A new Methodology to implement Security in a system based on **I**nternet **o**f **T**hings

Major Project - II

(CO – 821)

Thesis submitted in partial fulfilment of the requirements for the award of the degree of

Master of Technology in Software Technology

by

SURYA KANT JOSYULA (Roll No. 2K13/SWT/16)

Under the guidance of

Prof. Dr. DAYA GUPTA

Department of Computer Science & Engineering

Delhi **T**echnological **U**niversity

Shahbad Daulatpur, Main Bawana Road, New Delhi, Delhi 110042 (INDIA)

July, 2016

DECLARATION

I hereby want to declare that the thesis entitled **"A new Methodology to implement Security in a system based on Internet of Things"** being submitted to **Delhi Technological University** in partial fulfilment of the requirements for the award of the degree of **Master of Technology** in **Software Technology** is an authentic work carried out by me. The matter embodied in this thesis is original and has not been submitted for the award of any other degree or diploma anywhere.

SURYA KANT JOSYULA

Roll No. 2K13/SWT/16

Department of Computer Science & Engineering

Delhi Technological University

CERTIFICATE

This is to certify that the thesis entitled **"A new Methodology to implement Security in a system based on Internet of Things"** submitted by **Mr. Surya Kant Josyula (Roll No. 2K13/SWT/16),** in partial fulfilment of the requirements for the award of degree of Master of Technology in Software Technology to Delhi Technological University, Delhi is a record of the candidate's own work carried out by him under my supervision and guidance. The matter embodied in this thesis is original and has not been submitted for the award of any other degree or diploma anywhere as per best of my knowledge.

Date:

Prof. Dr. DAYA GUPTA

(Department of Computer Science & Engineering)

Delhi Technological University

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my guide Prof. Dr. Daya Gupta (Department of Computer Science & Engineering, Delhi Technological University) for her continuous support, expert guidance, and understanding throughout my study & research during this project. It is due to her vision, encouragement & valuable suggestions that I was able to complete this work.

I would also like to express gratitude to Mrs. Shruti Jaiswal (Research Scholar, Delhi Technological University) for providing me continuous support and guidance during this project.

I would also like to thank every DTU faculty member and the staff members who were directly or indirectly there to provide me valuable knowledge, guidance and support.

I would like to thank "Samsung" for providing me this option of higher studies and research opportunity simultaneously with the job. I am also thankful to my friends and colleagues who have supported me in the time of need.

I am really grateful to my family members for their unconditional love, support and understanding during this project.

Surya Kant Josyula
(2K13/SWT/16)

Table of Contents

List of Figures

List of Tables

ABSTRACT

The usability and popularity of the phenomenon of Internet of Things (IoT) is growing day by day. It is estimated that there will be billions of devices which will be inter-connected and working with each other using this concept. It has wide range of applications both industrial as well as residential. As the number of devices using this concept grows rapidly, security risks also grow with the same pace. This is because the network involved in it has different resource constraints and behaviour than that of the well-established networks like Internet network, WSN's, etc. So, security principles and concepts applicable to existing networks may not be directly applicable to these devices because of the constraints involved in IoT devices. Some of the constraints include having limited computational power, tight memory, limited energy budget, limited communication bandwidth being mobile and having heterogeneous architectures.

Personal data privacy, financial investments, safety hazards and human-life risks are some of the major stakes in IoT due to its working with everyday objects and human beings. More and more software vulnerabilities get exploited by attackers and other insane people. Then huge costs need to be spent to mitigate the problems that occur due to security issues.

Hence in this thesis, a new design methodology is proposed based on Security Engineering Framework [3] which considers security requirements to be a part the design decisions in the development life cycle. The inferred decisions can be used to mitigate security risks and can be easily adapted during each stage of software development to improve overall security and maintenance costs of the system.

CHAPTER – 1
INTRODUCTION

This chapter provides introduction and brief details about the work done in this thesis. It also provides details of the key terms being used and the background that led to this work.

1.1 INTRODUCTION

Internet of Things is a concept of further development of Internet intending to connect everyday objects to the Internet. It is a proposed network of "things" or physical objects embedded with sensors, electronics, software and network connectivity enabling these devices to communicate data with each other and humans. Abbreviation "**IoT**" is used to represent Internet of Things. IoT enables objects / things / devices / sensors to be controlled and sensed remotely across the network, creating more opportunities for interaction between computer-based systems and physical world resulting in more business opportunities, economic benefit and ease of living.

Kevin Ashton a British entrepreneur [2] first used the term in 1999 while working at Auto-ID Labs (which was originally called Auto-ID centres) - referring to a network of connected objects globally, based on RFID (Radio-frequency identification).

"Things" in Internet of Things, refers to a wide variety of sensors / devices like biochip transponders on farm animals, heart monitoring implants, automobiles with built-in sensors, electric clams in coastal waters, or field operation devices that help fire-fighters in rescue and search operations. These devices collect useful information with the help of several existing technologies. The collected data flows autonomously between other devices. Present market examples include washer/dryers and smart thermostat systems having built-in sensors, actuators, internet connectivity and ability to exchange data using Wi-Fi or other technologies. Virtually all devices can be part of IoT which can connect to internet but those that have sensing or actuating capabilities are preferred.

GSM Association (GSMA) [20] classified key components of IoT as Endpoint Ecosystems, Service Ecosystems, Network Operators and User Experience Systems.

If we knew everything to know about things in physical world located anywhere we can gather data which will help us to track, count, monitor, actuate, and greatly reduce waste, costs, and losses. We will also be able to know about faulty devices, when they need to be repaired or replaced, increasing productivity, autonomy, smartness and easy of life with unprecedented convenience.

IoT will help organizations and industries to reduce cost by improved productivity, efficiency and resource utilization. By using it many tasks could be performed remotely without risking human life or travel time providing real time insights, informed decisions can be taken all of which will in turn help in making smarter decisions creating more opportunities for people and industries.

As more and more things are connected to the Internet it calls for a security check. Due to the constraints involved with IoT devices and the high stakes associated with the working of these devices it can cause a lot of harm and financial loses if the security is not dealt early in the development cycle. It is said that security is one of the key factors and concern area for the success of IoT.

Providing security in IoT is more complex in comparison to network security. For most of existing IoT solutions are independent small networks, there are relatively few exploits can be attacked. With the sustained development of IoT, the small networks will merge into a large network. By then it would be more difficult to ensure the security. So it not only has security issues of internet but also the once with sensor networks, mobile networks, cloud network and others. Due to this integration of several networks privacy, access control & management, heterogeneous network authentication and others become more complex in IoT. Therefore solutions addressing each aspect of these security problems should be made.

Moreover, due to the resource-constrained network such as tight memory, limited communication bandwidth, limited computation and the limited energy budget available to Internet of Things devices, traditional security protocols and mechanisms

do not bode well for the Internet of Things. Thus, securing constrained devices requires optimal security mechanisms.

For serving this purpose and providing better security this thesis work proposes a new design methodology for IoT systems using the security engineering framework developed in earlier work [3] which proposed security check at every stage of development cycle. At each stage various security checks are done and design decisions are taken to incorporate security mechanisms during development phase itself.

This new design methodology also includes proposal of security requirements of IoT system and suitable security algorithms specific to IoT. Then it is inferred that using this methodology most of the security calamities can be avoided before they even happen.

1.2 MOTIVATION

IoT is an emerging phenomenon with rapid growth and estimated huge business value. Many organizations have started working on several projects individually as well as in collaboration [11, 12]. Due to this my main motivation for this work was to identify a key research area of this emerging phenomenon and narrow down to work on a sub-area of Internet of Things where lies my interest as well as there is scope of innovation so that something novel can be generated which will be useful to the society.

Motivation to choose security of IoT came from the idea after going through many research papers [1, 7, 8, 9, 10, 26, 27, 28] and it was found that there are gaps in security area related to Confidentiality, Identification, Authentication & Authorization, Liability, Privacy, Anonymity, Resilience, Fault Tolerance, Data freshness and Trust mainly due to the constraints and the heterogeneity in the IoT systems. Also it was inferred from these research papers that security was one of the major deciding factor for the growth of IoT.

1.3 RELATED WORK

Many researchers have identified several research challenges at various levels of Internet of Things. Research work of last few years which identifies key research areas and challenges of Internet of Things are consolidated and stated in this thesis.

Researchers in [1] analyses existing mechanisms and protocols for secure communications in the IoT. It provides information about how security in IoT for PHY layer, MAC layer, network-layer, routing, and application-layers can be achieved.

In another paper [7] researcher has conducted a study and compared existing IoT application layer protocols as well as protocols that are utilized to connect the things and end user applications. It addresses suitability of existing protocols, security, reliability, energy aspects of IoT.

Researchers in [8] review the current research in IoT, major IoT applications in industries, key enabling technologies, and identify research trends & challenges. Identified challenges are design of service-oriented architecture, scalability, management of heterogeneity, standardization, big data, data mining, interoperability, Security and Privacy. Also [9] reviews main challenges facing IoT middleware as Trust, Interoperability, Mobility, Heterogeneity, Scalability, Spontaneous Events, Unknown Data point availability, Random topology, Actuation conflicts, Bootstrapping, Security & Privacy, Extensibility, Modularity, Real-world integration.

Researchers in [10] intended to serve as a guideline and conceptual framework for future research in IoT. It provided details of existing research and suggested some significant research directions. It addressed that layered interoperability, multi-protocol communication support, modularity, sustainable business models, privacy, ownership and Security are the research challenges in IoT. Research in [25] states that the research challenges of IoT are Energy efficiency, reliability, Context awareness, Mobility awareness, handling big data.

Research challenges such as Identification, Interoperability, Autonomic networking, Security, Highly secure human body related services, Privacy protection, Plug and play, and Manageability are identified in [26]. IoT research challenges such as Identity and Naming management, standardization and interoperability, information privacy, objects security & safety, data confidentiality & encryption, network security, and Greening of IoT are addressed in [27].

Researchers in [28] firstly categories IoT systems and then addresses IoT research issues as challenges related to sensing & communication technology (interoperability, standardization, massive data issue, scalability, noisy data,) challenges in communication protocol (power consumption, special gateways, security challenges related to 6LowPAN, routing), challenges in middleware (abstraction, standardization, heterogeneity, scalability, zero infrastructure), challenges in QoS (resource constraints, scalability, heterogeneous traffic, dynamic infrastructure)

Researchers in [24] focuses on possible attacks on the RPL and 6LoWPAN network, counter measure against them, methods to mitigate the attacks, and also research opportunities in network layer security. It discusses attacks like Sinkhole attack, Selective Forwarding attack, Sybil attack, Clone ID attack, Hello Flooding attack, Hole attack, Wormhole attack, Denial of Service attack, Fragmentation attack, Authentication attack, Spoofing attack and confidentiality attacks.

Researchers in [13] discusses major issues raised while securing existing Web of Things as Concerns with regulatory frameworks, infrastructure protection, as security & privacy challenges such as data confidentiality & integrity, Trustworthiness, data privacy, identity management, access control. Researchers in [14] address several security issues of IoT such as key management and algorithms, security routing protocols, data fusion, authentication and access controls.

Consolidated list of important challenges and research areas of IoT are:

- Identification / Discovery
- Protocol Convergence & Managing Heterogeneity
- Standardization
- Interoperability

- Use of enabling technologies
- QoS, Robustness, Reliability
- Administration, Manageability
- Security, Privacy
- Data Management in Cloud / Big Data
- Data Mining
- Energy Efficiency
- Scalability
- Communication Spectrum issues

Consolidated security needs in IoT identified by researches in literature are:

- Naming & Identity Management
- Standardization
- Interoperability
- Privacy
- Physical security & Access Control
- Data confidentiality & Encryption
- Network security
- Real Time Response
- Integrity
- Trustworthiness

The research issues and challenges stated in this section were gathered from exhaustive searching in the surveys conducted in this field for last few years.

Security framework was proposed by some authors for improving security for generic software systems [21], Cloud systems [36], Big Data environments [37] but none has proposed a security framework for Internet of Things (IoT).

Security framework proposed by Gupta et al. [21] consisted of the following:

- Security Requirements Engineering Phase
- Security Design Engineering Phase
- Security Implementation

- Security Testing

They showed how security framework can be executed in parallel to software development life cycle. The sub-phases contained Security Requirements Elicitation, Security Requirements Analysis, Security Requirements Prioritization, Security Requirements Management, and Mapping of Security Requirements with Cryptographic Services, Security Design Analysis, Security Design Constraints, Security Design Structuring, Security Design Decisions, Security implementations and Security Testing.

Security problems were represented as Security Requirements by Firesmith in [3]. They were high-level requirements which give a specification of generic system behaviour which cannot be directly applied to IoT.

Several authors proposed hybrid encryption algorithms to improve efficiency, time taken and strength of security or save power [4, 5, 6, 38], but none has proposed a hybrid algorithm or set of algorithms which can single headedly address most of the security requirements & design constraints in IoT.

Researchers in [4] proposes mixed encryption algorithm to provide information integrity, confidentiality, non-repudiation, on data transmission of Internet of Things.

Researchers in [5] proposes new hybrid cryptographic algorithm using two asymmetric cryptographic techniques providing integrity, confidentiality, authentication using ECC, AES, RSA, Blowfish and MD5.

In [6] subasree & sakthivel propose algorithm to improve strength using combination of both symmetric and asymmetric cryptographic techniques. It provides integrity, confidentiality, and authentication with the help of ECC, Dual-RSA and MD5.

In [39] researchers address some of the available lightweight ciphers, compares between them and shows a new algorithm that can be applied for low computation devices. More details about the related work will be provided in the following chapters of this thesis.

1.4 PROBLEM STATEMENT

Protection in Internet of Things using existing technologies raises security challenges and opportunities for further research work. So, there is a need to improve existing technology to generate new technology which is more secure. The security issues and problems of the Internet of Things (IoT) are directly related to the wide range of application of its system. IoT as a combination of heterogeneous networks, not only has the same security problems with internet, sensor networks, and mobile communication networks, but also more such as privacy across the sub-networks, network authentication across the heterogeneous network, information store & management, access control & management problems and others. So, the research of IoT security is separate from that of Internet network security, for it is far more complicated. Therefore, we require targeted solutions for each aspect of security problems.

It is said that "even if one thing is able to prevent IoT from transforming the way we work and live, it will be a breakdown in security" [2] and hence this need to be addressed. Therefore, the problem of this thesis is as follows: "**A new Methodology to implement Security in a system based on Internet of Things**" which can effectively mitigate the vulnerabilities, threats or attacks expected on an IoT system.

To implement security in IoT system this thesis addresses existing security challenges and proposes new security requirements and suitable security algorithms. The security engineering framework used in this work considers security at each stage from the beginning to the end of the software development phase i.e. during requirements, design, implementations & testing phases. Security requirements for the system are proposed using well established methods and also suitable security algorithms (based on proposed security requirements) specific to IoT are proposed to mitigate maximum possible vulnerabilities, attacks or threats on the system. As per need and type of IoT system, proposed methodology can be applied to any IoT system to make it secure. Hence a new methodology is created for improving overall security of the system by addressing security during each phase of the development cycle.

1.5 SCOPE OF WORK & APPROACH

1.5.1 Scope of Work

The scope of this work includes thorough and deep understanding for the need of security engineering for Internet of Things, identification of key security and safety objectives of the IoT system. Various entities of the system are identified and analysed to understand their role in providing security. Using threat modelling threats and attacks possible on the system are prioritized and then security requirements are generated and prioritized.

Then during design phase design decisions are taken considering security requirements generated in previous phase. Each security requirement is mapped with the possible mitigation techniques and security algorithms. It will also consider several design constraints of the IoT system. Then security design decisions will be made using security design template. The created design decisions can then be used during implementation, testing and maintenance phases.

1.5.2 Approach

Firstly we need deep understanding of the IoT system. So we first understand the Internet of Things and its applications, its functioning and various usages. We then understand the security engineering framework concepts and how it can be applied to IoT system. A system based on IoT is considered and security requirements elicitation for this system is done. We identify all actors of the system using use case analysis and existing knowledge. Then we list all possible security vulnerabilities, possible attacks and threats of the IoT system. Assets are identified which get affected due to these vulnerabilities. Attacks / Threats are mapped to security requirements and security requirements are generated for the system. Using OWASP, threat modelling risk analysis is done which further help us to prioritize security requirements.

Security design of the system starts now. We list down design constraints of the IoT based system in consideration. Then we take into account the identified security requirements and list all possible security algorithms and mechanisms that can meet

the security requirements. Then security design template is made which helps in taking optimal security decisions. Using the results we propose security algorithms which can help to mitigate the identified problems and then finally comparisons of the proposed algorithms are done to help prioritize algorithms which will help to finalize the design decision.

1.6 ORGANIZATION OF THESIS

This section provides details of the chapters that follow:

Chapter 2 provides insight into the background of Internet of Things and its security. It also describes the related research done and research gaps present in this field.

Chapter 3 provides details of previously existing and proposed security engineering framework. It also discusses about Security Requirements Engineering, Security Design Engineering and Security implementation & Testing in IoT perspective.

Chapter 4 In this chapter firstly, an automotive vehicle tracking and control system based on IoT is discussed. Then all the steps involved in security requirements engineering phase of proposed framework are elaborated with respect to the IoT based automotive vehicle tracking and control system.

Chapter 5 In this chapter all the steps involved in security design engineering phase of proposed framework are discussed in detail with respect to the IoT based automotive vehicle tracking and control system described in previous chapter.

Chapter 6 In this chapter of thesis working of the implemented security engineering tool for IoT systems is described. It is based on the new security methodology proposed in this thesis.

Chapter 7 This chapter concludes this thesis work and provides insight into the future work.

Chapter 8 Provides References used in this thesis work.

CHAPTER – 2

BACKGROUND STUDY & RELATED RESEARCH

This chapter provides insight into the background & related research done for Internet of Things Security. It also describes gaps present in this field.

2.1 SECURITY ISSUES IN INTERNET OF THINGS

Security will be a fundamental enabling factor of most IoT applications. Internet of Things is a combination of heterogeneous networks, it not only involves the same security problems with Internet, mobile communication network, and the sensor networks, but also more particular ones, such as access control problems, authentication in heterogeneous network, privacy protection problem and management etc. The research of IoT security is different from that of Internet security, it is far more complicated. We can categorize the security issues in Internet of Things as follows:

Identification / Authentication / Authorization: Authentication in Internet of Things is very difficult as it involves heterogeneous network authentication. Things (sensors) / IoT Endpoint devices must be identified and authenticated properly before joining the network. IoT requires a global unique identifier (UID) for each entity in the network. Once identification and authentication is done authorization to the user must be provided i.e. set of rules which are permitted to him.

Confidentiality & Privacy: Need to ensure that the personal and sensitive information is not accessible to unauthorized users. Also, confidential & private messages should not reveal the content / information to eavesdroppers.

Resilience: In Internet of Things even if some interconnected nodes are compromised, still the system should be able to protect the network/ device / information from any threats or attacks.

Liability: In case of loss, misuse, theft or any other unusual event some liability or accountability should be taken.

Fault Tolerance: The system should be able to function with relevant security services in case of a fault such as a device compromise or failure.

Self-Healing: If a sensor in an Internet of Things application fails or compromised then, the other devices / components in it should be able to detect the failure and provide mechanisms to overcome the failure.

Heterogeneity / Standardization / Interoperability: The devices used in IoT are mostly standalone devices made for a specific purpose. Thousands of devices having different architectures and following various protocols constitute IoT network. Also there is lack of standardization between these devices. Interoperability between these devices is also a concern. These require a proper security design to be followed so as to mitigate security problem that arise from Heterogeneity, lack of Standardization and Interoperability in IoT.

Data Freshness: For an Internet of Things application to work in an efficient manner nodes / devices must have access to recent / latest messages or data as the system temporal measurements. For example to analyse the heart functionality of a patient by a doctor in a remove patient monitoring system needs the most recent ECG readings.

Big Data: When these devices communicate with themselves or with external entities large amount of data is generated. This data needs to be kept secure.

Constraints: IoT devices are constrained devices which have limited resources. Providing security with limited resources is also a security concern.

Trust: Trust should be present which determines a user's willingness to use the system. If a user is quite sure that the system is not compromised he is more willing to use the system.

Anonymity: In some cases the user does not want to disclose their identity to others. E.g. in a Remote Patient monitoring system, many medical patients would not want to disclose their identity or reports to anyone.

IoT can be compromised by various types of attacks. Many threats could be occurred during the manufacturing process. Eavesdropping attack compromise the authenticity, the integrity and the confidentiality of personal data. In addition, due to the eavesdropping attack, privacy of individual in the IoT is seriously menaced, especially if the data obtained by the attacker is important and contains personal information. Moreover, things are vulnerable to resource exhaustion attack such as Denial-of-service (DoS) attacks in which attackers send a mass saturation of incessant requests to specific things in order to deplete their resources. Thereby, network availability can be disrupted by flooding the network with a large number of packets.

Things should be authenticated before joining the network. However, unlike traditional networks, the thing's identity is not the equal as the identity of its implemented mechanisms that have different identification codes according to the object and its services. IoT requires a global unique identifier (UID) or a new & unique identification code to an entity in the network and a hierarchical identification schemes to indicate its location, such as the aggregation to IPv6 address.

Thus, things have unique identifiers called digital identity. In this context, a federated identity can offer a suitable solution for object identification and access control. Federated identity includes object's identity and its proprieties. Accordingly, authorizations specified for a given resource are no longer expressed in terms of object login IDs but in terms of requirements and conditions against object properties. Therefore, federated identity allows users of one security domain to securely access resources on another security domain, using the same account.

Recently, many researchers are focusing in the development of an infrastructure that allow object authentication in order to prove the identity. Consequently, Security requirements for the IoT subsume a wide range of security services including integrity, authentication, confidentiality, non-repudiation, authorization, and availability. These security services can be implemented using security protocols and combined cryptographic algorithms such as encryption algorithms, hash algorithms and digital signature that are used respectively to gain the confidentiality and the integrity and the non-repudiation.

The fragmentation of bigger packets like huge key exchange messages may be vulnerable to DoS attacks and cause fragmentation losses which degrade the overall network performance. Therefore, cryptographic techniques like public-key cryptography used in the traditional networks cannot be used with constrained devices like IoT End Point devices.

Security protocol and cryptography mechanisms ensuring secure communication must be reduced to be adapted to the highly constrained objects or completely new designed to be worth integrating into the Internet of Things. This adaptation must provide the same security level and security properties as the original mechanisms. Nowadays, the study of security of Internet of Things has caught many attentions of researchers. Therefore, there have been many researches trying to enhance the security of Internet of Things by improving the security protocol under the hardware limitations.

Many of IoT applications are expected to have large number of actuating and sensing devices, and in consequence its cost will be an important factor. On the other hand, cost limitations dictate constraints in terms of the resources available in sensing platforms, such as computational power and memory, while the unattended employment of many devices will also require the usage of batteries for energy storage.

Overall, all these factors motivate the design and adoption of better security mechanisms optimized for constrained sensing platforms, capable of providing its functionalities efficiently and reliably.

2.2 IOT SECURITY VS NETWORK / INTERNET SECURITY

Comparison of IoT security and Network Security is required due to the fact that IoT is a merger of several different types of networks whereas network security which is already existing and works on specific topologies. The other factor is that these devices are heterogeneous in nature i.e. they do not have same architecture and follow similar protocols. The last factor is that they devices are constrained devices and can be mobile. Table 2.1 elaborates the difference between the two.

Table 2.1: IoT Security vs. Network / Internet security

Design Parameters	IoT Security	Network Security
Memory constraints	Devices are memory constrained.	No memory constraints.
Speed of Computation	Low CPU Speed	High-speed CPU
Architecture	Heterogeneous	Mostly Homogeneous
Energy / Power Limitations	Limited battery power	There are no power limitations. They use power backups.
Scalability	There is an exponential increase in number of devices in IoT. Hence, selecting scalable security algorithm becomes a challenging task. A device can join or leave the network at anytime from anywhere.	They are connected by reliable wired links and have established wireless links also.
Cost	Low cost, long lived	May or may not be low cost and long lived
Communications Channel	IoT devices are connected to the Internet mainly through wireless links such as Zigbee, Z-Wave, Bluetooth, Bluetooth Low Energy, GSM, Wi-Fi, 3G/4G and Wi-Max. Hence, it is tough to have a security protocol which works for wireless links and provides security similar to wired links.	Have stable communication channels
Security Updates	Need to keep security protocols up-to-date arise to mitigate potential vulnerabilities, Automatic updation of security protocols is difficult.	They are having established a system for security.

2.3 SECURITY REQUIREMENTS BY FIRESMITH

Researcher Firesmith in [3] considered various security objectives like the following:

- Ensuring that client and user applications are identified and identities are properly verified.
- Ensuring that the client applications and users can access only the data & services which they are authorized.
- Ensuring that Intrusion detections by unauthorized client applications & users are possible.
- Ensuring that malicious & unauthorized programs, viruses do not infect application or its components.
- Ensuring that the data & communications are not intentionally altered.
- Ensure non-repudiation, privacy, confidentiality.
- Ensuring proper auditing by security personnel's.
- Ensuring that the applications and server survive attacks, protected against damage, destruction or theft.
- Ensure that maintenance of the system does not disrupt the security mechanism intentionally.

Based on these security objectives following Security Requirements were described:

- Identification Requirements

  Identification requirements are any security requirements that specify the extent to which a component, centre, business, or application, will identify its externals prior to interaction with them. E.g. an application will identify all of its client applications prior to allowing client applications to use the application's capabilities. This security requirement is used to mitigate the Identification security issue described in Section 2.1.

- Authentication Requirements

  Authentication requirements are any security requirements that specify the extent to which a component, centre, business, or application, will verify the identity of its externals prior to interaction with them. E.g. an application will verify the identity of

all of its users prior to allowing users to use the application's capabilities. This security requirement is used to mitigate the Authentication security issue described in Section 2.1.

- Authorization Requirements

  Authorization requirements are any security requirements that specify the access and usage privileges of authenticated client applications and the users. E.g. an application will allow each user / customer to obtain access to all of his / her own personal information. This security requirement is used to mitigate the Authorization security issue described in Section 2.1.

- Immunity Requirements

  Immunity requirements are any security requirements that specify the extent to which a component or an application will protect itself from damage or infection by unauthorized or undesirable programs. E.g. an application will protect itself from damage by scanning downloaded data to find for computer viruses and other harmful programs. This security requirement when used in combination of other requirements is used to mitigate the Fault Tolerance, Confidentiality and Privacy issues described in Section 2.1.

- Integrity Requirements

  Integrity requirements are any security requirements that specify the extent to which a component or an application will ensure its communications & data are not intentionally altered or modified. E.g. an application will prevent the unauthorized corruption of data that it sends to users and other customers. This security requirement when used in combination of other requirements is used to mitigate the Trust, Fault Tolerance, Confidentiality and Privacy issues described in Section 2.1.

- Intrusion Detection Requirements

  Intrusion detection requirements are any security requirements which specify the extent to which a component or an application will detect and log the attempted violation by unauthorised persons. E.g. an application will detect and log all attempted accesses that fail authentication, identification, or authorization requirements. This security requirement when used in combination of other

requirements is used to mitigate the Trust, Fault Tolerance, Resilience, Self-healing issues described in Section 2.1.

- Nonrepudiation Requirements

  Nonrepudiation requirements are any security requirements which specify the extent to which an application or business will prevent an entity to one of its interactions from denying having participated in some or all part of the interaction. E.g. an application will make and record tamper-proof logs / records which can be used to prove that the interaction was done by the entity only. This security requirement is used to mitigate the Trust and liability issues described in Section 2.1.

- Privacy Requirements

  Privacy requirements are any security requirements which specify the extent to which a component, business or an application will keep its sensitive data private form unauthorised parties. E.g. an application will not store of read sensitive user data being used in it. This security requirement is used to mitigate the Confidentiality, and Privacy issues described in Section 2.1.

- Security Auditing Requirements

  Security auditing requirements are any security requirements which specify the extent to which an application will enable security audits to check the status of the security mechanism being used. E.g. an application will collect, store, summarize, and regularly make report about the status of its security mechanisms being used. This security requirement when used in combination with other requirements is used to mitigate the Trust and liability issues described in Section 2.1.

- System Maintenance Security Requirements

  System maintenance security requirements are any security requirements which specify the extent to which a component or an application, component will prevent authorized updates or modifications from accidentally defeating its security mechanisms. E.g. an application will not violate its security requirements as a result of the system upgrade.

- Survivability Requirements

  These requirements state that an application should be able to survive an intentional destruction, damage or loss of a component. E.g. an application will not have a single point of failure. This security requirement is used to mitigate the Fault Tolerance & Resilience issues described in Section 2.1.

- Physical Protection Requirements

  These requirements state that the application should be able to protect itself from physical damage. E.g. an application should be able to protect its hardware from physical damage. This security requirement is used to mitigate the Fault Tolerance & Resilience issues described in Section 2.1.

2.4 SECURITY MECHANISMS

Some of the existing security mechanisms described by researchers in [21, 30, 36] are as follows:

- Ensuring Data Portability

  Due to the heterogeneity of the IoT Network achieving data portability is difficult but not impossible. We just need to make sure there are enough interfaces between the networks so that when a path fails the system can work through other path without lock-in attack. Lock-in threat can be avoided if data portability is ensured.

- Ensuring compliance with available standards and enforcing policies like Need to know Principle. As we know standardization is an issue of IoT. But if we deeply see IoT is a merger of sub-networks and these sub-networks does follow standards or we can say they can be standardized using existing mechanisms.

  However combination and standardization of the submerged network is in itself is a research work. Compliance should be ensured so as to avoid legal issues as well as to ensure better security of the system. We can also use principles like the Need to know principle in which the accesses and the authority given to persons is clear and make them liable so as to avoids theft / malicious insider problems by the abuse of power.

- Intrusion Detection & Prevention mechanisms

  There are several methods which can detect or prevent an intrusion. Regular checking of the network transmission or data being send or received is required for intrusion detection. Once detected, it can be prevented with the help of machine learning algorithms.

- Cryptographic Techniques

  Cryptography provides a way for securing data from various attacks. Sensitive data can be encrypted and protected against disclosure. Digital Certificates, Digital Signatures, Hash Functions, Authentication algorithms are all based on cryptography. They are divided into three basic types. All other resultant techniques are the combinations of these three basic types. They are:

  Symmetric Algorithms

  It uses same key to encrypt and de-crypt data. Sharing of keys is the major vulnerability of these systems. This is overcome by use of public key or Asymmetric key algorithms. Table 2.2 shows some popular symmetric ciphers.

Table 2.2: Some popular symmetric ciphers [21, 30]

Name of algorithm	Block size (bits)	Key size (bits)	Encryption speed (on 33 MHZ 486SX) (Kb/s)
DES	64	56	35
Blowfish	64	128	182
3DES (Triple DES)	64	168	12
IDEA	64	128	70
AES	128	128	60
CAST	64	128	53
RC5	64	128	86
RC4 (Stream cipher)	One byte at a time	256	164
SEAL (Stream cipher)	One byte at a time	160	381
PIKE (Stream cipher)	One byte at a time	160	62

  Asymmetric Algorithms

  In this two keys are used public and private. They are mathematically related and agreed between two parties. It doesn't have the key sharing vulnerability like the

symmetric one as no key needs to be shared. Table 2.3 shows some popular asymmetric ciphers.

Table 2.3: Some popular asymmetric ciphers [21, 30]

Name of algorithm	Encryption (ms)	Decryption (ms)	Sign (ms)	Verify (ms)
RSA (512 bits)	30	160	160	20
RSA (768 bits)	50	480	520	70
RSA (1024 bits)	80	930	970	80
ECDSA (160 bits)	797	281	150	230
ECDSA (233 bits)	882	385	250	521
ECDSA (283 bits)	928	400	25	580
HECDSA (81 bits)	668	191	60	31
HECDSA (83 bits)	893	224	56	32
ElGamal (512 bits)	330	240	250	1370

Hash Functions

Hash functions are one way functions which are collision-resistant. It is fixed-sized message digest or hash which is calculated on the basis of a hash function. Any change in size of message or data of message can be easily detected. Table 2.4 shows some popular Hash Functions.

Table 2.4: Some popular Hash Functions [21, 30]

Algorithm	Hash length (bits)	Encryption speed (on 33 MHZ 486SX) (Kb/s)
MD4	128	23
MD5	128	236
HAVAL	128	174
N-HASH	128	29
SHA1	160	75
SHA2	160	70

- Authentication Techniques

  Authentication is important because it helps decide the legitimate user / device. Basic authentication is provided by public key algorithms. Advanced methods like Two Factor Authentications, Multifactor Authentications, and Kerberos are available.

There are several advances like in two factor authentication in which there are two level authentications and it is assumed that the unauthorized user may not be able to provide both factors required for access correctly. In case of multi factor authentication the levels of authentications are added further. Kerberos was developed by MIT, it uses symmetric as well as asymmetric and trusted third party techniques for authentication.

- Digital Certificates assessment by tools like OCSP

  OCSP is Online Certificate Status Protocol. It provides checking of revocation of X.509 digital certificates. It can be vulnerable to replay attacks. It can be overcome by adding a "nonce" number. "Nonce" is a random or pseudo random number used only once.

- Vulnerability Assessment Tools

  There are many vulnerability assessment tools like Wire shark (www.wireshark.org), Metaspoilt (metasploit.com). They maintain a data base and analyse the inputs with respect to their known data base. This way they provide vulnerability assessment.

- Audit mechanisms e.g. SLA Strengthening

  SLA is Service Level Agreement document, it is a legal document and should be carefully made by the IoT service provider to avoid any penalties. Certain governments have very strict rules and penalties.

- Recovery Services

  These services are mostly made of machine learning algorithms which keep on learning once a failure occurs. In case of similar failure in future it will have the knowledge to overcome it.

- Maintenance Services

  These are regular services which are required for the proper functioning of the security system.

Researches have also proposed some hybrid cryptographic algorithms. Some of them are described as follows:

- It consists of work from Subasree [6] in which algorithms like ECC + Dual RSA + MD5 are used to make a hybrid cryptographic algorithm

- Elkady [31] in which text is first divided into two half's of N/2 each and then each half applies AES + ECC and Dual RSA respectively. It also applies HASH function.

- ECIES which stands for Elliptic Curve Integrated Encryption Scheme which consists of Key Agreement function, Key Derivation Function, Symmetric Encryption scheme, and Hash functions.

- A Mixed Encryption Algorithm [4] which is almost same as ECIES. But in ECIES we can choose sub-algorithms in each stage as per need. So, during security impact identification we are not considering this mixed encryption algorithm separately.

- A New Lightweight Hybrid Cryptographic Algorithm [39] for The Internet of Things addresses some of the available lightweight ciphers, then compares between them and describes a new algorithm which can be applied for low computation devices. It uses stream cipher to strengthen the security.

CHAPTER – 3

PROPOSED SECURITY ENGINEERING FRAMEWORK FOR IoT

This chapter provides details of the previously existing and proposed security engineering framework.

3. 1 PREVIOUS SECURITY ENGINEERING FRAMEWORK

Security engineering is a complicated process. It contains several security related activities parallel to the software development life cycle. They are security requirements engineering, security design engineering, implementation of security mechanisms and thorough testing.

Security requirements engineering further contains identification of security requirements, prioritization of security requirements and management of security requirements. Proper design decisions are made by proper security requirements elicitation. The design stage describes how requirements generated in previous phase can be implemented. By using security engineering we get a systematic and a better organized security functionality and design decisions. Security engineering framework as proposed by Gupta et al. [21] is shown in Figure 3.1.

The proposed framework has the following phases:

- ***Security Requirements Engineering***

  In this phase identification of stakeholders, assets, attacks & vulnerability was done then security requirements were elicited & analysed along with non-functional and functional requirements based on attacks and vulnerabilities on the assets of the system. Then they were prioritized and used for further checking.

- ***Security Design Engineering***

  In this phase cryptographic algorithms were identified, to mitigate the vulnerabilities and attacks on the system based on prioritized Security Requirements. This phase starts with mapping of security services with security requirements. Then threats,

attacks and vulnerabilities are mapped with assets and then considering various environment constraints design decisions security mechanisms best suited for a particular application are extracted out.

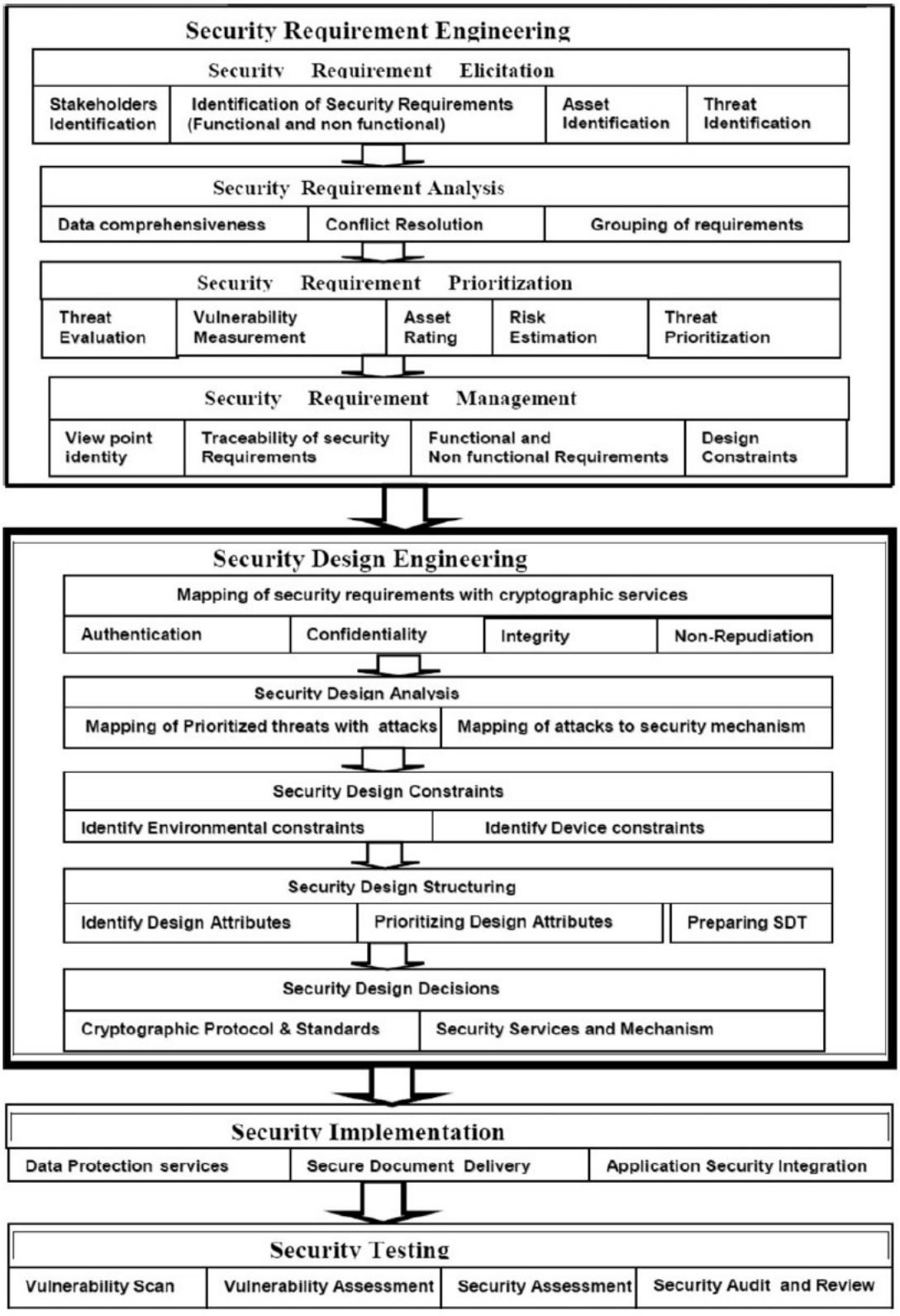


Figure 3.1: Previous Framework for Security Engineering Process [21]

- ***Security Implementation***

  In this phase the extracted security mechanisms in previous phase are implemented. It contains phases like Data protection services, secure document delivery an Application security Integration.

- ***Security Testing***

  In security requirements testing phase testing is done to check if the applied security mechanisms are working properly and mitigating all the threats and attacks and hence protecting valuable assets of the system.

The framework discussed in this section is generic and it cannot be applied directly to IoT based applications. So, in this thesis we have proposed a modified security framework targeted for IoT based applications.

3.2 PROPOSED SECURITY ENGINEERING FRAMEWORK FOR IOT

The proposed security framework for IoT Systems is depicted in Figure 3.2 which is modification of the security engineering framework discussed in previous section.

The proposed framework has following phases:

1. **Security Requirements Engineering Phase**: In this phase security requirements applicable to IoT systems are elicited, analysed and prioritized.

2. **Security Design Engineering Phase:** In this phase security design analysis is done and based on its results design decisions are taken.

3. **Security Implementation:** In this phase the design decisions taken in the previous phase are implemented. As per the taken decision decisions all the functionality is implemented related to security that were deduced in design phase i.e. security mechanisms are implemented in this phase. Related documentation is done as well. Integration of these with the existing system is also part of this phase only.

4. **Security Testing:** In this phase we test the implemented software whether all the security requirements are met. During security testing vulnerabilities are assessed again, possible attacks are tested and an assessment of the results are taken and considered for further action. Overall security assessment and verification of the system is done. If required, based on test results more features can be added for improving security. Finally, security audit and review can be done.

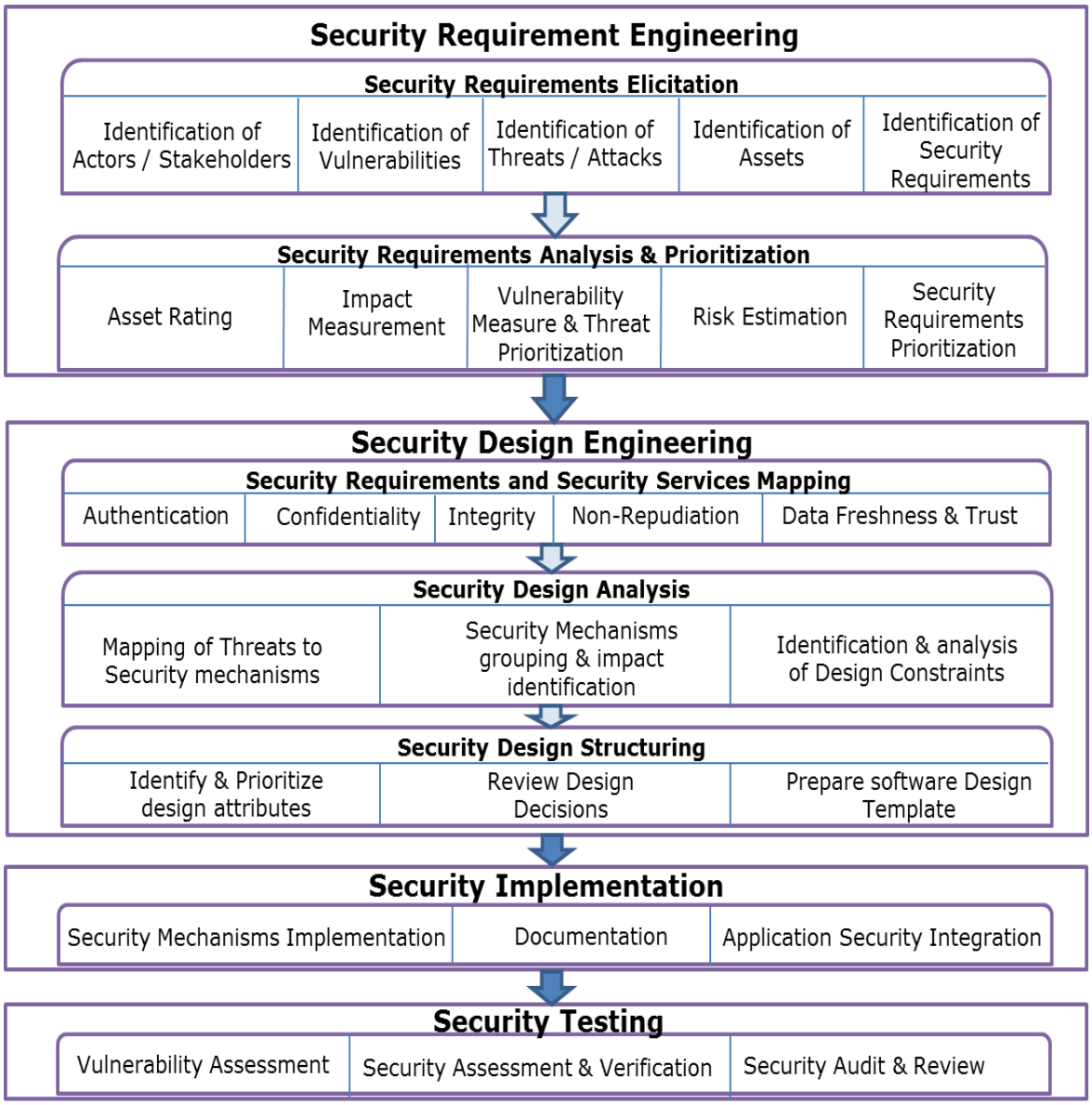


Figure 3.2: Proposed Security Engineering Framework

As mentioned there are four phases in the proposed framework, but in this thesis our focus is on the first two phases only which are discussed in the subsequent sections.

3.3 SECURITY REQUIREMENTS ENGINEERING

Security requirements are discovered, analysed and managed in this phase. There are three stages as depicted in Figure 3.3 using which Security Requirements specification is generated:

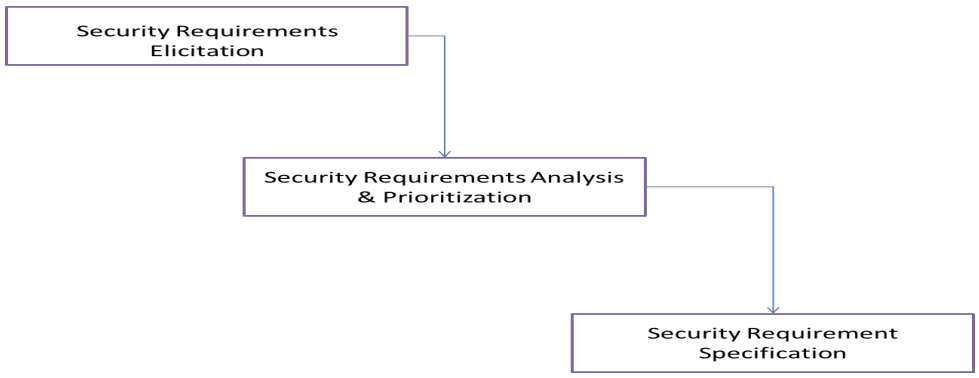


Figure 3.3: Security Requirements Engineering Process

- Security Requirements Elicitation
- Security Requirements Analysis & Prioritization
- Security Requirement Specification

Following is the working of the key stages of the Security Requirements Phase:

3.3.1 Security Requirements Elicitation

This stage involves identification and discovery of security requirements using various techniques or methods. The security requirements are drawn from different designs and organisational policies on security [22]. Security requirements are elicited for the system, based on the system requirements.

Following are the steps involved:

- Identification of various actors / stakeholders is done using view-point analysis [34, 35]. Humans, software system or hardware which is direct actors are identified. Software developer, administrators, regulators etc. are indirect actors and they are identified.
- Identification the vulnerabilities and attacks / threats.

- Identification of assets affected by threats / attacks and vulnerabilities.
- Identification Security Requirements associated with each vulnerability

3.3.2 Security Requirements Analysis & Prioritization

This phase is associated to quality control as there is consideration of some quality attributes of the gathered security requirements. Security requirements that are elicited should be able to mitigate all possible attacks on the functionality of the system. Grouping of the security requirements defined in previous stage is performed. Any ambiguities found are removed, analysis is done to check for completeness and consistency of the requirements. Risk analysis is performed using various methods for instance CRAMM, OWASP, AHP [23].

After security requirements analysis stage is complete, prioritization is performed on the basis of risk estimation. If the budget of the IoT application is low only medium to high risk security requirements may be considered for implementation. The rest of the requirements can be considered depending on the availability of resources.

Following are the steps involved:

- Assign value to corresponding vulnerability (0-100) and Impact (1-10) using OWASP.
- Prioritize assets and calculate asset rating
- Calculate Impact (a) using asset rating by taking average of the number of threats associated with the assets.

  (a) Calculate Impact = Average of (Asset Rating with respect to Threats)
- Calculate Threat rating by mapping vulnerabilities and threats on assets of the system.
- Calculate Risk value (b) based on Threat rating and impact

  (b) Estimate value of Risk = Average of (Threat Rating X Impact)
- Prioritize security requirements on measure of vulnerability, threats and risk value.

3.3.3 Security Requirements Specification

It is the output of the Security Requirements phase. The prioritized requirements generated in previous stages are specified so as keep track of all security requirements. These can be referred during design, implementation and testing phases for execution of these stages. Several methods can be used for documenting the details of the requirements phase [22]. This stage is not considered in this thesis work and will be part of future work.

3.4 SECURITY DESIGN ENGINEERING

Each stage considers the following design constrains specific to IoT systems during the process:

- Limited computation power
- Limited speed of computation
- Limited memory & bandwidth
- Heterogeneous Architecture

In this phase of software development life cycle software structure is designed to apply the specifications of the system. Firstly in security design life cycle we will do security service mapping with security requirements, then security analysis and security structuring. We will observe the impact of various factors like cryptographic techniques, coding standards and other related available techniques that need to be followed. Security mechanisms are mapped for mitigating identified security requirements. In this phase, any bad decision will lead to design failure making system vulnerable to attacks.

Following is the working of the key stages of the Security Design Phase:

3.4.1 Security requirements and security services mapping.

Security requirements which were prioritized are mapped to known security services like Non-repudiation, Integrity, Authentication, Confidentiality, Integrity, and etc. Real-time response, Data freshness & Trust are newly added for IoT. This later helps in specifying and mapping security mechanisms for specific security requirements.

3.4.2 Security design analysis

Prioritization of attacks / threats and affected assets are defined in this step.

It contains two sub-steps:

- Threats are mapped to Security Mechanisms
- Cryptography techniques and other security measures are identified to mitigate all the threats of the system. Impact of attack is accordingly evaluated.
- Identifying security design constraints. All the design constrains of the system should be considered in this stage for proper execution of this methodology.
- Security mechanisms are sorted and grouped based on constraints

3.4.3 Security design structuring

In this stage, design attributes are identified and prioritized.

It consists of two sub-steps:

- Identify design attributes and prioritizing them
- Design attributes like cost, choice of implementation platform, applicability of mitigating techniques, and priority of constraints are identified in this stage. E.g. Symmetric algorithms like AES, DES are suitable for confidentiality service requirements, as they are many times faster than asymmetric algorithms like RSA.
- Review design decisions
- Preparation of security design template (SDT)
- Security design template is made to take care of each security requirement as a design decision based on the process discussed so far. This will store all the specifications of the design constraints and mitigation techniques for the system in design.

3.4.4 Security design decisions

The output of the Security design engineering phase is the security design decisions listed in the Security design template. Using previous knowledge best suitable mechanisms are selected upon the attributes applied in the Security Design Template.

CHAPTER – 4

SECURITY REQUIREMENTS ENGINEERING FOR SECURING IoT

In this chapter firstly an automotive vehicle tracking and control system based on IoT is discussed then all the steps involved in security requirements engineering phase of proposed framework are discussed with respect to this system.

4.1 A SYSTEM BASED ON IoT AND APPLICATION OF NEW SECURITY METHODOLOGY TO IT

An automotive vehicle tracking and control system based on IoT is shown in Figure 4.1. It consists of the following components:

- User Interface Devices

  These are the devices using which clients, customers or the users of the IoT application can monitor, track, and sense or control various things in the IoT system. E.g. a user can check the speed of the running automobile on it or it can send a control command to switch off the vehicle's ignition in case of theft.

- Communication Medium

  Various communications are in use like Wi-Fi, 3G, 4G, and Ethernet. Main role of the communication medium is to send and receive data between IoT Endpoint devices and the Internet.

- Application / Cloud servers

  These are the servers hosting the IoT automotive vehicle tracking and control system or its server side application. They can be cloud based or a traditional servers used to run the application and store data.

- Internet

  It is the network using which an IoT Endpoint device can communicate to its application server or user interface application using the various communication mediums.

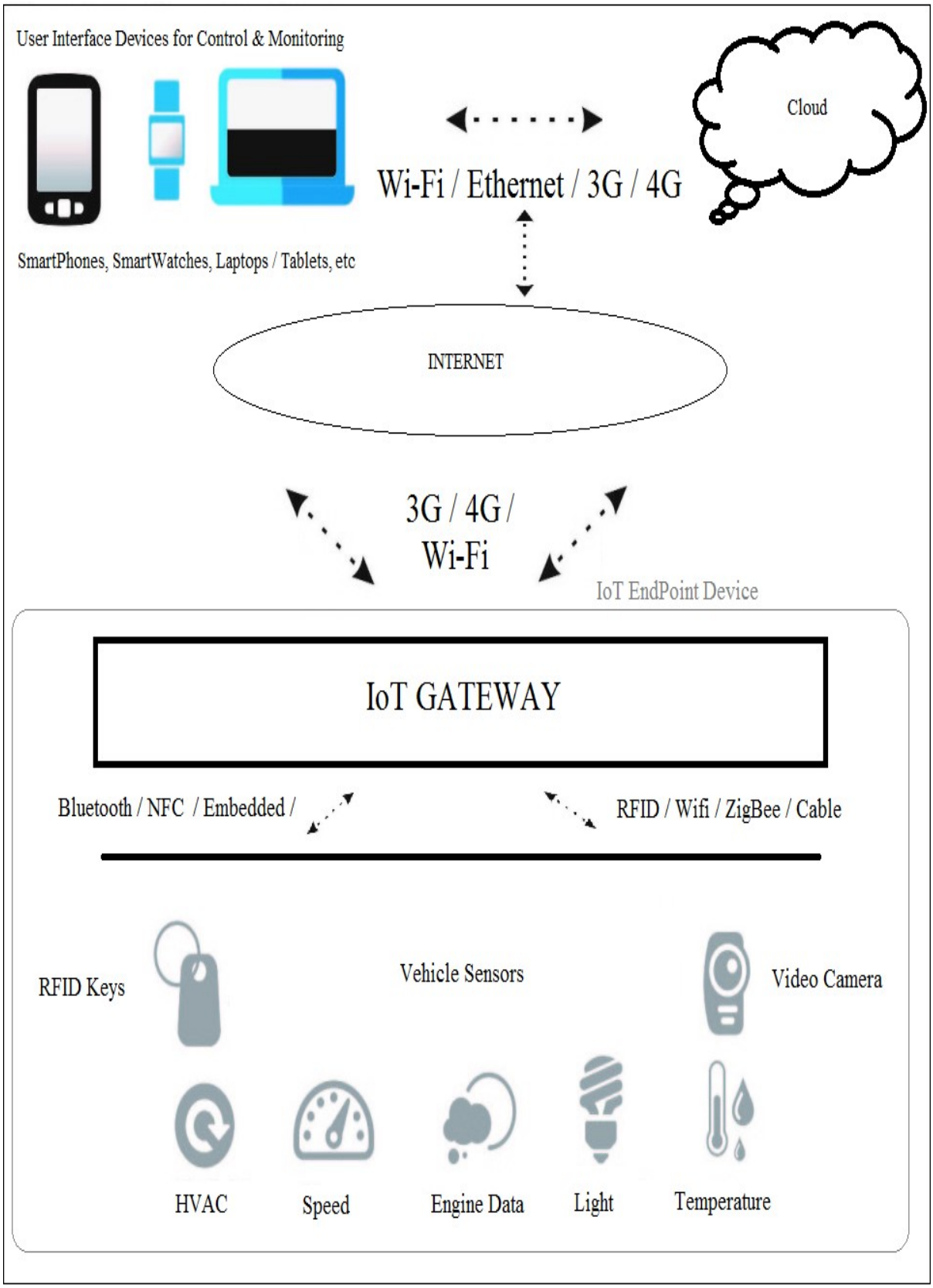


Figure 4.1: A system based on IoT – Automotive vehicle tracking and control system

- IoT Gateway

  These are the communication interfaces which help in sending and receiving data by providing interfaces which aren't available with the sensors.

- Sensors / Actuators

  These are the hardware components of the IoT End point devices. They are capable of sensing or controlling certain things or parameters. E.g. a sensor can be connected to speedometer to read the current speed. An actuation can be connected to vehicle ignition to switch it on or off.

4.2 SECURITY REQUIREMENTS ELICITATION

Firstly actors or stakeholders involved in the IoT based automotive vehicle tracking and control system shown in Figure 4.1 are identified using view-point analysis [34, 35]. Then vulnerabilities and attacks or threats on the assets of the IoT based automotive vehicle tracking and control system shown in Figure 4.1 are identified. Finally they are mapped with security requirements as defined by Firesmith in [3].

Following are the steps involved:

- Identification of actors / Stakeholders
- Identification of vulnerabilities related to actors
- Identification of threats based on vulnerabilities
- Identification of assets
- Identification of Security Requirements

4.2.1 Identification of actors or Stakeholders.

An actor specifies a role played by an external entity or user or any other system that interact with the subject. "Subject" is a classifier representing a software system, business, physical system or device under design, analysis, or consideration having some behaviour to which certain use cases gets applied. "Role" is informally used as some group, type, or particular facet of users that require specific services from the modelled subject with the associated use cases. Stakeholders are the actors which

have vested interest in the system who want something out of the system or liable for things happening in the system.

Stakeholders are identified using viewpoint approach. Two classes of actors are identified direct actors and indirect actors. Direct actors are those who play immediate role in the system and have vested interest or liability. Indirect actors are those who may not have vested interest or liability but they perform some role in the functionality of the system. For e.g. direct actors for the IoT based automotive vehicle and tracking system shown in Figure 4.1 are IoT customers, IoT users, IoT Service Providers, IoT End point devices, Peer devices Security administrators and auditors whereas indirect actors are firmware, IoT Servers, IoT Gateways and Internet. Details of the actors / stakeholders identified for IoT based automotive vehicle tracking and control system shown in Figure 4.1 is as follows:

- **IoT Customer**

  IoT customer is the one who uses the IoT system to actuate or monitor something. E.g. a user who wants to run washing machine located at home from a far off location.

- **IoT User**

  IoT user is the one who has access to use the features of IoT system subscribed by the IoT customer. A customer can also be a user. E.g. the person who wants to run washing machine in the previous example might want that his family members are also able to do the same. In this case the family members will be just IoT users whereas the subscriber is IoT customer as well as IoT user.

- **IoT service Provider**

  IoT service provider is the entity, who owns, manages and operates the IoT system to provide its services to the users. In some applications this can be just the network service providers. It is part of the IoT service ecosystem.

- **IoT security administrator**

  IoT administrator maintains security related functionalities.

- **IoT auditor**

  Manages audit and other things

- **IoT End Point Devices (Sensors & Devices)**

  These are the endpoint systems located at a remote location. They can be located at a fixed location or can be mobile.

- **Peer devices**

  These are the systems which do not have direct access to the Internet. They get connected to peers who have Internet connection using other network means like BT, Wi-Fi, NFC etc. and they communicate to and from Internet using the peer's connection.

- **IoT Gateways / Network Interfaces**

  These are the systems which help communicate the sensors & devices with the Internet

- **Internet**

  It is the Internet network used as a communication channel.

- **IoT Servers**

  These are the servers which are used by IoT service providers to store data and process user requests. They can be cloud based.

- **IoT user Interface Devices**

  These are the devices like computers or smartphones which can be used as an IoT user interface. These are used by the user for monitoring and controlling the IoT Endpoint Devices. They can be personal computers, tablets, smartphones, smart watch etc.

- **Firmware**

  The software contained in the IoT End point devices

4.2.2 Identification of vulnerabilities.

In computer security, vulnerabilities are the weaknesses of the system which allow attackers to reduce the system's information assurance. It is the intersection of three elements: a system attacker capability to exploit the flaw, susceptibility or flaw within the system and attacker access to the flaw. In this step vulnerabilities are identified corresponding to each actor specified in section 4.2.1 for the IoT based automotive vehicle tracking and control system shown in Figure 4.1. Various vulnerabilities are taken from literature survey [29, 36, 37]. Various vulnerabilities identified related to actors are depicted in Table 4.1. For convenience and easy distinction Vulnerabilities are prefixed with "V." throughout this thesis.

Table 4.1: Vulnerabilities related to Actors

Actors	Operations	Vulnerabilities
IoT User	An actual user who can initiate a monitor or actuation command	V.Untrained_Users V.Weak_Access_Control V.Legal_Audit_Issues V.System_Misuse
Firmware / Software	This is the OS or API's or software that the IoT Endpoints / IoT Servers / IoT User Interfaces are using	V.Monitoring_Absence V.Inadequate_Logging V.Physical_Security V.Misconfigurations V.Unsecured_API_Firmware V.Obsolete_System V.Lack_of_Standards V.Old_Data
IoT Network Interfaces	Communication of data between IoT End Point Devices and IoT applications / IoT User Interfaces via Internet	V.Weak_Access_Control V.Unencrypted_Data V.Breached_Firewall V.Inadequate_Logging V.Insecure_Interfaces V.Insecure_Network_services V.Insufficient_Security_Configurability V.Legal_Audit

		V.Intrution_Detection
Internet	Communication medium	V.Weak_Access_Control V.Unencrypted_Data V.Breached_Firewall V.Obsolete_System V.Insecure_Interfaces V.Insecure_Network_services V.Insufficient_Security_Configurability
IoT End Point Devices	Collection of Data by using inputs from the sensors, controlling attached peripherals as per user commands, communication of sensed and control data with the IoT Gateways	V.Weak_Access_Control V.Unencrypted_Data V.Monitoring_Absence V.Inadequate_Logging V.Physical_Security V.Misconfigurations V.Unsecured_API_Firmware V.Obsolete_System V.Insecure_Network_services V.Insecure_Interfaces V.Insufficient_Security_Configurability V.Remote_Access V.Resource_Isolation V.Poor_Key_Management V.Lack_of_Standards V.Old_Data V.Intrution_Detection
IoT Servers	Internet based servers running IoT applications. They serve as intermediary between IoT User interface and IoT End Point Devices	V.Weak_Access_Control V.Unencrypted_Data V.Breached_Firewall V.Misconfigurations V.Insecure_Interfaces V.Insufficient_Security_Configurability V.System_Misuse

		V.Intrution_Detection
Peer Devices	Collection of Data by using inputs from the sensors, controlling attached peripherals as per user commands, communication of sensed and control data with Peer nodes	V.Weak_Access_Control V.Unencrypted_Data V.Monitoring_Absence V.Inadequate_Logging V.Physical_Security V.Misconfigurations V.Unsecured_API_Firmware V.Obsolete_System V.Insecure_Network_services V.Insufficient_Security_Configurability V.Resource_Isolation V.Lack_of_Standards V.Old_Data V.Intrution_Detection
IoT Internet Gateways	Communication of data between IoT End Point Devices and IoT applications / IoT User Interfaces via Internet	V.Weak_Access_Control V.Unencrypted_Data V.Breached_Firewall V.Inadequate_Logging V.InsecureInterfaces V.Insecure_Network_services V.Insufficient_Security_Configurability V.Audit_Certification V.Intrution_Detection
IoT User Interfaces	There are again applications which are attached to the IoT application servers providing IoT services to the user	V.Untrained_Users V.Misconfigurations V.Unsecured_API_Firmware V.Obsolete_System V.Legal_Audit V.System_Misuse

4.2.3 Identification of threats or attacks.

In computer security, a threat is anything that has the potential to cause serious harm to a computer system. It is a possible danger that may or may not happen and might exploit a vulnerability to cause serious damage or harm. Threats can lead to attacks on networks computer systems. Attack is any attempt to expose, destroy, alter, steal, disable or unauthorized access. Various threats / attacks are taken from literature survey [24, 29, 36, 37]. In this step threats are identified corresponding to each vulnerability using Table 4.2. Table 4.2 shows vulnerability and threat mapping being constructed from extensive literature review. An "X" in the Table means that the threat is present / attack is possible due to the corresponding vulnerability. For e.g. threat T.Fraud is possible due to vulnerabilities V.Weak_Access_Control, V.Inadequate Logging, V.System_Misuse, & V.Insufficient_Security_Configuration. For convenience and easy distinction Threats are prefixed with "T." throughout this thesis.

Table 4.2: Vulnerabilities and threats mapping

Vulnerabilities → / *Threats ↓*	*V.Weak_Access_Control*	*V.Inadequate_Logging*	*V.Breached_Firewall*	*V.Unvalidated_Input*	*V.Unsecured_API_Firmware*	*V.Obsolete_System*	*V.Misconfiguration*	*V.Unencrypted_Data*	*V.Untrained_User*	*V.Monitoring_Absence*	*V.Unsecured_Network*	*V.Intrution_Detection*	*V.Physical_Security*	*V.Old_Data*	*V.System_Misuse*	*V.Legal_Audit_Issues*	*V.Lack_of_Standards*	*V.Resource_Isolation*	*V.Poor_Key_Management*	*V.Remote_Access*	*V.Insufficient_Security_Configurability*	*V.Insecure_Interfaces*
T.Change_Data	X	X	X					X	X		X			X	X	X	X	X	X		X	X
T.Data_Theft	X		X					X								X	X	X	X		X	X
T.Impersonate	X														X					X		
T.Fraud	X	X													X						X	
T.Repudiation_Receive		X												X	X				X		X	
T.Repudiate_Send		X												X	X				X		X	
T.Credential_Theft	X									X					X						X	X

T.Phishing	**X**							**X**	**X**				**X**			**X**				**X**
T.Insider	**X**	**X**		**X**	**X**		**X**	**X**		**X**			**X**			**X**				
T.Spoofing	**X**							**X**	**X**				**X**			**X**			**X**	**X**
T.Human_Error				**X**				**X**					**X**	**X**	**X**					
T.Disclose_Data								**X**	**X**				**X**				**X**		**X**	**X**
T.Privacy_Violated									**X**				**X**						**X**	**X**
T.DDoS										**X**		**X**		**X**			**X**			
T.Misuse_of_System_Resources									**X**				**X**	**X**	**X**	**X**				
T.Injection_Attack				**X**	**X**														**X**	**X**
T.Malware			**X**	**X**	**X**					**X**	**X**									**X**
T.Communication_Interception								**X**			**X**							**X**	**X**	**X**
T.Communication_Infiltration								**X**			**X**							**X**	**X**	**X**
T.Eavesdropping								**X**			**X**							**X**	**X**	**X**
T.Technical_Failure						**X**	**X**					**X**					**X**			
T.Power_Failure							**X**						**X**			**X**				
T.Network_Infrastructure_Failure							**X**						**X**			**X**				
T.Hardware_Failure						**X**						**X**	**X**	**X**			**X**			
T.Unavailability												**X**		**X**			**X**			
T.Vandalism													**X**		**X**				**X**	**X**
T.Operational_Issues						**X**											**X**			**X**
T.Console_Access_Attack					**X**		**X**						**X**	**X**			**X**		**X**	**X**
T.Chip_Access_Attack					**X**								**X**	**X**			**X**			**X**
T.Timing_Attack											**X**	**X**			**X**				**X**	
T.Hello_Flooding_Attack	**X**		**X**								**X**	**X**							**X**	**X**
T.Node_Capture													**X**				**X**			
T.Fake_Node							**X**				**X**		**X**				**X**			

4.2.4 Identification of assets

Asset can be any entity that has value to the organization. The assets of the system can be Private & Confidential Data, equipment's that are being monitored controlled by the IoT Endpoint devices, IoT Endpoint Devices, and others. Assets should be protected from illicit use, access, disclosure, destruction, alteration, and/or theft, resulting in losses to the organization. Assets are also identified on the basis of view-point analysis [34, 35].

Assets corresponding to actors identified in Section 4.2.1 are shown in Table 4.3.

Table 4.3: Identification of Assets

Actors	Operations	Assets
IoT User	An actual user who can initiate a monitor or actuation command	Personal Sensitive Data Personal Data Trust Service Delivery Network Logs Intellectual Property Credentials
IoT End Point Devices	Collection of Data by using inputs from the sensors, controlling attached peripherals as per user commands, communication of sensed and control data with the IoT Gateways	Personal Sensitive Data Personal Data Trust Service Delivery Network Logs Backup / Archive Data

Peer Devices	Collection of Data by using inputs from the sensors, controlling attached peripherals as per user commands, communication of sensed and control data with Peer nodes	Personal Sensitive Data Personal Data Trust Service Delivery Network Logs Backup / Archive Data
IoT Internet Gateways	Communication of data between IoT End Point Devices and IoT applications / IoT User Interfaces via Internet	Network Service Delivery Logs
IoT Network Interfaces	IoT network interfaces are the interfaces attached to the IoT End Point Devices. They can communicate to Internet via Internet Gateways or with Peer devices	Network Service Delivery Logs
Internet	Communication medium	Network Service Delivery Logs
IoT Servers	Internet based servers running IoT applications. They serve as intermediary between IoT User interface and IoT End Point Devices	Resources attached Customer Data Account information
IoT User Interfaces	There are again applications which are attached to the IoT application servers providing IoT services to the user	Network Personal Sensitive Data Credentials
Firmware / Software	This is the OS or API's or software that the IoT Endpoints / IoT Servers / IoT User Interfaces are using	Service Delivery Trust

4.2.5 Identification of Security Requirements

Security Requirements describe non-functional and functional requirements that need to be applied in order to achieve the security attributes of an IoT system. If security requirements are not properly defined, the resulting system cannot be evaluated for success or failure prior to implementation. Table 4.4 shows Security Requirements for IoT based on threats mapping for threats that were identified in Section 4.2.3. An "X" in the Table means that the threats present or attack possible can be overcome by the use of corresponding security requirement. For e.g. threat T.Fraud can be overcome by Security requirements Authentication and Security Auditing.

Table 4.4: Security Requirements based on Threats mapping

Security Requirements → / ***Threats ↓***	***Identification***	***Authentication***	***Authorization***	***Immunity***	***Integrity***	***Intrusion Detection***	***Non-Repudiation***	***Privacy***	***Security Auditing***	***Survivability***	***Physical Protection***	***System Maintenance***	***Real-Time Response***	***Data Freshness***	***Trust***
T.Change_Data	X	X	X											X	X
T.Data_Theft	X	X	X												X
T.Impersonate	X						X								
T.Fraud		X							X						
T.Repudiation_Receive							X						X		
T.Repudiate_Send							X						X		
T.Credential_Theft		X													X
T.Phishing		X	X												
T.Insider	X	X	X		X		X								X
T.Spoofing	X	X	X				X								
T.Human_Error					X										
T.Disclose_Data			X				X								X
T.Privacy_Violated		X	X		X			X		X					X
T.DDoS		X	X										X		X
T.Misuse_of_System_Reso			X	X		X								X	

urces														
T.Injection_Attack				**X**		**X**							**X**	
T.Malware					**X**									**X**
T.Communication_Interception					**X**							**X**		
T.Communication_Infiltration					**X**							**X**		
T.Eavesdropping														**X**
T.Technical_Failure									**X**			**X**	**X**	**X**
T.Power_Failure									**X**			**X**	**X**	**X**
T.Network_Infrastructure_Failure									**X**			**X**	**X**	**X**
T.Hardware_Failure									**X**			**X**	**X**	**X**
T.Unavailability										**X**		**X**	**X**	**X**
T.Vandalism									**X**					**X**
T.Operational_Issues												**X**	**X**	**X**
T.Console_Access_Attack									**X**			**X**	**X**	**X**
T.Chip_Access_Attack					**X**			**X**	**X**					
T.Timing_Attack					**X**			**X**				**X**		
T.Hello_Flooding_Attack														
T.Node_Capture									**X**		**X**			
T.Fake_Node	**X**													**X**

List of Security Requirements provided by Firesmith in [3] were provided in Section 2.3 of this thesis. All of those security requirements are applicable to IoT as well. In addition we have proposed new security requirements specific to IoT systems. They are as follows:

- **Real-Time Response**:

  Many IoT applications should give real time response. So, if there is any occurrence of security compromise mechanism should be there to get the incidence reported to concerned entities in real time.

- **Data Freshness**:
  The data sent should always be of the current time and latest.

- **Trust**:
  Implementing all security requirements will generate Trust. It will make people adopt to the IoT systems without much concern about security.

4.3 SECURITY REQUIREMENTS ANALYSIS & PRIORITIZATION

Security Requirements analysis is part of security requirements engineering. It is the process of determining security expectations for a modified or new application / product. The requirements must be relevant, detailed and quantifiable.

In this stage analysis of security requirements are done to estimate risk value related to each security requirement based on threat and assets ratings so as to use the estimated risk value to prioritize the security requirements.

Impacts and threats are classified according to OWASP methodology [29]. Assets are prioritized and asset rating is calculated. Threat rating is calculated by mapping vulnerabilities and threats on the assets of the system. Impact is calculated as per the average of threats on each valuable asset. Then risk value is calculated based on threat rating and impact. Finally security requirements are prioritized on the measure of vulnerabilities, threats and risk value.

Following are the steps involved:

- Prioritization of assets and calculation of asset rating
- Calculation of impact based on assets rating
- Calculation of threat rating
- Calculation of risk value
- Prioritizing security requirements

4.3.1 Prioritization of assets and calculation of asset rating

It is calculated as the count of the asset's importance with respect to the Actors. Table 4.5 shows the calculation of asset rating. An "X" in the Table means that the particular asset is valuable to the corresponding actor. Asset rating is calculated as the sum / count of the X. So, the asset which is valuable to maximum actors will have the highest asset rating and vice-versa. E.g. Backup data has asset rating of 2 as it is important to only IoT Endpoint devices and Peer devices.

Table 4.5: Calculation of asset rating

Actor → / *Assets ↓*	*IoT User*	*IoT Endpoint Devices*	*Peer Devices*	*IoT Internet Gateways*	*IoT Network Interfaces*	*Internet*	*IoT Servers*	*IoT User Interfaces*	*IoT Firmware / Software*	*Asset Rating*
Personal Sensitive Data	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**		**8**
Personal Data	**X**	**X**	**X**							**3**
Trust	**X**	**X**	**X**	**X**	**X**	**X**	**X**		**X**	**8**
Real time data delivery	**X**	**X**		**X**	**X**				**X**	**5**
Network				**X**	**X**	**X**				**3**
Credentials	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**		**8**
Resources attached		**X**					**X**			**2**
Account Information	**X**	**X**					**X**			**3**
Logs		**X**		**X**	**X**					**3**
Backup Data		**X**	**X**							**2**

4.3.2 Calculation of impact based on assets rating

Asset rating is taken from the Table 4.5 for each asset. Threats and assets are mapped and then impact rating is calculated as the average of asset rating of the affected

threats i.e. (Sum of Asset Ratings of assets getting affected by threats / Total number of Assets):

Calculate Impact = Average of (Asset Rating with respect to Threats)

Table 4.6 shows the calculation of Impact rating. E.g. Threat T.Fraud can affect assets Trust and Account information which has asset rating as 8 and 3 respectively. So, impact of threat T.Fraud will be (8+3) / (10) = 1.1

Table 4.6: Calculation of Impact rating

Asset Rating → / *Threats →*	*Personal Sensitive Data*	*Personal Data*	*Trust*	*Real Time data Delivery*	*Network*	*Credentials*	*Resources attached*	*Account Information*	*Logs*	*Backup Data*	*Impact Rating*
T.Change_Data	**8**	**3**	**8**			**8**		**3**			**3**
T.Data_Theft	**8**	**3**	**8**			**8**		**3**			**3**
T.Impersonate	**8**	**3**	**8**			**8**		**3**			**3**
T.Fraud			**8**					**3**			**1.1**
T.Repudiation_Receive					**3**						**0.3**
T.Repudiate_Send					**3**						**0.3**
T.Credential_Theft						**8**					**0.8**
T.Phishing				**5**							**0.5**
T.Insider	**8**	**3**	**8**			**8**		**3**			**3**
T.Spoofing				**5**							**0.5**
T.Human_Error								**3**			**0.3**
T.Disclose_Data	**8**	**3**	**8**			**8**		**3**			**3**
T.Privacy_Violated	**8**	**3**	**8**			**8**		**3**			**3**
T.DDoS				**5**	**3**						**0.8**
T.Misuse_of_System_Resources							**2**				**0.2**

T.Injection_Attack				**5**					**3**		**0.8**
T.Malware				**5**							**0.5**
T.Communication_Interception	**8**	**3**	**8**		**3**	**8**		**3**	**3**		**3.6**
T.Communication _Infiltration	**8**	**3**	**8**		**3**	**8**		**3**	**3**		**3.6**
T.Eavesdropping	**8**	**3**	**8**			**8**		**3**			**3**
T.Technical_Failure				**5**						**2**	**0.7**
T.Power_Failure				**5**						**2**	**0.7**
T.Network_Infrastructure_Failure				**5**	**3**					**2**	**1**
T.Hardware_Failure				**5**						**2**	**0.7**
T.Unavailability				**5**						**2**	**0.7**
T.Vandalism			**8**								**0.8**
T.Operational_Issues				**5**						**2**	**0.7**
T.Console_Access_Attack				**5**		**8**				**2**	**1.5**
T.Chip_Access_Attack							**2**		**3**		**0.5**
T.Timing_Attack				**5**	**3**		**2**				**1**
T.Hello_Flooding_Attack				**5**							**0.5**
T.Node_Capture	**8**	**3**	**8**					**3**			**2.2**
T.Fake_Node			**8**		**3**						**1.1**

4.3.3 Calculation of threat rating (Threat Prioritization)

It is calculated as the count of the vulnerabilities importance with respect to the Threats. An "X" in the Table means that the particular threat is possible due to the corresponding vulnerability. Threat rating is calculated as the sum count of the X. So, the threat which can be present due to most of the vulnerabilities will have the highest threat rating and vice-versa. E.g. threat T.Fraud can be present due to vulnerabilities V.Weak_Access_Control, V.Inadequate_logging, V.System_Misuse, V.Insufficient_Security_Configurability so it has threat rating of 4. Table 4.7 shows the calculation of Threat rating.

Table 4.7: Calculation of Threat rating

Vulnerability → / *Threats →*	*V.Weak_Access_Control*	*V.Inadequate_Logging*	*V.Breached_Firewall*	*V.Unvalidated_Input*	*V.Unsecured_API_Frimware*	*V.Obsolete_System*	*V.Misconfiguration*	*V.Unencrypted_Data*	*V.Untrained_User*	*V.Monitoring_Absence*	*V.Unsecured_Network*	*V.Intrution_Detection*	*V.Physical_Security*	*V.Old_Data*	*V.System_Misuse*	*V.Legal_Audit_Issues*	*V.Lack_of_Standards*	*V.Resource_Isolation*	*V.Poor_Key_Management*	*V.Remote_Access*	*V.Insufficient_Security_Configurability*	*V.Insecure_Interfaces*	*Threat Rating*
T.Change_Data	X	X	X					X	X		X			X	X	X	X	X	X		X	X	**14**
T.Data_Theft	X		X					X								X	X	X	X		X	X	**9**
T.Impersonate	X														X					X			**3**
T.Fraud	X	X													X						X		**4**
T.Privacy_Violated	X							X							X	X				X	X	X	**7**
T.Repudiation_Receive		X												X	X				X		X		**5**
T.Repudiate_Send		X												X	X				X		X		**5**
T.Credential_Theft	X									X					X						X	X	**5**
T.Phishing	X								X	X					X			X				X	**6**
T.Insider	X	X		X	X		X	X		X			X		X			X					**10**
T.Spoofing	X								X	X					X			X			X	X	**7**
T.Human_Error				X					X						X	X	X						**5**
T.Disclose_Data								X	X						X				X		X	X	**6**
T.Privacy_Violated									X						X						X	X	**4**
T.DDoS										X		X			X			X					**4**
T.Misuse_of_System_Resources									X						X	X	X	X					**5**
T.Injection_Attack				X	X																X	X	**4**
T.Malware			X	X	X					X	X											X	**6**

T.Communication_Interception							**X**			**X**							**X**			**X**	**X**	**5**
T.Communication_Infiltration							**X**			**X**							**X**			**X**	**X**	**5**
T.Eavesdropping							**X**			**X**							**X**			**X**	**X**	**5**
T.Technical_Failure						**X**	**X**				**X**						**X**					**4**
T.Power_Failure							**X**							**X**			**X**					**3**
T.Network_Infrastructure_Failure							**X**							**X**			**X**					**3**
T.Hardware_Failure						**X**					**X**	**X**		**X**			**X**					**5**
T.Unavailability											**X**			**X**			**X**					**3**
T.Vandalism												**X**			**X**					**X**	**X**	**4**
T.Operational_Issues						**X**											**X**				**X**	**3**
T.Console_Access_Attack					**X**		**X**					**X**		**X**			**X**		**X**	**X**		**7**
T.Chip_Access_Attack					**X**							**X**		**X**			**X**				**X**	**5**
T.Timing_Attack										**X**	**X**				**X**					**X**		**4**
T.Hello_Flooding_Attack	**X**		**X**							**X**	**X**									**X**	**X**	**6**
T.Node_Capture												**X**					**X**					**2**
T.Fake_Node							**X**			**X**		**X**					**X**					**4**

4.3.4 Calculation of risk value with respect to Threats

Risk value is calculated as the multiplication of the threat rating and impact rating of a particular threat. Impact rating and Threat ratings for threats are taken from Table 4.6 and Table 4.7 respectively. For e.g. Risk Value of T.Change_Data = 42 (14*3). Table 4.8 shows the calculation for Risk Estimation.

Estimate value of Risk = Threat Rating * Impact Rating

Table 4.8: Risk Estimation

Threats	*Threat Rating*	*Impact Rating*	*Risk Value*
T.Change_Data	**14**	**3**	**42**
T.Data_Theft	**9**	**3**	**27**
T.Impersonate	**3**	**3**	**9**
T.Fraud	**4**	**1.1**	**4.4**
T.Repudiation_Receive	**5**	**0.3**	**1.5**
T.Repudiate_Send	**5**	**0.3**	**1.5**
T.Credential_Theft	**5**	**0.8**	**4**
T.Phishing	**6**	**0.5**	**3**
T.Insider	**10**	**3**	**30**
T.Spoofing	**7**	**0.5**	**3.5**
T.Human_Error	**5**	**0.3**	**1.5**
T.Disclose_Data	**6**	**3**	**18**
T.Privacy_Violated	**4**	**3**	**12**
T.DDoS	**4**	**0.8**	**3.2**
T.Misuse_of_System_Resources	**5**	**0.2**	**1**
T.Injection_Attack	**4**	**0.8**	**3.2**
T.Malware	**6**	**0.5**	**3**
T.Communication_Interception	**5**	**3.6**	**18**
T.Communication _Infiltration	**5**	**3.6**	**18**
T.Eavesdropping	**5**	**3**	**15**
T.Technical_Failure	**4**	**0.7**	**2.8**
T.Power_Failure	**3**	**0.7**	**2.1**
T.Network_Infrastructure_Failure	**3**	**1**	**3**
T.Hardware_Failure	**5**	**0.7**	**3.5**
T.Unavailability	**3**	**0.7**	**2.1**
T.Vandalism	**4**	**0.8**	**3.2**
T.Operational_Issues	**3**	**0.7**	**2.1**

T.Console_Access_Attack	**7**	**1.5**	**10.5**
T.Chip_Access_Attack	**5**	**0.5**	**2.5**
T.Timing_Attack	**4**	**1**	**4**
T.Hello_Flooding_Attack	**6**	**0.5**	**3**
T.Node_Capture	**2**	**2.2**	**4.4**
T.Fake_Node	**4**	**1.1**	**4.4**

4.3.5 Prioritizing security requirements

Security requirements are prioritized based on the risk value associated with each of threats associated with it. Table 4.9 shows the security requirements prioritization for the IoT based automotive vehicle tracking and control system shown in Figure 4.1. The priority is calculated as the sum of risk value of the threats that can be overcome by Security Requirement. Risk value of threat is taken from Table 4.8. E.g. intrusion detection security requirement can overcome T.Misuse_of_System_Resources and T.Injection_Attack which has risk value of 1 and 3.2 respectively. So its priority will be 4.2 (1 + 3.2).

Table 4.9: Security Requirements Prioritization

SECURITY REQUIREMENTS	THREATS with Risk Value	PRIORITY
Identification	T.Change_Data (42) T.Data_Theft (27) T.Impersonate (9) T.Insider (30) T.Spoofing (3.5) T.Fake_Node (4.4)	115.9
Authentication	T.Change_Data (42) T.Data_Theft (27) T.Fraud (4.4) T.Credential_Theft (4) T.Phishing (3) T.Insider (30) T.Spoofing (3.5)	113.9

Authorization	T.Change_Data (42) T.Data_Theft (27) T.Phishing (3) T.Insider (30) T.Spoofing (3.5) T.Disclose_Data (18) T.Misuse_of_System_Resources (1) T.Privacy_Violated (12) T.DDoS (3.2)	139.7
Immunity	T.Misuse_of_System_Resources (1) T.Injection_Attack (3.2)	4.2
Integrity	T.Insider (30) T.Privacy_Violated (12) T.Human_Error (1.5) T.Malware (3) T.Communication_Interception (18) T.Communication _Infiltration (18) T.Chip_Access_Attack (2.5) T.Timing_Attack (4)	89
Intrusion Detection	T.Misuse_of_System_Resources (1) T.Injection_Attack (3.2)	4.2
Non-Repudiation	T.Impersonate (9) T.Insider (30) T.Spoofing (3.5) T.Disclose_Data (18) T.Repudiation_Receive (1.5) T.Repudiate_Send (1.5)	63.5
Privacy	T.Privacy_Violated (12) T.Chip_Access_Attack (2.5) T.Timing_Attack (4)	18.5
Security Auditing	T.Fraud (4.4)	4.4

Survivability	T.Privacy_Violated (12) T.Chip_Access_Attack (2.5) T.Node_Capture (4.4) T.Console_Access_Attack (10.5) T.Vandalism (3.2) T.Technical_Failure (2.8) T.Power_Failure (2.1) T.Network_Infrastructure_Failure (3) T.Hardware_failure(3.5)	44
Physical Protection	T.Unavailability (2.1)	2.1
System Maintenance	T.Node_Capture (4.4)	4.4
Data Freshness	T.Operational_Issues (2.1) T.Console_Access_Attack (10.5) T.Technical_Failure (2.8) T.Power_Failure (2.1) T.Network_Infrastructure_Failure (3) T.Hardware_failure(3.5) T.Unavailability (2.1) T.Misuse_of_System_Resources (1) T.Injection_Attack (3.2) T.Change_Data (42)	72.3
Real-Time Response	T.Repudiation_Receive (1.5) T.Repudiate_Send (1.5) T.DDoS (3.2) T.Communication_Interception (18) T.Communication _Infiltration (18) T.Data_Theft (27) T.Technical_Failure (2.8) T.Power_Failure (2.1) T.Network_Infrastructure_Failure (3) T.Hardware_failure(3.5)	99.3

	T.Unavailability (2.1) T.Operational_Issues (2.1) T.Console_Access_Attack (10.5) T.Timing_Attack (4)	
Trust	T.Fake_Node (4.4) T.Operational_Issues (2.1) T.Console_Access_Attack (10.5) T.Vandalism (3.2) T.Technical_Failure (2.8) T.Power_Failure (2.1) T.Network_Infrastructure_Failure (3) T.Hardware_failure(3.5) T.Unavailability (2.1) T.Eavesdropping (15) T.Disclose_Data (18) T.Privacy_Violated (12) T.DDoS (3.2) T.Malware (3) T.Change_Data (42) T.Data_Theft (27) T.Credential_Theft (4) T.Insider (30)	187.9

CHAPTER – 5

SECURITY DESIGN ENGINEERING FOR SECURING IoT

In this chapter all the steps involved in security design engineering phase of proposed framework are discussed in detail with respect to the IoT based automotive vehicle tracking and control system described in Section 4.1 of previous chapter.

5.1 MAPPING OF SECURITY REQUIREMENTS WITH SECURITY SERVICES

This section will consider the mapping of security requirements with security services & Security mechanisms. This is the starting step of the security design phase. In this phase security requirements are mapped with security services and security mechanisms as shown in Table 5.1. Security requirements were identified for IoT based automotive vehicle tracking and control system in the previous Chapter. Security services are the services that are provided by known stable and established security algorithms [36]. These algorithms and known methods which are used to improve security of a system form the security mechanisms. Most of the services are based on cryptographic techniques.

Table 5.1: Mapping of Security Requirements with Security Services

Security Services	Security Requirements	Security Mechanisms
Availability	Identification	Digital Certificates
	Authentication	Authentication Exchanges Two Factor Authentications Multi Factor Authentications Kerberos Key Agreement Protocols
	Authorization	Key Agreement Protocols RBAC (Role-Based Access Control) DAC (Discretionary Access Control) MAC (Mandatory Access Control)

	Non-Repudiation	Digital Signatures
	Intrusion Detection	Intrusion Detections & Prevention mechanisms Vulnerability Assessment Tools Cryptographic Techniques
	Survivability	Recovery Services Ensuring Data Portability
	Physical Protection	Recovery Services Secure Booting Cryptographic Techniques
	System Maintenance	Maintenance Services
	Real-Time Response	Vulnerability Assessment Tools Faster Cryptographic Techniques
	Data Freshness	Vulnerability Assessment Tools Faster Cryptographic Techniques
Confidentiality	Confidentiality (Privacy + Immunity)	Encryption mechanisms Transport Layer Security mechanisms (e.g. TLS / DTLS)
Integrity	Integrity	Hash Functions
Audit ability	Security Auditing	Auditing mechanisms Service Level Agreements Strengthening (SLA_ Strengthening)
Trust	Trust	Compliance mechanisms Need to know Principle Enforcement All of the above cryptographic techniques

Now we proceed to the next stage in the security design phase which is security design analysis.

5.2 SECURITY DESIGN ANALYSIS

Security design analysis consists of three stages:

- Mapping of Threats / attacks to Security Mechanisms
- Security Mechanisms grouping & Impact Identification
- Identification & Analysis of Design Constraints

5.2.1 Mapping of Threats / attacks to Security Mechanisms

In this step we will map identified threats to security mechanisms. Security mechanisms can be used alone or in combination to mitigate all the threats in an IoT system.

All the security mechanisms described in Section 2.4 have been mapped with threats and shown in Table 5.2. Table 4.4 and Table 5.1 were taken as reference for mapping Security requirements, attacks and Security Mechanisms.

Table 5.2: Mapping of Threats & Security mechanisms

Security Services	Security Requirements	Threats / Attacks	Security Mechanisms
Availability	Identification	T.Change_Data T.Data_Theft T.Impersonate T.Insider T.Spoofing T.Fake_Node	Digital Certificates
	Authentication	T.Change_Data T.Data_Theft T.Fraud T.Credential_Theft T.Phishing T.Insider T.Spoofing	Authentication Exchanges Two Factor Authentications Multi Factor Authentications Kerberos Key Agreement

			Protocols
	Authorization	T.Change_Data T.Data_Theft T.Phishing T.Insider T.Spoofing T.Disclose_Data T.Misuse_of_System_ Resources T.Privacy_Violated T.DDoS	Key Agreement Protocols RBAC (Role-Based Access Control) DAC (Discretionary Access Control) MAC (Mandatory Access Control)
	Non-Repudiation	T.Impersonate T.Insider T.Spoofing T.Disclose_Data T.Repudiation_Receive T.Repudiate_Send	Digital Signatures
	Intrusion Detection	T.Misuse_of_System_ Resources T.Injection_Attack	Intrusion Detections & Prevention mechanisms Vulnerability Assessment Tools Cryptographic Techniques
	Survivability	T.Privacy_Violated T.Chip_Access_Attack T.Node_Capture T.Console_Access_Att ack T.Vandalism T.Technical_Failure	Recovery Services Ensuring Data Portability

		T.Power_Failure T.Network_Infrastructure_Failure T.Hardware_failure	
	Physical Protection	T.Unavailability T.Spoofing	Recovery Services Secure Booting Cryptographic Techniques
	System Maintenance	T.Node_Capture	Maintenance Services
	Real-Time Response	T.Repudiation_Receive T.Repudiate_Send T.DDoS T.Communication_Interception T.Communication_Infiltration T.Data_Theft T.Technical_Failure T.Power_Failure T.Network_Infrastructure_Failure T.Hardware_failure T.Unavailability T.Operational_Issues T.Console_Access_Attack T.Timing_Attack	Vulnerability Assessment Tools Faster Cryptographic Techniques

	Data Freshness	T.Operational_Issues T.Console_Access_Attack T.Technical_Failure T.Power_Failure T.Network_Infrastructure_Failure T.Hardware_failure T.Unavailability T.Misuse_of_System_Resources T.Injection_Attack T.Change_Data	Vulnerability Assessment Tools Faster Cryptographic Techniques
Confidentiality	Confidentiality (Privacy + Immunity)	T.Privacy_Violated T.Chip_Access_Attack T.Timing_Attack T.Misuse_of_System_Resources T.Injection_Attack	Encryption mechanisms Transport Layer Security mechanisms (e.g. TLS / DTLS)
Integrity	Integrity	T.Insider T.Privacy_Violated T.Human_Error T.Malware T.Communication_Interception T.Communication_Infiltration T.Chip_Access_Attack T.Timing_Attack	Hash Functions

Audit ability	Security Auditing	T.Fraud	Auditing mechanisms Service Level Agreements Strengthening (SLA_ Strengthening)
Trust	Trust	T.Fake_Node T.Operational_Issues T.Console_Access_Attack T.Vandalism T.Technical_Failure T.Power_Failure T.Network_Infrastructure_Failure T.Hardware_failure T.Unavailability T.Eavesdropping T.Disclose_Data T.Privacy_Violated T.DDoS T.Malware T.Change_Data T.Data_Theft T.Credential_Theft T.Insider	Compliance mechanisms Need to know Principle Enforcement All of the above cryptographic techniques

5.2.2 Security Mechanisms grouping & Impact Identification

There are many security mechanisms and their alternatives. Our focus in this thesis is on various cryptographic techniques that were discussed in Section 2.4 of this thesis. Comprehensive evaluation of these is required with respect to the threat they mitigate. It can be observed from Table 5.2 that a single cryptographic technique cannot mitigate all the above threats. Thus grouping is required so as to find the best suite of algorithms that can be applied for the system. Either we need to use multiple cryptographic techniques or combine them to form a single technique having various features i.e. make them hybrid or use an existing hybrid algorithm if it can mitigate all possible threats to our system.

The grouping and the calculated impact is shown in Table 5.3. It is found that ECIES mitigates most of the threats for the IoT based automotive vehicle tracking and control system. The impact analysis depicts the applicability of each algorithm for a particular attack. A "Y" depicts that a particular algorithm mitigates particular attack and "N" depicts it does not.

Grouping is done on the basis of prior information about the security algorithm [36]. Impact of an algorithm is calculated on the basis of maximum Security requirements that an algorithm can meet. E.g. ECIES has impact value of 11 as it meets 11 security requirements.

Table 5.3: Security Mechanisms grouping & Impact Identification

Security Services	Security Requirements	Attacks / Threats	Security Mechanisms	Suitable Cryptographic Algorithms														
				Asymmetric Algorithms			Symmetric Algorithms			Hashing Algorithms		Signature Algorithms			Hybrid Algorithms			
				RSA	ECC	HECC	AES	DES	Triple DES	MD5	SHA1	RSA + DSA	ECDSA	HECDSA	ECC + DUAL RSA + MD5 (SUBASREE [6])	N/2 (AES + ECC) + N/2 (Dual RSA)) + HASH (ELKADY [31])	Lightweight Hybrid Cryptographic Algorithm Mouza Bani [39]	ECIES
Availability	Identification	T.Impersonate T.Fake_Node	Digital Certificates	Y	Y	Y	N	N	N	N	N	Y	Y	Y	Y	Y	N	Y
	Authentication	T.Fraud T.Credential_Theft T.Phishing T.Insider T.Spoofing	Authentication Exchanges Two Factor Authentications Multi Factor Authentications Kerberos Key Agreement Protocols	Y	Y	N	N	N	N	N	N	N	N	N	Y	Y	N	Y
	Authorization	T.Phishing T.Insider T.Spoofing T.Disclose_Data T.Misuse_of_System_Resources	Key Agreement Protocols RBAC (Role-Based Access Control)	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	N	Y

		T.Privacy_Violated T.DDoS	DAC (Discretionary Access Control) MAC (Mandatory Access Control)															
	Non-Repudiation	T.Impersonate T.Insider T.Spoofing T.Repudiation_Receive T.Repudiate_Send	Digital Signatures	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	N	Y
	Intrusion Detection	T.Misuse_of_System_Resources T.Injection_Attack	Intrusion Detections & Prevention mechanisms Vulnerability Assessment Tools Cryptographic Techniques	Y	Y	Y	N	N	N	Y	Y	Y	Y	Y	Y	Y	N	Y
	Physical Protection	T.Spoofing	Recovery Services Cryptographic Techniques	Y	N	N	Y	N	N	N	N	N	N	N	Y	Y	Y	Y
	Real-Time Response	T.Repudiation_Receive T.Repudiate_Send T.DDoS T.Communication_Interception T.Communication _Infiltration T.Operational_Issues T.Console_Access_Attack T.Timing_Attack	Vulnerability Assessment Tools Faster Cryptographic Techniques	N	Y	Y	Y	N	N	Y	Y	N	Y	Y	N	N	Y	Y

	Data Freshness	T.Operational_Issues T.Console_Access_Attack T.Misuse_of_System_Resources T.Injection_Attack T.Change_Data	Vulnerability Assessment Tools Faster Cryptographic Techniques	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y
Confidentiality	Confidentiality (Privacy + Immunity)	T.Privacy_Violated T.Chip_Access_Attack T.Timing_Attack T.Misuse_of_System_Resources T.Injection_Attack	Encryption mechanisms Transport Layer Security mechanisms (e.g. TLS / DTLS)	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y	Y
Integrity	Integrity	T.Insider T.Privacy_Violated T.Human_Error T.Malware T.Communication_Interception T.Communication _Infiltration T.Chip_Access_Attack T.Timing_Attack	Hash Functions	N	N	N	N	N	N	Y	Y	N	N	N	Y	Y	N	Y
Trust	Trust	T.Fake_Node T.Operational_Issues T.Console_Access_Attack T.Vandalism T.Technical_Failure T.Power_Failure T.Network_Infrastructure_Failure T.Hardware_failure T.Unavailability T.Eavesdropping T.Disclose_Data T.Privacy_Violated T.DDoS T.Malware	Compliance mechanisms Need to know Principle Enforcement All of the above Cryptographic techniques	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y

		T.Change_Data T.Data_Theft T.Credential_Theft T.Insider																
Total Impact				**6**	**6**	**5**	**4**	**2**	**2**	**4**	**4**	**4**	**5**	**5**	**9**	**9**	**4**	**11**

5.2.3 Identification & Analysis of Design Constraints

A design constraint refers to some limitation on the conditions under which a system is developed, or on the requirements of the system which can decide whether particular method can be considered or not. Table 5.4 shows the design constraints of the IoT based automotive vehicle tracking and control system shown in Figure 4.1.

Table 5.4: Security Design constraints in IoT

Design Parameter	Constraint
Memory	Devices are memory constrained. It may not have memory to execute complex security protocols.
Speed of Computation	Low CPU Speed, so finding a security solution without effecting real-time response of system is a complex task.
Architecture / Network Topology	IoT devices have Heterogeneous architectures and dynamic network topologies. Protocol convergence is an important factor and it should be well considered when choosing a security solution.
Mobility	Some IoT devices mobile in nature. Security solution must also consider this.
Energy / Power	Limited battery power. The security solution should cope up with this.
Scalability	There is an exponential increase in number of devices in IoT. Hence, selecting scalable security algorithm becomes a challenging task. A device can join or leave the network at anytime from anywhere.
Cost	Low cost, long lived
Communication Channel	IoT devices are connected to the Internet mainly through wireless links such as Zigbee, Z-Wave, Bluetooth, Bluetooth Low Energy, GSM, Wi-Fi, 3G/4G and Wi-Max. Hence, it is tough to have a security protocol which works for wireless links and provides security similar to wired links.
Security Updates	Need to keep security protocols up-to-date arise to mitigate potential vulnerabilities, Automatic updation of security protocols is difficult.

When impact analysis is combined with the design constraints it provides better options to the developer.

5.3 SECURITY DESIGN STRUCTURING

- Identification & Prioritization of design attributes
- Review design decisions
- Prepare software design Template

5.3.1 Identification & Prioritization of design attributes

Table 5.5: Security Design attributes identification & Prioritization

Quality Attribute	Design Attributes	IoT Service Providers	IoT End Point devices	IoT Peer devices	IoT User Interfaces
Performance	Memory	Medium	High	High	Medium
	Speed of Computation	Medium	High	High	Medium
	Energy / Power	Medium	High	High	Medium
	Run Time performance	Medium	High	High	Medium
	Communication Channel	Medium	High	High	Medium
Security	Security Objectives Security Updates	High	High	High	High
Usability	Mobility Compatibility	High	High	High	High
Scalability	Scalable without effecting current solution	High	High	High	Low
Cost	Cost of chosen solution	High	Low	Low	Low
Portability	Architecture / Network Topology	High	High	High	Low

Design attributes are characteristics or features of a system. Depending on the design constraints, some decisions need to be changed. We identify design attributes on the basis of actors used and the type of IoT network and its constraints. Then we prioritize these attributes. Identification and prioritization of design attributes for IoT based automotive vehicle tracking and control system is shown in Table 5.5. E.g. IoT End Point devices will have high priority for constraints like memory, power consumption etc., whereas IoT User Interfaces do not as IoT End Point devices are resource constrained devices whereas IoT User Interfaces may have better resource availability in comparison.

5.3.2 Review design decisions

Design decisions are making decision to formulate a plan and making decisions that are to be followed during development. In this step we review the taken design decisions of the system to check if it is enough secure. Depending on design attributes and the design constraints we may alter the design decisions in this step.

ECIES is a hybrid algorithm which mixes up several once. Depending on the quality attribute requirements we find that the following will be best suitable for IoT based automotive vehicle tracking and control system:

- For Key Agreement function – ECDH over others
- Key Derivation Function – ECDSA over others
- Symmetric Encryption scheme – AES 128 over others
- Hash function – MD5 over others

5.3.3 Preparation of security design template

Security design template is the reference document which takes care of each requirement. This will keep every specification of design constraint and design attribute of the specific environment. All mitigation techniques with respect to Quality attributes are listed in it. Based on design constraints of the specific application required mitigation technique can be chosen. Security design template with respect to IoT based automotive vehicle tracking and control system is shown in Table 5.6. It is made using the combination of Table 5.5 and Table 5.3.

Table 5.6: Security Design Template

Quality Attribute	Design Attributes	IoT Service Providers	IoT End Point devices	IoT Peer devices	IoT User Interfaces	Security Mechanisms & Techniques
Performance	Memory	Medium	High	High	Medium	Cryptographic Techniques RSA ECC HECC AES DES Triple DES MD5 SHA1 RSA+DSA ECDA HECDA ECDH Hybrid Subasree Hybrid Elkandy ECIES Data Portability Selection of architecture and topologies as per attributes & constraints Security Guidelines GSMA Iotivity Availability Techniques Two Factor Authentication Multi Factor Authentication Self-Healing and Resilience Mechanisms Vulnerability Assessment Tools Audit Mechanisms Recovery Services Maintenance Services
	Speed of Computation	Medium	High	High	Medium	
	Energy / Power	Medium	High	High	Medium	
	Run Time performance	Medium	High	High	Medium	
	Communication Channel	Medium	High	High	Medium	
Security	Security Objectives Security Updates	High	High	High	High	
Usability	Mobility Compatibility	High	High	High	High	
Scalability	Scalable without effecting current solution	High	High	High	Low	
Cost	Cost of chosen solution	High	Low	Low	Low	
Portability	Architecture / Network Topology	High	High	High	Low	

CHAPTER – 6

SECURITY ENGINEERING TOOL FOR IoT SYSTEMS

In this chapter of the thesis we will discuss implementation of a security engineering tool for IoT systems. It is based on the new security methodology proposed in this thesis. It can be used for any IoT system with or without minor modifications. However in this chapter we will show how it can be used with respect to the IoT based automotive vehicle tracking and control system that was discussed in Chapter 4.

6.1 INTRODUCTION

An IoT based automotive vehicle tracking and control system was shown in Chapter 4, Figure 4.1. The tool is made using Java Swing with JDK 1.8 on a Windows PC. It consists of eleven tabs each for specific stage of Security Engineering. Tab 7 labelled as "-----SRE RESULT-----" shows the results of Security Requirements Phase i.e. the prioritized security requirements. Tab 11 labelled as "-----SDE RESULT-----" shows the results of the Security Design Phase showing the selected algorithm that best suits the system. For convenience the term "SRE" will be used for Security Requirements Engineering & "SDE" for Security Design Engineering. Starting screen of the tool is as shown in Figure 6.1.

Figure 6.1: Starting screen of the developed tool

6.2 WORKING OF THE SECURITY ENGINEERING TOOL

6.2.1 Security Requirements Phase

Steps involved in Security Requirements engineering phase are first considered in the tool and Tabs 1 – Tabs 7 takes care of all the security requirements engineering steps.

Tab 1 is labelled as "SRE Step 1 - Elicitation". Identification of Actors, Assets, Vulnerabilities, Threats and Security Requirements is done as part of security requirements elicitation in this Tab. Figure 6.2 shows this tab of the Tool.

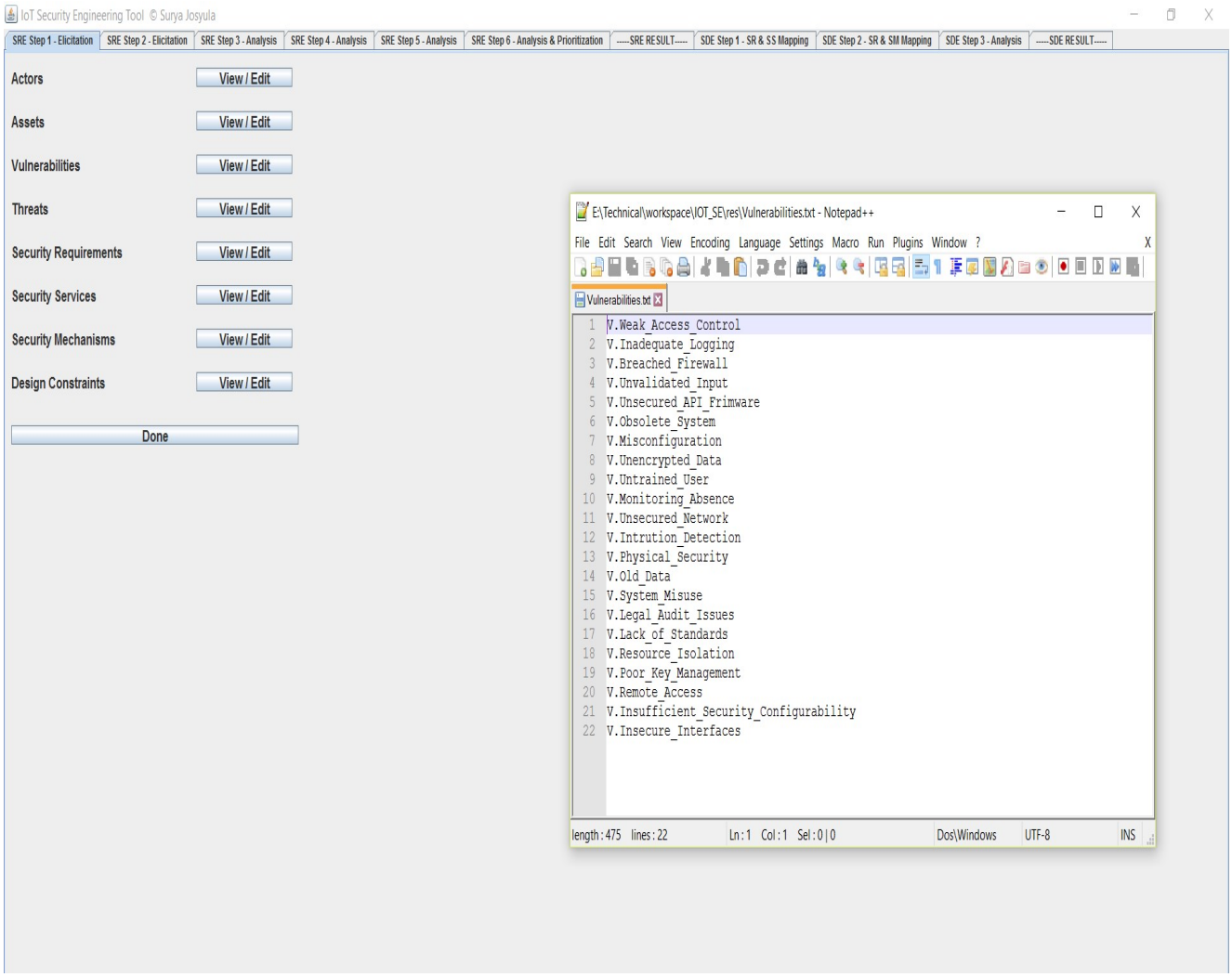
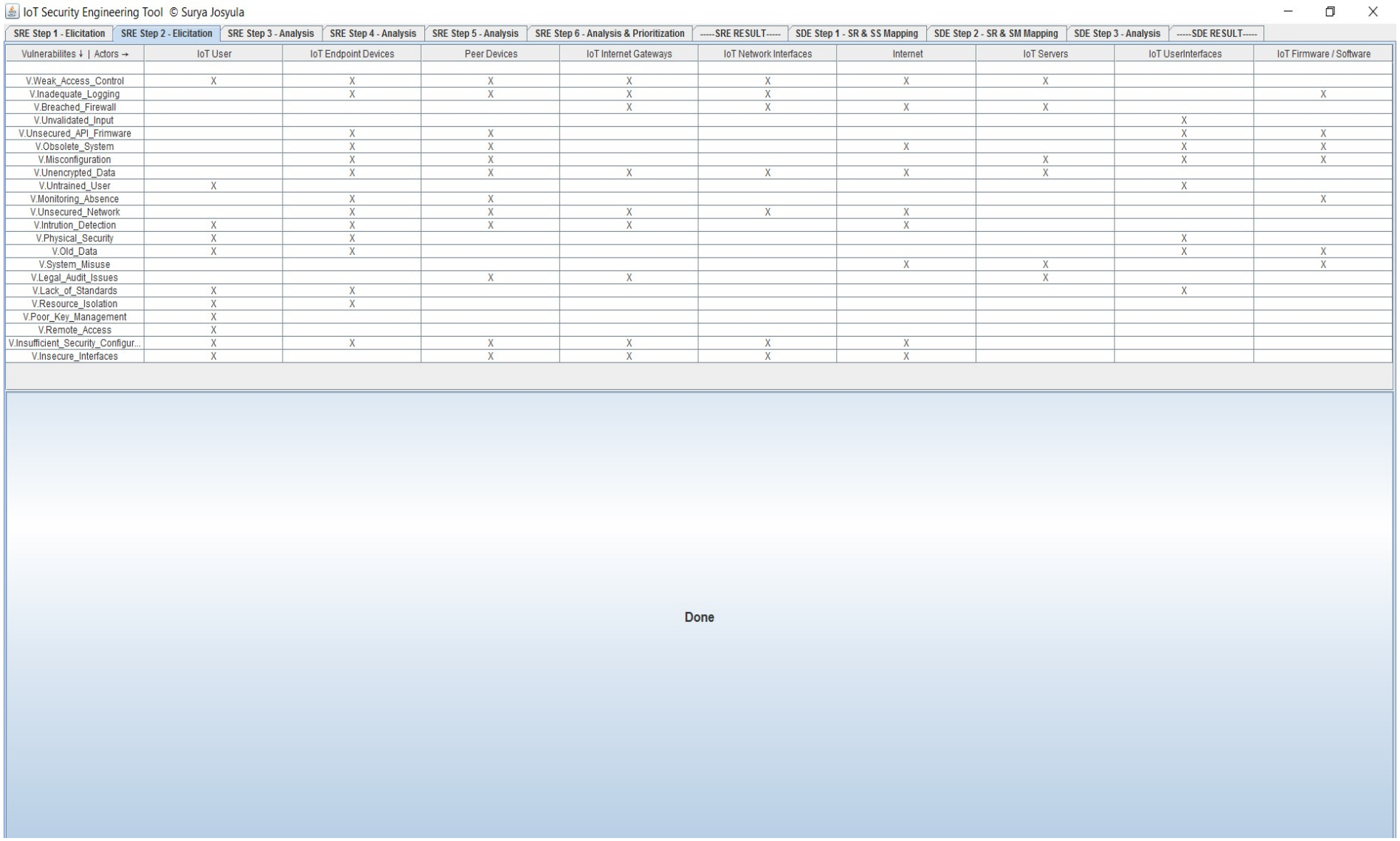


Figure 6.2: Tab 1 - SRE Step 1 – Elicitation

Tab 2 is labelled as “SRE Step 2 - Elicitation”. Figure 6.3 shows this tab of the Tool.

Vulnerabilities of actors are identified as a part of security requirements elicitation in this Tab.



Vulnerabilites ↓ | Actors →	IoT User	IoT Endpoint Devices	Peer Devices	IoT Internet Gateways	IoT Network Interfaces	Internet	IoT Servers	IoT UserInterfaces	IoT Firmware / Software
V.Weak_Access_Control	X	X	X	X	X	X	X		
V.Inadequate_Logging		X	X	X	X				X
V.Breached_Firewall				X	X	X	X		
V.Unvalidated_Input								X	
V.Unsecured_API_Frimware		X	X					X	X
V.Obsolete_System		X	X			X		X	X
V.Misconfiguration		X	X				X	X	X
V.Unencrypted_Data		X	X	X	X	X	X		
V.Untrained_User	X							X	
V.Monitoring_Absence		X	X						X
V.Unsecured_Network		X	X	X	X	X			
V.Intrution_Detection	X	X	X	X		X			
V.Physical_Security	X	X						X	
V.Old_Data	X	X						X	X
V.System_Misuse						X	X		X
V.Legal_Audit_Issues			X	X			X		
V.Lack_of_Standards	X	X						X	
V.Resource_Isolation	X	X							
V.Poor_Key_Management	X								
V.Remote_Access	X								
V.Insufficient_Security_Configur...	X	X	X	X	X	X			
V.Insecure_Interfaces	X		X	X	X	X			

Figure 6.3: Tab 2 - SRE Step 2 - Elicitation

Tab 3 labelled as “SRE Step 3 - Analysis”. Figure 6.4 shows this tab of the Tool.

Threats corresponding to Vulnerabilities are identified as a part of security requirements analysis in this Tab.

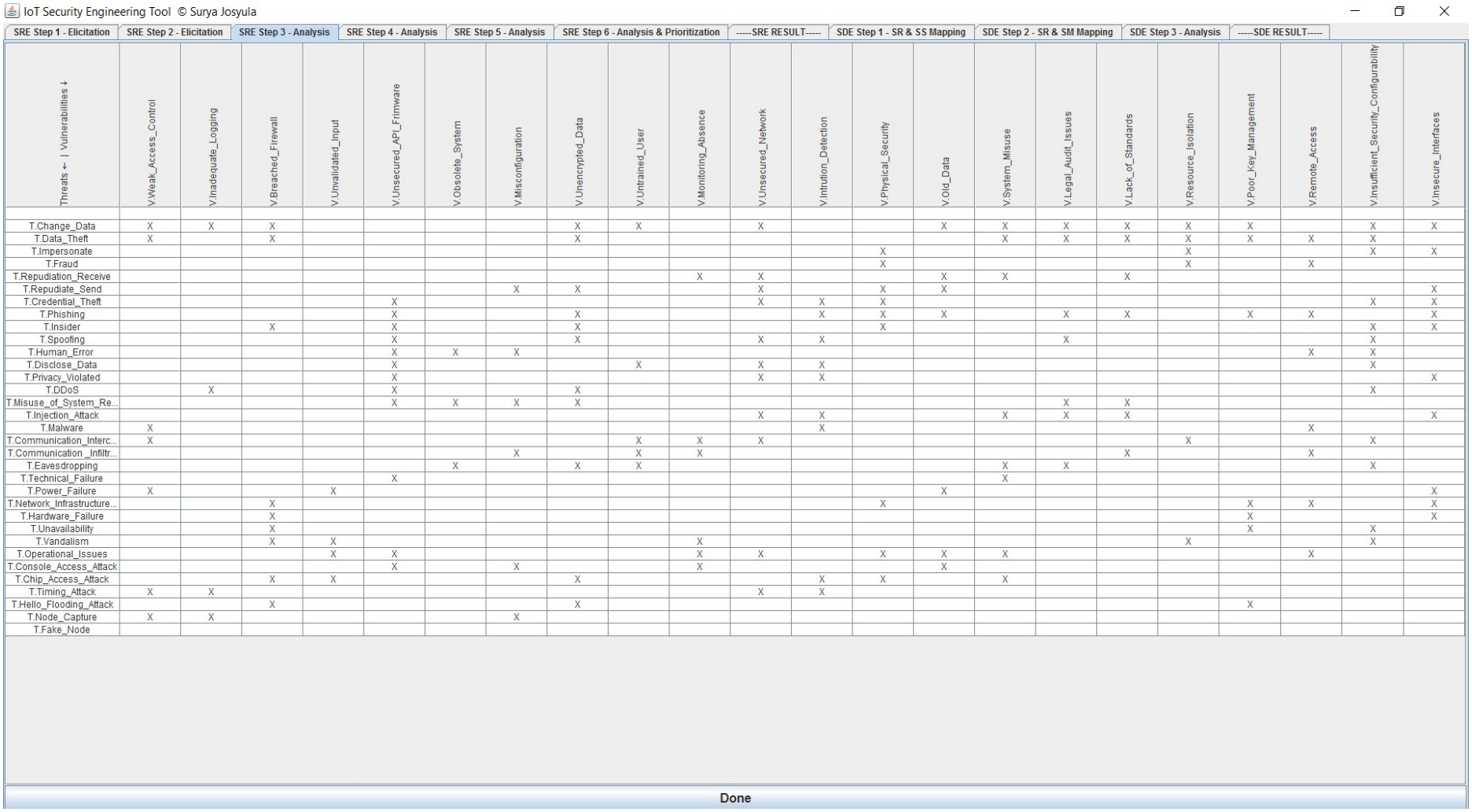
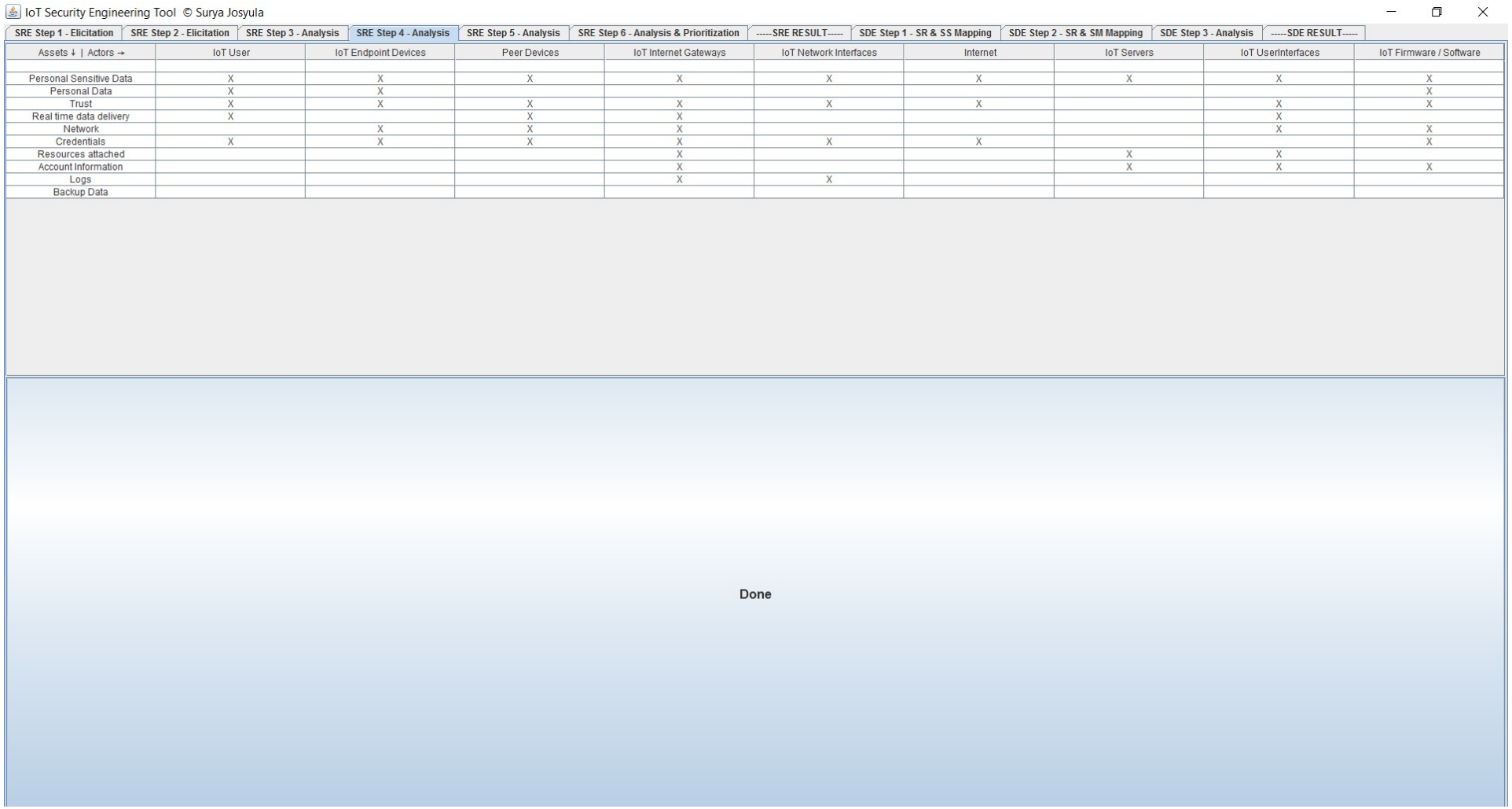


Figure 6.4: Tab 3 - SRE Step 3 - Analysis

Tab 4 labelled as “SRE Step 4 - Analysis”. Figure 6.5 shows this tab of the Tool.

Assets corresponding to Actors are mapped as a part of security requirements analysis in this tab.

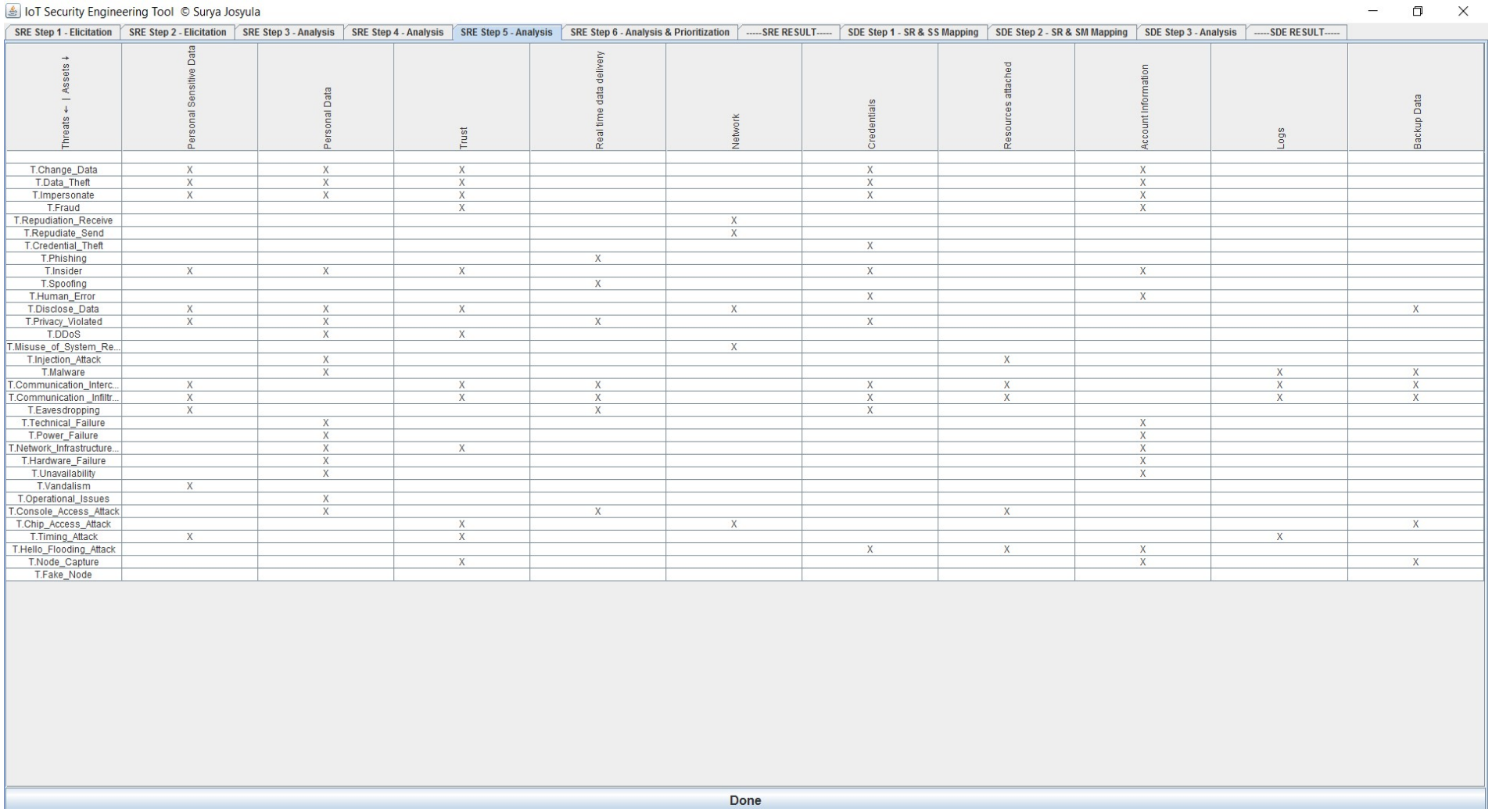


Assets ↓ | Actors →	IoT User	IoT Endpoint Devices	Peer Devices	IoT Internet Gateways	IoT Network Interfaces	Internet	IoT Servers	IoT UserInterfaces	IoT Firmware / Software
Personal Sensitive Data	X	X	X	X	X	X	X	X	X
Personal Data	X	X							X
Trust	X	X	X	X	X	X		X	X
Real time data delivery	X		X	X				X	
Network		X	X	X				X	X
Credentials	X	X	X	X	X	X			X
Resources attached				X			X	X	
Account Information				X			X	X	X
Logs				X	X				
Backup Data									

Figure 6.5: Tab 4 - SRE Step 4 - Analysis

Tab 5 is labelled as “SRE Step 5 - Analysis”. Figure 6.6 shows this tab of the Tool.

Threats corresponding to Assets are mapped as a part of security requirements analysis in this tab. It is used for calculating Asset rating.



Threats ← | Assets →	Personal Sensitive Data	Personal Data	Trust	Real time data delivery	Network	Credentials	Resources attached	Account Information	Logs	Backup Data
T.Change_Data	X	X	X			X		X		
T.Data_Theft	X	X	X			X		X		
T.Impersonate	X	X	X			X		X		
T.Fraud			X					X		
T.Repudiation_Receive					X					
T.Repudiate_Send					X					
T.Credential_Theft						X				
T.Phishing				X						
T.Insider	X	X	X			X		X		
T.Spoofing				X						
T.Human_Error						X		X		
T.Disclose_Data	X	X	X		X					X
T.Privacy_Violated	X	X		X		X				
T.DDoS		X	X							
T.Misuse_of_System_Re...					X					
T.Injection_Attack		X					X			
T.Malware		X							X	X
T.Communication_Interc...	X		X	X		X	X		X	X
T.Communication_Infiltr...	X		X	X		X	X		X	X
T.Eavesdropping	X			X		X				
T.Technical_Failure		X						X		
T.Power_Failure		X						X		
T.Network_Infrastructure...		X	X					X		
T.Hardware_Failure		X						X		
T.Unavailability		X						X		
T.Vandalism	X									
T.Operational_Issues		X								
T.Console_Access_Attack		X		X			X			
T.Chip_Access_Attack			X		X					X
T.Timing_Attack	X		X						X	
T.Hello_Flooding_Attack						X	X	X		
T.Node_Capture			X					X		X
T.Fake_Node										

Figure 6.6: Tab 5 - SRE Step 5 - Analysis

Tab 6 is labelled as “SRE Step 6 - Analysis & Prioritization” in which threats corresponding to Security Requirements are mapped as a part of security requirements analysis. It is used for calculating Threat rating. Figure 6.7 shows this tab of the Tool.

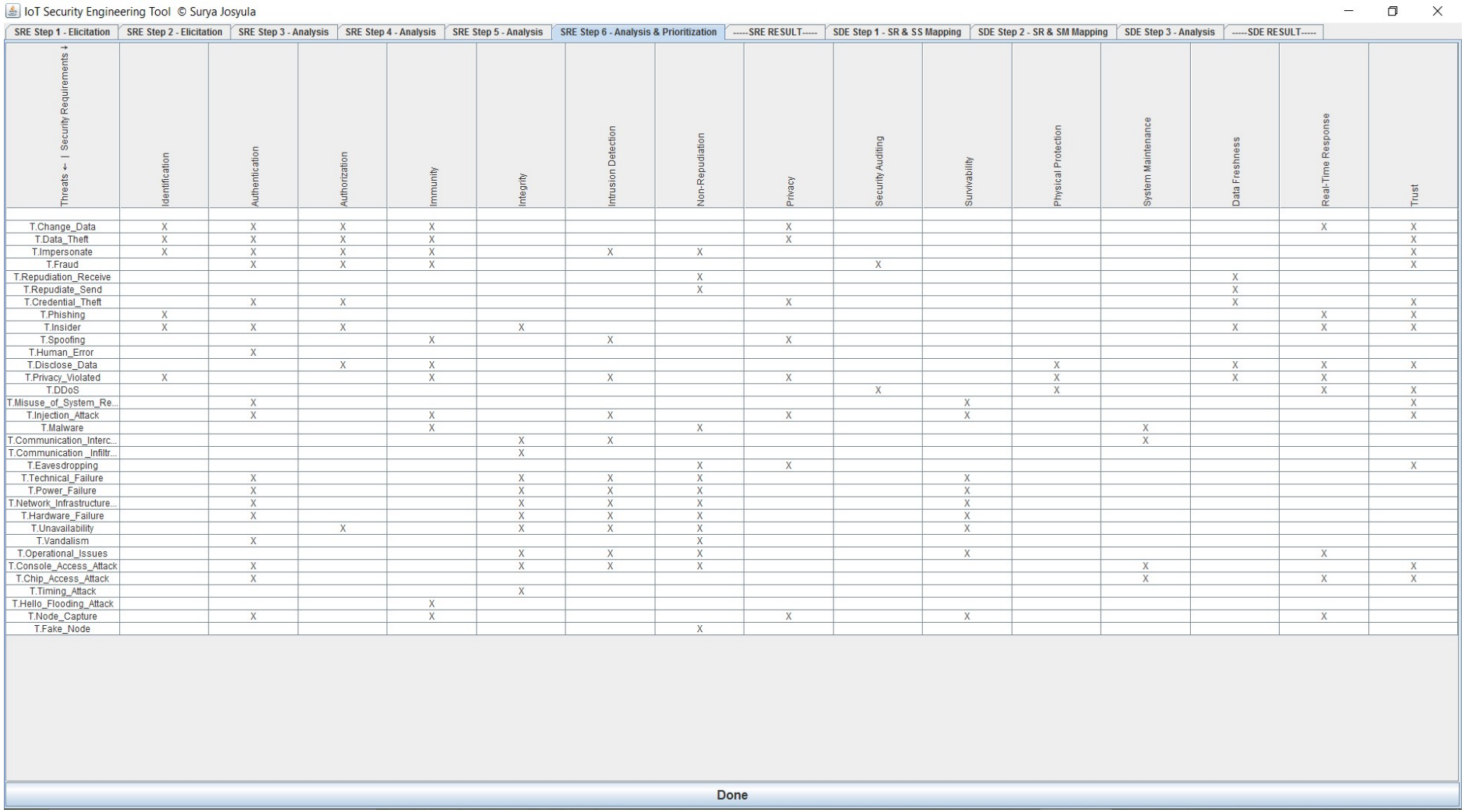
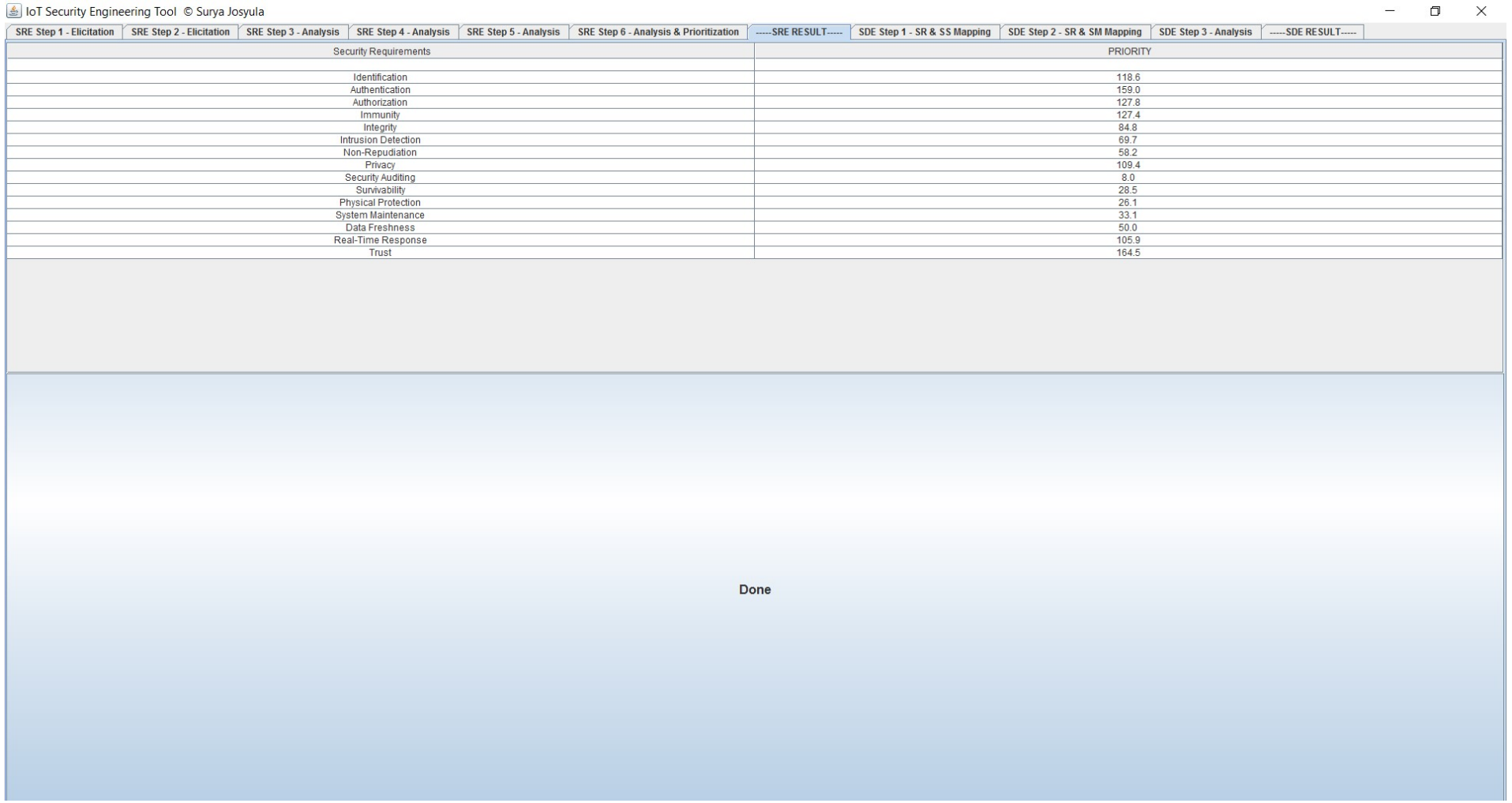


Figure 6.7: Tab 6 - SRE Step 6 - Analysis & Prioritization

Tab 7 labelled as "-----SRE RESULT-----" shows the result of SRE phase. It shows the prioritized security requirements.

Figure 6.8 shows this tab of the Tool.

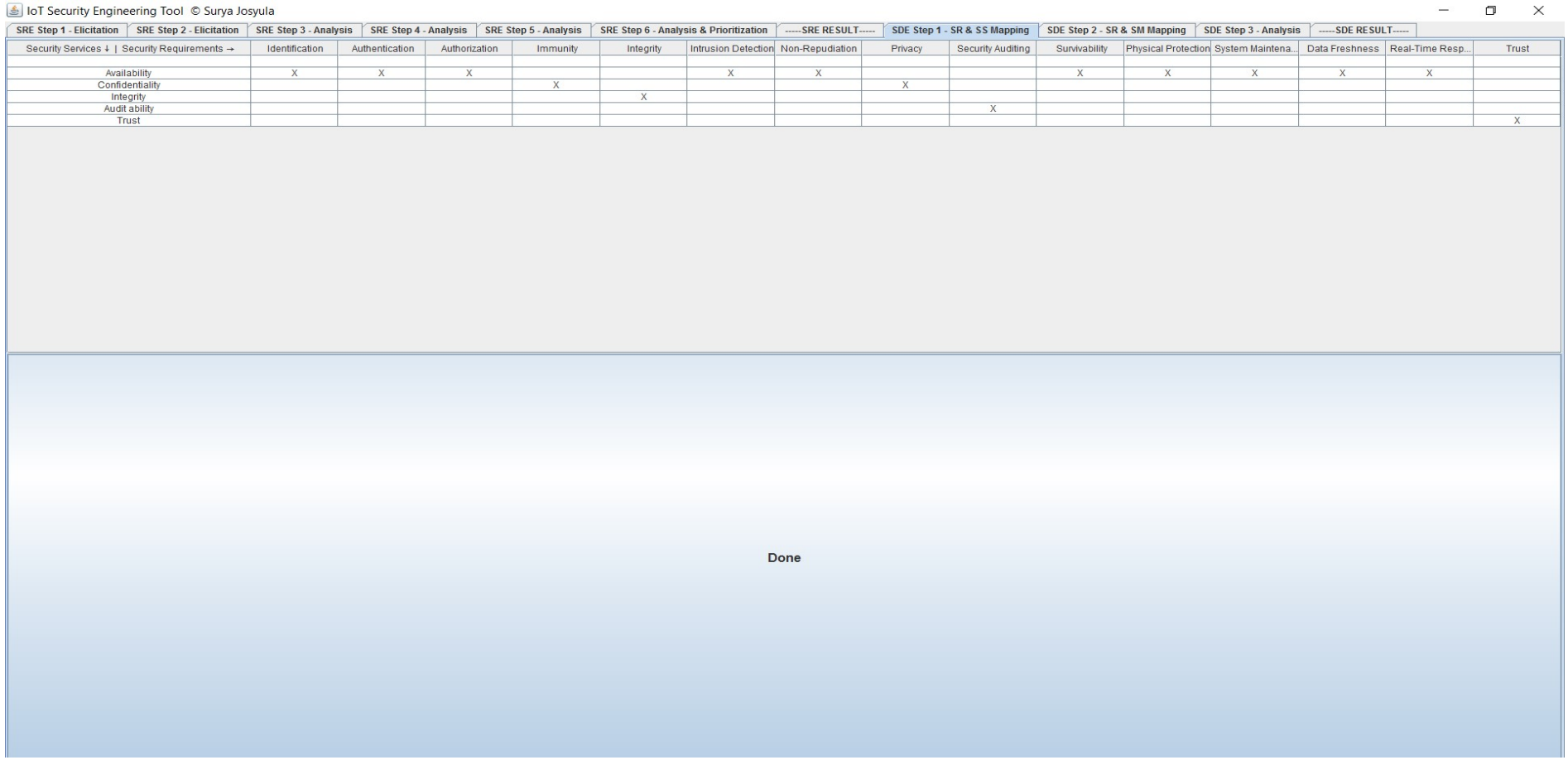


Security Requirements	PRIORITY
Identification	118.6
Authentication	159.0
Authorization	127.8
Immunity	127.4
Integrity	84.8
Intrusion Detection	69.7
Non-Repudiation	58.2
Privacy	109.4
Security Auditing	8.0
Survivability	28.5
Physical Protection	26.1
System Maintenance	33.1
Data Freshness	50.0
Real-Time Response	105.9
Trust	164.5

Figure 6.8: Tab 7 - SRE RESULT

6.2.2 Security Design Phase

Tabs from 8 to 11 are part of Security Design engineering phase. Tab 8 labelled as “SDE Step 1 - SR & SS Mapping” in which security requirements and security services are mapped as a part of security design phase. Figure 6.9 shows this tab of the Tool.



Security Services ↓ | Security Requirements →	Identification	Authentication	Authorization	Immunity	Integrity	Intrusion Detection	Non-Repudiation	Privacy	Security Auditing	Survivability	Physical Protection	System Maintena...	Data Freshness	Real-Time Resp...	Trust
Availability	X	X	X			X	X			X	X	X	X	X	
Confidentiality				X				X							
Integrity					X										
Audit ability									X						
Trust															X

Figure 6.9: Tab 8 - SDE Step 1 - SR & SS Mapping

Tab 9 is labelled as “SDE Step 2 - SR & SM Mapping” in which security requirements and security mechanisms are mapped as a part of security design phase. Figure 6.10 shows this tab of the Tool.

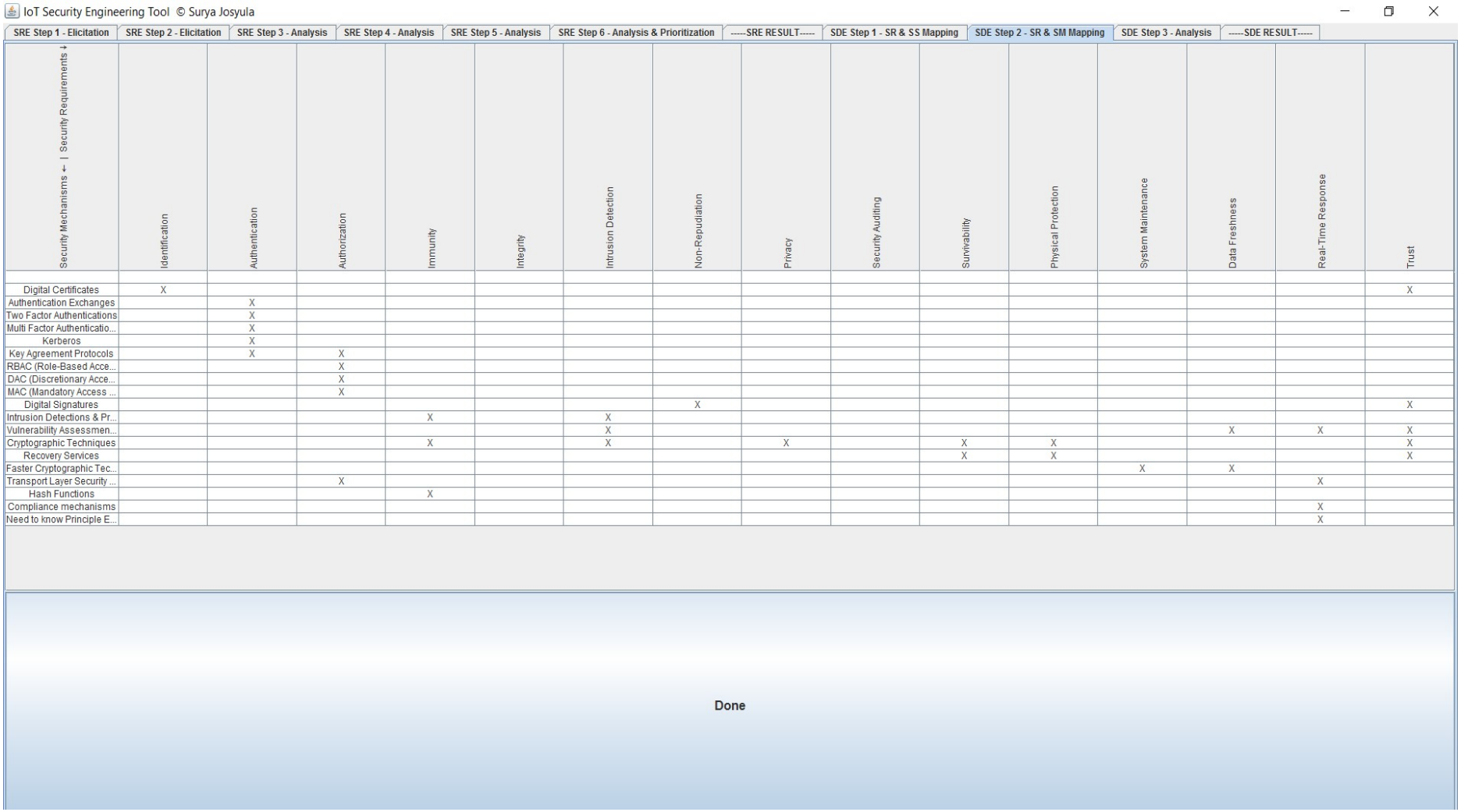


Figure 6.10: Tab 9 - SDE Step 2 - SR & SM Mapping

Tab 10 is labelled as “SDE Step 3 - Analysis” in which design constraints are identified and analysed. Figure 6.11 shows this tab of the Tool.

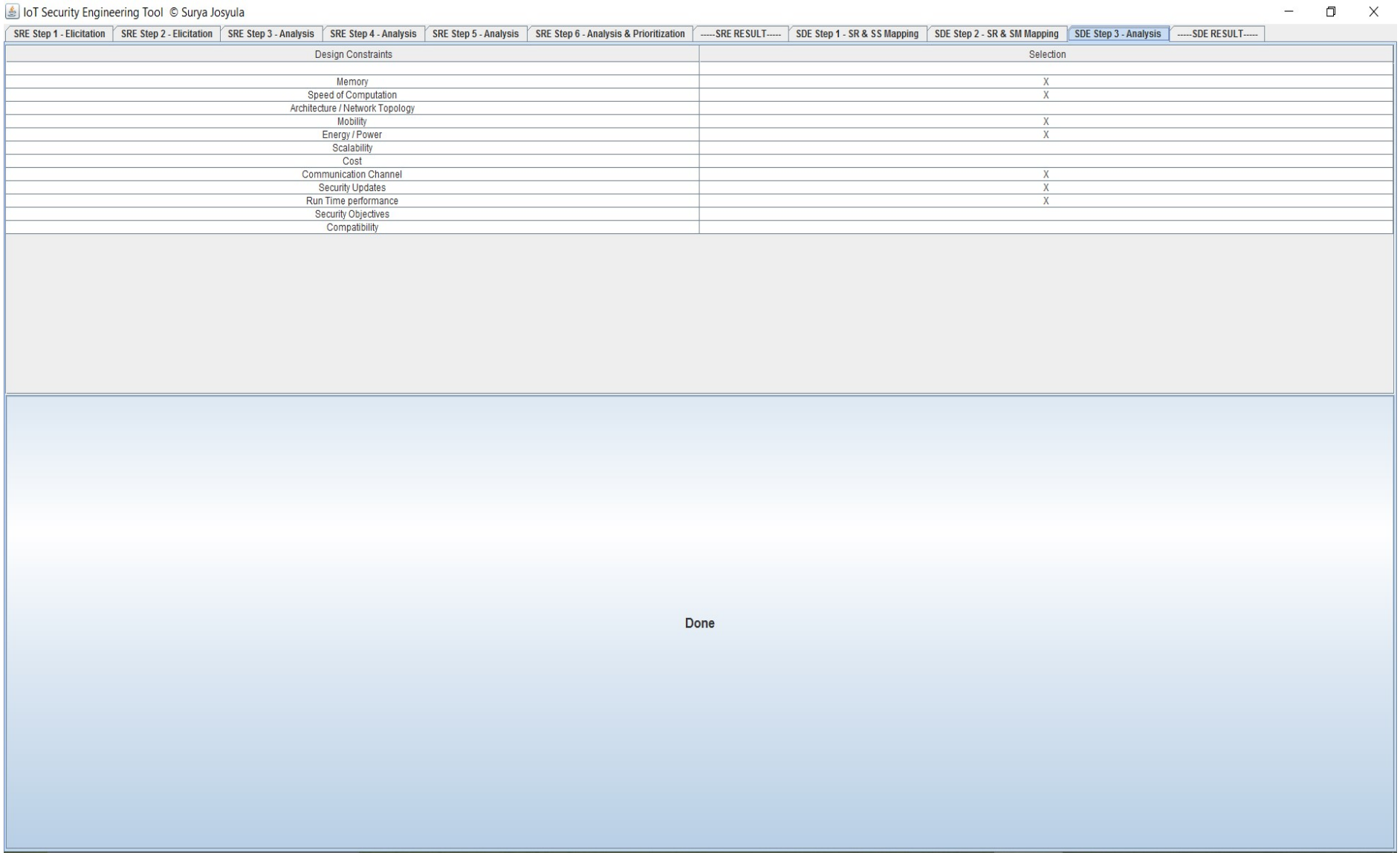


Figure 6.11: Tab 10 - SDE Step 3 - Analysis

Tab 11 labelled as "-----SDE RESULT-----" shows the result of SDE phase. It shows the selected algorithms that best suits the system. Figure 6.12 shows this tab of the Tool.

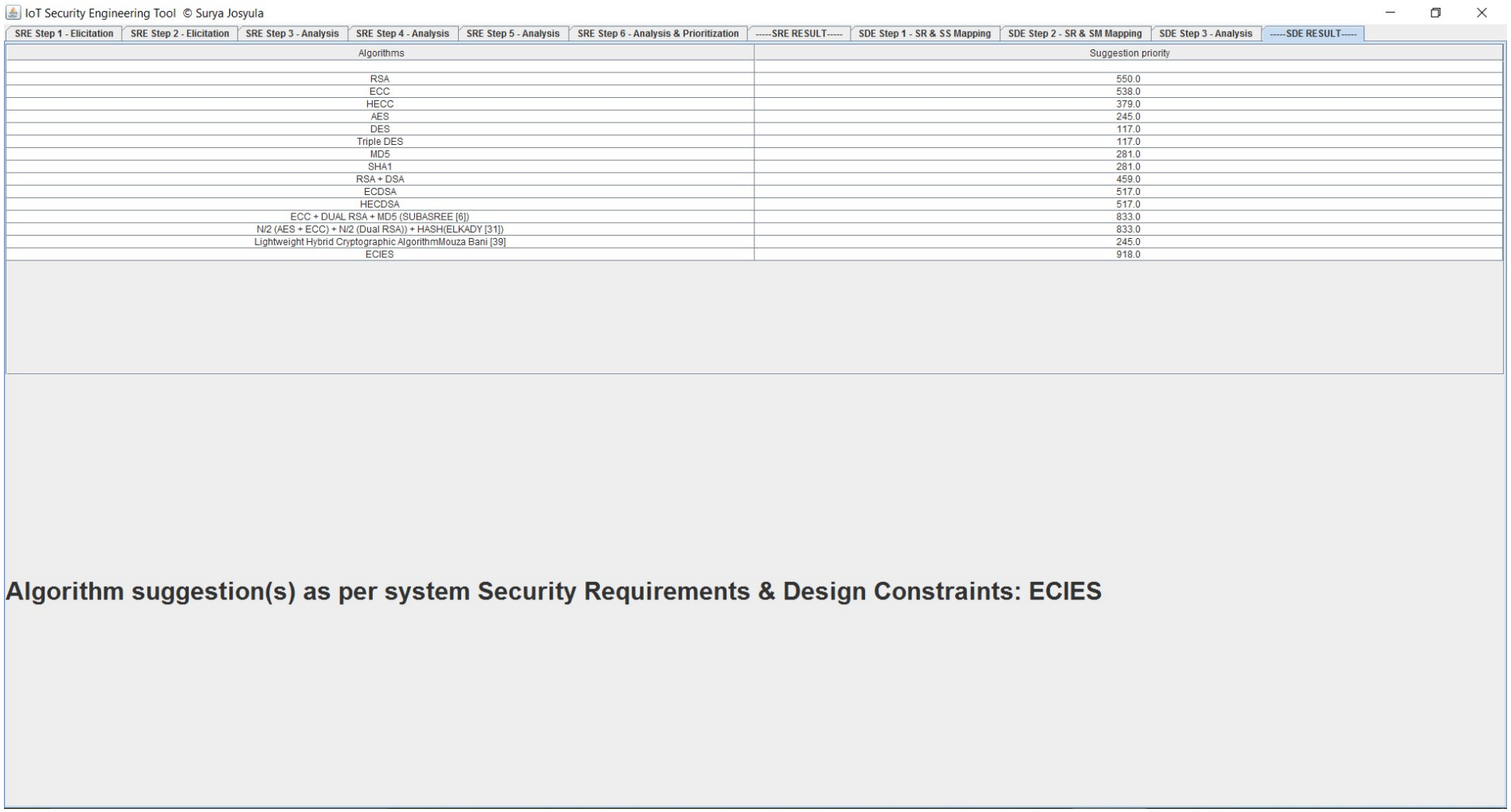


Figure 6.12: Tab 11 - SDE RESULT

CHAPTER – 7

CONCLUSIONS & FUTURE WORK

This chapter concludes this thesis work and provides insight into the future work.

7.1 CONCLUSIONS

In this project a new design methodology for securing an IoT system is introduced. The new methodology included proposal of a Security Engineering Framework specific to Internet of Things and suitable algorithms based on design decisions taken in the design stage.

Based on the well-defined steps present in the proposed Security Engineering Framework, security requirements elicitation is done by identifying stakeholders, their functional and non-functional requirements, identifying vulnerabilities, assets and all possible attacks on them and then they were mapped with security requirements. Then security requirement analysis was done by calculating risk using OWASP [29]. Further this work prioritized the attacks and threats.

In security design engineering, some security objectives were first identified namely Confidentiality, Integrity, Non-repudiation, Authentication, Message Privacy, Data freshness, Big Data and Trust to map security requirements. We then mapped prioritized attacks and their respective mitigation techniques with 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. Comparison of these cryptography algorithms against various attacks was done to identify best algorithm and combination of algorithms which were applicable. All design constraints were considered during the selection of these algorithms.

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 on an IoT based automotive vehicle tracking & control system. 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 new methodology to implement security in a system based on IoT provided certain ground breaking improvements and suggestions in the field of IoT security.

7.2 FUTURE WORK

- The tool made has further scope of improvement. It can include configurations of IoT systems which can be directly selected to check security. It will be part of future work.

- More case studies of IoT systems will be explored in future to verify whether our new methodology can be applied generically in IoT systems without any modifications.

- We will make a security requirements specification format which will contain all the security requirements and related design information just like a Software Requirements Specification (SRS).

- Optimization of proposed security algorithms in IoT.

- Active self detection and run-time mitigation of security attacks using machine learning algorithms.

CHAPTER – 8

REFERENCES

[1] Security for the Internet of Things: A Survey of Existing Protocols and Open Research issues, Jorge Granjal – 2015

[2] IoT Wikipedia, https://en.wikipedia.org/wiki/Internet_of_things

[3] Firesmith, D.G., Engineering Security Requirements, (2003), Journal of Object Technology, 2(1), pp.53-68

[4] A Mixed Encryption Algorithm Used in Internet of Things Security Transmission System, IEEE, 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery

[5] Performance Evaluation Of New Hybrid Encryption Algorithms To Be Used For Mobile Cloud Computing, Hatem M. Abdul Kader, 2014, international journal of technology enhancements and emerging engineering research, vol 2, issue 4

[6] Design of a new security protocol using hybrid cryptography algorithms, 2010, ijrras, subasree & sakthivel

[7] A Survey on Application Layer Protocols for the Internet of Things- 2015

[8] Internet of Things in Industries: A Survey, Li Da Xu (Senior Member, IEEE), Wu He, Shancang Li – 2015

[9] Middleware for Internet of Things: a study, Ghofrane Fersi – 2015

[10] The Emerging Internet of Things Marketplace From an Industrial Perspective: A Survey, Charith Perera – 2015

[11] https://www.iotivity.org/

[12] http://www.gsma.com/connectedliving/future-iot-networks/iot-security-guidelines/

[13] Luigi Catuogno, Stefano Turchi, "The dark side of the interconnection: security and privacy in the Web of Things", 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IEEE, 2015

[14] Kai Zhao, LinaGe, "A Survey on the Internet of Things Security", Ninth International Conference on Computational Intelligence and Security, 2013.

[15] http://www.goodworklabs.com/12-facts-worth-knowing-about-the-internet-of-things-or-iot/

[16] https://www.control4.com/blog/2014/03/the-internet-of-things-and-the-connected-home

[17] http://www.happiestminds.com/Insights/internet-of-things/

[18] http://www.ti.com/ww/en/internet_of_things/iot-applications.html

[19] http://blogs-images.forbes.com

[20] GSM Association http://www.gsma.com/connectedliving/future-iot-networks/iot-security-guidelines/

[21] "A framework for development of secure Software, Spinger ICT, 2013, K. Chatterjee & D. Gupta

[22] Jing Du, Ye Yang, Qing Wang, ―An analysis for understanding Software Security Requirement Methodologies‖, Third IEEE International Conference on Secure Software Integration and Reliability Improvement 2009

[23] Ashish Agarwal, Daya Gupta, ―Security Requirements Elicitation Using View Points for Online System, IEEE 2008

[24] A Survey Attacks on RPL and 6LoWPAN in IoT, Pavan Pongle, IEEE, 2015

[25] Sensing Services in Cloud-Centric Internet of Things: A Survey, taxonomy and challenges, Burak Kantarci – 2015

[26] "ITU Workshop on the Internet of Things - Trend and Challenges in Standardization", 2014

[27] "10th International Conference on Frontiers of Information Technology", Rafiullah Khan – 2012

[28] Internet of Things : Survey and Case Studies", Hetal B Pandya 2015

[29] OWASP, Open Web Application Security Project, https://www.owasp.org/

[30] Schneier B (1996) Applied cryptography, Wiley

[31] W. Ren, and Z. Miao, " A new security protocol using hybrid cryptography," In Proceedings of the 9th International Conference Computer Engineering Conference (ICEN-CO), 9th International, pp. 109-115, 2013.

[32] Microsoft, "Securing your Internet of Things from the ground up," [Online]. Available: http://download.microsoft.com/download/8/C/4/8C4DEF9B-041B-47F3-AD7F52F391B1D0AB/Securing_your_Internet_of_Things_from_the_ground_up_white_paper_EN_US.pdf

[33] D. Whitelegg, "Combating IoT cyber threats," 30 September 2015. [Online]. Available: https://www.ibm.com/developerworks/library/iot-security-best-practices-iot-apps/

[34] Kotonya G., Sommerville I., "Requirement Engineering with viewpoints", 1995

[35] Sommerville, I., "Software Engineering". Seventh edition 2003. ISBN - 8129708671. Pearson Education

[36] Design Methodology for Secure Cloud Systems, Jyotsna Sharma, 2015

[37] A framework for Security Requirements Engineering suitable for securing Big Data environments, Prudence, 2014

[38] A hybrid encryption algorithm in the application of equipment information management based on Internet of things, Peng Xu, Min Li, Yu-Jie He, Atlantis Press, 2013

[39] A New Lightweight Hybrid Cryptographic Algorithm for The Internet of Things, Mouza Bani, IEEE, 2012