PROJECT REPORT

ON

DESIGN AND IMPLEMENTATION OF DATA SECURITY IN INTERNET OF THINGS [IoT]

Submitted By

DEBDUT DAS, MEHELI RAY, NAYONIKA PAL, NILIMA PAUL & POULAMI DAS

Under the guidance of

Prof. SANJIB MITRA

Partial fulfillment of the B.Tech [2016-2020]
Degree in Electronics & Communication Engineering
Date of Submission: **30.04.2020**

Academy of Technology
Aedconagar, Hooghly-712121

CONTENTS

LIST OF FIGURE

LIST OF TABLE

LIST OF ABBREVIATION

Acronym	Full Form
AP	Access Point
ASCII	American Code for Information Interchange
AES	Advanced Encryption Standard
DES	Data Encryption Standard
ECC	Elliptic Curve Cryptography
GPIO	General Purpose Input/Output
IDE	Integrated Development Environment
IoT	Internet Of Things
IP	Internet Protocol
IT	Information Technology
NodeMCU	Node MicroController Unit
OT	Operational Technology
PCB	Printed Circuit Board
PKCS	Public Key Cryptography Standards
RAM	Random Access Memory
RFID	Radio Frequency Identification
SoC	System-on-Chip
SSID	Service Set Identifier
TCP/IP	Transmission Control Protocol/Internet Protocol
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity

ABSTRACT

In the modern world with the growth of Internet of Things (IoT), there is free flow of data between devices and cloud. As more people and devices are plugged in, the threat continues to increase manifold. Indeed, IoT comes with revolutionary evolution, but it has been seen that a large number of the IoT devices have serious security vulnerabilities opening up limitless opportunities for the attackers and scammers. Insecure web interface and data transfers have left the users vulnerable, and as multiple devices are connected, if one is breached, all others are easily accessible. So, some major aspect which remains unrevealed is the data security in IoT. The major focus at present in on the speed of data transfer and quick execution and rapid transfer of data over the internet. On this note, there is a huge section remaining in ruins is the data security in IOT.

In this project our aim is to secure the data transferred to the network from IoT devices. So that third party hacking the unsecured data can be stopped. We will design an encryption algorithm which will simply encode the message before reaching the cloud and by using reverse encryption-decryption algorithm we will retrieve the message in the output device. Only the authenticated user will have access to the correct decryption algorithm to view the message. Any third-party access trying to access the data would not be able to get the data because the message will be converted cipher text. Once the encrypted message or cipher text reaches the authenticated user only that person has the correct decryption process to view the message.

Implementation of data security in IoT has certain constraints of power, memory and processing speed which will be required for developing an optimized algorithm for securing the data. Our motive is to obtain a perfect algorithm which balances all the constraints and provides an optimized result. This encryption-decryption process can be applied to any field of IoT based various application.

Chapter1.

OVERVIEW

In today's world the primary concern is about data of every individual and its security. With the advancement of technology and its rapid fast growth the approach is to make processing and tasks faster. Quicker algorithms to solve problems and obtain optimized result. Moreover, advancements to scale up storage for large volumes of data. It has been observed the work on data and its security is growing but the urgency and needs are not completely getting fulfilled. This project basically aims to take a step ahead in this direction of data security particularly in the field of IOT (Internet Of Things). This domain of data security in IOT has very little work being done till date. It is an approach to excavate and unearth how can we implement data security in IOT with a hope to give a head start in this technological field.

1.1 Introduction

One defining trend of this century's IT landscape will be the extensive deployment of tiny computing devices. Not only will these devices feature routinely in consumer items, but they will form an integral part of a pervasive — and unseen communication infrastructure. With a massive number of devices connected to the Internet and the huge data associated with it, there remain concerns about the security. By security we mean the degree of resistance to, or protection of the IoT infrastructure and applications. Many of these devices are easy targets for intrusion because they rely on very few outside resources and are often left unattended. It is already recognized that such deployments bring a range of very particular security risks. Yet at the same time the cryptographic solutions, and particularly the cryptographic primitives, we have at hand are unsatisfactory for extremely resource-constrained environments. The concerns on data security is increasing manifold and unless we have a deep insight, it can possess a threat to our data and its privacy.

1.2 Project Background

IoT involves adding internet connectivity to a system of interrelated computing devices, mechanical and digital machines, objects, animals and/or people. Each "thing" is provided a unique identifier and the ability to automatically transfer data over a network. Allowing devices to connect to the internet opens them up to a number of serious vulnerabilities if they are not properly protected.

A number of challenges prevent the securing of IoT devices and ensuring end-to-end security in an IoT environment. Because the idea of networking appliances and other objects is relatively new, security has not always been considered top priority during a product's design phase. Additionally, because IoT is a nascent market, many product designers and manufacturers are more interested in getting their products to market quickly, rather than taking the necessary steps to build security in from the start. A major issue cited with IoT security is the use of hardcoded or default passwords, which can lead to security breaches. Even if passwords are changed, they are often not strong enough to prevent infiltration.

Another common issue facing IoT devices is that they are often resource-constrained and do not contain the compute resources necessary to implement strong security. As such, many devices do not or cannot offer advanced security features. For example, sensors that monitor humidity or temperature cannot handle advanced encryption or other security measures. Plus, as many IoT devices are "set it and forget it" -- placed in the field or on a machine and left until end of life -- they hardly ever receive security updates or patches. From a manufacturer's viewpoint, building security in from the start can be costly, slow down development and cause the device not to function as it should.

1.3 Statement

The statement is Design and Implementation of Data Security in IOT. The fundamental approach is to design an algorithm which encrypts a given message or data of the sender who sends it through the network and is decrypted correctly by the receiver.

1.4 Objective

In this project we are going to develop a new hardware-optimized block cipher that has been carefully designed with area and power constraints uppermost in our mind. Yet, at the same time, we have tried to avoid a compromise in security. In achieving this we have looked back at the pioneering work embodied in the different cryptography algorithms which demonstrated excellent performance in hardware. Though it is accepted that stream ciphers are potentially more compact yet there are couple of reasons why block ciphers are more versatile and compact. Firstly, by running a block cipher in counter mode we get a stream cipher. Secondly, and perhaps more importantly, the art of block cipher design seems to be a little better understood than that of stream ciphers. We suspect that a carefully designed block cipher could be a less risky undertaking than a newly designed stream cipher. Thus, we feel that a block cipher that requires similar hardware resources as a compact stream cipher could be of considerable interest.

In a nutshell the basic aim is achieving an optimized algorithm to obtain a perfect encryption and decryption process which balances strong security and running time. For achieving strong algorithm requires trade off of power consumption and time complexity. Our goal was to balance out all factors to obtain our desired result.

Chapter2.

PREVIOUS WORK REVIEW

Review of the previous work consists of various scholarly paper that presents the current knowledge including substantive findings as well as theoretical and methodological contributions to this particular topic. It is mostly associated with academic-oriented literature, such reviews are found in academic journals and are not to be confused with book reviews, which may also appear in the same publication.

[1] PRESENT: An Ultra-Lightweight Block Cipher-

A. Bogdanov, G. Leander, C. Paar, A. Poschmann, Horst-G¨ortz-Institute for IT-Security, Ruhr-University Bochum, Germany, M.J.B. Robshaw, Y. Seurin, Technical University Denmark, DK-2800 Kgs. Lyngby, Denmark and C. Vikkelsoe, L.R. Knudsen Technical University Denmark, DK-2800 Kgs. Lyngby, Denmark.

With the establishment of the AES the need for new block ciphers has been greatly diminished; for almost all block cipher applications the AES is an excellent and preferred choice. However, despite recent implementation advances, the AES is not suitable for extremely constrained environments such as RFID tags and sensor networks. In this paper we describe an ultra-lightweight block cipher, present. Both security and hardware efficiency have been equally important during the design of the cipher and at 1570 GE, the hardware requirements for present are competitive with today's leading compact stream ciphers. In this paper we have described the new block cipher present. Our goal has been an ultra-lightweight cipher that offers a level of security commensurate with a 64-bit block size and an 80-bit key. Intriguingly present has implementation requirements similar to many compact stream ciphers. As such, we believe it to be of both theoretical and practical interest. Like all new proposals, we discourage the immediate deployment of present but strongly encourage its analysis.

[2] Security in Internet of Things: Challenges, Solutions and Future Directions-

Sathish Alampalayam Kumar,Tyler Vealey, Coastal Carolina University, Conway, SC, USA and Harshit Srivastava, Guru Gobind Singh Indraprastha University, New Delhi, India.

Internet of Things (IoT) is an enabler for the intelligence appended to many central features of the modern world, such as hospitals, cities, grids, organizations, and buildings. The security and privacy are some of the major issues that prevent the wide adoption of Internet of Things. In this paper, with example scenarios, we are presenting review of security attacks from the perspective of layers that comprises IoT. In addition, a review of methods that provide solutions to these issues is presented along with their limitations. To overcome these limitations, we have provided future work recommendations with a framework. Further research and implementation of the framework and our recommendations will further enhance the robustness and reliability of the IoT and their applications against a variety of known attacks. In this paper, we have articulated that as more and more IoT based devices get connected to the Internet, it results in the extension of the surface area for external attacks. We classified those attacks based on the layers that make up IoT and discussed several such attacks with examples. We have also surveyed the literature on the existing methods to protect the IoT infrastructure and summarized these security methods on how they address the security issues in the IoT. We have summarized the limitations of the existing security methods and proposed future work recommendations to overcome these limitations. In order

for the customers to embrace the IoT technologies and the applications, these privacy and security issues and limitations need to be addressed and implemented immediately, so that potential of the IoT technology and their applications can be realized.

[3] Security on Internet of Things (IOT) with Challenges and Countermeasures –

R.Vignesh and A.Samydurai, Department of Computer Science and Engineering,Valliammai Engineering College SRM Nagar, Kattankulathur-603203, Tamil Nadu, India.

This paper confers a survey and an investigates of the current status and analysis of Internet of things (IoT) security. The IoT structure pursue to append anyone with anything, anywhere. As against to the fixed Internet, in addition to humans, an IoT fastens a large number of machines, resource-coerced devices and sensors using different wired and wireless networks. An IoT normally has a three imaginary layers consisting of realization, Network, and Application layers. This paper narrates security problems within and across these layers. Many security ideas that should be implemented at each layer are also furnished. Previous work specific to enforcing security for each IoT layer and matching countermeasures are also reviewed. Finally, the paper presents future orientations for acquiring the IoT. The IoT framework is vulnearble to attacks at each layer. Therefore, there are many security threats and requirements that need to be dispatched. Current state of research in IoT is mainly concentrated on authentication and access control protocols, but with the rapid growth of technology it is essential to consolidate new networking protocols like IPv6 and 5G to achieve the progressive mash up of IoT topology. The major developments supported in IoT are mainly on small scale including within companies and in some limited industries. To scale the IoT structure from one company to a discipline of different companies and different systems, various security interests need to be addressed. The IoT has great likely to transform the way we live today. But, the foremost discipline in recognition of completely smart structures is security. If security disciplines like privacy, confidentiality, authentication, access control, end-to-end security, trust management, global policies and standards are consigned completely, then a transformation of everything by IoT can be realised in the near future. There is need for new identification, wireless, software, and hardware technologies to resolve the currently open research threats in IoT like the standards for different devices, implementation of key management and identity establishment systems, and trust management hubs.

[4] ARO_EDGE: A Technique to Ensure Data Security in Internet of Things (IoT)-

A. Vithya Vijayalakshmi and L. Arockiam, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India

Recently, e-health care, smart home, smart city, smart car and smart car services have been receiving attention all over the world. In smart health care, there are many sensors are communicating between each other and connected to the global network connection. Therefore, there is a problem in securing the data sensed from the various medical IoT

devices. Lightweight and efficient way of providing secure communication in the IoT are the need of the hour. To overcome this problem, a technique has been proposed. This paper proposes a confidentiality technique, named ARO_EDGE to secure the data in IoT devices. This proposed confidentiality technique is based on data obfuscation technique to prevent the data from the attackers and unauthorized users. This paper has proposed a technique to ensure data security in internet of things based on data obfuscation technique namely ARO_EDGE. According to the proposed technique, the data are obfuscated before they are communicated among devices or local gateway. This technique obfuscates numerical values of the sensor data collected from the healthcare IoT devices. It uses different mathematical function to operate the original text into unintelligible text. The proposed technique reduces the size of the plaintext and ensures the confidentiality of the sensor data at the edge level.

Summary

The major threat in the field of IOT is security and privacy risk. These privacy and security issues and limitations in the field of IOT need to be addressed and implemented immediately, so that potential of the IoT technology and their applications can be realized. There is also an urgent need for the development of new wireless technology which is more secured to transmit data via IOT devices. The urgency of such devices is coming up in todays world with the rapid pace of the development of IOT in various sectors. So, there is a goal to design an ultra-lightweight algorithm that offers a high level of security. This algorithm may vary but the approach to minimize the threat remains to be the prime objective. Once the algorithm is implemented then we can say a shield has been applied to the data being transmitted to through the network. Then we can say that IOT will be furnished in the modern world to its full potential.

Chapter3.

THEORETICAL BACKGROUND

The theoretical background consists of various concepts, definitions, technological details associated with our project. It mostly deals with the basic concepts of Internet Of Things, Data Security, encryption, decryption and its types, various components and software used in the implementation of the project.

3.1 Internet of Things [IoT]

The 'Thing' in IoT can be any device with any kind of built-in-sensors with the ability to collect and transfer data over a network without manual intervention. The embedded technology in the object helps them to interact with internal states and the external environment, which in turn helps in decisions making process.

Fig 3.1: Major Components of IoT

In a nutshell, IoT is a concept that connects all the devices to the internet and let them communicate with each other over the internet. IoT is a giant network of connected devices – all of which gather and share data about how they are used and the environments in which they are operated. By doing so, each of your devices will be learning from the experience of other devices, as humans do. IoT is trying to expand the interdependence in human- i.e. interact, contribute and collaborate to things. I know this sounds a bit complicated, let's understand this with an example. A developer submits the application with a document containing the standards, logic, errors & exceptions handled by him to the tester. Again, if there are any issues Tester communicates it back to the Developer. It takes multiple iterations & in this manner a smart application is created. Similarly, a room temperature sensor gathers the data and send it across the network, which is then used by multiple device sensors to adjust their temperatures accordingly. For example, refrigerator's sensor can gather the data regarding the outside temperature and accordingly adjust the refrigerator's temperature. Similarly, your air conditioners can also adjust its temperature accordingly. This is how devices can interact, contribute & collaborate.

3.2 Data Security

Data security refers to protective digital privacy measures that are applied to prevent unauthorized access to computers, databases and websites. Data security also protects data from corruption. Data security is an essential aspect of IT for organizations of every size and type. Data Security is also known as information security (IS) or computer security.

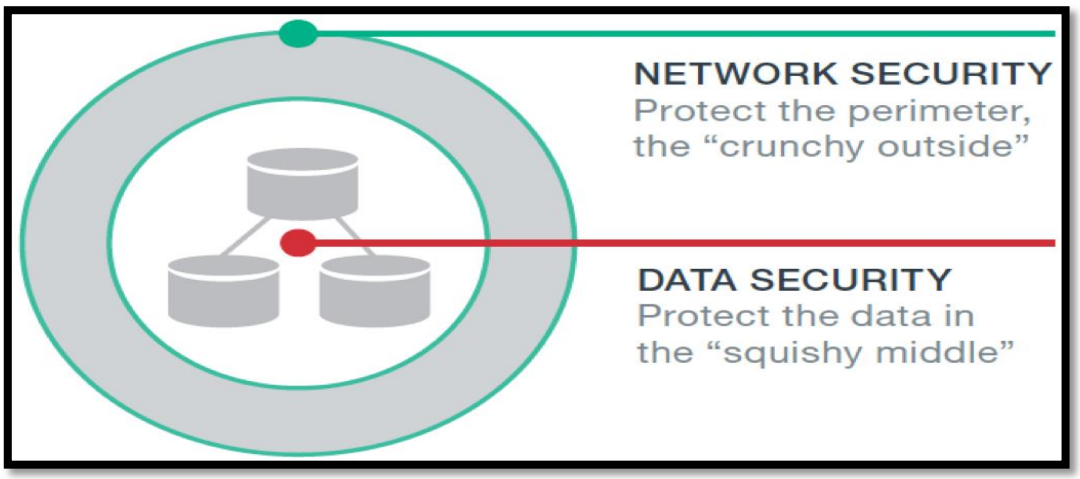


Fig 3.2: Data security

Most organizations put an immense focus on implementing endpoint, application, perimeter, and network security—and for good reason. Preventing intrusion into the network is a critical part of securing the infrastructure. Some companies see hundreds of thousands of intrusions attempts against their network every single day. But focusing only on network security is like creating a hard shell around a soft, squishy middle. The truth is, no network perimeter will ever be impenetrable. There are likely bad actors already in the network. Some of the biggest data breaches have occurred because an insider got the keys to the kingdom. And, the number of incidents involving internal factors is increasing. The numbers vary, but in general, internal factors are involved in 25 percent of all breaches. In the healthcare industry, insiders are responsible for 68 percent of breaches. Unfortunately, many systems are vulnerable to such attacks because they only have all-or-none data access rather than fine-grained security controls.

The different data security technologies are given below. Data security technology comes in many shapes and forms and protects data from a growing number of threats. Many of these threats are from external sources, but organizations should also focus their efforts on safeguarding their data from the inside, too. Ways of securing data include:

- **Data encryption**: Data encryption applies a code to every individual piece of data and will not grant access to encrypted data without an authorized key being given
- **Data masking**: Masking specific areas of data can protect it from disclosure to external malicious sources, and also internal personnel who could potentially use the data. For example, the first 12 digits of a credit card number may be masked within a database.

- **Data erasure**: There are times when data that is no longer active or used needs to be erased from all systems. For example, if a customer has requested for their name to be removed from a mailing list, the details should be deleted permanently.
- **Data resilience**: By creating backup copies of data, organizations can recover data should it be erased or corrupted accidentally or stolen during a data breach.

3.3 Data Security in IoT

IoT security is the technology area concerned with safeguarding connected devices and networks in the internet of things (IoT). The IoT has created new values by connecting various devices to the network, but has also led to security threat becoming important issues as seen in the recent reports of illegal surveillance camera manipulation and automobile hacking etc. The Information-technology Promotion Agency of Japan (IPA) has ranked "Exteriorization of vulnerability of IoT devices" as 8th in its report entitled "The 10 Major Security Threats of 2017." IoT involves adding internet connectivity to a system of interrelated computing devices, mechanical and digital machines, objects, animals and/or people. Each "thing" is provided a unique identifier and the ability to automatically transfer data over a network. Allowing devices to connect to the internet opens them up to a number of serious vulnerabilities if they are not properly protected.

A number of challenges prevent the securing of IoT devices and ensuring end-to-end security in an IoT environment. Because the idea of networking appliances and other objects is relatively new, security has not always been considered top priority during a product's design phase. Additionally, because IoT is a nascent market, many product designers and manufacturers are more interested in getting their products to market quickly, rather than taking the necessary steps to build security in from the start.

A major issue cited with IoT security is the use of hardcoded or default passwords, which can lead to security breaches. Even if passwords are changed, they are often not strong enough to prevent infiltration.

Another common issue facing IoT devices is that they are often resource-constrained and do not contain the compute resources necessary to implement strong security. As such, many devices do not or cannot offer advanced security features. For example, sensors that monitor humidity or temperature cannot handle advanced encryption or other security measures. Plus, as many IoT devices are "set it and forget it" -- placed in the field or on a machine and left until end of life -- they hardly ever receive security updates or patches. From a manufacturer's viewpoint, building security in from the start can be costly, slow down development and cause the device not to function as it should.

Connecting legacy assets not inherently designed for IoT connectivity is another security challenge. Replacing legacy infrastructure with connected technology is cost-prohibitive, so many assets will be retrofitted with smart sensors. However, as legacy assets that likely have not been updated or ever had security against modern threats, the attack surface is expanded.

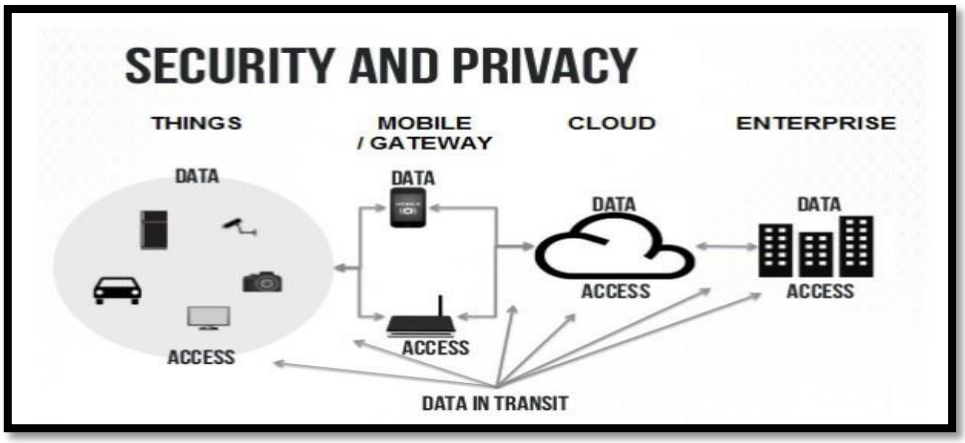


Fig 3.3: IoT security challenges

In terms of updates, many systems only include support for a set timeframe. For legacy and new assets, security can lapse if extra support is not added. And as many IoT devices stay in the network for many years, adding security can be challenging.

IoT security is also plagued by a lack of industry-accepted standards. While many IoT security frameworks exist, there is no single agreed-upon framework. Large companies and industry organizations may have their own specific standards, while certain segments, such as industrial IoT, have proprietary, incompatible standards from industry leaders. The variety of these standards makes it difficult to not only secure systems, but also ensure interoperability between them.

The convergence of IT and operational technology (OT) networks has created a number of challenges for security teams, especially those tasked with protecting systems and ensuring end-to-end security in areas outside their realm of expertise. A learning curve is involved, and IT teams with the proper skill sets should be put in charge of IoT security.

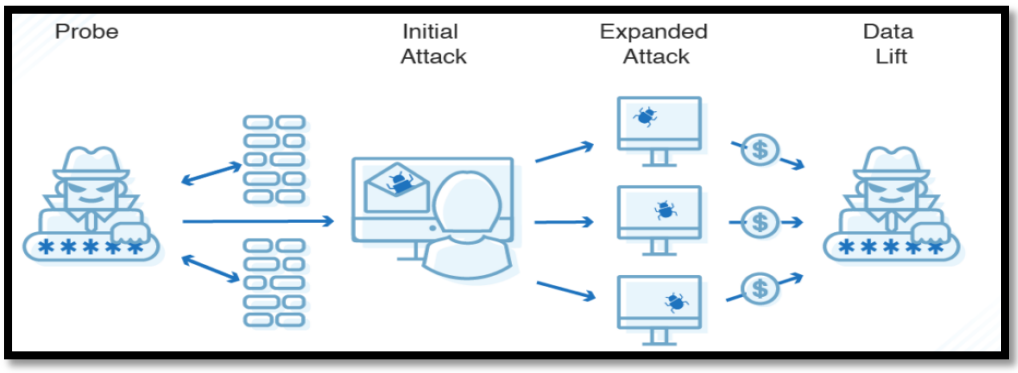


Fig 3.4: Data Breach Occurrence

Organizations must learn to view security as a shared issue, from manufacturer to service provider to end user. Manufacturers and service providers should prioritize the security and privacy of their products, and also provide encryption and authorization by default, for example. But the onus does not end there; end users must be sure to take their own

precautions, including changing passwords, installing patches when available and using security software.

3.4 Technological Details

3.4.1 Encryption

Encryption is a process which transforms the original information into an unrecognizable form. This new form of the message is entirely different from the original message. That's why a hacker is not able to read the data as senders use an encryption algorithm. Encryption is usually done using key algorithms.

Data is encrypted to make it safe from stealing. However, many known companies also encrypt data to keep their trade secret from their competitors.

Fig 3.5: Encryption Process

3.4.2 Decryption

Decryption is a process of converting encoded/encrypted data in a form that is readable and understood by a human or a computer. This method is performed by un-encrypting the text manually or by using keys used to encrypt the original data.

Fig 3.6: Decryption process

3.4.3 Key

In cryptography, a key is a variable value that is applied using an **algorithm** to a string or **block** of unencrypted text to produce **encrypted** text, or to decrypt encrypted text. The length of the key is a factor in considering how difficult it will be to decrypt the text in a given message. The different types of keys are as follows:

- **Symmetric Key:** Symmetric-key encryption are algorithms which use the same cryptographic keys for both encryption of plaintext and decryption of ciphertext.
- **Asymmetric Key:** Asymmetric encryption uses 2 pairs of key for encryption. Public key is available to anyone while the secret key is only made available to the receiver of the message. This boots security.
- **Public Key:** Public key cryptography is an encryption system which is based on two pairs of keys. Public keys are used to encrypt messages for a receiver.
- **Private Key:** Private key may be part of a public/ private asymmetric key pair. It can be used in asymmetric encryption as you can use the same key to encrypt and decrypt data.
- **Pre-Shared Key:** In cryptography, a pre-shared key (PSK) is a shared secret which was earlier shared between the two parties using a secure channel before it is used.

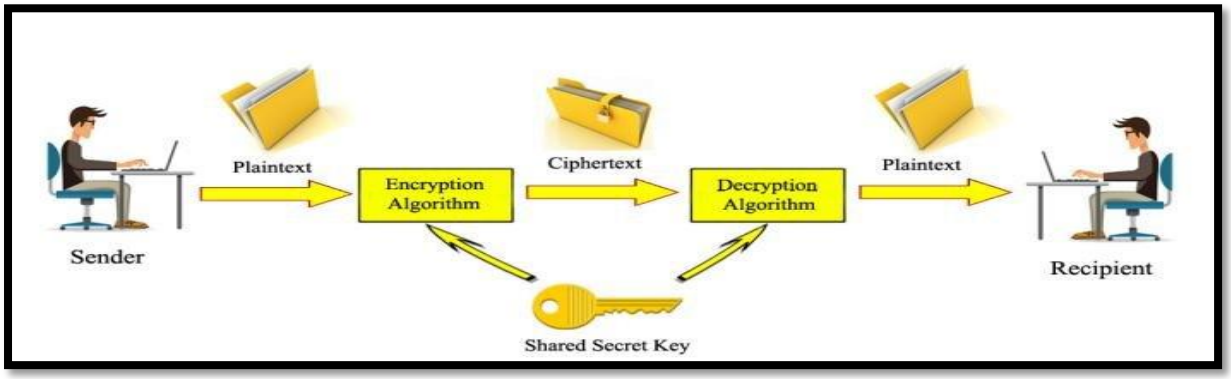


Fig 3.7: Data Security Steps

3.4.4 Symmetric Encryption

This is the simplest kind of encryption that involves only one secret key to cipher and decipher information. Symmetrical encryption is an old and best-known technique. It uses a secret key that can either be a number, a word or a string of random letters. It is a blended with the plain text of a message to change the content in a particular way. The sender and the recipient should know the secret key that is used to encrypt and decrypt all the messages. Blowfish, AES, RC4, DES, RC5, and RC6 are examples of symmetric encryption. The most widely used symmetric algorithm is AES-128, AES-192, and AES-256.The main disadvantage of the symmetric key encryption is that all parties involved have to exchange the key used to encrypt the data before they can decrypt it.

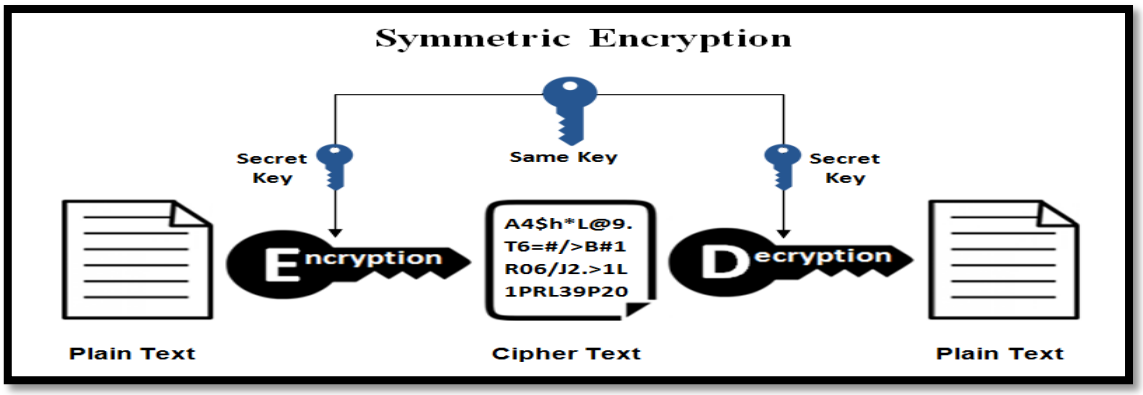


Fig 3.8: Symmetric Encryption Process

3.4.5 Asymmetric Encryption

Asymmetrical encryption is also known as public key cryptography, which is a relatively new method, compared to symmetric encryption. Asymmetric encryption uses two keys to encrypt a plain text. Secret keys are exchanged over the Internet or a large network. It ensures that malicious persons do not misuse the keys. It is important to note that anyone with a secret key can decrypt the message and this is why asymmetrical encryption uses two related keys to boosting security. A public key is made freely available to anyone who might want to send you a message. The second private key is kept a secret so that you can only know.

A message that is encrypted using a public key can only be decrypted using a private key, while also, a message encrypted using a private key can be decrypted using a public key. Security of the public key is not required because it is publicly available and can be passed over the internet. Asymmetric key has a far better power in ensuring the security of information transmitted during communication. Asymmetric encryption is mostly used in day-to-day communication channels, especially over the Internet. Popular asymmetric key encryption algorithm includes EIGamal, RSA, DSA, Elliptic curve techniques, PKCS.

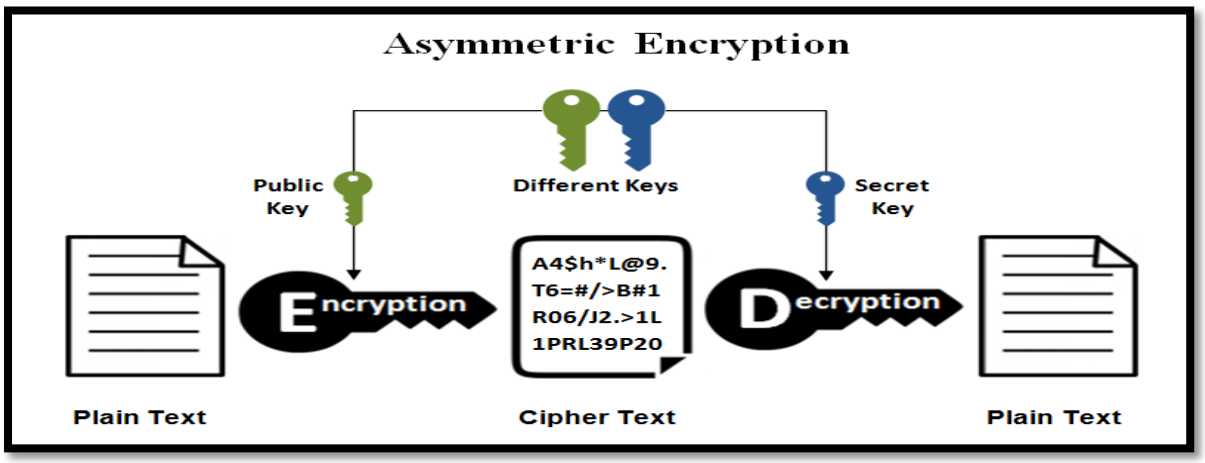


Fig 3.9: Asymmetric Encryption Process

3.4.6 Security and Privacy Risk:

Data Security in IoT using encryption and decryption process helps us to protect the confidential data such as passwords and login id. It provides confidentiality of private information. It helps to ensure that that the document or file has not been altered. Encryption process also prevents plagiarism and protects IP. It is helpful for network communication (like the internet) and where a hacker can easily access unencrypted data. It is an essential method as it helps to securely protect data that one doesn't want anyone else to have access. Some security risks and privacy risks are as follows

- IoT devices are connected to your desktop or laptop. Lack of security increases the risk of your personal information leaking while the data is collected and transmitted to the IoT device.
- IoT devices are connected with a consumer network. This network is also connected with other systems. So if the IoT device contains any security vulnerabilities, it can be harmful to the consumer's network. This vulnerability can attack other systems and damage them.
- Sometimes unauthorized people might exploit the security vulnerabilities to create risks to physical safety.
- As the mode of communication between these devices will be wireless and through the Internet, the devices will be vulnerable to eavesdropping attacks as the devices will generally be left unattended. In this attack scenario, sensors in the smart home or m-health domain that are compromised can send push notification to users and try to collect private information from the users.
- Huge chunks of data containing vital information of the user will need to be stored on storage devices or on cloud, both of which can be attacked and the data may be compromised or changed to incorrect details. The replication of the data coupled with the access of data to different types of people results in the increased surface area for the attacks.
- In IoT, devices are interconnected with various hardware and software, so there are obvious chances of sensitive information leaking through unauthorized manipulation.
- All the devices are transmitting the user's personal information such as name, address, date of birth, health card information, credit card detail and much more without encryption.

3.4.7 NodeMCU ESP8266

NodeMCU is an open source LUA based firmware developed for ESP8266 Wi-Fi chip. By exploring functionality with ESP8266 chip, NodeMCU firmware comes with ESP8266 Development board/kit i.e. NodeMCU Development board. ince NodeMCU is open source platform, their hardware design is open for edit/modify/build. NodeMCU Dev Kit/board consist of ESP8266 Wi-Fi enabled chip. The ESP8266 is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol.There is Version2 (V2) available for NodeMCU Dev Kit i.e. NodeMCU Development Board v1.0 (Version2), which usually comes in black colored PCB.NodeMCU is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-

controller unit). The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits.

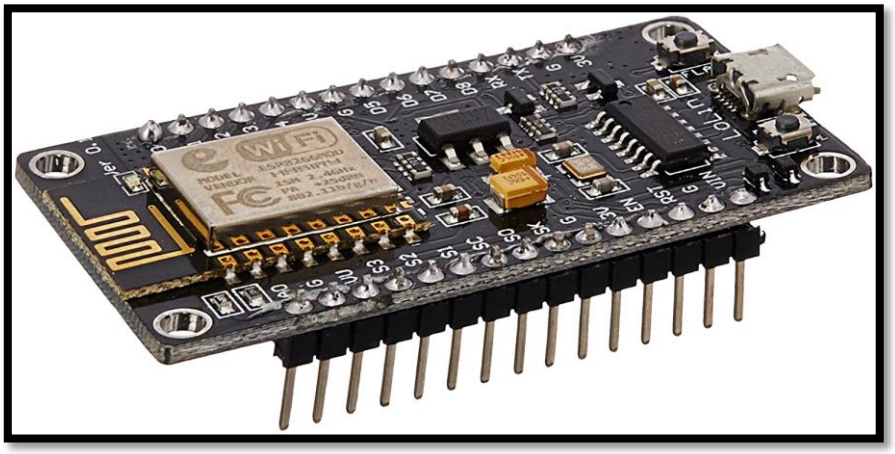


Fig 3.10: NodeMCU ESP8266

The firmware uses the Lua scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented.

The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The design was initially was based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications.

NodeMCU operates on a voltage of 3.3V.It has Wi-Fi direct[P2P], soft Access Point. It consumes current in the range of 10uA-170mA. It has a maximum flash memory of 16MB. It follows integrated TCP/IP protocol stack. It has a 32bit processor named as Tensilica L106 with a processing speed of 80MHz-160MHz. It has a RAM size of 32K+80K. It has 17 GPIO pins. The maximum concurrent TCP connections are 5.

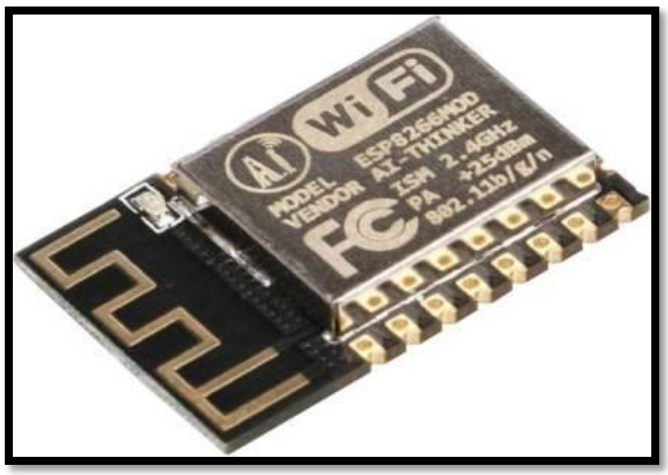


Fig 3.11: ESP8266 Wi-Fi Module

The **ESP8266** is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller capability, produced by Espressif Systems in Shanghai, China. The chip first came to the attention of Western makers in August 2014 with the **ESP-01** module, made by a third-party manufacturer Ai-Thinker. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using Hayes-style commands. However, at first there was almost no English-language documentation on the chip and the commands it accepted.[2] The very low price and the fact that there were very few external components on the module, which suggested that it could eventually be very inexpensive in volume, attracted many hackers to explore the module, the chip, and the software on it, as well as to translate the Chinese documentation. The **ESP8285** is an ESP8266 with 1 MiB of built-in flash, allowing the building of single-chip devices capable of connecting to Wi-Fi. The successor to these microcontroller chips is the ESP32, released in 2016.

3.4.8 Cost Estimation

The below table depicts the total cost incurred for this project. It is absolutely cheap in comparison to various other projects specially in IOT domain.

Component	Quantity	Cost (in rupees)	Total (in rupees)
NodeMCU ESP8266	1	395.00	395.00
USB Cable	1	30.00	30.00
		Total Cost	425.00

Table 3.1: Cost Estimation

3.4.9 Arduino IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Fig 3.12: Arduino IDE Logo

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store programs (or sketches). The sketches in the sketchbook can be opened from the **File > Sketchbook** menu or from the **Open** button on the toolbar. The first time the Arduino software is run, it will automatically create a directory for the sketchbook. One can view or change the location of the sketchbook location from with the **Preferences** dialog. Tabs, Multiple Files, and Compilation allows to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the **Sketch > Import Library** menu. This will insert one or more **#include** statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with the sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its **#include** statements from the top of the code.

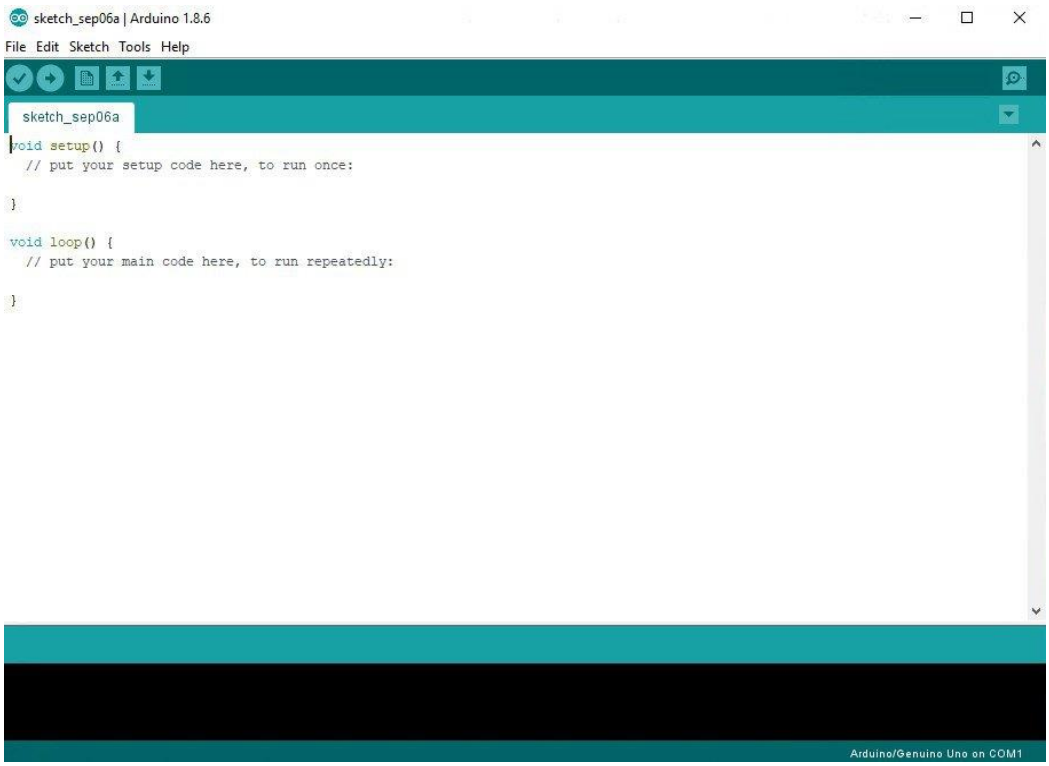


Fig 3.13 Arduino Editor to write Sketches

Serial Monitor displays serial sent from the Arduino or Genuino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial.begin in the sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when one connects with the serial monitor. Please note that the Serial Monitor does not process control characters; if the sketch needs a complete management of the serial communication with control characters, one can use an external terminal program and connect it to the COM port assigned to the Arduino board.

The board selection has two effects: it sets the parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches; and sets and the file and fuse settings used by the burn bootloader command. Some of the board definitions differ only in the latter, so even if one has been uploading successfully with a particular selection one will want to check it before burning the bootloader. Arduino Software (IDE) includes the built in support for the boards in the following list, all based on the AVR Core. The Boards Manager included in the standard installation allows to add support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, Galileo and so on.

Chapter4.

DESIGN AND SIMULATION

The basic design process incudes choice of the best algorithm for our design among various others available. Not only a perfect choice can result in a robust and optimized algorithm but also it requires one's innovation, skills and improvisations to make it a unique algorithm

4.1 Elliptic Curve Cryptography (ECC)

Elliptic curve cryptography is used to implement public key cryptography. It was discovered by Victor Miller of IBM and Neil Koblitz of the University of Washington in the year 1985. ECC popularly used an acronym for Elliptic Curve Cryptography. It is based on the latest mathematics and delivers a relatively more secure foundation than the first-generation public key cryptography systems for example RSA.

Elliptic Curves

In 1985, cryptographic algorithms were proposed based on elliptic curves. An elliptic curve is the set of points that satisfy a specific mathematical equation. They are symmetrical.

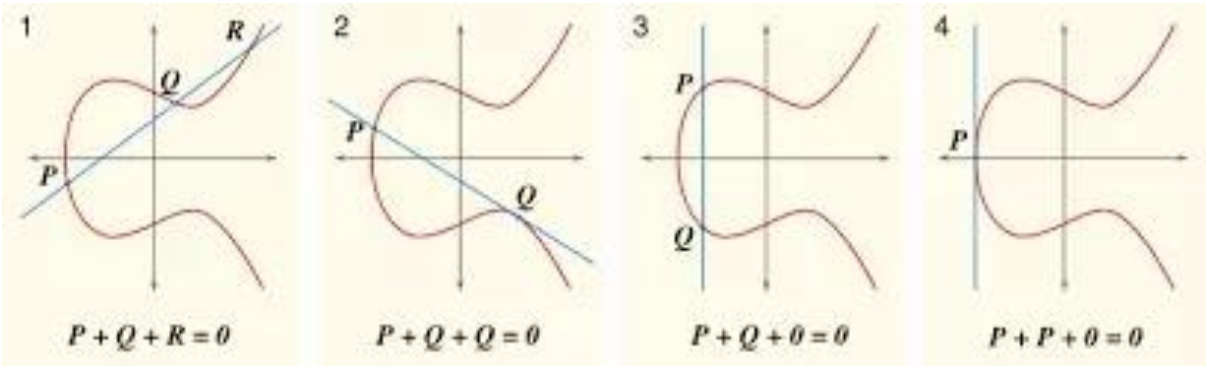


Fig 4.1 Various Types of Elliptic Curve

Uses

- Websites make extensive use of ECC to secure customers' hypertext transfer protocol connections.
- It is used for encryption by combining the key agreement with a symmetric encryption scheme.
- It is also used in several integer factorization algorithms like Lenstra elliptic-curve factorization.
- Time stamping uses an encryption model called a blind signature scheme. It is possible using Elliptic Curve Cryptography.

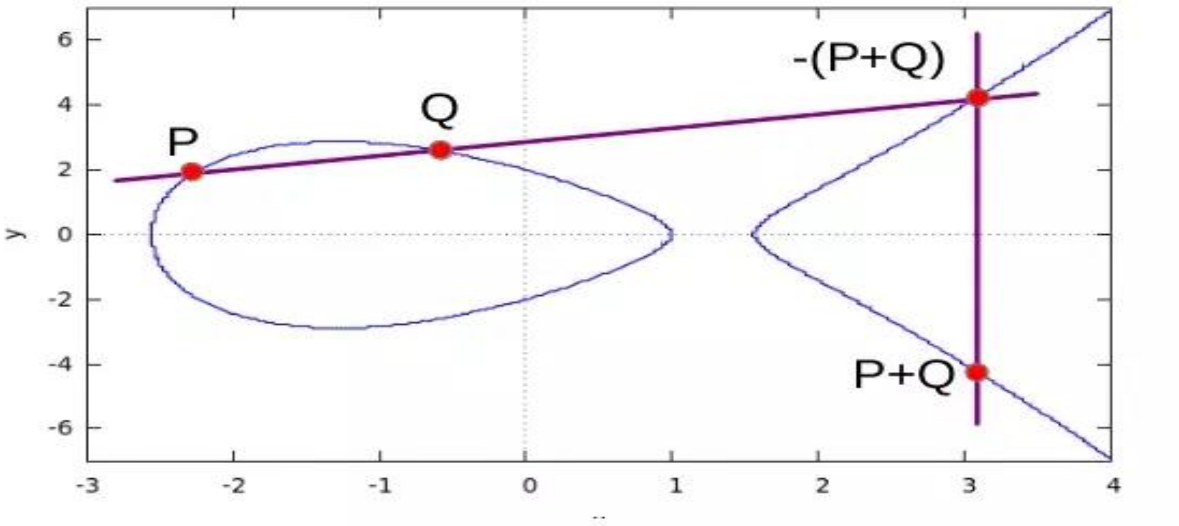


Fig 4.2 Graphical Representation of Elliptic Curve

4.2 El Gamal Encryption and Decryption Algorithm

El Gamal encryption is an public-key cryptosystem. It uses asymmetric key encryption for communicating between two parties and encrypting the message.

This cryptosystem is based on the difficulty of finding discrete logarithm in a cyclic group that is even if we know g^a and g^k, it is extremely difficult to compute g^{ak}.Idea of Suppose Alice wants to communicate to Bob.

1. Bob generates public and private key:
   - Bob chooses a very large number q and a cyclic group F_q.
   - From the cyclic group F_q, he chooses any element g and an element a such that gcd (a, q) = 1.
   - Then he computes $h = g^a$.
   - Bob publishes F, $h = g^a$, q and g as his public key and retains a as private key.

2. Alice encrypts data using Bob's public key:
   - Alice selects an element k from cyclic group F such that gcd (k, q) = 1.
   - Then she computes $p = g^k$ and $s = h^k = g^{ak.}$
   - She multiples s with M.
   - Then she sends (p, M*s) = (g^k, M*s).
   -
3. Bob decrypts the message:
   - Bob calculates $s' = p^a = g^{ak}$.
   - He divides M*s by s′ to obtain M as s = s′.

4.3 Algorithm

The optimized algorithm of the above code is being described in details. This algorithm covers all the basic variable description and analysis of detail functions and procedures implemented in our given code. Moreover, the operations on the given message and the description of the various keys are also depicted in the below algorithm. Precisely, it gives a basic understanding of the entire code which gives anyone a crystal-clear idea about the aim of the code and its implementation.

1.We initially choose two random large integer numbers q and g which are public keys.

2.gcd method finds out gcd of two numbers

3.In the genkey method we are calculating the private key as gcd(key,q) should ==1 that is co-prime numbers

4.power method calculates modular exponentiation that is power=a^b%c.

5.Now we are finding our private key using genkey method

6.Next we are calculating h another large number publically available h=g^key

7.Now we are calling the encryption function which performs as follows:

8.It receives m,q,h,g

9.it finds out its private number k=genkey(q)

10. encryption functions finds out two large numbers p=g^k and s=h^k

11.now we do some mathematical operation on the message m with s value

12. form encrypted message em

13.now encryption function has to send em and p to decryption function

14.decryption function receives em,p,key,q . em and p from encryption function q publicly available and key is the private key

15.Decryption function works as follows

16.It calculates its own h value as h=p^key

17.On some mathematical operation with the encrypted message em using h value we obtain the decrypted message which is our original message as answer.

4.4 Methods Invoked

The various functions utilized in our code is as described below. The table of functions comprises of the prototype, function description and the example. This table gives a vivid picture about all the functions used in our code and helps in proper understanding of the project in details.

Prototype	Function Description	Example
random()	The random function generates pseudo-random numbers.	int x=random(2,10);
pow()	Calculates the value of a number and raised to a power.	int j=pow(2,12);
length()	Returns the length of the String, in characters.	String s=”Hello World”; Int l=s.length();
charAt()	Access a particular character of the String.	String u=”India my native land”; ch=m.charAt(10);

void setup()	Used to initialize variables, pin modes. It run’s once after each powerup or reset of the Arduino board.	void setup()
		{
		//contents of setup elements
		}

void loop()	Runs consecutively allowing our program to change and respond	void loop()
		{
		//contents of execution //elemnsts
		}

delay()	Pauses the program for the amount of time (in milliseconds) specified as parameter	delay(1000);
millis()	Returns the number of milliseconds passed since the Arduino board began running the current program. This number will overflow (go back to zero), after approximately 50 days.	ti= millis(); Serial.print(ti);

Serial.begin()	Sets the data rate in bits per second (baud) for serial data transmission. For communicating with Serial Monitor, make sure to use one of the baud rates listed in the menu at the bottom right corner of its	void setup() {
		Serial.begin(9600);
		// opens serial port, sets data
		//rate to 9600 bps
		}
		
		void loop() {}

	screen. You can, however, specify other rates - for example, to communicate over pins 0 and 1 with a component that requires a particular baud rate. An optional second argument configures the data, parity, and stop bits. The default is 8 data bits, no parity, one stop bit.	

WiFi.begin()	Initializes the Wi-Fi library's network settings and provides the current status
	
	ssid: the SSID (Service Set Identifier) is the name of the Wi-Fi network you want to connect to
	
	pass: WPA encrypted networks use a password in the form of a string for security.
		#include <WiFi.h>
		
		//SSID of your network
		char ssid[] = "yourNetwork";
		//password of your WPA Network
		char pass[] = "secretPassword";
		void setup()
		{
		WiFi.begin(ssid, pass);
		}
		void loop ()
		{
		}

Serial.println()	Prints data to the serial port as human readable ASCII text followed by a carriage return character (ASCII 13 or'\r') and a newline character (ASCII 13 or'\r')	Serial.println(value);

server.on()	Tells the server to begin listening for incoming connections.	void setup() {
		// initialize serial:
		Serial.begin(9600);
		Serial.println("Attempting to connect to WPA network...");
		Serial.print("SSID: ");
		Serial.println(ssid);
		
		status = WiFi.begin(ssid, pass);
		if (status != WL_CONNECTED) {
		Serial.println("Couldn't get a wifi connection");

		while(true);
		}
		else {
		server.begin();
		Serial.print("Connected to wifi. My address:");
		IPAddress myAddress = WiFi.localIP();
		
		Serial.println(myAddress);
		
		}
		}

server.send()	it sends the webpage to the server for display	server.send(200, "text/html", webpage);

server.begin()	Tells the server to begin listening for incoming connections.	void setup()
		{
		Ethernet.begin(mac, ip, gateway, subnet);
		server.begin();
		}
		void loop()
		{
		EthernetClient client = server.available();
		if (client == true) {:
		server.write(client.read());
		}
		}

server.handleClient()	looks for pending requests, matches the requested address and executes the code that was registered in setup.	server.handleClient();

Table 4.1: Functions Definition

4.5 Methodology for Execution

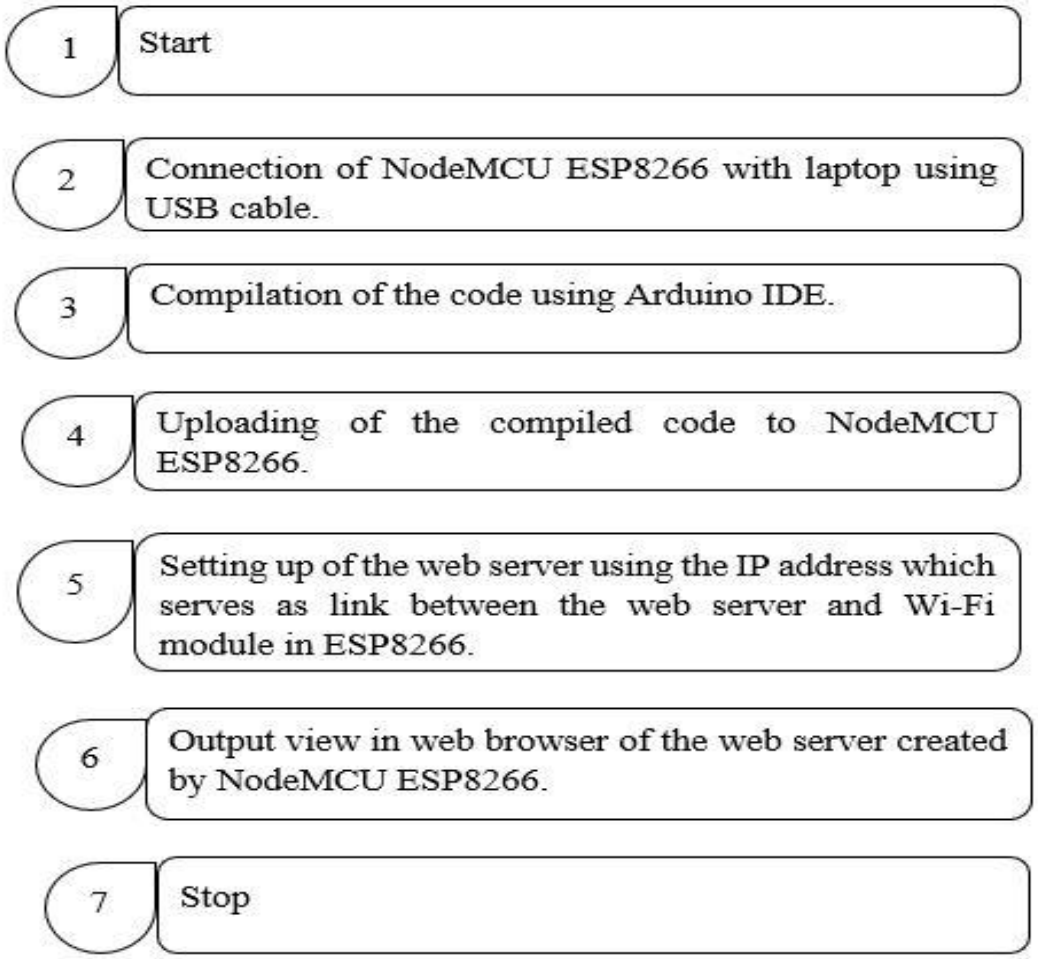


Fig 4.3: Execution Steps

Chapter5.

RESULT AND ANALYSIS

After an optimization and correct execution, it was possible to obtain the desired result and output. The result basically deals with the output html page we have designed in the Arduino code to be displayed on the webpage. This webpage is basically run on an Android based smartphone which basically provides the source of the network connection or Wi-Fi hotspot. This means the internet connection for the webpage to host is provided by the smartphone.

Each smartphone has capability of providing Wi-Fi connectivity to other devices. This technology is being used in this project. It is also known that each smartphone has a SSID and Password to get connected.

Now the first task is to provide the correct SSID and password in our given Arduino code. This will ensure that the desired mobile IP address will be used to start the web server. This will contain our output web page which will be displayed on the screen of the smart phone.

The next task is compilation of the code this will ensure there are no compilation errors in the code and it is ready for execution. Errors might come up while compilation which basically includes syntax related errors. These errors have to be resolved for further steps ahead.

It has to be ensured that the baud rate is correct. Also, it should be checked that the NodeMCU should be properly connected via the USB to the laptop. As a precaution measure the IC temperature should not be too large.

Now the code is to be uploaded to the NodeMCU which will execute it and if proper connection and other aspects are perfect then the Arduino IDE will display 100% uploaded properly. If errors occur in uploading then either there is a mismatch of USB connection. It might also happen the SSID and password are incorrect which may also result in not proper uploading of the code.

It is to be noted that proper IP address should be hit on the web browser or else proper simulation output cannot be obtained.

Also it has to be ensured that the network connection should have proper internet facility otherwise web server will not be started.

Now, after correct execution of the code the serial monitor will display connected to Wi-Fi network successful and now on entering the IP of the smartphone in the web browser of smartphone, we will get the desired output on screen.

5.1 Output at Web Server End

The output snippet of the given Arduino code is as follows:

Message displayed on Webserver: Encryption without Decryption

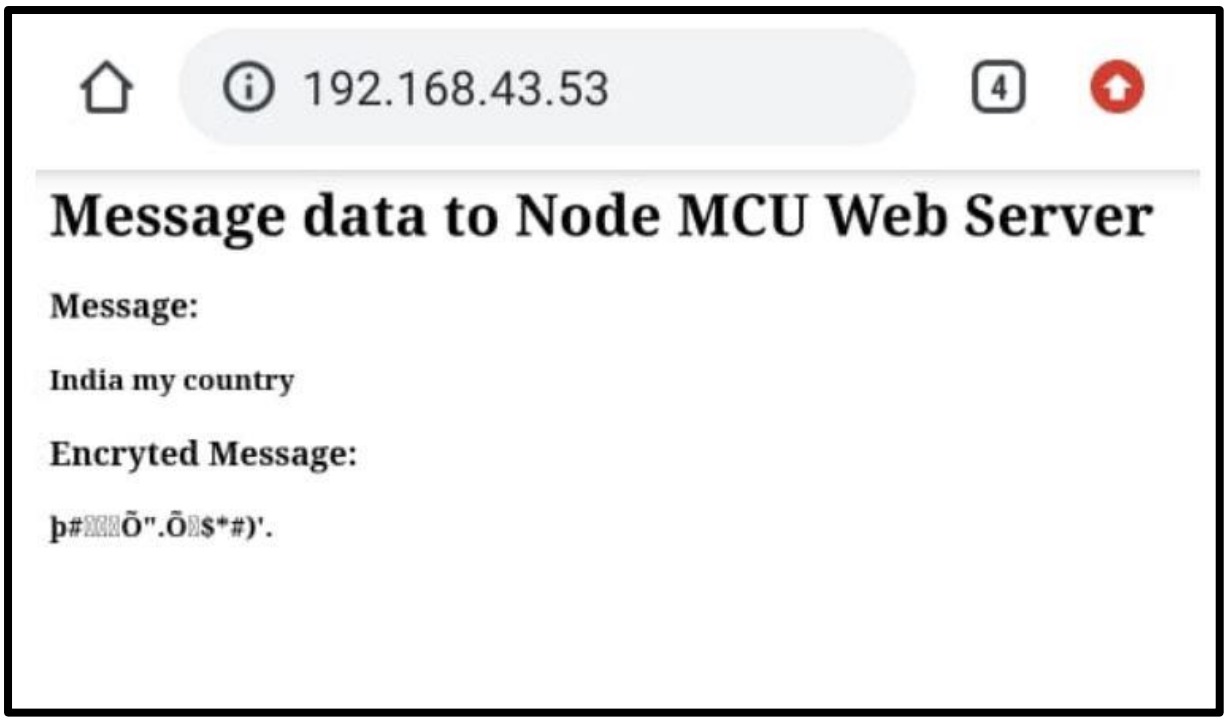


Fig 5.1: Output 1: Encryption without Decryption

Now on running the code for encryption with decryption it is observed as follows

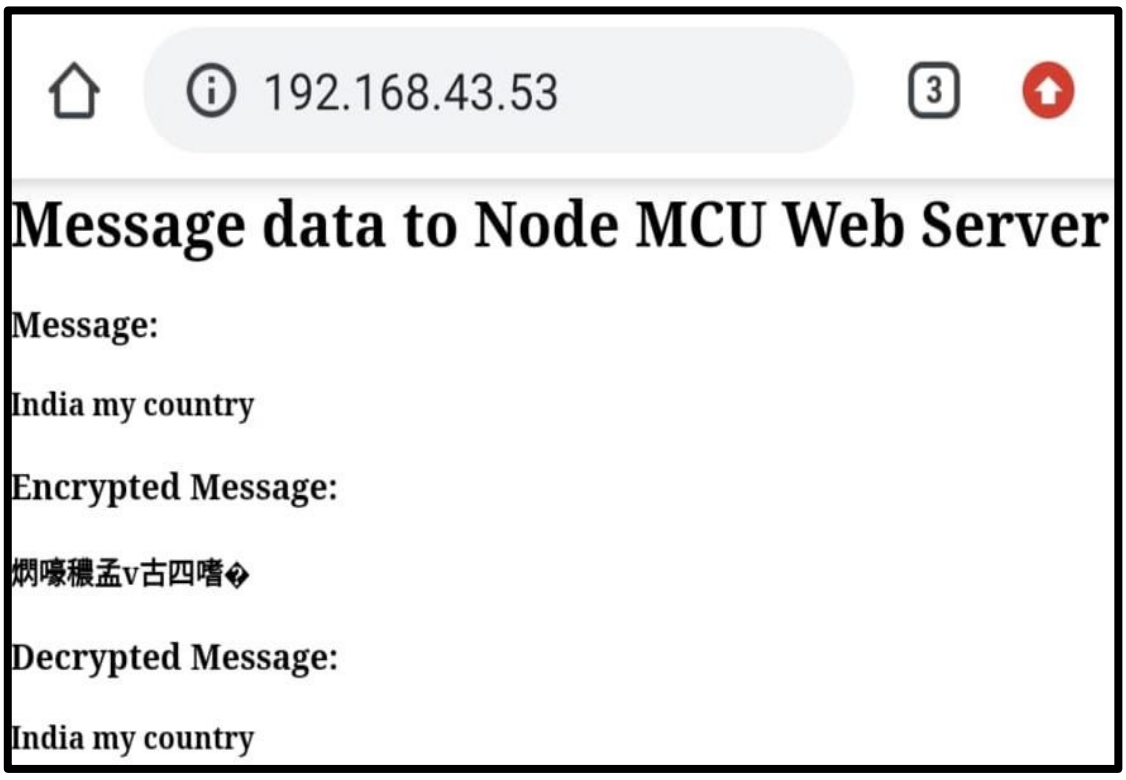


Fig 5.2: Output2: Encryption with Decryption

5.2 Discussion

The various analysis which can been done on the outputs obtained in the above two snippets is as follows

- The webpage shows the IP address of the utilized network of the smartphone indicating the NodeMCU has created the webpage and started it. This is clearly evident from the snippet.
- The two outputs obtained as run on two different time intervals is proven by the encrypted message.
- It can be observed closely that the encrypted message at two different intervals of time are completely different which suggests that the way of encryption is different at different execution time.
- We can also infer the fact that different way of encryption indicates different key generations both the private and public key at each execution time. This ensures that the hacking of the key is very tough.
- If on each time of execution, it is found that the key changes from one particular value to another then it is almost impossible to decode by a bob at a later stage.
- Even if by chance the bob or thief trying to hack the code gets the key after the complete operation is performed that key is of no use in second execution because the second execution starts with a newly generated key. And encryption occurs with the new key. So there is no functioning of the previous key. This ensures a high level of security.
- Also, the IP of the desired smartphone should be hit on the web browser launched by that smartphone. Any third party if try to access this data will not be able to do so. This ensures a great deal of security.
- From the output we can find out the original message, the encrypted message and the decrypted message which shows the desired result.
- We find that the output is obtained on the web browser as soon as the uploading of the code is complete.
- The html page displayed on the webpage is a simple html page with header tags for displaying the original encrypted and decrypted message.

5.3 Performance Analysis

The performance analysis basically deals with the entire evaluation of how the code performs. It also covers the details about the total time requirement. The total space requirement. It also gives an idea about the clock speed and the scalability.

- Memory usage: For both client and server, this is an improved experience, streamlining the connection and simplifying the process. Space required is 3 KB.
- Time requirement: Time complexity of the code: 21973 milliseconds to 45193 milliseconds using the time() function in Arduino IDE which returns the number of

milliseconds passed since the Arduino board began running the current program. The running time varies a bit due to the certain uploading delay at some execution start time.

- Power requirement: ECC consumes less computing power and battery resource. As, a result it is quite effective in terms of power requirement. Power requirement is 3.3V.
- Clock speed: Clock speed is 80 MHz
- Scalability: The ECC algorithm works on Elliptic Curve Discrete Logarithm Problem (ECDLP) that is hard to crack for hackers. Less network overhead, allowing faster performance and a better customer or user experience. A slower growth in bit size over time, which makes it more scalable, potentially, for the Internet of Things.

This above analysis shows that the space requirement is negligible in today's world. The time complexity is an important factor in determining the running time of the Arduino code. It is quite a justified time requirement for the entire process of encryption, decryption and web server activity. The clock speed is also quite acceptable leading to fast response of NodeMCU for the uploading and executing purpose.

On a total analysis it is depicted and proven that that total performance is quite effective and optimized. Eventually the objective of designing lightweight with good security has been achieved to a great extent.

Chapter6.

FUTURE WORK AND CONCLUSION

6.1 Future Work

The Arduino code basically has a fixed length which is quite large but still it is always recommended to increase the length of the key as long as possible. This will ensure that the various attacks and hacking of the key by any bob or thief minimizes to a large extent. So, one of the future approaches is to make the key as strong as possible. This will create great impact on the data security which is the prime focus of our project. So, more work and research will be done in future for the analysis of a highly secured key.

Although this will have a trade-off with the power consumption of the NodeMCU. So there has to be a balance between security and the power utilization. Basically, at this moment the key size and power consumption both are optimized and a good balance of these two performance factors.

In terms of time complexity, we have almost achieved our desired result and minimized the time as far as possible. On the other hand, the space complexity is negligible which does not require much attention in the future. The processing speed is also fast and NodeMCU is the ultimate choice. The only prime motive is to increase the size of the keys both public and private. Now this can also impact a little on the running time of the code, its execution and also its uploading to the NodeMCU.

It is not possible to make a complete efficiency in terms of all the factors but with trial and experimentation a much more optimized security in hands with the power consumption can be achieved.

6.2 Conclusion

It's no secret that IoT security is a problem. That's why there are so many regulations and initiatives aimed at fixing the issue. But even with the right measures in place, networking professionals still need to be careful how they deploy IoT. To those ends, a number of best practices have been published to guide IoT deployments. IoT security is divided into three parts specifically:

Securing Devices includes making hardware tamper resistant, providing for firmware updates/patches, performing dynamic testing, specifying procedures to protect data on device disposal. Securing Networks includes strong authentication, use strong encryption and secure protocols, minimizing device bandwidth, dividing networks into segments Securing the Overall System includes Protecting sensitive information, encouraging ethical hacking and discouraging blanket safe harbour, instituting an IoT Security and Privacy Certification Board.

Thus, we believe that with this project the threats which the world is suffering to data security in IoT can be minimized to a good extent. It would also create an impact on taking significant approach in protecting data to a high level for IoT devices such that they are not easily

accessible to the third parties. To send and receive data at a fast pace is not only the primary concern but also protecting it is a duty while implanting new technologies in IoT. In a nutshell, our project will serve as the basic steps in achieving our goal to secure data and restrict free access of it in the future aspect.

REFERENCES

[1] PRESENT: An Ultra-Lightweight Block Cipher

[2] Security in Internet of Things: Challenges, Solutions and Future Directions

[3] Security on Internet of Things (IOT) with Challenges and Countermeasures

[4] ARO_EDGE: A Technique to Ensure Data Security in Internet of Things (IoT)

[5] https://www.stoodnt.com/blog/future-trends-in-cyber-security/

[6] https://www.guru99.com/difference-encryption-decryption.html

[7] https://www.geeksforgeeks.org/elgamal-encryption-algorithm/

[8] https://www.arduino.cc/en/main/software

[9] https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf

[10] https://nodemcu.readthedocs.io/en/master/

[11]https://internetofthingsagenda.techtarget.com/definition/IoT-security-Internet-of-Things-security

[12]https://www.researchgate.net/publication/305631381_Data_Security_and_Privacy_in_the_Internet_of_Things_IoT_Environment

[13] https://www.arduino.cc/reference/en

[14] https://www.techopedia.com/definition/26464/data-security

APPENDIX: A

NodeMCU ESP8266 Specification-

• Voltage:3.3V.

• Wi-Fi Direct (P2P), soft-AP.

• Current consumption: 10uA~170mA.

• Flash memory attachable: 16MB max (512K normal).

• Integrated TCP/IP protocol stack.

• Processor: Tensilica L106 32-bit.

• Processor speed: 80~160MHz.

• RAM: 32K + 80K.

• GPIOs: 17 (multiplexed with other functions).

• Analog to Digital: 1 input with 1024 step resolution.

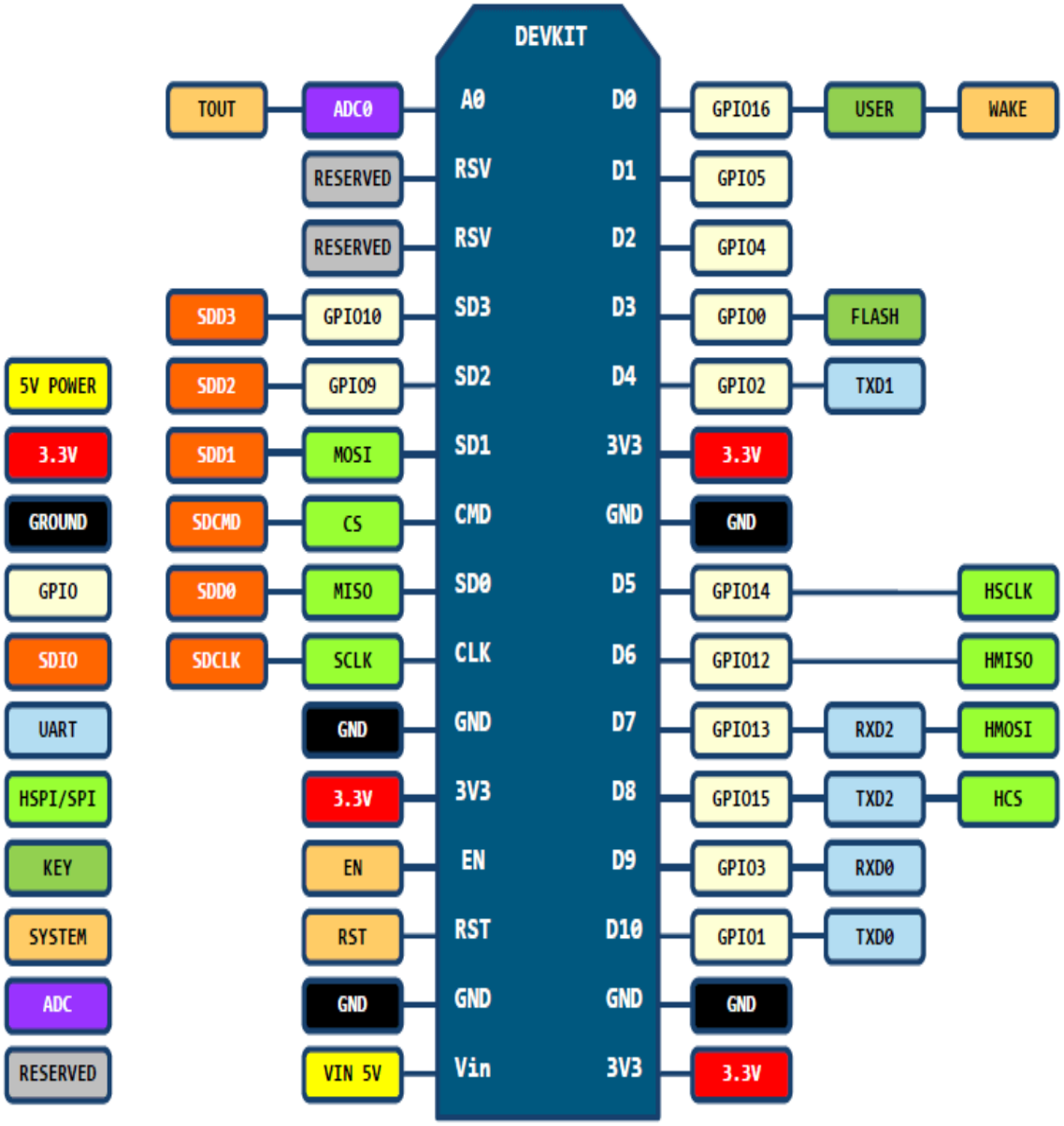
• +19.5dBm output power in 802.11b mode

• 802.11 support: b/g/n.

• Maximum concurrent TCP connections: 5.

APPENDIX: B

NodeMCU ESP8266 Pin Configuration-



APPENDIX: C

NodeMCU ESP8266 Pin Description-

Pin Names on NodeMCU Development Kit	ESP8266 Internal GPIO Pin number
D0	GPIO16
D1	GPIO5
D2	GPIO4
D3	GPIO0
D4	GPIO2
D5	GPIO14
D6	GPIO12
D7	GPIO13
D8	GPIO15
D9/RX	GPIO3
D10/TX	GPIO1
D11/SD2	GPIO9
D12/SD3	GPIO10