A Framework for the incorporation and measurement of Security in IoT based Systems

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

ABHISHEK KUMAR VISHWAKARMA (Roll No. 2K15/SWT/503)

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)

DECLARATION

I hereby want to declare that the thesis entitled **"A Framework for the incorporation and measurement of Security in IoT based Systems."** 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.

ABHISHEK KUMAR VISHWAKARMA

Roll No. 2K15/SWT/503

Department of Computer Science & Engineering

Delhi Technological University

CERTIFICATE

This is to certify that the thesis entitled **"A Framework for the incorporation and measurement of Security in IoT based Systems."** submitted by **Mr. Abhishek kumar Vishwakarma (Roll No. 2K15/SWT/503),** 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 present 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 present 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.

Abhishek kumar Vishwakarma
(2K15/SWT/503)

Table of Contents

List of Tables:

List of Figures:

ABSTRACT

Internet of things is one of the widely researched topics these days. It is believed there will be billions of IOT devices connected to the internet in the near future with varied applications in varied fields. But with the increase in the number of devices, there will be far more security leaks possible. Security concerns are the biggest challenges in front of IOT systems. There are various limitations associated with IOT devices and connectivity constraints are also there such as less computation power, low memory, low battery storage, heterogeneous architectures, and limited communication bandwidth being mobile. As these system have several of the limitation of so the formal security concepts cannot be directly applied to these.

Chapter 1: Introduction

This chapter includes the background of the research, scope of work, approach and methodology of the research, and the covered topics of the thesis report.

1.1. About Internet of Things

The concept of the Internet of Things (IoT) is the extended usage of the internet with everyday objects. The objective of IoT is to operate these everyday objects remotely via the internet or network connectivity. To execute this setup the physical objects loaded with sensors, hardware, embedded software and internet connectivity to communicate with each other and with other network devices and ultimately with humans.

The phrase "Internet of Things" first coined by Kevin Ashton (Gabbai, 2015) in the year 1999 and still, there is a huge scope of new researches on this topic. There are three components of IoT are - hardware, software and communication/protocols. Hardware components which are sensors/devices which are connected directly to the objects which needs to be observed or controlled such as biochip transponders on farm animals, heart monitoring implants, automobiles with built-in sensors, and in many other devices for various business use cases. These devices collect useful information with the help of various technologies.

Though IoT was introduced in the year 1999 still it is unexplored area which can help in to improve the life style of the personal life, health care, and provide to improve various business operations and process to create more opportunities for the industries and people. IoT directly helps in to improve productivity, efficiency, and appropriate resource utilisation.

The key component of the IoT is communication/protocols, it contributes 40%, hardware and software contributes 30% each in the creation of any IoT system [https://internetofthingswiki.com/internet-of-things-definition]. As there is no separate communication system needs to be introduced and it works on low power consumption, so it reduces the financial implications greatly. As the market size of IoT globally by the year 2020 will be of €8.1Billion to sustain $14 Trillion this means more and more devices are connecting to the internet. This increases the challenges as well to implement the IoT, the key challenges are - security (37%), data privacy

(27%), access management (9%), external attacks (9%), hardware (6%) and others (12%).

Any function within the system can be studied and regulate using IoT for tracking, count, monitoring, actuate, and to control the cost and losses. We can easily find the faulty devices or component which needs overhauling, replacement which in turn increases the overall productivity and efficiency.

IoT is providing help to organisations, industries, and even common man by improving productivity, efficiency, and resource utilisation. By using IoT many tasks and operations can be performed remotely without risking human life or consuming the time on travel to provide the real insights, informed decision can be taken all of which will help in turn help in to take smart decisions.

As more and more things are connecting to internet is also increasing the possibilities of loss because of the security problems in the IoT via internet. To control security issues most of the researches are focused on software components whereas hardware component contributes 40% to the system which is critical.

As IoT is a complex system compare to the simple network so the security is also comparatively complex. It involves different kind of networks within single IoT systems such as sensor network, mobile network, cloud network and others. Due to the integration of these various network privacy, access control and management, heterogeneous authentication and others become more complex in IoT. Even the various components of the hardware are so integrated where the possibilities of intruding the network is easy. The last factor is that the devices are constrained devices and can be mobile. So this needs to be taken care so that the rest of the security breach can be controlled.

The focus area of this thesis is to provide proper framework for IoT system to take meaningful design decision for security feature implementation.

1.2. Related Works

There are a lot of challenges associated with IoT and a lot of research has been carried out in this field and many challenges are discussed here as given by various researchers.

In the IoT system researchers (Granjal, 2015) has provided the light on the currently present protocols and techniques available for secure communication. It also discusses the various layers of protocol stack with respect to IoT so that secure communication is also possible in IoT systems.

In another research paper (A Survey on Application Layer Protocols for the Internet of Things, 2015) research has been carried out to compare the existing application layer protocols and other protocols are also compared that provide end to end connectivity from user application to things based on the reliability, suitability and energy efficiency aspects of the IoT.

Researchers in (Li Da Xu, 2014) review the existing research work done in the field of IoT, important technologies, industries applications and identify some kind of trends & challenges. Identified challenges are design of service-oriented architecture, interoperability, scalability, big data management, heterogeneity management, data mining, standardization, Security and Privacy. Also (GhofraneFersi, 2015) reviews main challenges facing IoT middleware as Modularity, Trust, Mobility, Heterogeneity, Bootstrapping, Scalability, Spontaneous Events, Random topology, Interoperability, Actuation conflicts, Security & Privacy, Extensibility, Real-world integration, Unknown Data point availability.

Researchers in (CharithPerera, 2015) provided way forward for further researches in the field of IoT. It suggested some significant research directions and provided existing research details. It addressed that multi-protocol communication support, layered interoperability, sustainable business models, modularity, privacy, ownership and Security as the challenges in IoT research field. Research in (Kantarci, 2015) states various challenges of IoT, which are reliability, energy consumption optimization, awareness for mobile devices, big data management and context awareness.

Challenges in research in the field of IoT such as Security, Identification, Privacy protection, Interoperability, secure services for humans, Manageability, instant operations are referred in (ITU Workshop on the Internet of Things - Trend and Challenges in Standardization, 2014). Other research challenges were also identified such as objects security & safety, Identity and Naming management, data confidentiality & encryption, interoperability, privacy of information, standardization, Green IoT and network security are mentioned in (Khan, 2012).

Researchers in (Pandya, 2015) has provided a complete categorization of IoT based systems and then provided the further challenges in sensors technologies, protocol for communication, middleware challenges and Quality of Services challenges with IoT based systems.

Researchers in (Pongle, 2015) is has mainly concentrated on the various attacks that are possible in an IoT system with 6LoWPAN as well as RPL networks. They have also provided insight into the measures to avoid the attacks and provided research opportunities in the security of network layers.

Researchers in (Luigi Catuogno, 2015) has provided light in the field of security of IoT with regulation frameworks, protection of infrastructure and challenges such as trustfulness, data confidentiality, access control and privacy of data. Researchers in (Kai Zhao, 2013) give security issues of IoT such as Traffic Controls, access control, management of keys and security algorithms

(Gupta, 2013) in her research has given a security framework which provide the security for the software during the SDLC itself.
Security framework proposed (Gupta, 2013) has below mentioned phases :

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

It is presented in the paper that security can be incorporated into the software development life cycle and security framework can run parallel to it. There are several sub-phases proposed for the above phases which are shown in Figure 1-1.

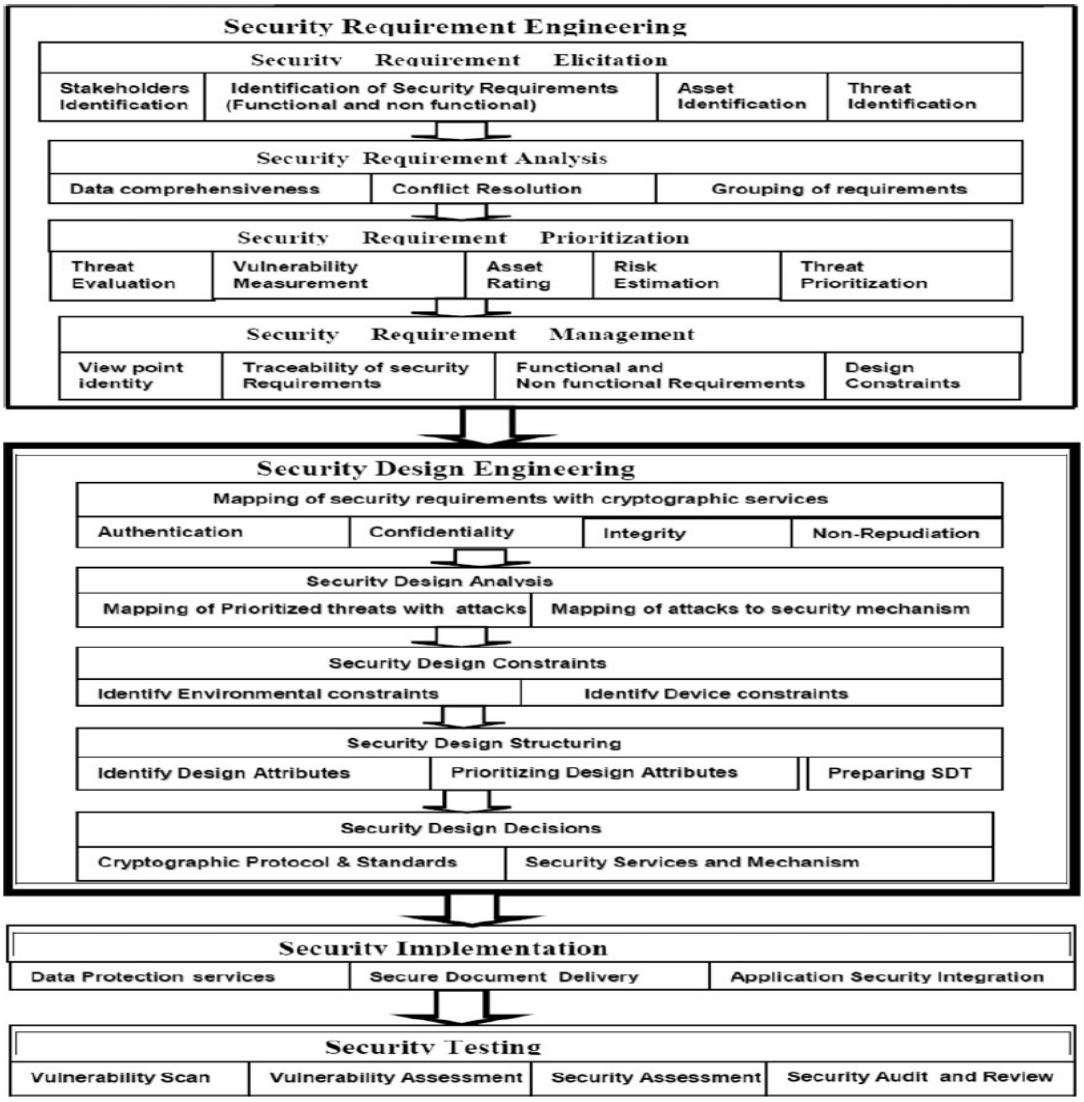

Figure 1-1 Security Engineering Framework for Software development (Gupta, 2013)

(Firesmith, 2003) in his work provides the security requirements as high level requirements which a system must fulfil in order to make a system highly secure. He considered various security objectives like the following:

Based on these security objectives following Security Requirements were described:

"

- Identification Requirements
- Authentication Requirements
- Authorization Requirements
- Immunity Requirements
- Integrity Requirements
- Intrusion Detection Requirements
- Nonrepudiation Requirements
- Privacy Requirements
- Security Auditing Requirements
- System Maintenance Security Requirements
- Survivability Requirements
- Physical Protection Requirements

(S. K. Josyula, 2017) has previously provided a security engineering framework for IoT system. But it doesn't provide anything for the security testing phase. Thus, in this thesis a complete security framework is proposed which provide detailed steps for each phase including security testing phase.

The different methodologies for implementing security mechanisms in IoT are discussed below:

- **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 1-3 shows some popular symmetric ciphers. The Table has been prepared from various sources such as (Nadeem, 2005) and (Tamimi, n.d.)

Table 1-1 Some popular symmetric ciphers (Gupta, 2013), (B, 1996)

Algorithm	Key size(bits)	Block size(bits)	Encryption speed (kb/s)
Rijndael	128	256	61
Blowfish	64	128	182
AES	128	128	60
3DES	64	168	12
(3DES)DES-XEX3	128	128	20(mb/s)
DES	64	56	35
CAST	64	128	53
RC5	64	128	86
RC4	1 byte	256	164
PIKE	1 byte	160	62
SEAL	1 byte	160	381

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 1-4 shows some popular asymmetric ciphers.

Algorithm	File size(Kb)	Encryption(sec)	Decryption(sec)
RSA	68	0.4	25
	105	0.7	50
	124	0.9	54
	235	1.5	95
ElGamal	68	1	2
	105	1.3	5
	124	0.1	6
	235	3	8
Paillier	68	0.3	60
	105	0.5	65
	124	0.5	140
	235	2.7	360

Table 1-2 Performance comparison of some popular asymmetric algorithms (SHAHZADI FARAH, 2012)

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 1-5 shows some popular Hash Functions.

Algorithm	Hash size(Kb)	Encryption speed(kb/sec)
MD4	128	23
MD5	128	236
HAVAL	128	174
N-HASH	128	29
SHA1	160	75
SHA2	160	70

Table 1-3 Some popular Hash Functions (Gupta, 2013), (B, 1996)

Hybrid Algorithms

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

1. It consists of work from (sakthivel, 2010) in which algorithms like ECC + Dual RSA + MD5 are used to make a hybrid cryptographic algorithm

2. Elkady (W. Ren, 2013) 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.

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

4. A Mixed Encryption Algorithm (A Mixed Encryption Algorithm Used in Internet of Things Security Transmission System, 2015) 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.

5. A New Lightweight Hybrid Cryptographic Algorithm (MouzaBani, 2012) 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.

- 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.

- 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.

1.3. Problem Statement for the work in this thesis

Internet of things is one of the fastest growing technology and rapid adoption among industries and masses gives a dire need for researches in the field of IoT security. From the literature survey it is clear that the security aspect is of the utmost important in case of IoT systems.

Thus, in this thesis we have provided an end to end security framework. This thesis provides detailed steps to find the security requirements of IoT based systems and then these requirements are analysed in the security design phase to find out the security mechanisms to be implemented and then these mechanisms are validated in the next step by calculating the Security Index.

1.4. Scope of Work

This work is about incorporation of security into IoT systems and to mitigate all possible threats. Below points provides a detailed scope followed throughout the work:

a) Security requirements are identified and mapped with corresponding threats and vulnerabilities present in the system which can harm the system assets. Security requirements are then prioritized based on the importance it serves in order of the threats mitigated once these requirements are implemented
b) Mapping of cryptographic services or algorithms is done with the security requirements, and then afterwards security design analysis and security design structuring is done based on constraints of the system.
c) We calculate the effectiveness of the security algorithms chosen and check if the system is under safe state after the implementation of chosen security services. If the system is still unsafe we review our decisions for choosing security services.

1.5. Working Approach

In this work firstly we have identified the current problems in IoT security as detailed in section 1.3. Based on this we have proposed an end to end framework for the incorporation of security into IoT systems. For this we have to identify the security requirements of an IoT based system.

The security Engineering framework for IoT provided by (S. K. Josyula, 2017) listed Security requirements such as Identification, Authentication, Authorization, Immunity, Integrity, Intrusion Detection, Non-Repudiation, Privacy, Security Auditing, Survivability, Physical Protection, System Maintenance, Real time response, Data freshness and trust. We have identified some more security requirements as per case study of IoT based electric metering system discussed in section 2.1. These security requirements are given below:

- **Firmware Security**

IoT devices manufactured through unsecured manufacturing processes gives criminals opportunities to change production runs, to introduce unauthorized code or produce additional units that can be sold on black market. One way to firmware

security processes is to use hardware security modules (HSMs) and supporting security software to inject cryptographic keys and digital certificates and to control the number of units built and the code incorporated into each unit.

- **Traffic control**

Traffic Controls needs to be robust and secure to not allow sniffing of data. Routing protocols needs to be designed to handle more traffic as there would be billions of devices connected to the network which will be far greater in number than the total devices connected to internet today.

- **Standardization of IoT protocol stack**

Standardization of IoT protocols stack in physical and data link layers is also a security challenge as different vendors will provide different implementation for these layers which will create a chance for security loopholes in some vendor implementations.

- **Self-healing**

When an IoT node breaks the whole of the IoT system must have a mechanism to detect the node malfunction and provide some mechanism to communicate with the IoT service provider or user about the breakdown.

- **Secure communication between devices**

The communication between devices or things in IoT should also be secure and encrypted. The encryption algorithm needed to secure communication should require less computational power and storage as IoT devices are already constrained on these factors.

The above security requirements are then mapped with security/cryptographic mechanisms which can fulfil these requirements if implemented. The security mechanisms are taken from literature survey in section 1.2. These are symmetric key algorithms, asymmetric key algorithms, hashing algorithms, signing algorithms and hybrid algorithms. Based on the priority of security requirements and constraints a design decision is taken.

The design decision taken is then reviewed by security testing. A Security Index (SI) value is calculated for the based on the design decision based on threats mitigated and live threats that still remain in the system. As per our work we chose ECIES to be the

cryptographic mechanism to be implemented and worked out that the SI comes out to be 0.017 which is significantly lower than the assumed epsilon value of 1.5. Thus, after implementing ECIES algorithm it can be ascertained that the system is in safe state. Table 1-4 shows the generated report after the security testing phase in section 5.4, which tests the design decision of implementing the ECIES algorithm to mitigate security threats.

Table 1-4 Test report for design decision to implement ECIES

IOT based electric metering system	
Security Algo applied	ECIES
Threats Identified and risk measure	T.Change_Data T.Data_Theft T.Impersonate T.Fraud T.Repudiation_Receive T.Repudiate_Send T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Human_Error T.Disclose_Data T.Privacy_Violated T.DDoS T.Misuse_of_System_Resources T.Injection_Attack T.Malware T.Communication_Interception T.Communication _Infiltration T.Eavesdropping T.Technical_Failure T.Power_Failure T.Network_Infrastructure_Failure T.Hardware_Failure T.Unavailability T.Operational_Issues T.Console_Access_Attack T.Chip_Access_Attack T.Timing_Attack T.Hello_Flooding_Attack T.Fake_Node T. Node Capture T. Vandalism

Threats mitigated	T.Change_Data T.Data_Theft T.Impersonate T.Fraud T.Repudiation_Receive T.Repudiate_Send T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Human_Error T.Disclose_Data T.Privacy_Violated T.DDoS T.Misuse_of_System_Resources T.Injection_Attack T.Malware T.Communication_Interception T.Communication _Infiltration T.Eavesdropping T.Technical_Failure T.Power_Failure T.Network_Infrastructure_Failure T.Hardware_Failure T.Unavailability T.Operational_Issues T.Console_Access_Attack T.Chip_Access_Attack T.Timing_Attack T.Hello_Flooding_Attack T.Fake_Node
Threats not mitigated	T. Node Capture T. Vandalism
Result	SI=0.017, so system is secure
Remark	SI value can be 0 if the system employs all the security mechanisms

1.6. Organisation of Thesis

This section expresses the details of the chapters that follows:

Chapter 2 provides details of previously existing and then proposes a framework for the incorporation of security in IoT. It also discusses about phases involved in detail with respect to IoT.

Chapter 3 In this chapter firstly, an IOT based electric metering system is discussed as a case study for the application of our proposed security framework. Then all the steps involved in security requirements engineering phase of proposed framework are elaborated with respect to the proposed case study.

Chapter 4 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 electric metering system and control system described in previous chapter.

Chapter 5 In this chapter all the steps involved in the testing of design based on the security services implements for the security requirement is done and a test report is generated.

Chapter 6 gives the conclusion and future work

Chapter 7 provides the references used in this thesis

Chapter 2: SECURITY FRAMEWORK FOR IOT SYSTEMS

This chapter provides details of IoT based Electric metering system as case study taken in this thesis and based on this case study a security engineering framework is proposed for the IoT systems.

2.1. AN ELECTRIC METERING SYSTEM BASED ON IOT (Case study)

An IoT based electric metering system architecture is shown in Figure 2-1. It consists of the following components:

- **UI Devices**

  UI Devices such as smart phones, laptops, smart watches can be used by the IOT users to monitor the various information provided by the IOT devices and can take action based on that information.

- **Communication Medium**

  Communication medium can be anything such as Wi-Fi, 3G, 4G, and Ethernet which can be used for communication between Things in IOT and the cloud

- **IoT Cloud servers**

  A cloud server is a server hosting the server application for the IoT systems. The IoT things update the data on the cloud and the UI devices access the data from the cloud to show the user updates.

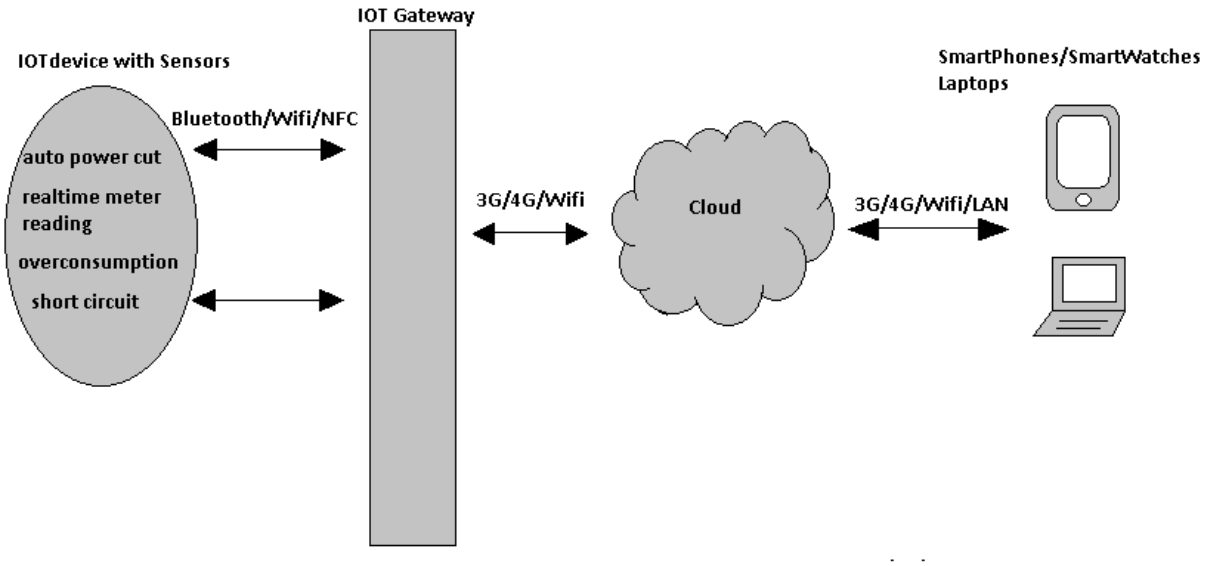

Figure 2-1 IoT based electric metering system

- **IoT Gateway**

  These acts as interfaces between the cloud and the IoT devices and can communicate with the cloud with various communication technologies available such as 3G/4G/Wifi etc.

- **IOT devices with Sensors**

  These are actual devices installed on the site. They contain sensors to sense the actual parameters values on real time basis required by the IOT user. For ex- In our electric metering system there can be sensors for meter reading that can provide realtime power consumption to user on their phones.

2.2. A PROPOSAL OF FRAMEWORK FOR SECURITY IN IOT SYSTEMS

The proposed framework for security in IoT Systems is an extension of the security framework discussed in section 1.2 and the security framework discussed for IoT in (S. K. Josyula, 2017).

The proposed framework has following phases:

1. Security Requirements Engineering For IoT systems:

In this phase security requirements that are applicable to IOT are specified, prioritized and afterwards validated upon.

2. Security Design Engineering for IoT systems:

In this phase mapping of cryptographic services or algorithms is done with the security requirements obtained for IoT systems in phase 1, and then afterwards security design analysis and security design structuring is done based on constraints of the system.

3. Security Testing for IoT systems:

In this phase we generate the test scenarios and then based on each functionality check the threats mitigated and live threats. Then we calculate a security index based on the threats mitigated and live threats and we create a test report. If the live threats are greater than a tolerable epsilon value then design decisions are reconsidered and we go back to security design phase.

4. Security Implementation:

Security mechanisms based on cryptographic algorithms are implemented along with documentation. This is out of the scope of this thesis and can be included in the future work.

2.2.1. **Security Requirements Engineering for IoT systems**

Security requirements are specified, analysed prioritized and validated in this phase. There are three stages as depicted in Figure 2-2 using which Security Requirements specification is generated:

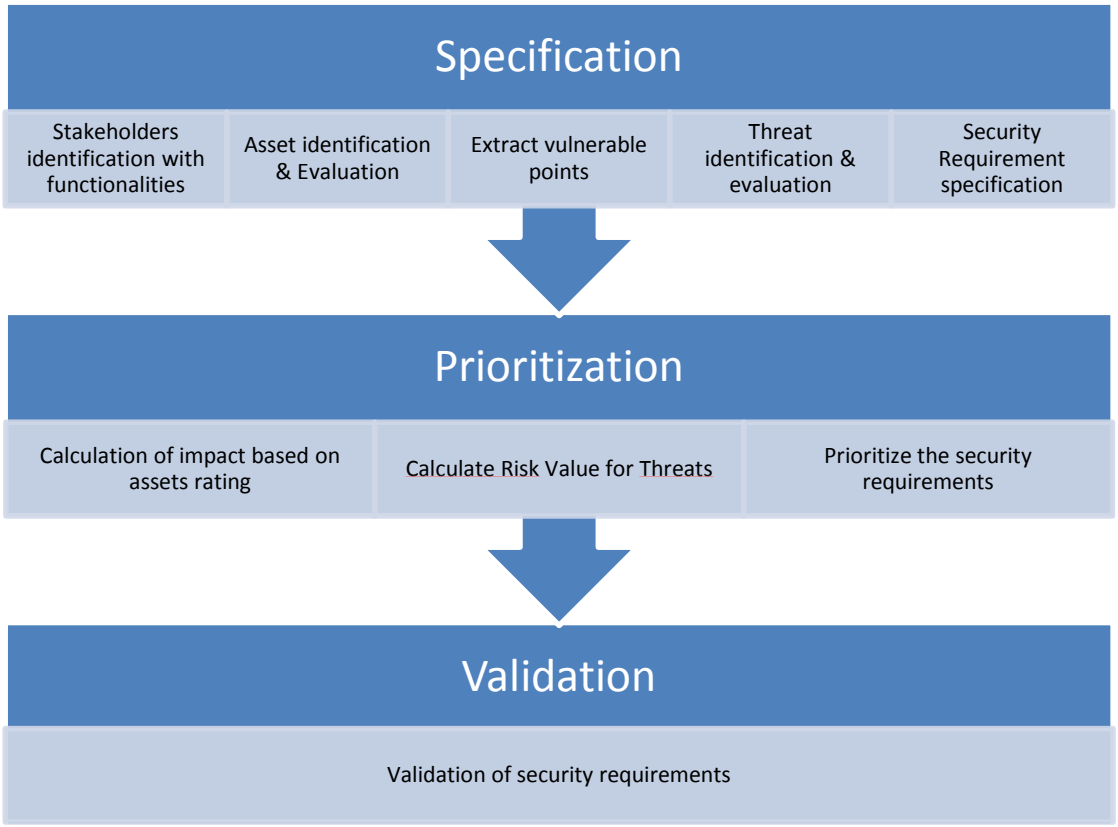

Figure 2-2 Security Requirements Engineering Process

"

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

- **Specification**

This stage involves specification of security requirements and using the requirements a threat mitigation plan is developed.

The steps are as discussed below:

- The security requirements are identified based on the case study
- Actors and stakeholders that are directly or indirectly involved in the system is done using view-point analysis (Kotonya G., 1996), (Sommerville, Seventh edition 2003). Humans, software system or hardware which is direct actors are identified. Software developer, administrators, regulators etc. are indirect actors and they are identified.
- Functionality requirement are elicited for all direct stakeholders
- Identification of assets associated with different functionalities and calculation of Asset value is done.
- Vulnerable points are extracted and a vulnerability rating is assigned
- Threats are identified and evaluated by assigning threat ratings.
- Security requirement specifications are then specified as a final step.

- **Prioritization**

A prioritization mechanism is designed for the security requirements identified in the specification step which is derived from the risk values. 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.

The major steps are discussed as below:

- An impact is calculated which is based on assets ratings of the assets
- Calculate the Risk value for Threats.

- Prioritize the Security Requirements according to the sum of risk values of threats that are being mitigated if the security requirement is implemented.

- **Validation**

Validation of security requirements are done to remove various loopholes in the them as sometimes there may be the case that some security requirements are even not needed, so in order to remove such scenarios validation is essential.

2.2.2. **Security Design Engineering for IoT systems**

Below design constraints are there for IOT system, which are considered in each step of the design phases:

- Low computation power
- Low memory & bandwidth
- Heterogeneous Architecture of the IOT systems.

In this part of software development life cycle software structure is designed to use the specifications of the system. Firstly mapping of security services with security requirements are done, then security analysis and security structuring is done. We'll consider the impact of all the factors like cryptographic techniques, coding standards and alternative connected available techniques that require to be followed. Security mechanisms are mapped for mitigating identified security requirements. During this phase, any bad call can result in design failure creating system liable to attacks.

Figure 2-3 shows the steps involved in security design engineering

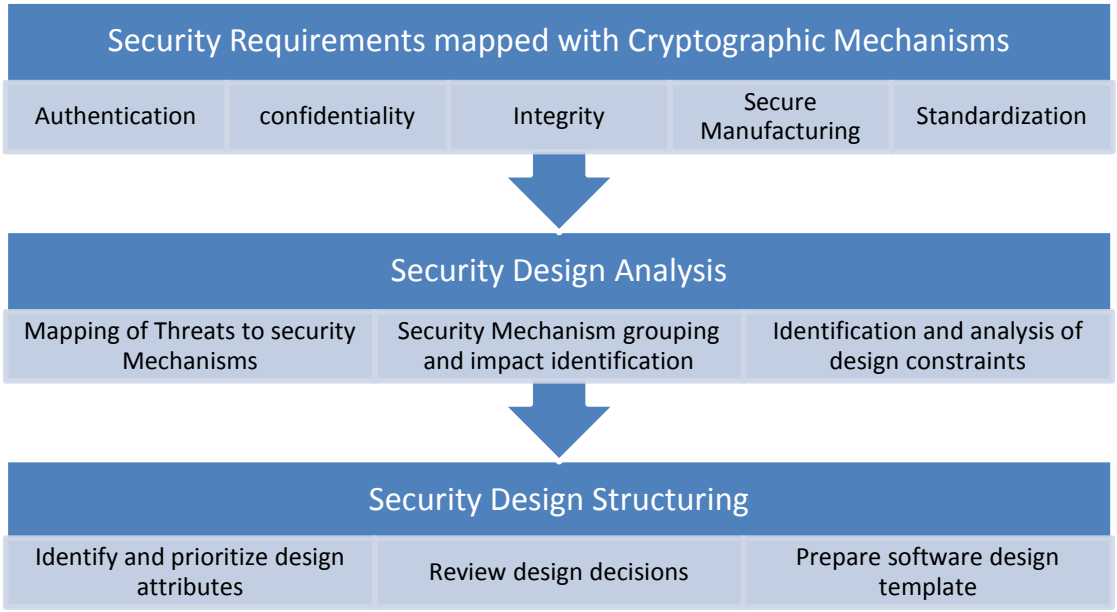

Figure 2-3 Security Design Engineering Process

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

- **Security requirements mapped with Cryptographic Mechanisms.**

In this, security requirements which were prioritized are mapped to known cryptographic services Security requirements like Non-repudiation, Integrity, Authentication, and Confidentiality etc., Real-time response, Trust and Data freshness. Firmware security, Traffic Controls, standardization, self-healing are also added in this thesis for IOT. This later helps in specifying and mapping security mechanisms for specific security requirements.

- **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

- **Security design structuring**

Design attributes are identified and prioritized in this stage.

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.

- **Security design decisions**

The output of the Security design phase is the security design decisions listed in the Security Design Template (SDT). Using previous knowledge best suitable mechanisms are selected based on the values of attributes in Security Design Template.

2.2.3. **Security Testing for IoT systems**

Security testing is evaluation of the security mechanisms implemented for the mitigating the threat in the system. The steps in this phase are shown in Figure 2-4

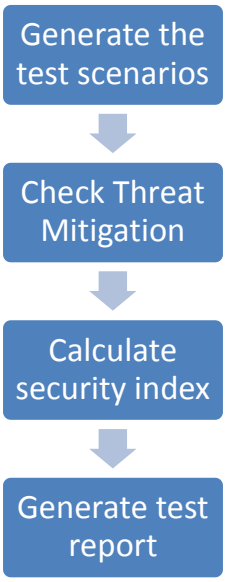

Figure 2-4 Security testing process

Following are the key stages of Security testing process:

- **Generate the Test Scenarios**

We need to generate the test scenarios in this stage by creating a sequence diagram. Scenarios are generated for all possible threats for all functionalities on the vulnerable points. We also mention the risk of each threat as well as assets which may be harmed.

- **Checking Threat Mitigation and Live Threats level**

Now checking is done for all the threats that will be mitigated if a particular cryptographic mechanism is implemented. The threats that are not removed from the system are called as live threats.
A vulnerability matrix is calculated correspondingly for each threat that is still live in the system, which shows the corresponding risk value.

- **Calculate the Security Index**

Then an index called as Security Index (Si) is calculated which infers the live vulnerabilities in the system. Security index can be calculated by the equation given as below as given by (Shruti Jaiswal, 2018), Security Index value 0 means no security lapses exist in the system.

$$SI = \frac{\sum_1^N Vi}{\sum_1^N Ri}$$

Where,

V_i is the vulnerability metric of all the active threat for all functionality F_i

R_i is the total risk value corresponding to functionality F_i

N is the number of Functionality considered

Now if (SI ≥ Epsilon)

Where Epsilon is the maximum tolerable risk value

Then the system is unsafe.

Else the system is in safe mode

- **Generate Test Report**

Then a testing report is created for the IOT system indication the summary of the overall Testing Phase. The template is taken from (Shruti Jaiswal, 2018) and test report is generated as per the results. The template has the following fields:

- System name under test
- Security Algorithms applied
- Threats identified with measure of risk
- Threats mitigated
- Live threats
- Results
- Remarks

Chapter 3: SECURITY REQUIREMENTS ENGINEERING FOR IOT Systems

In the chapter, all the steps involved in security requirements engineering phase of proposed framework are discussed based on the IOT system. Security requirements are specified, analysed, prioritized and validated along with the traditional requirements i.e. both functional and non-functional as per (Shruti Jaiswal, 2018).
The main steps in the process are given below:

- **Specification**
  - Security requirements are identified for the IoT based electric metering system taken as case study.
  - Direct as well as indirect actors or stakeholders are identified using view-point analysis (Kotonya G., 1996), (Sommerville, Seventh edition 2003) Humans, software system or hardware which is direct actors are identified. Software developer, administrators, regulators etc. are indirect actors and they are identified.
  - Functionality requirement are elicited for all direct stakeholders
  - Identification of assets associated with different functionalities and calculation of Asset value is done.
  - Vulnerable points are extracted and a vulnerability rating is assigned
  - Threats are identified and evaluated by assigning threat ratings.
  - Security requirement mapping with threats.

- **Prioritization**
  - Calculation of impact based on assets rating
  - Calculate the Risk value for Threats.
  - Prioritize the Security Requirements according to the risk values of threats that are being mitigated.

- **Validation**

- Validation of security requirements is done to remove various loopholes in them as sometimes there may be the case that some security requirements are even not needed, so in order to remove such scenarios validation is essential.

3.1. Specification

3.1.1. **Security Requirements Identification**

(Firesmith, 2003) gave the security requirements which have already been listed in section 1.2 of this thesis which are Identification, Authentication, Authorization, Immunity, Integrity, Intrusion Detection, Non-Repudiation, Privacy, Security Auditing, Survivability, Physical Protection, and System Maintenance. These security requirements are generic but mostly the IoT system also has the same set of security requirements. Security requirements such as Real time response, Data freshness and trust are also defined by (S. K. Josyula, 2017). We have identified some more security requirements as per case study of IoT based electric metering system discussed in section 2.1. These security requirements are already discussed in section 1.5 and they are:

- **Firmware Security**

IoT devices manufactured through unsecured manufacturing processes gives criminals opportunities to change production runs, to introduce unauthorized code or produce additional units that can be sold on black market. One way to firmware security processes is to use hardware security modules (HSMs) and supporting security software to inject cryptographic keys and digital certificates and to control the number of units built and the code incorporated into each unit.

- **Traffic control**

Traffic Controls needs to be robust and secure to not allow sniffing of data. Routing protocols needs to be designed to handle more traffic as there would be billions of devices connected to the network which will be far greater in number than the total devices connected to internet today.

- **Standardization of IoT protocol stack**

Standardization of IoT protocols stack in physical and data link layers is also a security challenge as different vendors will provide different implementation for these layers which will create a chance for security loopholes in some vendor implementations.

- **Self-healing**

When an IoT node breaks the whole of the IoT system must have a mechanism to detect the node malfunction and provide some mechanism to communicate with the IoT service provider or user about the breakdown.

- **Secure communication between devices**

The communication between devices or things in IoT should also be secure and encrypted. The encryption algorithm needed to secure communication should require less computational power and storage as IoT devices are already constrained on these factors.

3.1.2. Stakeholders identification with functionality required

An actor interacts directly or indirectly with the system. An actor which interacts directly is called a direct actor and the one who directs indirectly is called an indirect actor. Stakeholders are the actors that have some vested interest or who wish some output from the system or accountable for the things happening within it.

Viewpoint approach is followed to find stakeholders. Both direct actors and indirect actors are identified. Details of the actors / stakeholders identified for IoT based electric metering system shown in Figure 2-1 is as follows:

i) Direct stakeholders

- **IoT service Provider**

  IoT service provider provides the whole IoT based systems to facilitate IoT customer to use IoT services.

- **IoT Users**

IoT Customer are the real user of the IoT systems which they can use for monitoring or control purposes. Ex – In our electric metering system a person can monitor real-time power consumption.

- **IoT security administrator**

  IoT administrator is responsible for security of the IoT system.

ii) Indirect stakeholders

- **IoT devices with sensors**

  These are the actual IoT devices at the remote location with sensors installed on them which are responsible for providing real-time data.

- **Peer devices**

  These are peer device to the IoT device with sensor and it may or may not have internet connection so it can communicate with IoT device via Bluetooth, wi-fi, NFC, to send data to the internet.

- **IoT Gateways**

  IoT gateways acts as an interface between the internet and the IoT devices. A possible scenario can be that IoT devices connect doesn't have a direct internet connection and can connect to IoT gateways to connect to the internet.

- **Internet**

  Internet used for communication between IoT devices and IoT Users

- **IoT Cloud**

  IoT cloud servers are available to process request and store the data.

- **IoT UI Devices**

"

UI Devices such as smart phones, laptops, smart watches can be used by the IOT users to monitor the information provided by the IOT devices and can take action based on that information.

3.1.3. **Asset identification and Evaluation of Asset rating**

Assets can be defined as anything of value to the actors and can be identified using view-point analysis (Kotonya G., 1996), (Sommerville, Seventh edition 2003) and asset evaluation is done by determining the asset value for each asset.

Assets corresponding to actors identified in Section 1 are shown in Table 3-1.

In the work done by (Shruti Jaiswal, 2018) only direct actors were considered but in this work both the direct and indirect actors are considered for asset identification and evaluation of Asset rating.

Table 3-1 Identification of Assets

Actors	Functionality	Assets
IoT Users	1.Monitor data 2.Give command to IoT endpoint device	Credentials Personal Data Personal Sensitive Data Trust Service Delivery
IOT service provider	1.Provides the IoT device with sensor installed 2.Provides the UI based application for users	Trust Service Delivery Network Logs Intellectual Property Credentials

IOT security administrator	1.Provides all the security related features to the IOT users	Personal Sensitive Data Personal Data Trust Service Delivery Network Logs Intellectual Property Credentials Backup / Archive Data Account information
IOT devices with sensors	1.Data collection from sensors 2.controlling attached peripherals as per commands by IoT user 3.send/receive data to/from IoT Gateways	Personal Sensitive Data Personal Data Trust Service Delivery Network Logs Backup / Archive Data
Peer Devices	1.Data collection from sensors 2.controlling attached peripherals as per commands by IoT user 3.Communication with peer device	Personal Sensitive Data Personal Data Trust Service Delivery Network Logs Backup / Archive Data

IoT Gateways	1.Data transfer between IoT Devices with sensors and IoT UI devices	Network Service Delivery Logs
Internet	1.Data transfer medium	Network Service Delivery Logs
IoT Cloud	1.Run IOT application intermediatory to the IOT UI devices and IOT end point device	Resources attached Customer Data Account information
IoT UI devices	1. Take user command from to IOT servers 2. Provide IoT services to the user 3. Provide the data from IOT device to the user for monitoring.	Network Personal Sensitive Data Credentials

3.1.4. **Asset ratings are calculated**

Assets as described in the previous section, is something valuable to stakeholders. As different stakeholders can value assets differently, so we have drawn a table 3-2 where we have marked "I" for each asset which is important or of value to that user. So, asset value can be defined as the count of "I". Asset ratings mainly show that to how many actors an asset is valuable. Hence most valuable asset will have higher value of asset rating. E.g. Backup data has asset rating of 2 as it is important to only IoT Endpoint devices and Peer devices.

Table 3-2 Calculation of asset rating

Actors → ***↓ Assets***	***IoT User***	***IoT Service Provider***	***IoT security administrator***	***IoT devices with sensors***	***Peer Devices***	***IoT Gateways***	***Internet***	***IoT Cloud***	***IoT UI devices***	***Asset Rating***
Personal Sensitive Data	**I**	**I**	**I**	**I**	**I**	**I**	**I**	**I**	**I**	**9**

"

Personal Data	**I**			**I**					**I**	**3**
Trust	**I**		**I**	**I**	**I**	**I**	**I**	**I**		**7**
Real time data delivery	**I**	**I**		**I**		**I**		**I**	**I**	**6**
Network			**I**			**I**	**I**			**3**
Credentials	**I**	**I**	**I**	**I**	**I**	**I**	**I**	**I**	**I**	**9**
Resources attached		**I**		**I**				**I**		**3**
Account Information	**I**		**I**	**I**				**I**	**I**	**5**
Logs			**I**	**I**		**I**				**3**
Backup Data		**I**	**I**		**I**				**I**	**4**

3.1.5. **Vulnerability points identification**

Vulnerability points to the flaws that may exist in the system for the attackers to exploit as an entry point for an attack. Vulnerabilities can be anything from physical flaw to software bugs or security defects. Vulnerabilities are identified using the the sequence diagram drawn for each functionalities of actors for the IoT based electric metering system. The table 3-3 shows the vulnerabilities of all the functionalities of actors. Some vulnerabilities are referred from (Open Web Application Security Project), (Sharma, 2015), (Prudence, 2014). For convenience Vulnerabilities are prefixed with "V." throughout this thesis.

Table 3-3 Vulnerabilities related to Actors

Actors	Functionality	Vulnerabilities
IoT User	1.Monitor data 2.Give command to IoT endpoint device	V.Unencrypted_Data V.Untrained_Users V.Weak_Access_Control V.System_Misuse
IOT service provider	1.Provides the IoT device with sensor installed 2.Provides the UI based application for users	V.Weak_Access_Control V.Unencrypted_Data V.Breached_Firewall V.Inadequate_Logging V.Insecure_Interfaces

		aV.Insecure_Network_services V.Insufficient_Security_Configurability V.Legal_Audit V.Intrution_Detection
IOT security administrator	1.Provides all the security related features to the IOT users	V.Weak_Access_Control V.Unencrypted_Data V.Breached_Firewall V.Obsolete_System V.Insecure_Interfaces V.Insecure_Network_services V.Insufficient_Security_Configurability V.Weak_Access_Control V.Legal_Audit_Issues V.System_Misuse
Internet	1.Data transfer 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 devices with sensors	1.Data collection from sensors 2.controlling attached peripherals as per commands by IoT user 3.send/receive data to/from 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 Cloud	1.Run IOT application intermediatory to the IOT UI devices and IOT devices with sensors	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	1.Data collection from sensors 2.controlling attached peripherals as per	V.Weak_Access_Control V.Unencrypted_Data V.Monitoring_Absence V.Inadequate_Logging

	commands by IoT user 3.Communication with peer device	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 Gateways	1.Data transfer between IoT Devices with sensors and IoT UI devices	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 UI devices	1.Take user command from to IOT servers 2.Provide IoT services to the user 3.Provide data from IOT device to the user for	V.Untrained_Users V.Misconfigurations V.Unsecured_API_Firmware V.Obsolete_System V.Legal_Audit

	monitoring.	V.System_Misuse

3.1.6. **Threat identification & evaluation of Threat rating**

A threat is something that can harm the overall system as a whole or some of its operation. A threat can be caused to the system by making use of some of the vulnerability as an entry point. Threats are taken from references such as (Pongle, 2015), (Khan, 2012), (Sharma, 2015), (Prudence, 2014). Threats are mapped to the corresponding Vulnerability in Table 3-4. Mapping is done on the basis as if due to particular vulnerability a threat is possible then the corresponding element is marked with "X". Also Threat rating is calculated based on Table 3-4 which is the sum of "X" for a particular threat i.e. just count the number of "X" in the row for obtaining the threat rating for any threat. For convenience and easy distinction Threats are prefixed with “T.” throughout this thesis.

Table 3-4 Calculation of Threat rating

Vulnerability → / ***Threats →***	***V Weak Access Control***	***V Inadequate Logging***	***V Reached Firewall***	***V Unvalidated Input***	***V Unsecured API Firmware***	***V Obsolete System***	***V Misconfiguration***	***V Poor Key Management***	***V Unencrypted Data***	***V Untrained User***	***V Monitoring Absence***	***V Unsecured Network***	***V Lack of Standards***	***V Intrusion Detection***	***V Physical Security***	***V Old Data***	***V System Misuse***	***V Legal Audit Issues***	***V Resource Isolation***	***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**	**12**
T.Data_Theft	**X**		**X**					**X**	**X**				**X**					**X**			**X**	**X**	**8**
T.Impersonate	**X**																**X**			**X**			**3**
T.Fraud	**X**	**X**		**X**													**X**				**X**		**5**
T.Repudiation_Receive		**X**	**X**					**X**		**X**						**X**	**X**				**X**		**7**
T.Repudiate_Send		**X**	**X**					**X**		**X**						**X**	**X**				**X**		**7**
T.Credential_Theft	**X**					**X**					**X**						**X**				**X**	**X**	**6**
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**	**X**	**8**
T.Human_Error				**X**						**X**			**X**				**X**	**X**					**5**
T.Disclose_Data	**X**							**X**	**X**	**X**							**X**				**X**	**X**	**7**
T.Privacy_Violated										**X**							**X**			**X**	**X**	**X**	**5**
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**			**X**	**7**
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**		**X**		**4**
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**	**X**	**4**

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**

3.1.7. **Security Requirement and Threat mapping**

Security requirements of an IoT system identified in section 3.1.1 are mapped to the threat as shown in Table 3-5. These security requirements represents the overall security requirements of the system which if implemented will make the system secure from corresponding threats and is shown by marking "X" in the corresponding column in Table 3-5. For e.g. threat T.Impersonate can be overcome by Security requirements Identification and Non-Repudiation.

Table 3-5 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***	***Firmware security***	***Traffic controls***	***Standardization***	***Self-healing***	***Secure communication between devices***

"

T.Data_Theft	**X**	**X**	**X**											**X**	**X**	**X**	**X**		**X**
T.Impersonate	**X**						**X**										**X**	**X**	
T.Fraud		**X**							**X**						**X**		**X**		
T.Repudiation_Receive							**X**						**X**				**X**		**X**
T.Repudiate_Send							**X**						**X**				**X**		**X**
T.Credential_Theft		**X**												**X**					
T.Phishing		**X**	**X**																
T.Privacy_Violated		**X**	**X**		**X**			**X**		**X**				**X**	**X**	**X**	**X**		
T.Change_Data	**X**	**X**	**X**										**X**	**X**	**X**	**X**	**X**		
T.Insider	**X**	**X**	**X**		**X**		**X**							**X**	**X**				**X**
T.Human_Error					**X**												**X**		
T.Disclose_Data			**X**				**X**							**X**	**X**				
T.DDoS		**X**	**X**										**X**		**X**			**X**	
T.Misuse_of_System_Resources			**X**	**X**		**X**							**X**				**X**		

T.Injection_Attack				**X**		**X**								**X**					
T.Spoofing	**X**	**X**	**X**				**X**									**X**	**X**			**X**
T.Malware					**X**										**X**			**X**		
T.Communication_Interception					**X**								**X**				**X**			**X**
T.Communication _Infiltration					**X**								**X**				**X**			
T.Eavesdropping															**X**	**X**	**X**			**X**
T.Technical_ Failure										**X**			**X**	**X**	**X**			**X**	**X**	
T.Power_Failure										**X**			**X**	**X**	**X**				**X**	
T.Network_Infrastructure_ Failure										**X**			**X**	**X**	**X**			**X**	**X**	
T.Hardware_ Failure										**X**			**X**	**X**	**X**			**X**	**X**	**X**
T.Unavailability											**X**		**X**	**X**	**X**					
T.Vandalism										**X**					**X**					

T.Operational_Issues												**X**	**X**	**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																		**X**	
T.Node_Capture									**X**		**X**							**X**	**X**
T.Fake_Node	**X**													**X**	**X**			**X**	**X**

3.2. Prioritization

Security requirements prioritization is a process to order security requirements based on some metric. As per the metric value the higher value will indicate the higher priority of security requirement. The metric we are using is risk value. Risk value is a function of threat rating and asset ratings.

Below are the major steps:

- Impact calculation
- Calculation of Risk Value with respect to threats
- Prioritize the security requirements

3.2.1. **Impact Calculation**

Asset rating is taken from the Table 3-2 for each asset. Table 3-4 shows the mapping of threats with assets and then impacts can be derived from the below formula:

Impact = Avg (Asset Rating w.r.t. Threats)

Or

Impact = sum of asset rating of assets which will be impacted if threat occurs/ Total no. of assets

Table 3.6 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 3-6 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	9	3	7			9		5			**3.3**
T.Data_Theft	9	3	7			9		5			**3.3**
T.Impersonate	9	3	7			9		5			**3.3**
T.Fraud			7					5			**1.2**
T.Repudiation_Receive					3						**0.3**
T.Repudiate_Send					3						**0.3**
T.Credential_Theft						9					**0.9**

T.Phishing				6							**0.6**
T.Insider	9	3	7			9		5			**3.3**
T.Spoofing				6							**0.6**
T.Human_Error								5			**0.5**
T.Disclose_Data	9	3	7			9		5			**3.3**
T.Privacy_Violated	9	3	7			9		5			**3.3**
T.DDoS				6	3						**0.9**
T.Misuse_of_System_Resources							3				**0.3**
T.Injection_Attack				6					3		**0.9**
T.Malware				6							**0.6**
T.Communication_Interception	9	3	7		3	9		5	3		**3.9**
T.Communication_Infiltration	9	3	7		3	9		5	3		**3.9**
T.Eavesdropping	9	3	7			9		5			**3.3**

T.Technical_Failure				6						4	**1**
T.Power_Failure				6						4	**1**
T.Network_Infrastructure_Failure				6	3					4	**1.3**
T.Hardware_Failure				6						4	**1**
T.Unavailability				6						4	**1**
T.Vandalism			7								**0.7**
T.Operational_Issues				6						4	**1**
T.Console_Access_Attack				6		9				4	**1.9**
T.Chip_Access_Attack							3		3		**0.6**
T.Timing_Attack				6	3		3				**1.2**
T.Hello_Flooding_Attack				6							**0.6**
T.Node_Capture	9	3	7					5			**2.4**
T.Fake_Node			7		3						**1**

3.2.2. **Calculation of risk value with respect to Threats**

Risk value is derived as the product of the threat and impact rating of a threat. Impact rating and Threat ratings for threats are taken from Table 3-6 and Table 3-4 respectively. For e.g. Risk Value of T.Change_Data = 42 (14*3). Below table shows the calculation for Risk Value.

Risk value = Threat Rating * Impact Rating

Table 3-7 Risk Estimation

Threats	Threat Rating	Impact Rating	Risk Values
T.Change_Data	12	3.3	39.6
T.Data_Theft	8	3.3	26.4
T.Impersonate	3	3.3	9.9
T.Fraud	5	1.2	6
T.Repudiation_Receive	7	0.3	2.1
T.Repudiate_Send	7	0.3	2.1
T.Credential_Theft	6	0.9	5.4
T.Phishing	6	0.6	3.6
T.Insider	10	3.3	33
T.Spoofing	8	0.6	4.8
T.Human_Error	5	0.5	2.5
T.Disclose_Data	7	3.3	23.1
T.Privacy_Violated	5	3.3	16.5
T.DDoS	4	0.9	3.6
T.Misuse_of_System_Resources	5	0.3	1.5
T.Injection_Attack	4	0.9	3.6
T.Malware	7	0.6	4.2
T.Communication_Interception	5	3.9	19.5
T.Communication _Infiltration	5	3.9	19.5
T.Eavesdropping	5	3.3	16.5
T.Technical_Failure	4	1	4
T.Power_Failure	3	1	3
T.Network_Infrastructure_Failure	4	1.3	5.2

T.Hardware_Failure	5	1	5
T.Unavailability	3	1	3
T.Vandalism	4	0.7	2.8
T.Operational_Issues	4	1	4
T.Console_Access_Attack	7	1.9	13.3
T.Chip_Access_Attack	5	0.6	3
T.Timing_Attack	4	1.2	4.8
T.Hello_Flooding_Attack	6	0.6	3.6
T.Node_Capture	2	2.4	4.8
T.Fake_Node	4	1	4

3.2.3. **Prioritize the security requirements**

We have already mapped the security requirements with associated threats in Table 3-5. Table 3-8 shows the security requirements prioritization for the IoT based electric metering system shown in Figure 2-1, in accordance with the Priority Value. Risk values of threats are taken from Table 3-7 are added to obtain the Priority Values of security requirements. E.g. Intrusion detection security requirement can overcome T.Misuse_of_System_Resources and T.Injection_Attack which has risk value of 1.5 and 3.6 respectively. So its priority will be 5.1 (1.5 + 3.6).

Table 3-8 Security Requirements Prioritization

SECURITY REQUIREMENTS For IoT	**PRIORITY VALUE**
Firmware security	169.9
Traffic control	147
Standardization of IoT protocol stack	124.8
Self-healing	47.1
Identification	117.7
Authentication	118.8
Authorization	152.1
Immunity	5.1
Integrity	103
Intrusion Detection	5.1
Non-Repudiation	75

Privacy	24.3
Security Auditing	6
Survivability	57.6
Physical Protection	3
System Maintenance	4.8
Data Freshness	82.2
Real-Time Response	115.5
Trust	212.6
Secure communication between devices	118.2

3.3. Validation

Validation of security requirements are done to remove various loopholes in the them as sometimes there may be the case that some security requirements are even not needed, so in order to remove such scenarios validation is essential.

Chapter 4: SECURITY DESIGN ENGINEERING FOR SECURING IoT

In this chapter, security design engineering phase of proposed framework steps are discussed in detail with reference to the IoT based electric metering system described in Section 2.1.

4.1. IDENTIFYICATION OF CRYPTOGRAPHIC MECHANISMS

Security Design Engineering is mainly concerned about mapping of security requirements identified in previous chapter to the identified security services and cryptographic mechanisms as shown in Table 4-1. Various Cryptograph mechanisms which are used for mapping are taken from (Sharma, 2015) .

Table 4-1 Security Requirements mapped with Cryptographic mechanisms

Security Services	Security Requirements	Cryptographic Mechanisms
Availability	Firmware security	Recovery Services
		Secure Booting
		Cryptographic Techniques
	Identification	Digital Certificates
	Authentication	Authentication Exchanges
		Two Factor Authentications
		Multi Factor Authentications
		Kerberos
		Key Agreement Protocols
	Authorization	DAC (Discretionary Access Control)
		Key Agreement Protocols
		MAC (Mandatory Access Control)
		RBAC (Role-Based 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	(Privacy + Immunity)	Encryption mechanisms
		Transport Layer Security mechanisms (e.g. TLS / DTLS)
	Traffic Controls	Encryption mechanisms
		Transport Layer Security mechanisms (e.g. TLS / DTLS)
	Secure communication between devices	Encryption mechanisms
		Transport Layer Security mechanisms (e.g. TLS / DTLS)
Integrity	Integrity	Hash Functions
Auditability	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
Standardization	Standardization of IoT protocol stack	Standard security protocols
Self-healing	Self-healing	Authentication Exchanges
		Key Agreement Protocols
		Recovery Services

4.2. SECURITY DESIGN ANALYSIS

Security design analysis consists of the following stages:

- Threats are mapped with Cryptographic Services
- Grouping of Cryptographic mechanisms and impact calculation

- Design constraints are computed and analysed

4.2.1. **Threats mapped with Cryptographic Services**

In this step as the heading suggests we map the threats to the security services or cryptographic mechanisms. These mechanisms are either used independently or in conjunction with others to cancel all the possible threats.

All the security mechanisms described in Section 3.1.1 have been mapped with threats and shown in Table 3-2 Table 3-4 and Table 4-1 were taken as reference for mapping Security requirements, attacks and Security Mechanisms.

Table 4-2 Threat mapping with Security mechanisms

Security Services	Security Requirements	Threats	Security Mechanisms
Availability	firmware security	T.Data_Theft (27)	Recovery Services
		T.Fraud (4.4)	Secure Booting
		T.Privacy_Violated (12)	Cryptographic Techniques
		T.Change_Data (42)	
		T.Insider (30)	
		T.Disclose_Data (18)	
		T.Spoofing (3.5)	
		T.Eavesdropping (15)	
		T.Fake_Node (4.4)	
	Identification	T.Change_Data	Digital Certificates
		T.Spoofing	
		T.Insider	
		T.Fake_Node	
		T.Data_Theft	
		T.Impersonate	
	Authentication	T.Change_Data	Authentication Exchanges
		T.Spoofing	Two Factor Authentications
		T.Insider	Multi Factor Authentications

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

"

		T.Technical_Failure	
		T.Power_Failure	
		T.Network_Infrastructure_Failure	
		T.Hardware_failure	
	Physical Protection	T.Unavailability	Recovery Services
		T.Spoofing	Secure Booting
			Cryptographic Techniques
	System Maintenance	T.Node_Capture	Maintenance Services
	Real-Time Response	T.Repudiation_Receive	Vulnerability Assessment Tools
		T.Repudiate_Send	Faster Cryptographic Techniques
		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	
	Data Freshness	T.Operational_Issues	Vulnerability Assessment Tools
		T.Console_Access_Attack	Faster Cryptographic Techniques
		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	
Confidentiality	Confidentiality (Privacy + Immunity)	T.Privacy_Violated	Encryption mechanisms
		T.Chip_Access_Attack	Transport Layer Security mechanisms (e.g. TLS / DTLS)
		T.Timing_Attack	
		T.Misuse_of_System_Resources	
		T.Injection_Attack	
	Traffic Controls	T.Data_Theft (27)	Encryption mechanisms
		T.Repudiation_Receive (1.5)	Transport Layer Security mechanisms (e.g. TLS / DTLS)
		T.Repudiate_Send (1.5)	
		T.Privacy_Violated (12)	
		T.Change_Data (42)	
		T.Spoofing (3.5)	
		T.Communication_Interception (18)	
		T.Communication _Infiltration (18)	
		T.Eavesdropping (15)	
Integrity	Integrity	T.Insider	Hash Functions
		T.Privacy_Violated	
		T.Human_Error	
		T.Malware	
		T.Communication_Interception	
		T.Communication _Infiltration	
		T.Chip_Access_Attack	
		T.Timing_Attack	
Audit ability	Security Auditing	T.Fraud	Auditing mechanisms
			Service Level Agreements Strengthening

			(SLA_ Strengthening)
Trust	Trust	T.Disclose_Data T.Fake_Node	Compliance mechanisms
		T.Operational_Issues	Need to know Principle Enforcement
		T.Console_Access_Attack	All of the above cryptographic techniques
		T.Vandalism	
		T.Change_Data	
		T.Technical_Failure	
		T.Power_Failure	
		T.Network_Infrastructure_Failure	
		T.Hardware_failure	
		T.Unavailability	
		T.Eavesdropping	
		T.Privacy_Violated	
		T.DDoS	
		T.Malware	
		T.Data_Theft	
		T.Credential_Theft	
		T.Insider	
Standardization	Standardization of IoT protocol stack	T.Data_Theft (27)	Standard security protocols
		T.Impersonate (9)	
		T.Fraud (4.4)	
		T.Privacy_Violated (12)	
		T.Change_Data (42)	
		T.Human_Error (1.5)	
		T.Misuse_of_System_Resources (1)	
		T.Malware (3)	
		T.Technical_Failure (2.8)	
		T.Network_Infrastructure_Failure (3)	
		T.Hardware_failure(3.5)	
		T.Operational_Issues (2.1)	
Self-healing	Self-healing	T.Impersonate (9)	Authentication Exchanges
		T.DDoS (3.2)	Key Agreement

			Protocols
		T.Technical_Failure (2.8)	Recovery Services
		T.Power_Failure (2.1)	
		T.Network_Infrastructure_Failure (3)	
		T.Hardware_failure(3.5)	
		T.Operational_Issues (2.1)	
		T.Hello_Flooding_Attack (3)	
		T.Node_Capture (4.4)	
		T.Fake_Node (4.4)	

4.2.2. **Grouping of Cryptographic mechanisms and impact calculation**

Cryptographic mechanisms for providing security can grouped together to provide a better mix of threat mitigation as we can see from Table 4-2, that none of the cryptographic algorithm alone is able to mitigate all the threat thus we need to either group them or get any hybrid algorithm if any.

The grouping and the calculated impact is shown in Table 4-3. As observed in the table that ECIES is the best cryptographic mechanism and can avoid most threats for the IoT based electric metering 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.

We have grouped the cryptographic mechanisms based on (sharma,2015). Impact of any mechanism can be given by the number of security requirements it fulfils.

Table 4-3 Security Mechanisms grouping & Impact Identification

Security Services	Security Requirements	Suitable Cryptographic Algorithms													
		Asymmetric Algorithms			Symmetric Algorithms			Hashing Algorithms		Signing Algorithms			Hybrid Algorithms		
		RSA	ECC	HECC	AES	DES	Triple DES	MD5	SHA1	RSA + DSA	ECDSA	HECDSA	ECC + DUAL RSA + MD5 (sakthivel, 2010)	N/2 (AES + ECC) + N/2 (Dual RSA)) + HASH (W. Ren, 2013)	Lightweight Hybrid Cryptographic Algorithm (MouzaBani, 2012)	ECIES
Availability	Firmware security	Y	N	N	Y	N	N	N	N	N	N	N	Y	Y	Y	Y
	Identification	Y	Y	Y	N	N	N	N	N	Y	Y	Y	Y	Y	N	Y
	Authentication	Y	Y	N	N	N	N	N	N	N	N	N	Y	Y	N	Y
	Authorization	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	N	Y
	Non-Repudiation	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	N	Y
	Intrusion Detection	Y	Y	Y	N	N	N	Y	Y	Y	Y	Y	Y	Y	N	Y

	Physical Protection	Y	N	N	Y	N	N	N	N	N	N	N	Y	Y	Y	Y
	Real-Time Response	N	Y	Y	Y	N	N	Y	Y	N	Y	Y	N	N	Y	Y
	Data Freshness	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y
Confidentiality	(Privacy + Immunity)	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y	Y
	Traffic Controls	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y	Y
	Secure communication between devices	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y	Y
Integrity	Integrity	N	N	N	N	N	N	Y	Y	N	N	N	Y	Y	N	Y
Auditability	Security Auditing	N	N	N	N	N	N	N	N	Y	Y	Y	N	N	N	N
Trust	Trust	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y

"

Standardization	Standardization of IoT protocol stack	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	N
Self-healing	Self-healing	N	N	N	N	N	N	Y	Y	Y	Y	Y	N	N	N	N
Total Impact		**9**	**8**	**7**	**7**	**4**	**4**	**5**	**5**	**7**	**8**	**8**	**13**	**13**	**8**	**14**

4.2.3. **Design Constraints are computed and analysed**

These are the constraints or limitations of the system that needs to be taken into consideration before taking any of the security mechanisms as a solution to mitigate threats. Below are the design constraints of the IoT based electric metering system

1) **Memory**

   IoT things are always available with limited memory to keep low cost and viability

2) **Computation power**

   Low CPU Speed, so finding a security solution without effecting real-time response of system is a complex task.

3) **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.

4) **Mobility**

   IoT devices are mobile in nature. Security solution must also consider this.

5) **Energy / Power**

   Limited battery power is available in IoT devices. The security solution should consider this.

6) **Scalability**

   There is an exponential increase in number of devices in IoT. So, we have to choose a scalable security algorithm. A device can join or leave the network at anytime from anywhere.

7) **Cost**

   IoT devices should be Low cost as there will be billions of devices installed as well as devices should be durable

8) **Communication Channel**

IoT things are mainly connected wirelessly through various wireless communication technologies such as Bluetooth, Bluetooth Low Energy, GSM, Wi-Fi, 2G/3G/4G and Wi-Max. So, providing security is a tough ask.

9) **Security Updates**

Security updates needs to be up to date to provide better security

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

4.3. FINALIZING SECURITY DESIGN DECISION

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

4.3.1. Identification & Prioritization of design attributes

Table 4-4 Security Design attributes identification & Prioritization

Quality Attribute	Design Attributes	IoT SP's	IoT devices with sensors	IoT Peer devices	IoT UI device
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 constraints are prioritised for all direct actors and is labelled as high, medium and low. These prioritized design constraints are called design attributes of the system .Table 4-5 shows the prioritization of design attributes for IoT based electric metering system. E.g. IoT things have a high priority for battery consumption and memory constraint.

4.3.2. **Review design decisions**

In this step the design decisions taken after performing all the above steps are reviewed again that whether the security mechanisms is providing enough security so that threats are avoided to a certain tolerable limit. If it is not the case then more security mechanism are incorporated into design. Also based on design constraints certain security mechanisms are not feasible so those are also reconsidered.

4.3.3. **Security design template preparation**

A security design template is created that shows the design attributes along with the security mechanisms that manages these attributes well. Based on design constraints

of the specific application required mitigation technique can be chosen. Below table shows the security design template for IoT electric metering system.

Table 4-5 Security Design Template

Quality Attribute	Design Attributes	IoT SP's	IoT devices with sensors	IoT Peer devices	IoT User Interfaces	Cryptographic 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
	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	Self-Healing and Resilience Mechanisms Vulnerability Assessment Tools Audit Mechanisms Recovery Services Maintenance Services
Portability	Architecture / Network Topology	High	High	High	Low	

Analysis:

Depending on the constraints we find that the following cryptographic algorithms will be best suitable for IoT based electric metering system.

- Hybrid Algorithm – ECIES over others
- Signing Algorithm – ECDSA over others
- Symmetric Encryption – AES 128 over others
- Hash function – MD5 over others

ECIES is a hybrid algorithm which mixes up several cryptographic algorithms as one. As per Table 4-3 ECIES is the best algorithm to implement security requirements with Impact Value of 14.

Chapter 5: Security Testing for IoT systems

In this chapter the testing of design decisions taken is done that is the security mechanisms selected for threat mitigation on the previous chapter is evaluated and checked if the threats are mitigated below the tolerable value of the IoT system. This value is referred as Epsilon value and an admin can decide over its value. Below are the steps in this phase:

5.1. Test Scenarios generation

We generate various test scenarios on the basis of sequence diagram drawn during the extraction of vulnerable point during the security requirement specification phase. These scenarios generation is done for all the functionalities with probable vulnerability points through which a threat or attack is possible .

Based on the sequence diagram the following table is generated which maps all the threats on vulnerable points associated with the functionality identified for all the users.

Table 5-1 Vulnerablities, Threats, Risk Value for different functionalities

Actors	Functionality	Vulnerabilities	Threats	Risk Value

IoT User	1.Monitor data	V.Untrained_Users V.Weak_Access_Control	T.Change_Data T.Phishing T.Spoofing T.Human_Error T.Disclose_Data T.Privacy_Violated T.Misuse_of_System_Resources T.Impersonate T.Fraud T.Credential_Theft T.Insider T.Hello_Flooding_Attack T.Data_Theft	175.9
	2.Give command to IoT endpoint device	V.Weak_Access_Control V.Legal_Audit_Issues V.System_Misuse	T.Change_Data T.Impersonate T.Fraud T.Credential_Theft T.Disclose_Data T.Phishing T.Insider T.Spoofing T.Hello_Flooding_Attack T.Human_Error T.Misuse_of_System_Resources T.Vandalism T.Timing_Attack T.Repudiation_Receive T.Repudiate_Send T.Privacy_Violated T.DDoS T.Power_Failure T.Network_Infrastructure_Failure T.Hardware_Failure T.Unavailability T.Console_Access_Attack T.Chip_Access_Attack T.Data_Theft	223.8

IOT service provider	1.Provides the IoT device with sensor installed	V.Weak_Access_Control V.Unencrypted_Data V.Breached_Firewall V.Insecure_Interfaces V.Insufficient_Security_Configurability V.Legal_Audit V.Intrution_Detection	T.Change_Data T.Impersonate T.Fraud T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Hello_Flooding_Attack T.Disclose_Data T.Communication_Interception T.Communication_Infiltration T.Eavesdropping T.Malware T.Privacy_Violated T.Injection_Attack T.Vandalism T.Operational_Issues T.Chip_Access_Attack T.Repudiation_Receive T.Repudiate_Send T.Console_Access_Attack T.Timing_Attack T.Human_Error T.Misuse_of_System_Resources T.DDoS T.Technical_Failure T.Hardware_Failure T.Unavailability T.Data_Theft	286.9

"

	2.Provides the UI based application for users	V.Weak_Access_Control V.Inadequate_Logging V.Insecure_Network_services V.Insufficient_Security_Configurability V.Legal_Audit V.Intrution_Detection	T.Change_Data T.Impersonate T.Fraud T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Hello_Flooding_Attack T.Disclose_Data T.Repudiation_Receive T.Repudiate_Send T.Malware T.Communication_Interception T.Communication_Infiltration T.Eavesdropping T.Timing_Attack T.Fake_Node T.Privacy_Violated T.Injection_Attack T.Vandalism T.Console_Access_Attack T.Human_Error T.Misuse_of_System_Resources T.DDoS T.Technical_Failure T.Hardware_Failure T.Unavailability T.Data_Theft	283.9

IOT security administrator	1.Provides all the security related features to the IOT users	V.Weak_Access_Control V.Unencrypted_Data V.Breached_Firewall V.Obsolete_System V.Insecure_Interfaces V.Insecure_Network_services V.Insufficient_Security_Configurability V.Legal_Audit_Issues V.System_Misuse	T.Change_Data T.Impersonate T.Fraud T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Hello_Flooding_Attack T.Disclose_Data T.Communication_Interception T.Communication_Infiltration T.Eavesdropping T.Malware T.Privacy_Violated T.Injection_Attack T.Vandalism T.Operational_Issues T.Chip_Access_Attack T.Repudiation_Receive T.Repudiate_Send T.Console_Access_Attack T.Timing_Attack T.Human_Error T.Misuse_of_System_Resources T.Technical_Failure T.Hardware_Failure T.DDoS T.Power_Failure T.Network_Infrastructure_Failure T.Unavailability T.Data_Theft	295.1

Internet	1.Data transfer 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	T.Change_Data T.Impersonate T.Fraud T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Hello_Flooding_Attack T.Disclose_Data T.Communication_Interception T.Communication_Infiltration T.Eavesdropping T.Malware T.Privacy_Violated T.Injection_Attack T.Vandalism T.Operational_Issues T.Chip_Access_Attack T.Repudiation_Receive T.Repudiate_Send T.Console_Access_Attack T.Timing_Attack T.Technical_Failure T.Hardware_Failure T.Data_Theft	276.3

IOT devices with sensors	1.Data collection from sensors	V.Weak_Access_Control V.Unencrypted_Data V.Monitoring_Absence V.Physical_Security V.Misconfigurations V.Unsecured_API_Firmware V.Obsolete_System V.Insecure_Interfaces V.Insufficient_Security_Configurability V.Lack_of_Standards V.Intrution_Detection	T.Change_Data T.Impersonate T.Fraud T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Hello_Flooding_Attack T.Disclose_Data T.Communication_Interception T.Communication_Infiltration T.Eavesdropping T.DDoS T.Malware T.Hardware_Failure T.Vandalism T.Console_Access_Attack T.Chip_Access_Attack T.Node_Capture T.Fake_Node T.Technical_Failure T.Power_Failure T.Network_Infrastructure_Failure T.Injection_Attack T.Operational_Issues T.Privacy_Violated T.Repudiation_Receive T.Repudiate_Send T.Timing_Attack T.Human_Error T.Misuse_of_System_Resources T.Unavailability T.Data_Theft	303.9

	2.controlling attached peripherals as per commands by IoT user	V.Monitoring_Absence V.Inadequate_Logging V.Physical_Security V.Misconfigurations V.Unsecured_API_Firmware V.Obsolete_System V.Insufficient_Security_Configurability V.Lack_of_Standards V.Old_Data	T.Credential_Theft T.Phishing T.Spoofing T.DDoS T.Malware T.Change_Data T.Fraud T.Privacy_Violated T.Repudiation_Receive T.Repudiate_Send T.Hardware_Failure T.Vandalism T.Console_Access_Attack T.Chip_Access_Attack T.Node_Capture T.Fake_Node T.Technical_Failure T.Power_Failure T.Network_Infrastructure_Failure T.Injection_Attack T.Operational_Issues T.Data_Theft T.Disclose_Data T.Communication_Interception T.Communication_Infiltration T.Eavesdropping T.Timing_Attack T.Hello_Flooding_Attack T.Insider T.Human_Error T.Misuse_of_System_Resources	291

	3.send/receive data to/from IoT Gateways	V.Weak_Access_Control V.Unencrypted_Data V.Monitoring_Absence V.Inadequate_Logging V.Obsolete_System V.Insecure_Network_services V.Insecure_Interfaces V.Remote_Access V.Resource_Isolation V.Poor_Key_Management V.Lack_of_Standards V.Old_Data V.Intrution_Detection	T.Change_Data T.Impersonate T.Fraud T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Communication_Interception T.Communication_Infiltration T.Eavesdropping T.DDoS T.Malware T.Repudiation_Receive T.Repudiate_Send T.Technical_Failure T.Hardware_Failure T.Operational_Issues T.Timing_Attack T.Fake_Node T.Hello_Flooding_Attack T.Privacy_Violated T.Injection_Attack T.Vandalism T.Chip_Access_Attack T.Console_Access_Attack T.Misuse_of_System_Resources T.Power_Failure T.Network_Infrastructure_Failure T.Unavailability T.Disclose_Data T.Node_Capture T.Human_Error T.Data_Theft	303.9

IoT Cloud	1.Run IOT application intermediatory to the IOT UI devices and IOT devices with sensors	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	T.Change_Data T.Impersonate T.Fraud T.Credential_Theft T.Phishing T.Insider T.Eavesdropping T.Malware T.Technical_Failure T.Power_Failure T.Network_Infrastructure_Failure T.Console_Access_Attack T.Fake_Node T.Operational_Issues T.Chip_Access_Attack T.Hello_Flooding_Attack T.Repudiation_Receive T.Repudiate_Send T.Spoofing T.Privacy_Violated T.Injection_Attack T.Communication_Interception T.Communication_Infiltration T.Vandalism T.Timing_Attack T.Human_Error T.DDoS T.Misuse_of_System_Resources T.Hardware_Failure T.Unavailability T.Disclose_Data T.Data_Theft	299.1

Peer Devices	1.Data collection from sensors	V.Weak_Access_Control V.Unencrypted_Data V.Monitoring_Absence V.Physical_Security V.Misconfigurations V.Unsecured_API_Firmware V.Obsolete_System V.Insufficient_Security_Configurability V.Lack_of_Standards V.Intrution_Detection	T.Credential_Theft T.Phishing T.Insider T.Spoofing T.DDoS T.Privacy_Violated T.Malware T.Technical_Failure T.Hardware_Failure T.Operational_Issues T.Change_Data T.Human_Error T.Misuse_of_System_Resources T.Vandalism T.Console_Access_Attack T.Chip_Access_Attack T.Node_Capture T.Fake_Node T.Power_Failure T.Network_Infrastructure_Failure T.Injection_Attack T.Fraud T.Repudiation_Receive T.Repudiate_Send T.Disclose_Data T.Communication_Interception T.Communication_Infiltration T.Eavesdropping T.Timing_Attack T.Hello_Flooding_Attack T.Impersonate T.Unavailability T.Data_Theft	303.3

	2.controlling attached peripherals as per commands by IoT user	V.Monitoring_Absence V.Inadequate_Logging V.Physical_Security V.Misconfigurations V.Unsecured_API_Firmware V.Obsolete_System V.Insufficient_Security_Configurability V.Lack_of_Standards V.Old_Data	T.Data_Theft T.Credential_Theft T.Phishing T.Insider T.Spoofing T.DDoS T.Malware T.Technical_Failure T.Hardware_Failure T.Operational_Issues T.Change_Data T.Human_Error T.Misuse_of_System_Resources T.Vandalism T.Console_Access_Attack T.Chip_Access_Attack T.Node_Capture T.Fake_Node T.Power_Failure T.Network_Infrastructure_Failure T.Injection_Attack T.Fraud T.Repudiation_Receive T.Repudiate_Send T.Disclose_Data T.Communication_Interception T.Communication_Infiltration T.Eavesdropping T.Timing_Attack T.Hello_Flooding_Attack T.Unavailability T.Privacy_Violated	294

	3.Communication with peer device	V.Weak_Access_Control V.Unencrypted_Data V.Monitoring_Absence V.Obsolete_System V.Insecure_Network_services V.Remote_Access V.Resource_Isolation V.Lack_of_Standards V.Old_Data V.Intrution_Detection	T.Credential_Theft T.Phishing T.Insider T.Spoofing T.DDoS T.Malware T.Technical_Failure T.Hardware_Failure T.Change_Data T.Human_Error T.Misuse_of_System_Resources T.Impersonate T.Fraud T.Hello_Flooding_Attack T.Disclose_Data T.Communication_Interception T.Communication_Infiltration T.Eavesdropping T.Timing_Attack T.Fake_Node T.Power_Failure T.Network_Infrastructure_Failure T.Unavailability T.Operational_Issues T.Console_Access_Attack T.Chip_Access_Attack T.Node_Capture T.Repudiation_Receive T.Repudiate_Send T.Data_Theft	281

IoT Gateways	1.Data transfer between IoT Devices with sensors and IoT UI devices	V.Weak_Access_Control V.Unencrypted_Data V.Breached_Firewall V.Inadequate_Logging V.InsecureInterfaces V.Insecure_Network_services V.Insufficient_Security_Configurability V.Legal_Audit V.Intrution_Detection	T.Change_Data T.Fraud T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Hello_Flooding_Attack T.Disclose_Data T.Communication_Interception T.Communication_Infiltration T.Eavesdropping T.Malware T.Repudiation_Receive T.Impersonate T.Repudiate_Send T.Privacy_Violated T.Injection_Attack T.Vandalism T.Operational_Issues T.Chip_Access_Attack T.Timing_Attack T.Fake_Node T.Console_Access_Attack T.Human_Error T.Misuse_of_System_Resources T.DDoS T.Technical_Failure T.Hardware_Failure T.Unavailability T.Data_Theft	290.9

IoT UI devices	1.Take user command from to IOT servers	V.Misconfigurations V.Unsecured_API_Firmware V.Obsolete_System V.Legal_Audit V.System_Misuse	T.Technical_Failure T.Hardware_Failure T.Operational_Issues T.Change_Data T.Human_Error T.Misuse_of_System_Resources T.Vandalism T.Timing_Attack T.Impersonate T.Fraud T.Repudiation_Receive T.Repudiate_Send T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Disclose_Data T.DDoS T.Power_Failure T.Network_Infrastructure_Failure T.Unavailability T.Console_Access_Attack T.Chip_Access_Attack T.Privacy_Violated T.Fake_Node T.Injection_Attack T.Malware T.Data_Theft	240

	2.Provide IoT services to the user	V.Untrained_Users V.Obsolete_System V.Legal_Audit V.System_Misuse	T.Technical_Failure T.Hardware_Failure T.Operational_Issues T.Change_Data T.Human_Error T.Misuse_of_System_Resources T.Vandalism T.Timing_Attack T.Impersonate T.Fraud T.Repudiation_Receive T.Repudiate_Send T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Disclose_Data T.Privacy_Violated T.DDoS T.Power_Failure T.Network_Infrastructure_Failure T.Unavailability T.Console_Access_Attack T.Chip_Access_Attack T.Data_Theft	228.2

	3.Provide data from IOT device to the user for monitoring.	V.Untrained_Users V.Misconfigurations V.Unsecured_API_Firmware V.Obsolete_System V.Legal_Audit V.System_Misuse	T.Technical_Failure T.Hardware_Failure T.Operational_Issues T.Change_Data T.Human_Error T.Misuse_of_System_Resources T.Vandalism T.Timing_Attack T.Impersonate T.Fraud T.Repudiation_Receive T.Repudiate_Send T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Disclose_Data T.Privacy_Violated T.DDoS T.Power_Failure T.Network_Infrastructure_Failure T.Unavailability T.Console_Access_Attack T.Chip_Access_Attack T.Fake_Node T.Injection_Attack T.Malware T.Data_Theft	240

5.2. Threat Mitigation Level Check

The threats that are identified are checked for mitigation from the security mechanism grouping in Table 5.2. The threats that remain in the system are called as live threats. Suppose we take the ECIES algorithm as the security Algorithm for threat mitigation. The threats that are mitigated is shown in Table 5.2 .A vulnerability metric of live threats are calculated which is actually the risk value associated with each threat. All live threats along with their corresponding vulnerability metric are shown in Table 5.3

Table 5-2Threats mitigated and security requirements for ECIES.

Cryptographic Technique	Threats Mitigated	Security Requirements implemented
ECIES	T.Change_Data	Firmware security
	T.Fake_Node	Identification
	T.Impersonate	Authentication
	T.Fraud	Authorization
	T.Repudiation_Receive	Non-Repudiation
	T.Repudiate_Send	Intrusion Detection
	T.Credential_Theft	Physical Protection
	T.Phishing	Real-Time Response
	T.Insider	Data Freshness
	T.Spoofing	(Privacy + Immunity)
	T.Human_Error	Traffic Controls
	T.Disclose_Data	Secure communication between devices
	T.Privacy_Violated	Integrity
	T.DDoS	Trust
	T.Misuse_of_System_Resources	
	T.Injection_Attack	
	T.Malware	
	T.Communication_Interception	
	T.Communication _Infiltration	
	T.Eavesdropping	
	T.Technical_Failure	
	T.Power_Failure	
	T.Network_Infrastructure_Failure	
	T.Hardware_Failure	
	T.Unavailability	
	T.Operational_Issues	
	T.Console_Access_Attack	
	T.Chip_Access_Attack	
	T.Timing_Attack	
	T.Hello_Flooding_Attack	
	T.Data_Theft	

S.No	Live Threats	Vulnerability Matrix

"

1	T. Node Capture	4.8
2	T. Vandalism	2.8

Table 5-3 Live threats for ECIES

5.3. Security Index Calculation

Security index can be calculated by the equation given as below as given by (Shruti Jaiswal, 2018), Security Index value 0 means no security lapses exist in the system.

$$SI = \frac{\sum_1^N Vi}{\sum_1^N Ri}$$

Where,
$V\,i$ is the vulnerability metric of all the active threat for all functionality F_i
R_i is the total risk value corresponding to functionality F_i
N is the number of Functionality considered

Now if (SI ≥ Epsilon)
Where Epsilon is the maximum tolerable risk value
Then the system is unsafe.
Else the system is in safe mode

5.3.1. **SI Value when ECIES is employed**

SI=(0+2.8+2.8+2.8+2.8+2.8+7.6+7.6+7.6+2.8+7.6+7.6+4.8+2.8+2.8+2.8+2.8)/
175.9+223.8+286.9+283.9+295.1+276.3+303.9+291+303.9+299.1+303.3+294+281+
290.9+240+228.2+240

SI= 0.015

Let assume the Epsilon value to be 1.5, so the SI way below this value so the system is in safe state.

5.3.2. **SI Value when Hybrid Algorithm ECC + DUAL RSA + MD5 is employed**

The Live Threats when ECC+DUAL RSA+ MD5 is employed

S.No	Live Threats	Vulnerability Matrix
1	T. Node Capture	4.8
2	T. Vandalism	2.8
3	T.Hello_Flooding_Attack	3.6
4	T.Malware	4.2
5	T.Human_Error	2.5

SI =

(6.1+8.9+13.1+13.1+13.1+10.6+17.9+17.9+17.9+13.1+17.9+17.9+15.1+13.1+9.5+5.3+9.5)/175.9+223.8+286.9+283.9+295.1+276.3+303.9+291+303.9+299.1+303.3+294+281+290.9+240+228.2+240

SI= 0.047

Thus based on the SI value it can be concluded that ECIES is the best suited algorithm for IoT Based electric metering system.

5.4. Generate Test Report

The generated test report is shown in Table 5.4

Table 5-4 Test report for ECIES implementation

IOT based electric metering system	
Security Algo applied	ECIES

Threats Identified and risk measure	T.Change_Data T.Data_Theft T.Impersonate T.Fraud T.Repudiation_Receive T.Repudiate_Send T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Human_Error T.Disclose_Data T.Privacy_Violated T.DDoS T.Misuse_of_System_Resources T.Injection_Attack T.Malware T.Communication_Interception T.Communication _Infiltration T.Eavesdropping T.Technical_Failure T.Power_Failure T.Network_Infrastructure_Failure T.Hardware_Failure T.Unavailability T.Operational_Issues T.Console_Access_Attack T.Chip_Access_Attack T.Timing_Attack T.Hello_Flooding_Attack T.Fake_Node T. Node Capture T. Vandalism

Threats mitigated	T.Change_Data T.Data_Theft T.Impersonate T.Fraud T.Repudiation_Receive T.Repudiate_Send T.Credential_Theft T.Phishing T.Insider T.Spoofing T.Human_Error T.Disclose_Data T.Privacy_Violated T.DDoS T.Misuse_of_System_Resources T.Injection_Attack T.Malware T.Communication_Interception T.Communication _Infiltration T.Eavesdropping T.Technical_Failure T.Power_Failure T.Network_Infrastructure_Failure T.Hardware_Failure T.Unavailability T.Operational_Issues T.Console_Access_Attack T.Chip_Access_Attack T.Timing_Attack T.Hello_Flooding_Attack T.Fake_Node
Threats not mitigated	T. Node Capture T. Vandalism
Result	SI=0.017, so system is secure
Remark	SI value can be 0 if the system employs all the security mechanisms

Chapter 6: CONCLUSIONS & FUTURE WORK

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

6.1. CONCLUSIONS

This project proposes a new security incorporation framework for Internet of Things in which the security is incorporated along with the development of the IoT system. The phases include security requirements engineering, security design engineering, and security testing.

In the security requirements engineering the security the security requirements are specified, prioritized and then validated. Security requirement specification include identification of stakeholders as direct and indirect stakeholders. Valuable assets to the system are identified and vulnerable points are identified from sequence diagram. After that threat identification is done and corresponding security requirements are identified Prioritization of security requirements is done by calculating risk value of threats and then security requirements are prioritized based on the threat risk values for particular requirement. At last validation of security requirements is done.

In security design engineering, Security requirements are mapped with cryptographic mechanisms. After this security analysis is done which involves mapping of security mechanisms to threats, security mechanisms grouping and impact identification and all the design constraints of the system are identified such as for IoT based system design constraints can be computational power, storage and power consumption etc. The last step in security design engineering is security design structuring which involves Identification and prioritization of design attributes, review of design decisions and preparation of design template.

In security testing phase, the steps involves generation of test scenarios based on sequence diagram drawn in security requirement phase, then a suitable security mechanism is picked and mitigated threats as well as live threats are checked for the chosen security mechanism. Then security index is calculated which if below a

particular epsilon value, then security mechanisms is chosen for implementation otherwise security mechanisms are selected so that security index comes below the epsilon value.

The framework provided in this thesis will serve as a benchmark for incorporating security into the IoT system during the production phase only. Using this framework a secure, more robust system can be made easily and it helps the IoT system engineers to take best design decisions with the available hardware and software constraints

The framework and the methodology proposed in this project can be taken as a generic model for enhancing security in many IoT Applications, as shown in case study how we can achieve in an IoT based electric metering system.

6.2. FUTURE WORK

- Many more IoT based case studies can be checked for the proposed framework so that a generic and more refined version of the framework can be obtained.

- Machine learning algorithms can also be applied for the framework to make informed design decisions

- Security implementation is not much of a discussion point in this thesis and can be incorporated in further work.

Chapter 7: References

(n.d.). Retrieved from Iotivity: https://www.iotivity.org/

ITU Workshop on the Internet of Things - Trend and Challenges in Standardization. (2014, feb 25). Geneva, Switzerland: ITU.

A Mixed Encryption Algorithm Used in Internet of Things Security Transmission System. (2015). *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery.* IEEE.

A Survey on Application Layer Protocols for the Internet of Things. (2015). *Transaction on IoT and Cloud Computing*.

AshishAgarwal, D. G. (2008). Security Requirements Elicitation Using View Points for Online System. *IEEE*.

B, S. (1996). *Applied cryptography.* Wiley.

CharithPerera. (2015). The Emerging Internet of Things Marketplace From an Industrial Perspective: A Survey.

Firesmith, D. (2003). Engineering Security Requirements. *Journal of Object Technology*, 53-68.

Gabbai, A. (2015, January 01). *Innovation*. Retrieved from Smithsonianmag: https://www.smithsonianmag.com/innovation/kevin-ashton-describes-the-internet-of-things-180953749/

GhofraneFersi. (2015). Middleware for Internet of Things: a study.

Granjal, J. (2015). Security for the Internet of Things: A Survey of Existing Protocols and Open Research issues.

GSMA IoT security guidelines and Assessment. (n.d.). Retrieved from GSMA: http://www.gsma.com/connectedliving/future-iot-networks/iot-security-guidelines/

Gupta, K. C. (2013). A framework for development of secure Software. *Springer-Verlag*.

Kader, H. M. (2014). Performance Evaluation Of New Hybrid Encryption Algorithms To Be Used For Mobile Cloud Computing. *International journal of technology enhancements and emerging engineering research*, vol 2, issue 4.

Kai Zhao, L. (2013). A Survey on the Internet of Things Security. *Ninth International Conference on Computational Intelligence and Security.*

Kantarci, B. (2015). Sensing Services in Cloud-Centric Internet of Things: A Survey, taxonomy and challenges. *IEEE*.

Khan, R. (2012). 10th International Conference on Frontiers of Information Technology.

Kotonya G., S. I. (1996). Requirement Engineering with viewpoints. *Software Engineering Journal*, 5-18.

Li Da Xu, W. H. (2014). Internet of Things in Industries: A Survey. *IEEE*, 2233 - 2243.

Luigi Catuogno, S. T. (2015). 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.

MouzaBani. (2012). A New Lightweight Hybrid Cryptographic Algorithmfor The Internet of Things. *IEEE*.

Open Web Application Security Project. (n.d.). Retrieved from OWASP: https://www.owasp.org/

Pandya, H. B. (2015). Internet of Things : Survey and Case Studies.

PengXu, M. L.-J. (2013). *A hybrid encryption algorithm in the application of equipment information management based on Internet of things.* Atlantis Press.

Pongle, P. (2015). A Survey Attacks on RPL and 6LoWPAN in IoT. *IEEE*.

Prudence. (2014). A framework for Security Requirements Engineering suitable for securing Big Data environments.

S. K. Josyula, D. G. (2017). A new security methodology for internet of things. *2017 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 613-618). Greater Noida: ICCCA.

sakthivel, s. &. (2010). Design of a new security protocol using hybrid cryptography algorithms. *IJRRAS*.

Sharma, J. (2015). *Design Methodology for Secure Cloud Systems.*

Shruti Jaiswal, D. G. (2018). Measuring Security:A Step Towards Enhancing Security of System. *International Journal of Information Systems in the Service Sector*.

Sommerville, I. (Seventh edition 2003). *Software Engineering.* Pearson Education.

W. Ren, a. Z. (2013). A new security protocol using hybrid cryptography. *9th International Conference Computer Engineering Conference (ICEN-CO)*, (pp. 109-115).