Secure Multicast Communication in IoT

Submitted By

Makkad Asim Mahamadrafiq

15MCEI13

DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2017

Secure Multicast Communication in IoT

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering (Information and Network Security)

Submitted By

Makkad Asim Mahamadrafiq

(15MCEI13)

Guided By

Dr. Vijay Ukani

DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2017

Certificate

This is to certify that the major project entitled **"Secure Multicast Communication in IoT"** submitted by **Makkad Asim Mahamadrafiq (Roll No: 15MCEI13)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering (Information and Network Security) of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this thesis, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Vijay Ukani
Guide & Associate Professor,
Computer Engineering Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Sharada Valiveti
Associate Professor,
Coordinator M.Tech - CSE (INS)
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Sanjay Garg
Professor and Head,
Computer Engineering Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad.

Statement of Originality

I, **Makkad Asim Mahamadrafiq**, Roll. No. **15MCEI13**, give undertaking that the Major Project entitled "**Secure Multicast Communication in IoT**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science and Engineering (Information and Network Security)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Dr. Vijay Ukani

(Signature of Guide)

Acknowledgements

It is indeed a pleasure for me to express my sincere gratitude to those who have always helped me throughout my project work.

I would thank to my guide to **Dr. Vijay Ukani**, Associate Professor, Computer Engineering Department, Institute Technology, Nirma University, Ahmadabad whose keen interest and excellent knowledge base helped me to finalize the topic of the dissertation work. His constant support, encouragement, and constructive criticism has been invaluable assets through my project work. He has shown keen interest in this dissertation work right from beginning and has been a great motivating factor in outlining the flow of my work.

As a token of my deepest gratitude, I would like to thanks **Dr. Sanjay Garg**, Hon'ble Head of Computer Engineering Department, Institute of Technology, Nirma University, Ahmadabad and **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmadabad who have directly or indirectly helped me during this dissertation work.

I am also thankful to all faculty members of Department of Computer Engineering, Nirma University, Ahmadabad for their special attention and suggestions for work. The blessings of God and my family made the way for this dissertation work. I am very much grateful to them. The friends, who always bear and motivated me throughout this training, I am thankful to them.

- Makkad Asim Mahamadrafiq

15MCEI13

Abstract

Internet of Things (IoT) or Web of Things (WoT) is a wireless network between smart products or smart things connected to the Internet. IoT is emerging technology. IoT can be classified into software components and hardware components. The focus of the proposed project is on CoAP (Constraint Application Protocol) protocol which falls under software components. CoAP is web-based protocol which provides Datagram Transport Layer Security (DTLS) security. DTLS only provide security in uni-cast message, because DTLS do not support multicast. To provide security in multicast messages, the proposed solution is to distribute session keys using key distribution center. Using provided session keys user encryption or decryption multicast messages.

Abbreviations

6LoWPAN	Ipv6 over low power wireless personal Area network
ACK	Acknowledgement
AMQP	Advanced Message Queuing Protocol
Cf	Californium
CoAP	Constraint Application Protocol
CON	Confirm-able
CU	Copper
DTLS	Datagram Transport Layer Security
DSA	Data Storage and Analysis
E-mail	Electronic mail
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IPsec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
KDC	Key Distribution Center
KSA	Key Scheduling Algorithm
LWM2M	Lightweight M2M
M2M	Machine to Machine
MQTT	Message Queue Telemetry Transport
NAT	Network address translation
NFC	Near Field communication
NON	Non Confirm-able
OS	Operating System
PSK	Pre-Shared Key
PRGA	Pseudo Random Generation Algorithm
QoS	Quality of Statement
RESTFUL Services	Representational State Transfer
RFID	Radio Frequency Identification

RPL	Routing Protocol For Low Lossy Network
SMS	Short Message Service
TCP	Transmission Control Protocol
ubicomp	Ubiquitous Computing
UDGM	Unit Disk Graph Medium
UDP	User Data-gram Protocol
WoT	Web of Things
WSN	Wireless Sensor Network
XMPP	Extensible Messaging and Presence Protocol

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Security is main problem in every communication. There are many different type of attacks like Man-in-middle attack, eavesdropping, data modification, application layer attacks, sniffer attacks, IP spoofing, password based attack and denial-of-service attack use to interrupt communication. The IoT is based on a wide range of semiconductor advancements, including power administration gadgets, sensors and microchips. Execution and security prerequisites differ impressively starting with one application then onto the next[1]. One thing is consistent as it were. That is the fact that the success of smart homes, connected cars, industries and factories hinges on user confidence in robust, easy-to-use, fail-safe security capabilities. To overcame those attacks user use different types of cryptographic algorithms.

As Internet developing quickly and numerous of gadgets are interfacing step by step. So we can utilize this network and can make those individual articles to impart that will share information and can handle generally can provide for a client or can store some place where a client can utilize that information to settle on the choice[2]. IoT is required to offer propelled availability of gadgets, frameworks and administrations that go past machine-to-machine (M2M) correspondences and spreads an assortment of protocols, domains and applications. IoT contains low power gadgets and IPv6 availability amongst every single gadget.

In this report, our aim is to provide security in multicast communication. We analyze one problem in CoAP security DTLS and we design cryptographic algorithm as a proposed solution to provide multicasting security. Also implement algorithm in Cooja simulator (Contiki OS built in Ubuntu) and give some output in Wireshark and Copper (CU) web

browser.

1.1 Motivation

IoT is emerging technology in all over world. IoT is one of those technologies which transforms Internet to M2M basis. Hence, IoT can seamlessly connect to the real world and cyberspace via physical objects that are embedded with various types of intelligent sensors. A question is why we use IoT? Because IoT enables all kinds of devices to connect together and share information seamlessly and number of things connected to the Internet is more than people present on earth[3]. Number of things connect using some set of rules to exchange messages with other internet point called protocol. There are many protocols in different layer shown in Table 1.1. It shows IoT protocol stack that include application layer protocol (like CoAP, MQTT, XMPP, AMQP), Transport layer protocol (like UDP, DTLS), Internet layer Protocol (like RPL, 6LoWPAN) and Network/Link layer Protocol (like IEEE 802.15 Series and IEEE 802.11 Series). This thesis is more inclined toward web-based protocols that is application layer protocol. In application layer protocol work is being done on CoAP and its security DTLS. The CoAP protocol is used for application like health-care, parking system and home management etc.

Layer	Protocols
Application Layer	CoAP, MQTT, XMPP, AMQP
Transport Layer	UDP, DTLS
Internet Layer	RPL, 6LoWPAN
Network/Link Layer	IEEE 802.15 Series, IEEE 802.11 Series

Table 1.1: IoT protocol stack

The hardware side of IoT is the primary motivation to work on it and make smart things, than study about protocols in IoT and we decide work in CoAP protocol and its security protocol DTLS. Its been concluded that work would be done in security of CoAP. Based on the study of various research paper was carried out about DTLS protocol in CoAP and identify problem related secure communication in IoT.

1.2 Research Objectives

Security is one of the need of every application or software. Cryptography algorithm (encryption and decryption algorithm) provide network security or provide security in communication called end to end security. Every application or software have different algorithms. IoT need low constrain, complexity and power, so here we design low complexity and low power consumption algorithm for secure group communication or multicast in IoT.

Based on the study of existing work of CoAP, following research objective are defined:

- Develop CoAP based or multicast communication in simulator.
- Develop cryptography algorithm to secure multicast communication.
- Implement and Test algorithm in simulator.
- Analyse overall performance of framework.

1.3 Thesis Outline

The rest of the thesis is organized as follow:

Chapter 2 (*Literature Survey*) In this chapter, fundamental of IoT and its protocol are described. Issues and challenges in CoAP and DTLS are also described. elaborates examples related to CoAP, advantages and limitations of CoAP and DTLS.

Chapter 3 (*Problem Statement*) This chapter give problem related to secure multicast communication in CoAP. Summarize existing scenario of this project and discuss effort given by other author.

Chapter 4 (*Proposed Solution*) This chapter show effort given by us and show algorithm for secure group communication.

Chapter 5 (*Implementation*) In this chapter, we describe the simulator and its functionality. Run program of CoAP related to IoT. In this program, we add some sensor mote and Communicate each other via CoAP. All this implementation is performed in

Contiki OS and Cooja simulator. We also present performance analysis of whole framework.

Chapter 6 (*Future Work*) This chapter include the major conclusion of this research work. Future direction in this area is also outlined in this chapter.

Chapter 2

Literature Survey

2.1 Internet of Things

IoT is an environment where small smart devices are connected always, anytime and anywhere with each other. All the devices are connected to the Internet and have the capability to sense and send data over the cloud. These small devices are called things in IoT. These small devices could be any object like mobile devices, home appliances, tablets etc. Once these devices are connected to the network then more and more data and services will be available which helps for different types of applications[4]. IoT creates an application in a variety of fields such as agriculture, logistics and home automation.

IoT in simple meaning, "The Internet to do everything smart". The IoT turn into a dream where genuine items are a piece of the Internet: each protest is exceptionally recognized, and available to the system, its position and status were known. IoT enables all kinds of devices to connect together and share information seamlessly and the number of things connected to the Internet is more than people present on earth.

IoT has small embedded device that must consume less power and with low cost also. To provide and overcome those problems, we need to research so we can come up with a solution that allows users to develop applications for IoT. The idea of IoT came based on RFID tags in which each RFID tag is uniquely identifiable and the information is embedded in each RFID tag so we can identify each RFID tag and information embedded into that. Same is applicable to IoT in which all objects are uniquely identifiable in the network and has some information where we can identify each object in the network and we can get some information from that particular object. Devices are capable of sense

data that has built in sensors and processing power also has storage capability increased and size is reduced. These all devices have sensors connected which sense data and send data to the cloud where we can process all the data and make some conclusion.

2.1.1 Applications in IoT

IoT uses every where application like medical store, infrastructure (Example: smart-classrooms, smart library), health-care (hospital) and many more. Some of those describe below [5].

- **Assisted Driving :**
  Today's different kind of transportation like autos, buses and transports alongside the street and the rails. To give better route and security. They utilize furnished with sensors and actuators. Those prepared give gainful data to the driver and/or travelers (i.e. accidents, temporary and/or permanent road closures, traffic congestion).

- **Authentication and Identification :**
  Authentication and Identification are two terms that portrayed the preparatory periods of the security procedure in IoT frameworks which could apply to social insurance. For example: persistent ID to lessen destructive episodes to patient, current electronic restorative record upkeep and baby distinguishing proof in doctor's facilities to avert bungling.

- **Mobile Ticketing :**
  Electronic publications or announcements gives information in setting to transportation administrations which can be furnished with the NFC tag. The client can get to information from the web by either drifting their cell phone over the NFC tag or indicating the cell phone the visual markers.

- **Thefts :**
  An application cautions the client to if an important object is moved from a limited region, which demonstrates that the question is stolen. In such sort of case, this sort of occasion must be advised quickly to the proprietor and/or security protects through SMS, call, email, and so on.

2.1.2 Elements of IoT

There are three IoT components which enable seamless ubi-comp(Ubiquitous Computing):
a) Hardware: They made up with actuators, sensors and embedded communication hardware.
b) Middle-ware: on-demand storage and computing tools for information analytic.
c) Presentation: novel easy to understand visualization and interpretation tools which can be easily accessed on distinct platforms and which can be designed for distinct applications[6].
That five elements which will make up the three components are stated below[6].

- RFID (Radio Frequency Identification)
- WSN (Wireless Sensor Networks)
- Addressing Schema
- DSA (Data Storage and Analysis)
- Visualization

2.1.3 IoT Architecture

To communicate between two nodes internet used TCP/IP protocol same devices in IoT will use TCP/IP network for communication with different devices. That will increase more traffic in network but with advancement in device capability. It will also increase storage and network capability.

As shown Figure 2.1, IoT consist of five layers and each layer has its own role to support IoT. These layers are:

- **The Perception Layer:**

  The perception layer is also known as device layer. It consists of physical device and sensors. The sensors are of any type that can be location, temperature, pressure, acceleration etc. These sensors sense the value based on type of sensor and they will pass it to network layer from where they can transmit it to network. The information is passed to network layer where network layer provides security to send data to cloud.

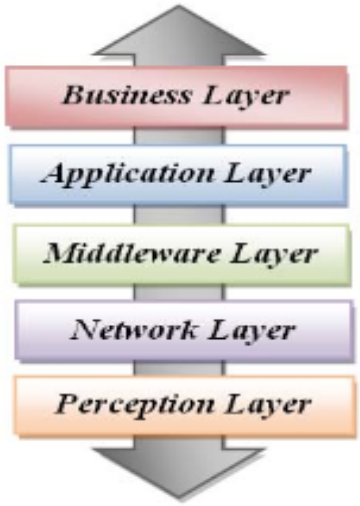

Figure 2.1: The IoT generic architecture [2]

- **Network Layer:**

  Network layer is also called as transmission layer which is used to transfer data collected by sensor to processing device through wired or wireless technology. To transfer information interface can be WiFi, Bluetooth, ZigBee, 3G etc. This layer is connectivity layer between sensor devices and processing devices where we can process all the data and can make some conclusion.

- **Middle-ware Layer:**

  The middle-ware layer is also called as storage layer. In which all the data collected from network this layer will store it into database or cloud from where application can use the data stored in database.

- **Application layer:**

  Application layer has applications where they can use data stored in database or at middle-ware layer and can make use of the data. For example the application implemented in IoT can be smart health, smart home, smart city and intelligent transportation. This layer collects data from cloud or storage and can give data to user.

- **Business Layer:**

  This layer is responsible for the management of overall IoT system including the application and services. It builds business models with graph or flowchart of whole

system. Based on the analysis of results one can take the decision and can take future actions and decide business strategies.

2.1.4 IoT challenges

There are various challenges present in IoT as information is passed from network and anybody can access information. Various challenges in IoTs are:

- **Naming and Identity management:**
  The IoT will connect billions of devices or objects and each should be uniquely identifiable in network. Each object or sensor has unique identity over the Internet[3]. The efficient naming and identity management system required that can dynamically assign and manage unique identity for such a large number of objects.

- **Standardization:**
  There are different manufacturer involved in manufacturing different devices with their own technology where in some cases these devices cannot communicate with device that is manufactured by different manufacture so the standardization between all the devices is needed so all the devices can communicate easily.

- **Data confidentiality and encryption:**
  Sensors will sense the data and send it to network so confidentiality of information is very important where we can use some standard encryption or decryption to encrypt the data and send over network and other device can decrypt it.

- **Network Capability:**
  The challenge regarding network capability is there are many sensors and devices connected with network and the data from sensor device will be sent through wired or wireless interface. The transmission system or network should be able to collect all the data from sensors and make sure that no data loss occur due to network congestion.

- **Low power:**
  The main challenge for IoT is low power of devices as embedded devices used in IoTs are deployed at many places and that has limited power capacity in this case its very important to save power whenever its possible[7]. So there is need of mechanism

where we can power off the devices when there is no need of power and can power up again whenever needed.

- **Reliability:**

  The main challenge in IoT is reliability. When one IoT node send data to more than one server, if one of the server will crash or goes down then it is very hard to get original file. If file will be deleted at server side it can not be reconstructed so, data will be lost.

2.1.5 Protocols in IoT

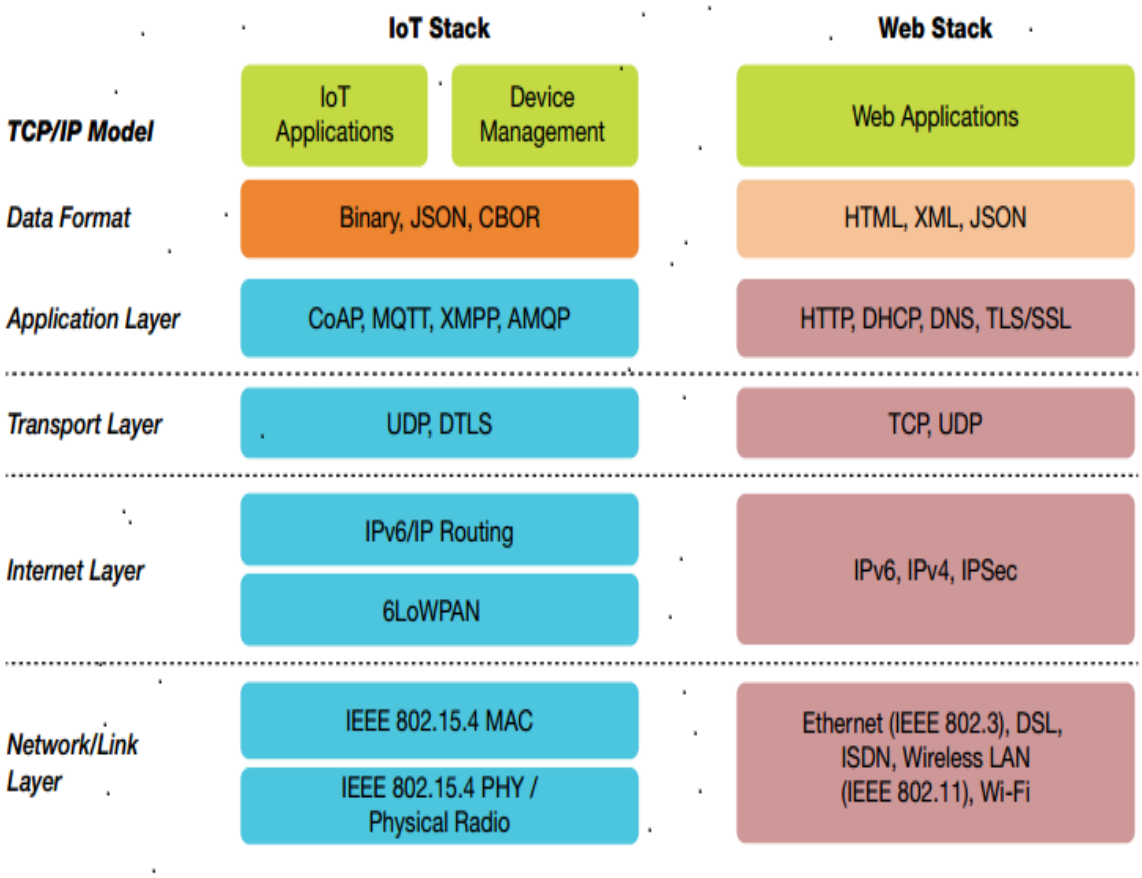

Figure 2.2: Comparison between IoT and web protocols [6]

Figure 2.2 shows different between IoT protocol stack and web protocols stack. It shows the which types of data format support in both of model. There are many protocols of different layer. It shows the TCP/IP protocol stack just replace by IoT protocol stack. In IoT protocol stack there is different layer like Network/Link layer(In this layer protocols

are IEEE 802.11 series and IEEE 802.15 series), Internet layer (In this layer protocols are RPL, 6LoWPAN, etc...), Transport layer (In this layer protocols are UDP, DTLS, etc...) and Application layer (In this layer protocols are CoAP, MQTT, AMQP, XMPP, etc...). Here, in this thesis focus on only application layer protocol and its security.

Application layer Protocols

This section of the protocol is used for message forwarding in IoT application layer proposed by various standardization organizations. Many of the IP applications, including IoT applications utilize TCP or UDP for transport[3]. However, there are many message distribution functions that are same among many IoT applications; it is expected that these functions could be implemented in an inter-operable standard ways by distinct applications. There is list of application layer protocols and its security given by Table 2.1. It shows, all application layer protocols and which type of architecture, transport protocol and security they use for communication, check the quality of statement (QoS) for every protocol.

Application Layer protocol	Transport	QoS	Architecture	Security
CoAP	UDP	YES	Request/Response	DTLS
MQTT	TCP	YES	Publish/Subscribe	TLS/SSL
XMPP	TCP	NO	Request/Response Publish/Subscribe	TLS/SSL
RESTFUL	HTTP	NO	Request/Response	HTTPS
AMQP	TCP	YES	Publish/Subscribe	TLS/SSL
Web socket	TCP	NO	Publish/Subscribe Client/Server	TLS/SSL

Table 2.1: Application layer protocols

By reading papers regarding application layer protocol, for this thesis decide work on CoAP and its security protocol DTLS.

2.2 Constraint Application Protocol

There are many communication protocols available to communicate with different devices currently there are many communication protocols available such as MQTT and CoAP. CoAP is used for low power and low memory embedded devices where it can be used for communication instead of HTTP[3]. Currently, there is HTTP protocol available with request and response paradigm but HTTP has many features and more footprint.

HTTP runs over TCP where TCP will need more resources due to three-way handshake and much more complex mechanism. Now for low power embedded devices, there is no need of this heavy protocols and we can optimize it to run over TCP. TCP also contains congestion and reliability mechanism so it will take more time and more resources. CoAP is easy protocol implementation for communication between embedded devices. There is packet format that can support both reliability and no reliable applications.

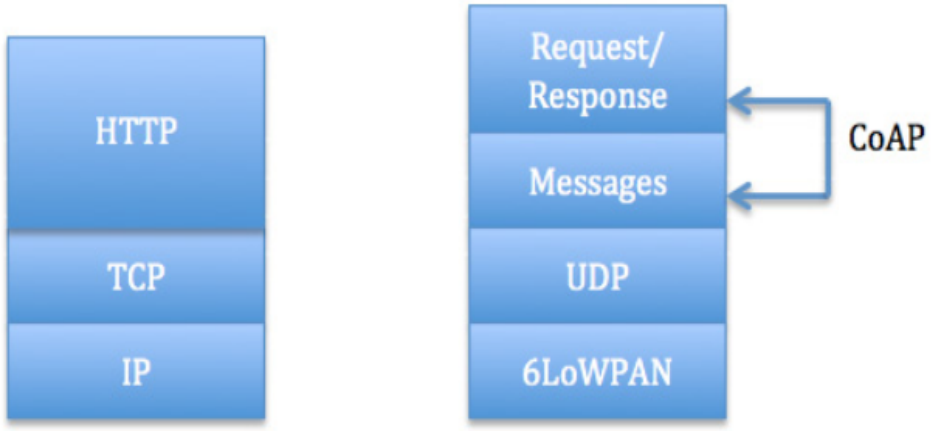

Figure 2.3: HTTP and CoAP protocol stack [5]

As CoAP is a RESTFUL (Representational State Transfer) web exchange convention for use with compelled systems. CoAP utilizes request/response model of approach same as HTTP. It is intended for obliged systems with low overhead and lower impression. A few focuses for CoAP that improves convention contrasted with HTTP is:

- CoAP runs over UDP (User Datagram Protocol) that helps to avoid costly TCP handshake before data transmission.
- CoAP protocol is only 4-byte header and provides reliable transfer and no reliable transfer as it uses four types of messages.
  - Show above figure, Its support four types of message 1) Confirmable, 2) Non-Confirmable, 3) Acknowledgement and 4) Reset.
  - Request/Response layer used those message and classified in 1) Piggy-backed (successful and failure response results of GET method) show Figure 2.4, 2) Separate response (A Get request with a separate response) show Figure 2.5

and 3) Non confirmable request and response, communicate with each other show Figure 2.6[5].

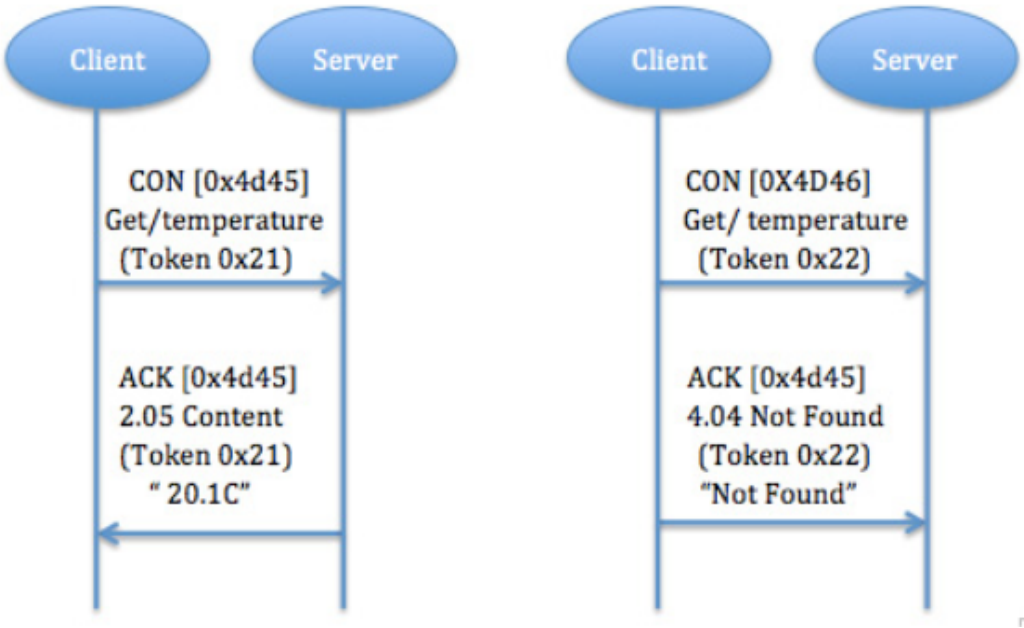

Figure 2.4: In piggy-backed, successful and failure response results of GET method [8]

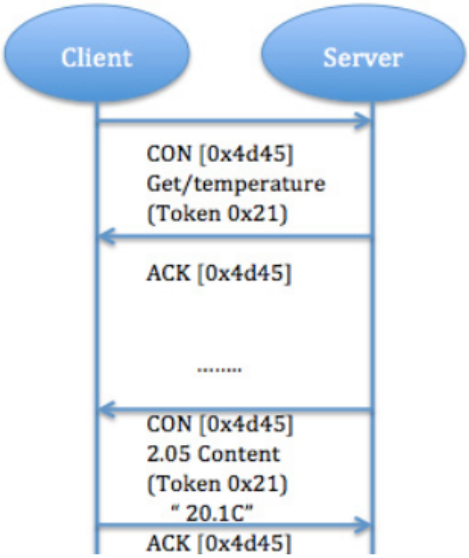

Figure 2.5: A GET request with a separate response [8]

- CoAP has minimal header format that saves IoT of power for constrained nodes compared to running HTTP in that constraint nodes.
- CoAP provides both reliability and non-reliability support that allows CoAP to use in both kind of use case or applications.

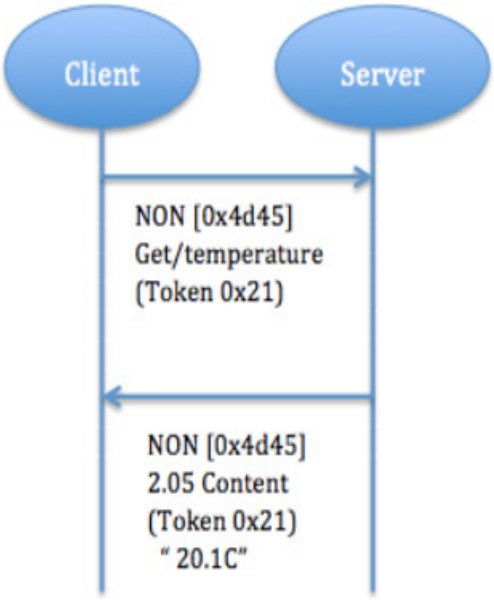

Figure 2.6: Non confirmable request and response [8]

2.2.1 Features of CoAP

- A very efficient RESTFUL protocol.
- Small simple 4 byte header.
- Asynchronous transaction model
- Easy to proxy to/from HTTP.
- URL support.
- Security binding to DTLS.
- Support reliability and multicast.

2.2.2 Security in CoAP

Security is important to protect the communication. DTLS means Datagram Transport Layer Security. It is transport layer protocol. DTLS protect end-to-end communication in application layer. DTLS mainly focus on CoAP protocol to provide security show the Figure 2.7. DTLS have three main element integrity, authentication and confidentiality[5].

DTLS used the LESS algorithm which protect the session key during exchange. The outgoing message are forwarded to DTLS which then forward them to the destination in

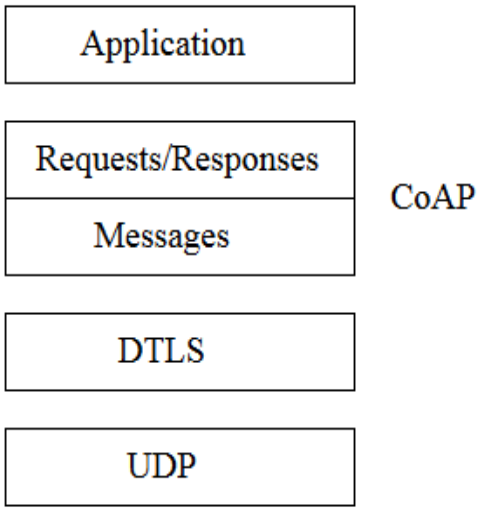

Figure 2.7: Abstract layering of DTLS-secured CoAP

protected mode. Incoming CoAP messages will be protect by DTLS layer then it will be headed to CoAP layer.

2.3 Existing Work in CoAP

There are many problem occur during communicate with one mote to another mote. There are some problems solved using CoAP.

- Scalable Cloud Service with CoAP
- NAT Issues
- Web based Monitoring in Health care
- End-to-End Security

2.3.1 Web Based Monitoring in Health-care

Health care is one of the application in IoT. This application have many problem regarding communication between sensor and server. To those problem over come with CoAP. CoAP used to remote health-care monitoring system that provides the patient 's condition through web browser[9]. There are sensor collect data of patient and transferred to various IP end-devices. Those are communication use 6LoWPAN network to communicate with server[10].

Health-care application like heart rate, ECG, blood pressure, level of glucose or oxygen, all this application problem solved by CoAP. They used contiki os with cooja sim-

ulator and for result in mozilla firefox they use Copper (CU) add-on. Study was done on CoAP based communication papers. Most of paper have one common problem is the security. Secure data, secure end-to-end communication, hardware security and many problem regarding to security.

2.3.2 End-to-End Security

The architecture appears below is for uni-cast correspondence (customer interfaces with at least one servers). The customer likewise has to know which endorsement or crude open key it needs to use with a particular server[5].

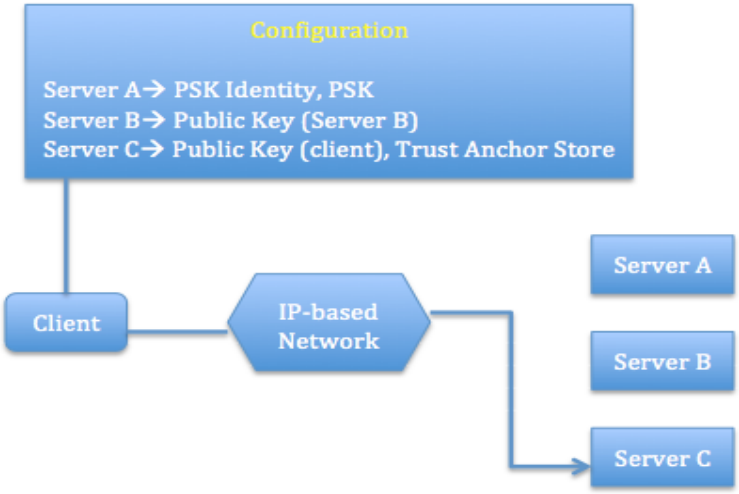

Figure 2.8: Uni-cast communication model [11]

Security is important to protect the CoAP communication. Generally CoAP used DTLS with PSK for security. According to survey DTLS used LESS algorithm which protect the session key during exchange, the outgoing messages are forward to DTLS which then forewords them to the destination in protected mode[12]. The incoming CoAP messages will protect by DTLS layer at first then it will be headed to CoAP layer[12]. LESS algorithm is nothing but non-blocking algorithm. This method is existing scenario of uni-cast message security in end-to-end communication.

Show Figure 2.9, DTLS secure single (client-server) CoAP message. There are two layers in DTLS. The bottom one contains Record convention. The top one incorporate three conventions which are ready, handshake and application information, in some condition Change Cipher Spec convention may supplant one of them[5].

There are many problems in DTLS like LESS algorithm only secure single message.

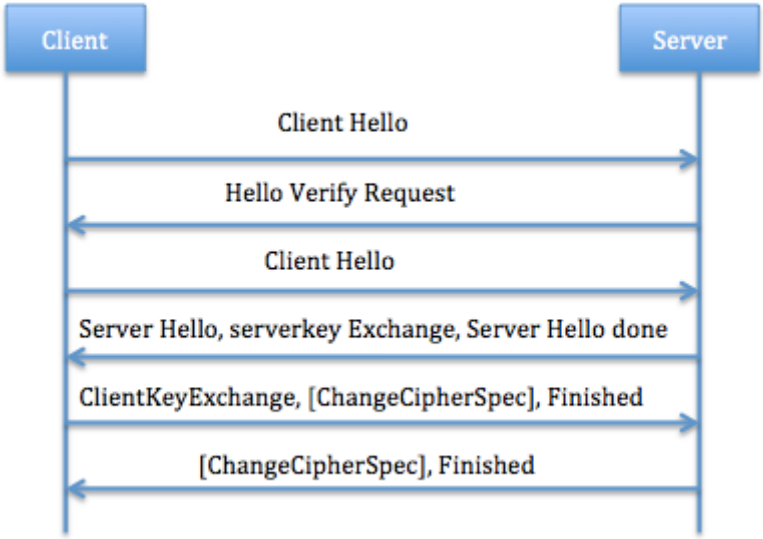

Figure 2.9: DTLS handshake [13]

problem is there not secure path (way of message) and multiple messages not secure. Like vise many challenges are there robust key management in CoAP. In all this problem it was chosen multicast communication in CoAP using IoT frame work.

2.4 Summary

The survey above all those CoAP and DTLS security-related paper. Learn from them protocols is important in IoT it is a main part of IoT. In that protocol, it was chosen CoAP (Constraint Application Protocol, Application Layer Protocol). Also, learn about DTLS (Security protocol of Application Layer).

CoAP generally uses in health care, home management and many more applications. In health care used monitoring patient 's condition through a web browser. The problem in those paper they cannot provide security for M2M communication.

Then learn DTLS based paper those who focus in CoAP protocol. In this DTLS only secure message and uni-cast message. DTLS uses LESS algorithm to encryption and decryption a message. The problem in those paper they cannot secure way of a message (path) and multicast messages.

Chapter 3

Problem Statement

Security is the main issue in everywhere and needs of every application or software and hardware. There are the different type of security available for defending thread like third party authentication (the third party provide secure log-in like Google, Facebook) and use some anti-virus or anti-malware software. In this thesis, secure communication between one to many devices in IoT. For securing communication author already secure uni-cast CoAP message for communication between two devices. Secure uni-cast message author uses LESS algorithm (or non-blocking algorithm) as per show previews chapter how author secure single client-server communication. Security is wide research area as per previews chapter author cannot secure path or multicast communication for IoT devices. In next section provide problem statement and what we do in this thesis (objective).

3.1 Secure multicast communication

By studying all survey paper conclude that, DTLS only secure uni-cast message(or single client-server communication). which is not provide multicast communication service because DTLS do not support multicast. This research provide secure multicast service in IoT.

In this thesis first create multicast communication in simulator then develop cryptographic algorithm that provide security in multicast communication. Finally simulate cryptography algorithm and analysis performance.

3.2 Summary

In this chapter, Understand Problem in existing work. Give problem statement of this thesis and describe the objective of this thesis.

Chapter 4

Proposed Work

Multicast communication is generally one to many communication and one of the best examples of multicast communication is multimedia. The CoAP is similar to client/server HTTP model. In multicasting, the client can send multiple requests to the server but in IoT, CoAP is request/response model so, here client node send multiple requests to server node and also server response to the client[14]. To understand all this scenario show Figure 4.1.

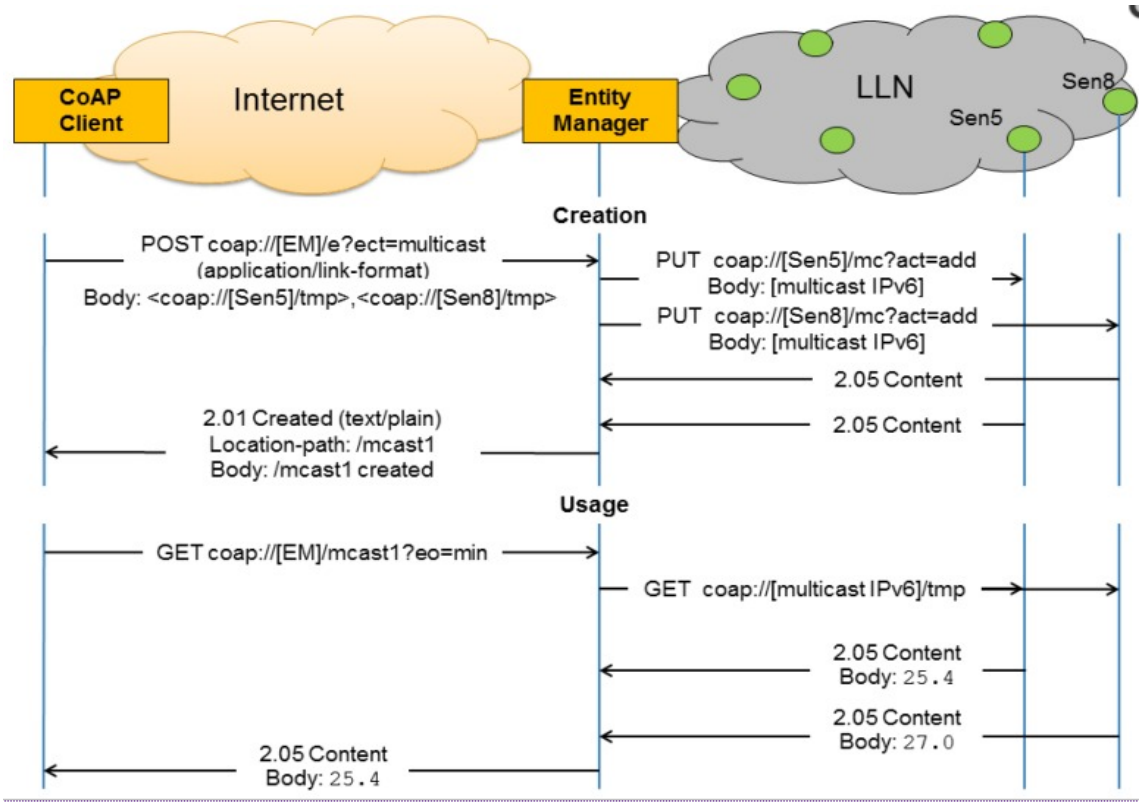

Figure 4.1: Multicast communication model [14]

To secure multicast communication we have to generate session keys for every node. In this chapter, define the algorithm to secure multicast communication in IoT. For securing communication make KDC (Key Distributed Center) distribute session keys to every nodes or sensor. After generating session keys for every node do encryption for each note and create encrypted messages[15]. The same method at decryption side using same KDC algorithm generate session keys then decryption with the encrypted message.

IoT needs low complexity, network capability and standardization to implement any IoT-based algorithm and equipment. That's why design algorithm contains low complexity. In IoT communication work under CoAP protocol and CoAP provide DTLS protocol based security. For multicast communication, DTLS do not support multicast, so here used TLS/SSL protocol based security for securing multicast communication.

4.1 Session Keys Generation Algorithm

Algorithm 1 Key Scheduling Algorithm

1: Array Initialization
2: for (i=0 to 255) do
3: S[i] = i;
4: T[i] = K[i mod keylen]
5: Initial Permutation
6: The initialized array S[256] is now run through the KSA. uses the secret key to scramble the array.
7: j = 0;
8: for (i=0 to 255) do
9: J = (J + S[i]+T[i]) mod 250;
10: swap (S[i],S[j]);

Algorithm 2 Stream Generation Algorithm

1: KSA Scramble S[256] array is used to generate the PRGA. This is actual key stream
2: i,j = 0 ;
3: While(true)
4: {
5: i = (i+1) mod 256
6: j = (j+S[i]) mod 256
7: }
8: swap(S[i],S[j])
9: t = (S[i],S[j]) mod 256
10: k = S[t]

Here, use two different algorithms first key scheduling algorithm (KSA) and second Pseudo-Random Generation Algorithm (PRGA). In KSA initialize an array or decide key length for session keys. After this using array scramble and generate values for PRGA. Then swap both values one question why we use swap, because of encryption of swap position use protect sensitive information.

In PRGA is an actual key stream, using both values of KSA and using loop generate multiple values. In the first loop, two values generated and do modulo with key length like vice loop generate multiple values then do modulo with key length then get some values those are session keys.

After generating session key do encryption using any low complexity algorithm and same method apply at decryption side. First using one key and this algorithm creates multiple session keys and like encryption do decryption.

4.2 Summary

In this Chapter, learn about the working model of multicast communication. Using KSA and PRGA algorithms generate session keys and secure multicast communication in IoT framework.

Chapter 5

Implementation

To implement preview chapter cryptographic algorithm we use as a tool Contiki OS and Cooja simulator section 5.1 describe about implementation tool and section 5.2 shows performance analysis.

5.1 Implementation Tools

In this section, describe about implementation tools used by implement secure multicast communication in IoT. So we used Contiki OS, for simulator Cooja simulator and for result Copper (CU) and Wireshark.

5.1.1 Contiki OS

Contiki is open source operation system. It is lightweight OS and implement in C programming language. It is purposely developed and created for the low-power devices in constrained environment. It is connects constraint node to Internet. It provide powerful low power Internet communication[16]. It supports IPv6 and IPv4 standard along with 6LoWPAN, CoAP and RPL. It can be used in commercial, non-business and full source code is simple accessible. Contiki application are written is C programming language and used cooja simulator for simulation purpose so that network can be emulated before burned into hardware[17].
Features of Contiki below:

- Memory Allocation:
  Contiki is only used for constraint device which is having limited memory like kilobytes. Contiki is highly memory efficient and provides mechanism for memory

allocation.

- Full IP Networking:
  Contiki provide full IP networking support with standard IP protocol UDP, TCP, HTTP and new low power standard 6LoWPAN, RPL, CoAP.

- 6LoWPAN, RPL, CoAP:
  Contiki support 6LoWPAN protocol, RPL IPv6 multi-hope routing protocol and CoAP RESTFUL application layer protocol.

- Power Awareness:
  Contiki basically used for constraint node which low power system. Contiki provide a mechanism for estimating power consumption.

5.1.2 Cooja Simulator

Cooja is network simulator for Contiki which allowed larger and small network for simulation. Cooja control and analyse contiki system via few function. In front-end interface, cooja used combination of java codes[18]. It is a cross-level simulator, built in Contiki OS[19]. It Provide the simulation on network level, OS level and machine code level. It is network simulator for contiki which allowed big and tiny network for simulation.

5.1.3 Copper (CU)

CU is CoAP user agent. It is CoAP plug in for fire-fox only. It allows the user to Browse and Interact with Internet of Thing Devices. CU plug in show in Figure 5.2.
Features of Copper :

- URI handling for the 'CoAP' scheme (address bar and links).

- method are GET, POST, PUT, and DELETE.

- Resource discovery

- Block-wise transfers

- Observing resources

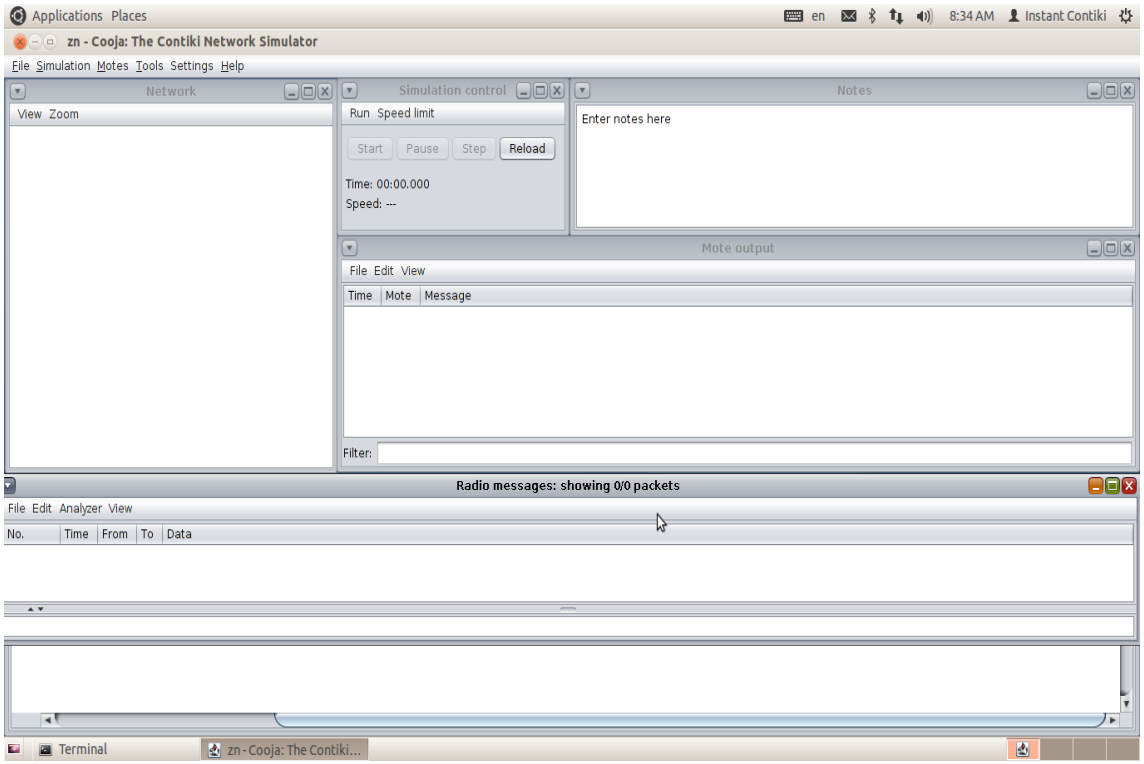

Figure 5.1: Blank cooja simulator interface

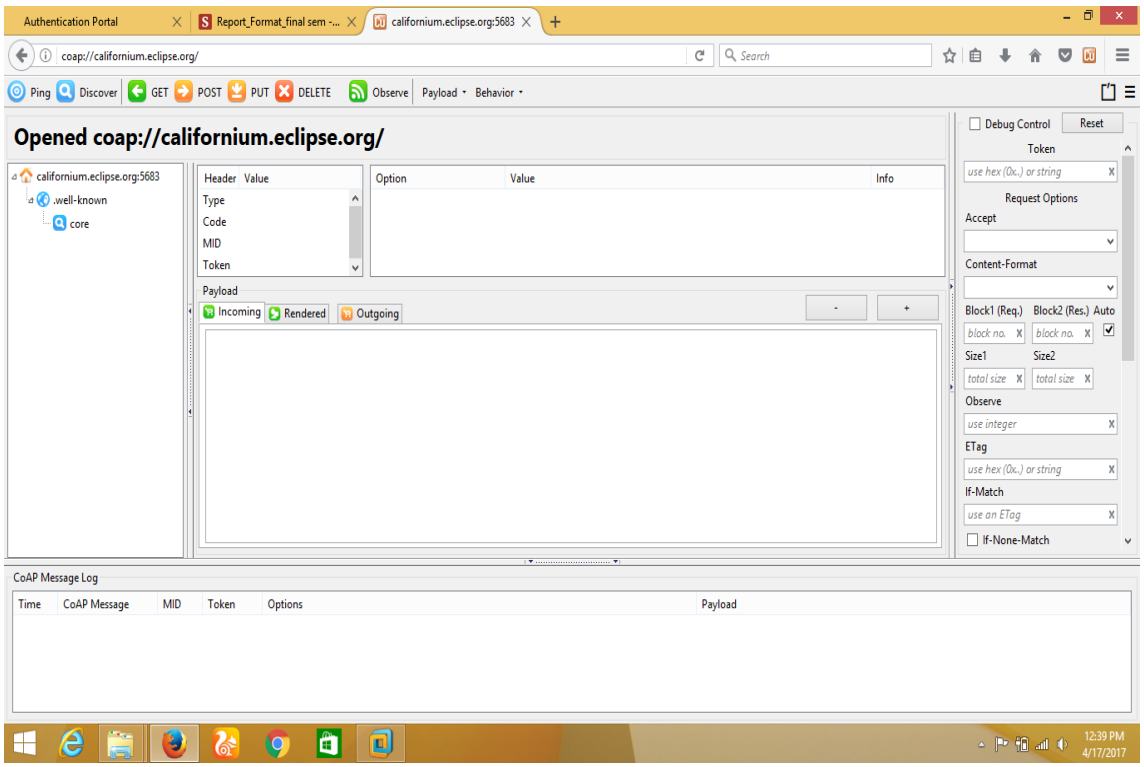

Figure 5.2: Copper window

5.2 Implementation Result and Analysis

In Cooja simulator, there are 5 window available:

1) Network: In this window we make topology using modes. Here we used SKY mode to implement multicast communication.

2) Simulation control: In this window user can start, stop and reload simulator.

3) Radio messages: Show the communication between each and every mode and transfer some messages.

4) Timeline: It shows radio transceiver each mode in colour code, if color is White then transceiver is off, if Gray it is on.

5) Mote output: This window similar to Wireshark, its shows all output like it shows encryption then decryption between two modes then stop simulation its disconnection.

Like mote output, Wireshark is inbuilt with contiki OS and cooja simulator. Whatever we see output in mote output window same as show in wireshark or Wireshark give more information then mote output. Mote output and Wireshark also do packets analysis in contiki OS. For simulator multicasting communication, in topology we used radio environment (UDGM), radio traffic, 10m grid in background, IPv6 IP addresses, as a node SKY mode.

In mote output window or Wireshark shows some output, those show as tabular from in Table 5.1 and Table 5.2.

No. of nodes	Encryption Time (ms)	Decryption Time (ms)
1	1108.00	1255.00
2	1230.00	1325.00
3	1375.00	1500.00
4	1495.00	1735.00
5	1714.00	1897.00
6	1830.00	2050.00
7	1925.00	2214.00
8	2177.00	2485.00
9	2255.00	2644.00
10	2388.00	2790.00

Table 5.1: Encryption and Decryption time for every node

Table 5.1, it shows encryption and decryption time for every nodes. it shows for

File size (Bytes)	Encryption Time (ms)	Decryption Time (ms)
10	1108.00	1250.00
20	1238.00	1358.00
30	1360.00	1482.00
40	1493.00	1612.00
50	1622.00	1744.00
60	1748.00	1874.00
70	1878.00	2096.00
80	2095.00	2226.00
90	2190.00	2349.00
100	2325.00	2465.00

Table 5.2: For 1 node encryption and decryption time in different file size

1 node first client encrypt message then send to server or 1st node then server receive message they start decryption, complete decryption they show message. After encrypt messages or decrypt message client or server distribute key using KDC or algorithm show in section 4.1.

Table 5.2, it shows encryption and decryption time for 1 node for different file size. it show that client encrypt 10bytes message then send to server, server receive message and start decryption.

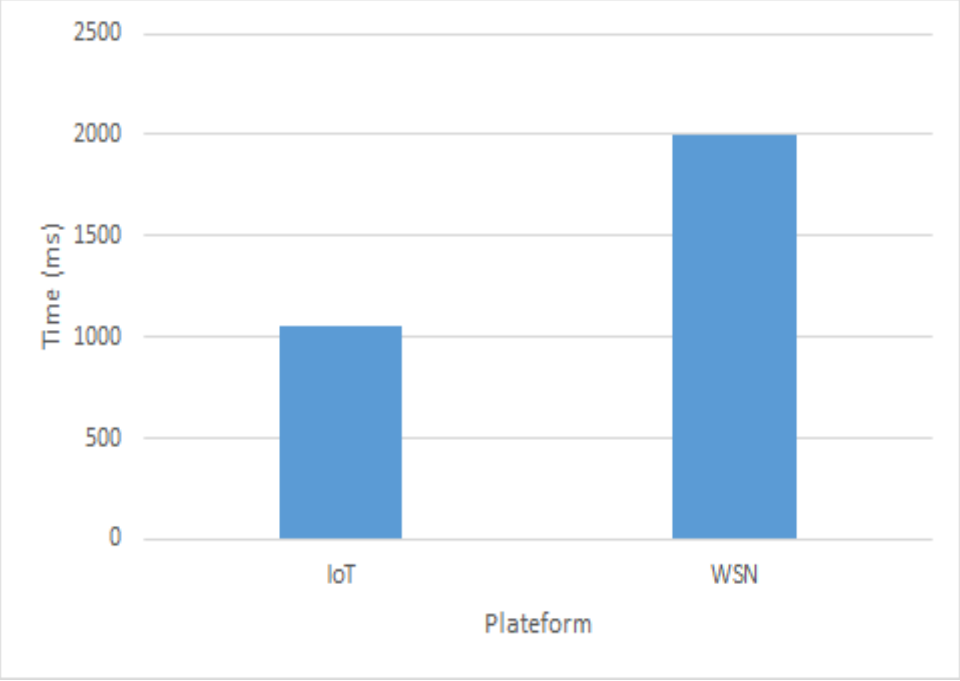

Figure 5.3: comparison IoT and WSN for key distribution

In Figure 5.3, it shows different between IoT and WSN. In WSN, key distribution for contiki OS take much time then IoT, reason behind that IoT used low complexity and low power consumption for every node and WSN is part of IoT. Like key distribution, WSN also take more time then IoT in encryption and decryption. compare to WSN, IoT is fast and low power consumption and it is more reliable then WSN. In IoT, WSN play pivotal role on bridging gap between the physical and virtual worlds.

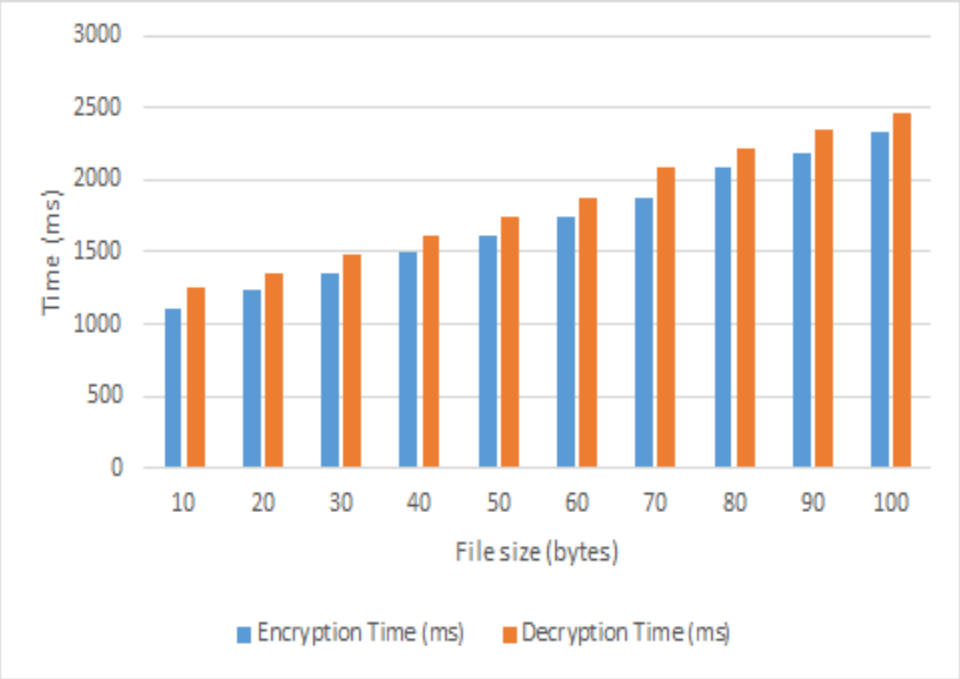

Figure 5.4: File size VS Time

The above graph each node encryption time and decryption time for every node client after distributed session keys client do encryption then server received message then decrypt message. Second graph show for 1 node to encryption and decryption depend on file size here, file size in bytes. In both graph represent as first provided algorithm distributed session key at client and server both side then start encryption at client side then decryption at server side.

As shows Figure 5.6, node number 4 encryption time is zero, because node 4 joint communication after key distribution. In this algorithm has disadvantage to once start communication if any node joint or leave in between communication, it not encrypt that node.

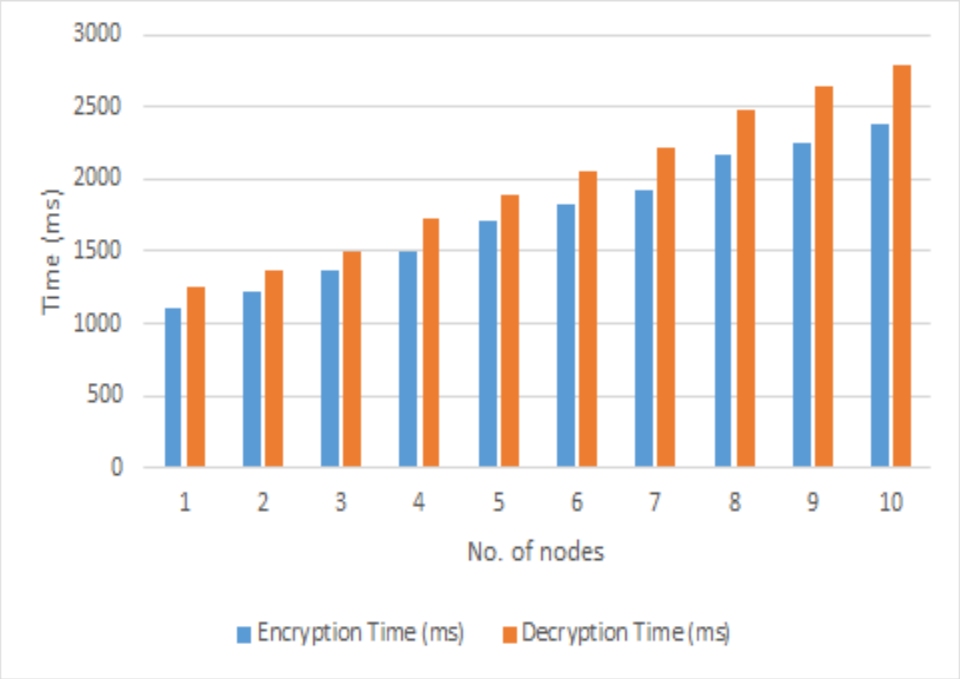

Figure 5.5: No. of node VS Time

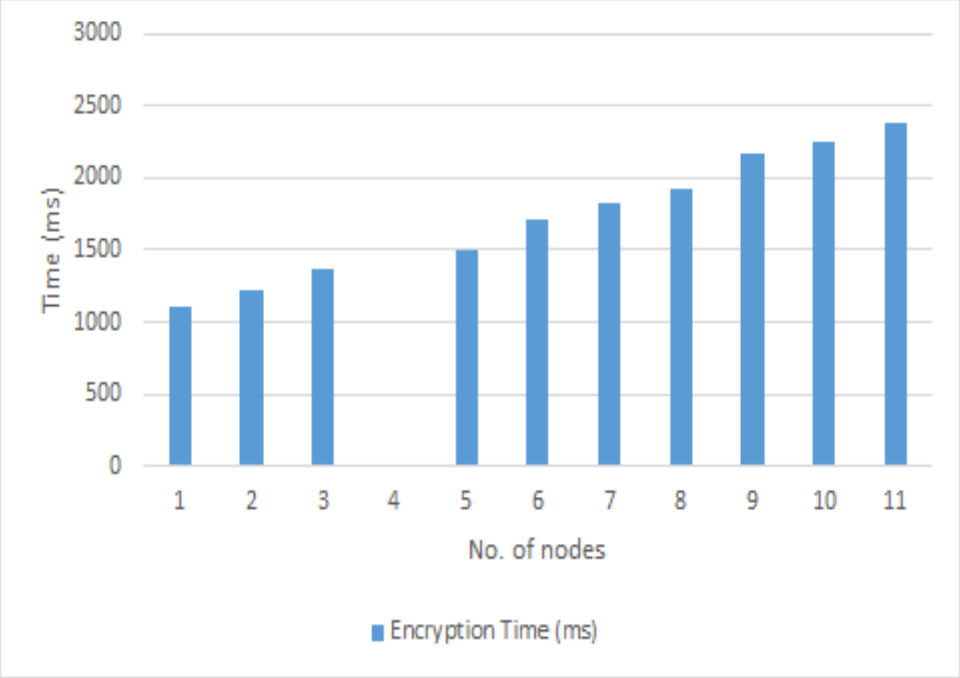

Figure 5.6: After key distribution joint one node

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

There are three major components for implementing IoT on different applications: Security, Privacy and Trust. While increasing the growth of IoT, security is more important for reliable data transferred among the billions of smart objects. In these research, we concentrate on CoAP protocol application layer protocol on IoT devices. Having light weight and consume low energy, CoAP is used on many applications of IoT. To secure data transferred, CoAP combined with DTLS protocol named as Datagram Transport Layer Security protocol as the security agent. The heavy weight of DTLS protocol used to protect the communication between smart objects on IoT.

On some of the area, DTLS may not secure for reliable data transferred and can be considered as the threat for the protocol. DTLS do not supporting for multicast messages in communications on IoT. Having a lack of security in DTLS protocol, Random session key generation stream algorithm is used to protect the communication among the object's security. The various implementation of CoAP protocols on IoT may lead towards the secure communications among smart objects.

6.2 Future Scope

This thesis focused on only cryptography technique. The main emphasis revolved around two KSA and PRGA algorithm implement in cooja simulator for securing multicast algorithm. In this algorithm have the problem like leave node and joint node. If any node joint communication that node is not encrypted at that time and if any leave commu-

nication that node is also encrypted. So, in future work to overcome that problem in securing multicast communication for IoT frame work.

Bibliography

[1] S. Kraijak and P. Tuwanut, "A survey on iot architectures, protocols, applications, security, privacy, real-world implementation and future trends," in *11th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015)*, pp. 1–6, Sept 2015.

[2] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized protocol stack for the internet of (important) things," *IEEE Communications Surveys Tutorials*, vol. 15, pp. 1389–1406, Third 2013.

[3] V. Karagiannis, P. Chatzimisios, F. Vázquez-Gallego, and J. Alonso-Zarate, "A Survey on Application Layer Protocols for the Internet of Things," *Transaction on IoT and Cloud Computing (TICC), 2015*, vol. 1, Jan. 2015.

[4] M. Brachmann, O. Garcia-Morchon, and M. Kirsche, "Security for practical coap applications: Issues and solution approaches," *Proceedings of the 10th GI/ITG KuVS Fachgespraech Sensornetze (FGSN11), Paderborn, Germany*, pp. 15–16, 2011.

[5] X. Chen, "Constrained application protocol for internet of things,"

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645 – 1660, 2013.

[7] administration, *Internet of Things*, 2016 (accessed November 3, 2016).

[8] administration, *Constrained Application Protocol*, 5 may 2017.

[9] D. Ugrenovic and G. Gardasevic, "Coap protocol for web-based monitoring in iot healthcare applications," in *2015 23rd Telecommunications Forum Telfor (TELFOR)*, pp. 79–82, Nov 2015.

[10] H. A. Khattak, M. Ruta, E. D. Sciascio, and D. Sciascio, "Coap-based healthcare sensor networks: A survey," in *Proceedings of 2014 11th International Bhurban Conference on Applied Sciences Technology (IBCAST) Islamabad, Pakistan, 14th - 18th January, 2014*, pp. 499–503, Jan 2014.

[11] "Iot messaging protocols," 31 march 2015.

[12] R. A. Rahman and B. Shah, "Security analysis of iot protocols: A focus in coap," in *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*, pp. 1–7, IEEE, 2016.

[13] M. Kovatsch, S. Duquennoy, and A. Dunkels, "A low-power coap for contiki," in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*, pp. 855–860, IEEE, 2011.

[14] M. I. D. P. Ishaq I, Hoebeke J, "Experimental evaluation of uni-cast and multicast coap group communication," *Sensors (Basel, Switzerland)*, 16.7 (2016).

[15] D. M. Mani, "Secure multicasting for wireless sensor networks," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 14, no. 11, p. 70, 2014.

[16] F. Osterlind, "A sensor network simulator for the contiki os," *SICS Technical Report*, Feb 2006.

[17] *Get Started with Contiki, Instant Contiki and Cooja*.

[18] *Contiki tutorials - Contiki*, 4 November 2016.

[19] A. Sehgal, *Using the Contiki Cooja Simulator*, 29 October 2013.