ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Encryption and decryption Implementation method using network socket programming

Vidhi B patel and Chandresh Parekh
Cyber Security Dept., Raksha-Shakti University
Ahmadabad, India

Reena M Patel
Electronics and Communication Dept., LDRP – Institute of Technology and Research, India

Abstract: Like mobile phones, a Wi-Fi network makes use of radio waves to transmit information across a network. The computer should include a wireless adapter that will translate data sent into a radio signal. This same signal will be transmitted, via an antenna, to a decoder known as the router.The ESP8266 is the name of a micro controller designed by Espressif Systems. Espressif is a Chinese company based out of Shanghai. The ESP8266 advertises itself as a self-contained Wi-Fi networking solution offering itself as a bridge from existing micro controller to Wi-Fi and is also capable of running self-contained applications. The number of GPIO pins exposed, the amount of flash memory provided, the style of connector pins and various other considerations related to construction. A socket is one endpoint of a two-way communication link between two programs running on the network. Main focus in this project work is to develop high security for data transferring process using encryption and decryption customize method.

Keywords: ESP8266; Wi-Fi; AT-Commands; Access point, socket

I. INTRODUCTION

Wi-Fi stands for Wireless Fidelity. Wi-Fi It is based on the IEEE 802.11 family of standards and is primarily a local area networking (LAN) technology designed to provide in-building broadband coverage.Wi-Fi is a technology that allows electronic devices to connect to a wireless LAN (WLAN) network, mainly using the 2.4 gigahertz (12 cm) UHF and 5 gigahertz (6 cm) SHF ISM radio bands. One significant advantage of Wi-Fi over WiMAX and 3G is its wide availability of terminal devices. A vast majority of laptops shipped today have a built-in Wi-Fi interface.

Wi-Fi modes:

ACCESS POINTS (APs) - devices performing wireless-to-wired bridging function.

STATION POINTS (STAs) - device with wireless network interface communicating with other similar devices via Aps. [2]

Wi-Fi standards

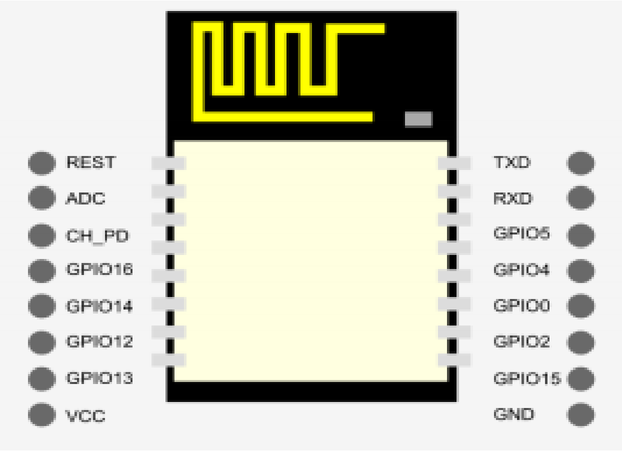
- 802.11a technology has a range of 5.725 GHz to 5.850GHz with a data rate of 54Mbps.
- 802.11b with a data rate of 11Mbps at 2.4GHz
- 802.11e addresses Qos issues and is excellent for streaming, quality of video, audio and voice channels.
- 802.11f addresses multivendor interoperability
- 802.11g deals with higher data rate extension to 54Mbps in the 2.4GHz.
- 802.11h deals with dynamic frequency selection and transmit power control for operation of 5GHz products.
- 802.11i addresses enhanced security issues.
- 802.11j addresses channelization in Japan's 4.9GHz band.
- 802.11k enables medium and network resources more efficiently.
- 802.11 deals with Wireless Network Management which is still in progress. [1]

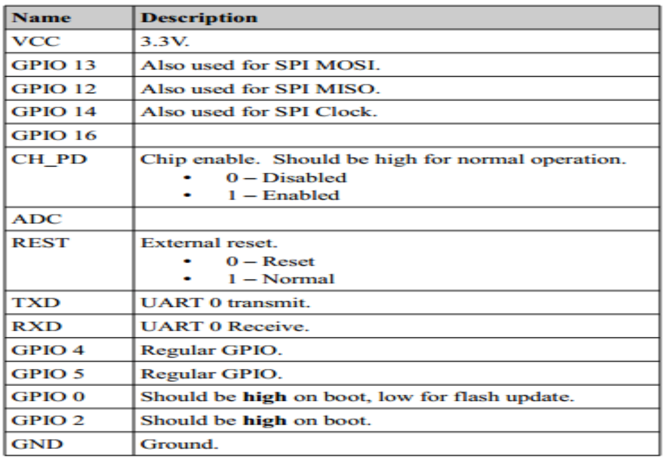
ESP8266 Wi-Fi module

ESP8266 Modules the ESP8266 integrated circuit comes in a small package, maybe five millimetres square. The ESP8266 is designed to be used with a partner memory module and this is most commonly flash memory. There are a variety of board styles available. The two that I am going to focus on have been given the names ESP-1 and ESP-12. It is important to note that there is only one ESP8266 processor and it is this processor that is found on ALL breakout boards.

ESP-12 is the current most popular and flexible configuration available today. It exposes the most GPIO pins for use. The basic ESP-12 module really needs its own expander module to make it breadboard and 0.1" strip board friendly.

Figure 1 ESP8266 module[3]

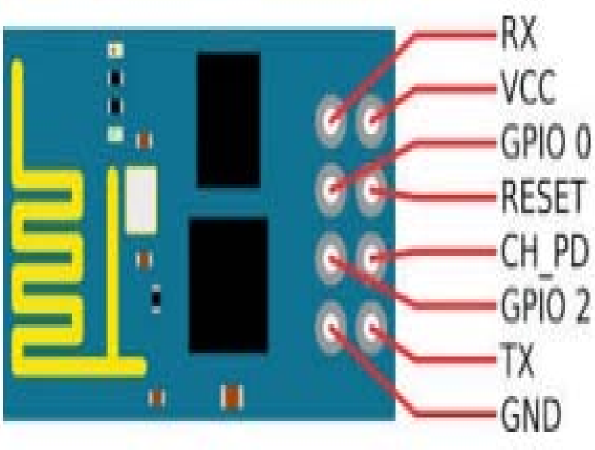


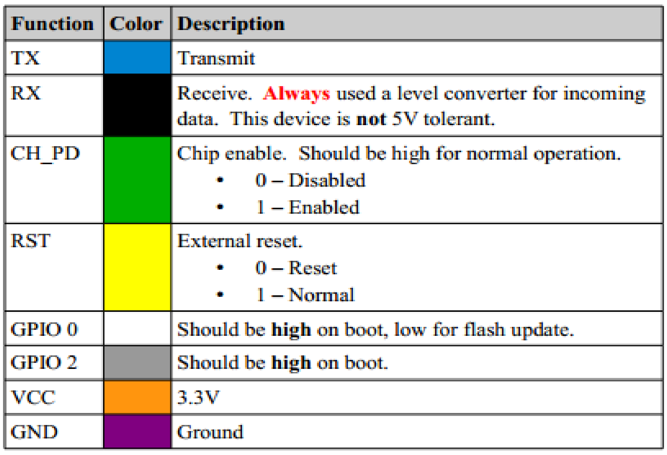


Name	Description
VCC	3.3V.
GPIO 13	Also used for SPI MOSI.
GPIO 12	Also used for SPI MISO.
GPIO 14	Also used for SPI Clock.
GPIO 16	
CH_PD	Chip enable. Should be high for normal operation. • 0 – Disabled • 1 – Enabled
ADC	
REST	External reset. • 0 – Reset • 1 – Normal
TXD	UART 0 transmit.
RXD	UART 0 Receive.
GPIO 4	Regular GPIO.
GPIO 5	Regular GPIO.
GPIO 0	Should be **high** on boot, low for flash update.
GPIO 2	Should be **high** on boot.
GND	Ground.

Figure 2 ESP12 module pin discription [7]

ESP-1 board is an ESP8266 on an 8 pin board. It is not at all breadboard friendly but fortunately we can make adapters for it extremely easily.





Function	Color	Description
TX	🟦	Transmit
RX	⬛	Receive. **Always** used a level converter for incoming data. This device is **not** 5V tolerant.
CH_PD	🟩	Chip enable. Should be high for normal operation. • 0 – Disabled • 1 – Enabled
RST	🟨	External reset. • 0 – Reset • 1 – Normal
GPIO 0	⬜	Should be **high** on boot, low for flash update.
GPIO 2	▨	Should be **high** on boot.
VCC	🟧	3.3V
GND	🟪	Ground

Figure 3 ESP1 module pin description [7]

Connecting to the ESP8266

The ESP8266 is a Wi-Fi device and hence we will eventually connect to it using Wi-Fi protocols but some bootstrapping is required first. The device doesn't know what network to connect to, which password to use and other necessary parameters. This of course assumes we are connecting as a station, if we wish the device to be an access point or we wish to load our own applications into it, the story gets deeper.This implies that there is a some way to interact with the device other than Wi-Fi. The ESP8266 has a dedicated UART interface with pins labeled TX and RX. The TX pin is the ESP8266 transmission (outbound from ESP8266) and the RX pin is used to receive data (inbound into the ESP8266). These pins can be connected to a UART (serial) partner. This is used extensively when working with the AT commands.

A second purpose of the UART is to receive binary data used to "flash" the flash memory of the device to record new applications for execution. When the ESP8266 is performing the role of an access point, it is likely that you will want it to also behave as a DHCP server so that connecting stations will be able to be automatically assigned IP addresses and learn their subnet masks and gateways. TCP/IP is the network protocol that is used on the Internet. It is the protocol that the ESP8266 natively understands and uses with Wi-Fi as the transport. [3]

AT Command Programming

The quickest and easiest way to get started with an ESP8266 is to access it via the AT command interface. When we think about an ESP8266 device we find that it has a built in UART (Serial) connection. This means that it can both send and receive data using the UART protocol. We also know that the device can communicate with Wi-Fi. This would then allow us to use the ESP8266 without ever having to know the programming languages that are native to the device. This is exactly what a program that has so far been found to be pre-installed on the ESP8266 does for us. The program is called the "AT command processor" named after the format of the commands sent through the serial link. All "AT" instructions end with the "\r\n" pair.

The services of the ESP8266 as a client and the ESP8266 as a server capable of servicing those commands as a server, then the client sends strings of characters through the UART connection to the server and server responds with the outcome. [3].

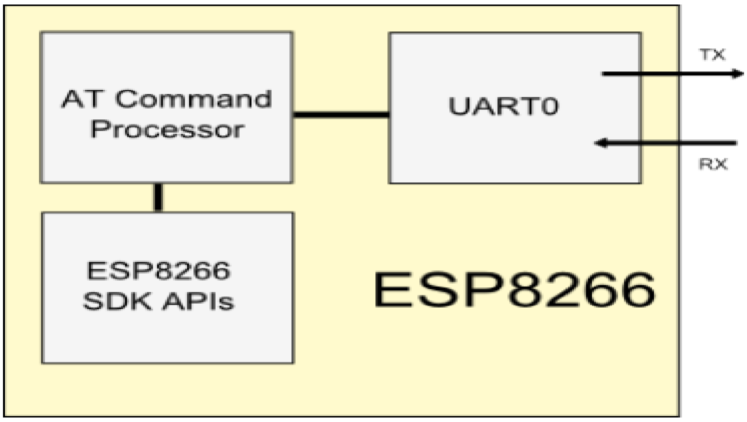


Figure 4 At command process [3]

The services of the ESP8266 as a client and the ESP8266 as a server capable of servicing those commands as a server, then the client sends strings of characters through the UART connection to the server and server responds with the outcome. [4]

The following command mostly use for Wi-Fi connection:

1 AT

2 AT+WD

3 AT+NSET = ipadress

4 AT+CWMODE_CUR=<mode>

5 AT+RST [9]

What is socket programming

A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to. An endpoint is a combination of an IP address and a port number.

A socket is just a logical endpoint for communication. They exist on the transport layer. You can send and receive things on a socket, you can bind and listen to a socket. A socket is specific to a protocol, machine, and port, and is addressed as such in the header of a packet.[8]

II. PROPOSED METHOD

We study different paper and manual, then we understand and implement the more security and encryption method for Wi-Fi modules for secure network. We implement Wi-Fi security like password security, data security, easy GOI programming, platform option.

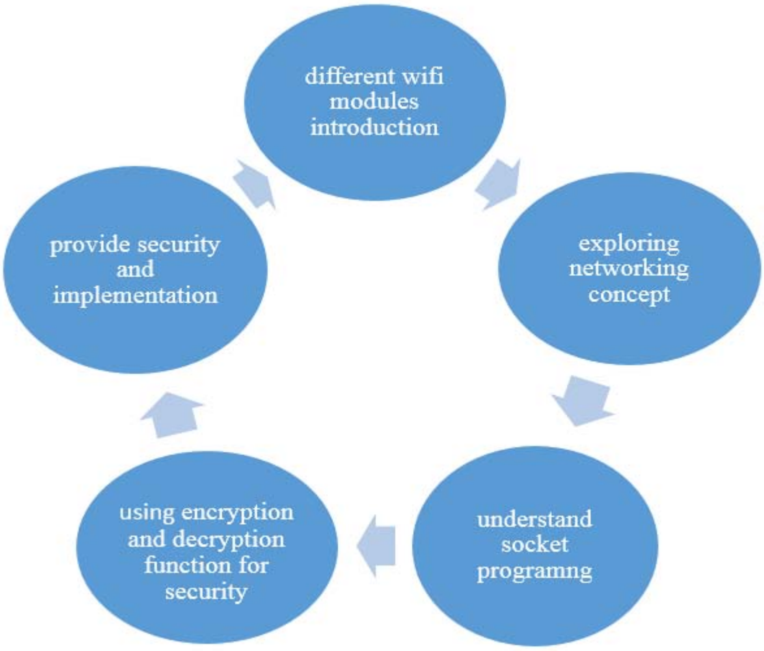

different wifi modules introduction
exploring networking concept
understand socket programng
using encryption and decryption function for security
provide security and implementation

Figure 5 diagram of project work

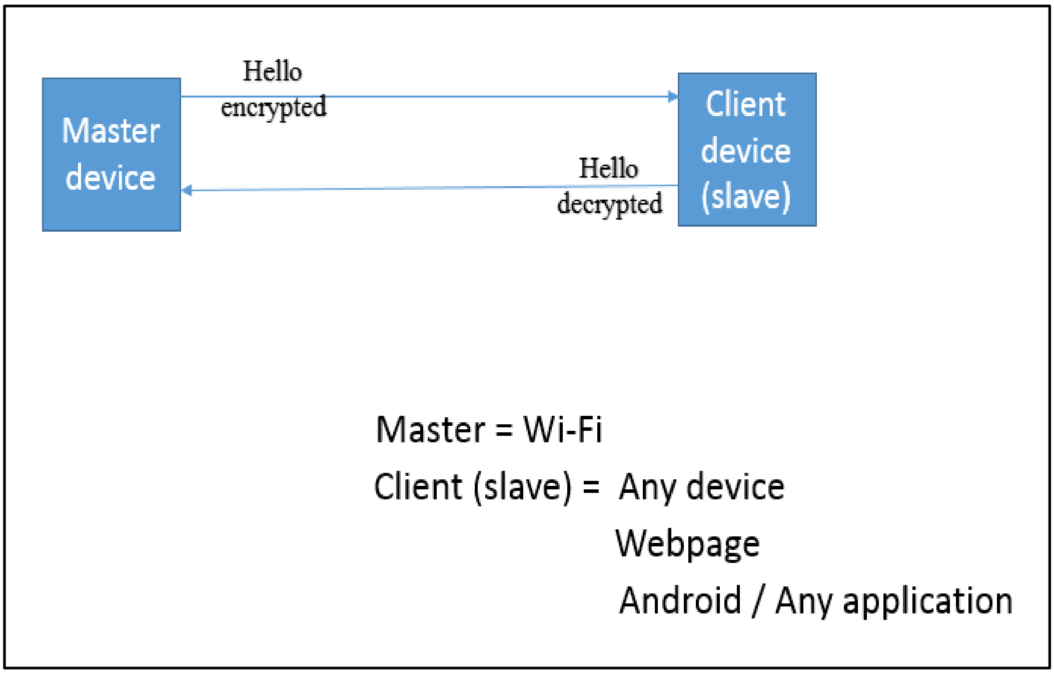


Figure 6 Master and Client device transferring secure data

I. Master device means any Wi-Fi module which can be used by ESP8266.

II. Master device send data in encrypted form to client device (slave).

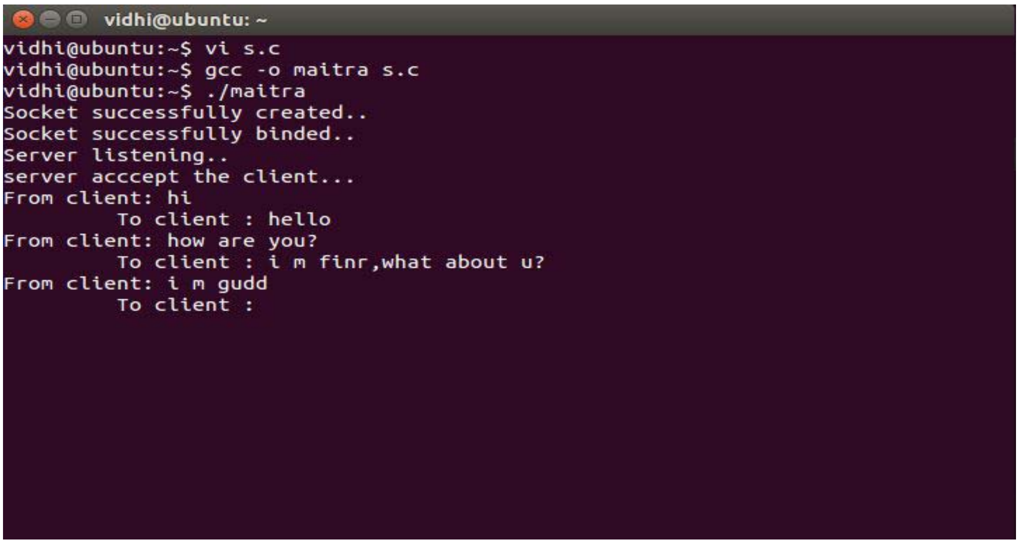
III. Client (slave) device receive data from master device in encrypted form then client convert data in decrypted form (original form).

IV. This type communication Master and Client secure connection using encryption and decryption socket programming.

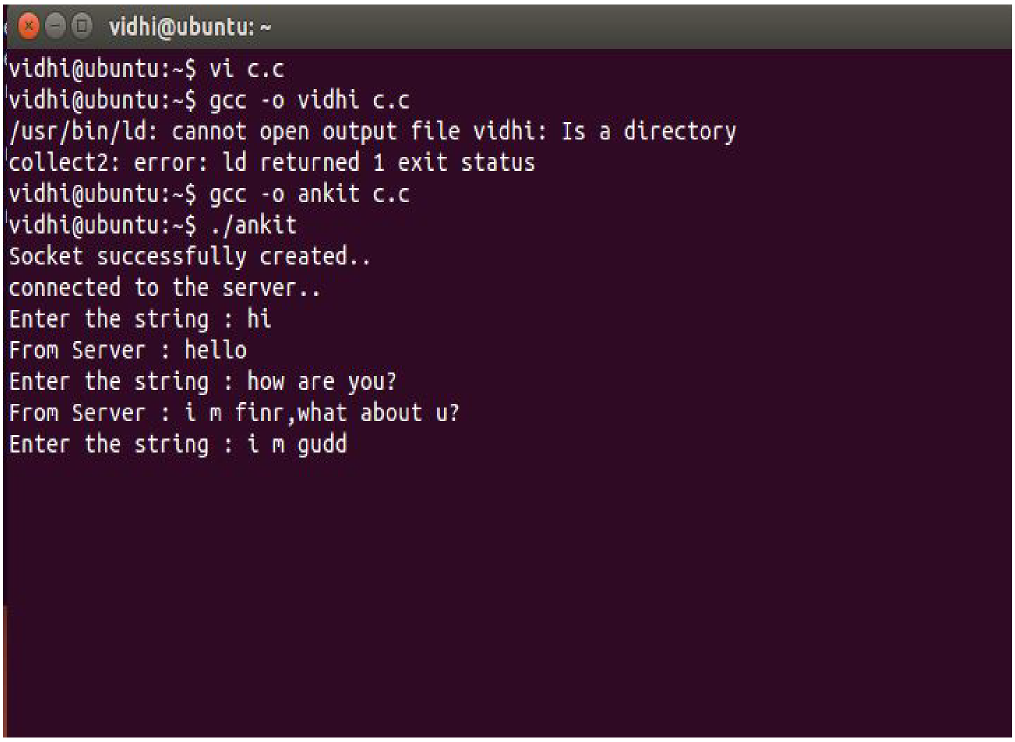
III. RESULT AND ANALYSIS

Wi-Fi security using encryption and decryption customize method and socket programming in Linux.

- Simple server and client data transfer program. [4][6]

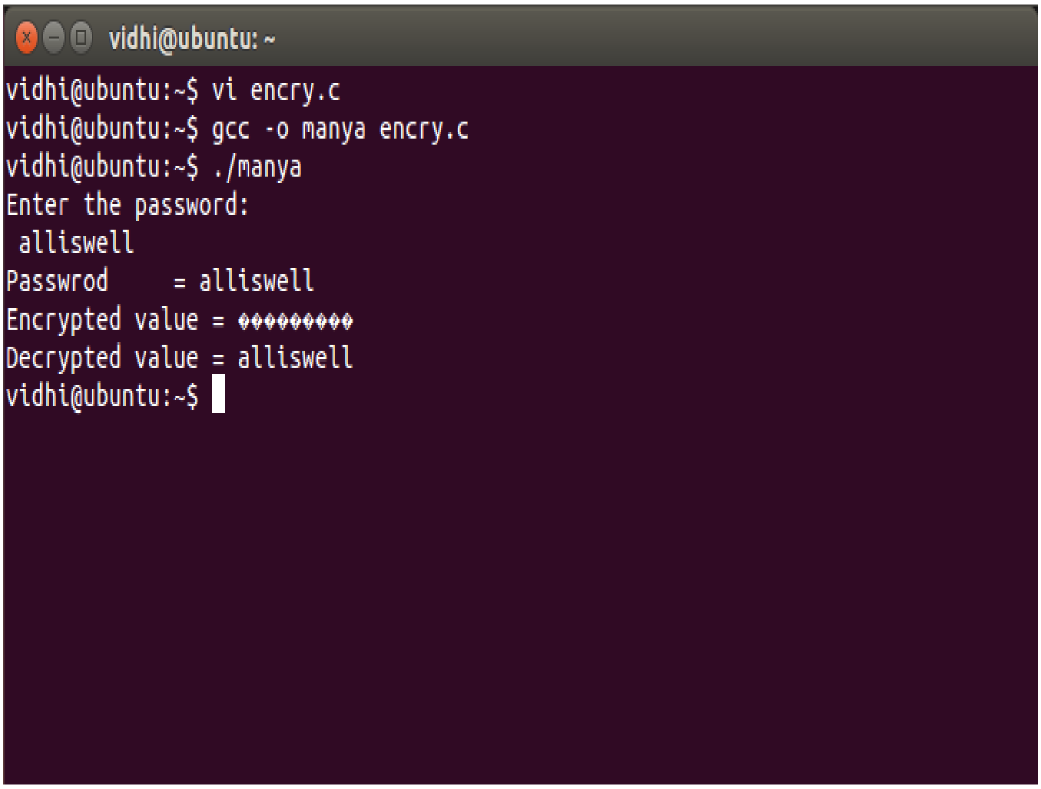


```
vidhi@ubuntu:~$ vi s.c
vidhi@ubuntu:~$ gcc -o maitra s.c
vidhi@ubuntu:~$ ./maitra
Socket successfully created..
Socket successfully binded..
Server listening..
server acccept the client...
From client: hi
         To client : hello
From client: how are you?
         To client : i m finr,what about u?
From client: i m gudd
         To client :
```

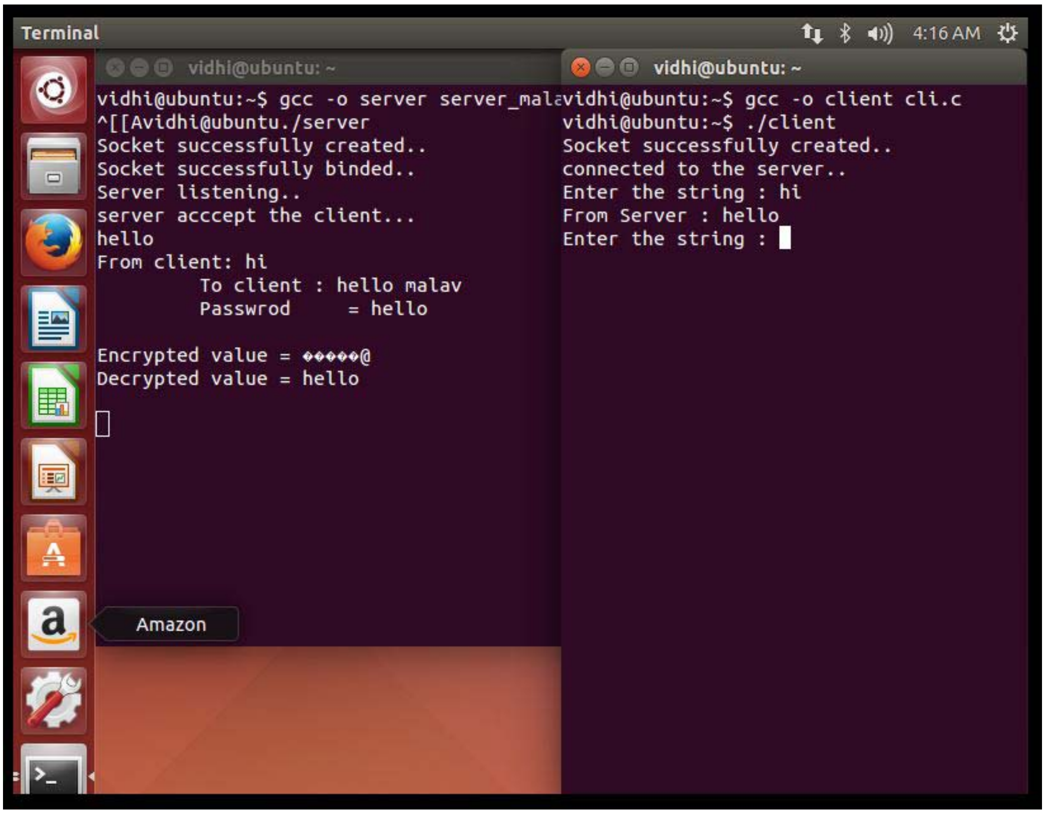


```
vidhi@ubuntu:~$ vi c.c
vidhi@ubuntu:~$ gcc -o vidhi c.c
/usr/bin/ld: cannot open output file vidhi: Is a directory
collect2: error: ld returned 1 exit status
vidhi@ubuntu:~$ gcc -o ankit c.c
vidhi@ubuntu:~$ ./ankit
Socket successfully created..
connected to the server..
Enter the string : hi
From Server : hello
Enter the string : how are you?
From Server : i m finr,what about u?
Enter the string : i m gudd
```

- Encryption and decryption program for data transfer security.[5]



```
vidhi@ubuntu:~$ vi encry.c
vidhi@ubuntu:~$ gcc -o manya encry.c
vidhi@ubuntu:~$ ./manya
Enter the password:
 alliswell
Passwrod      = alliswell
Encrypted value = ���������
Decrypted value = alliswell
vidhi@ubuntu:~$
```

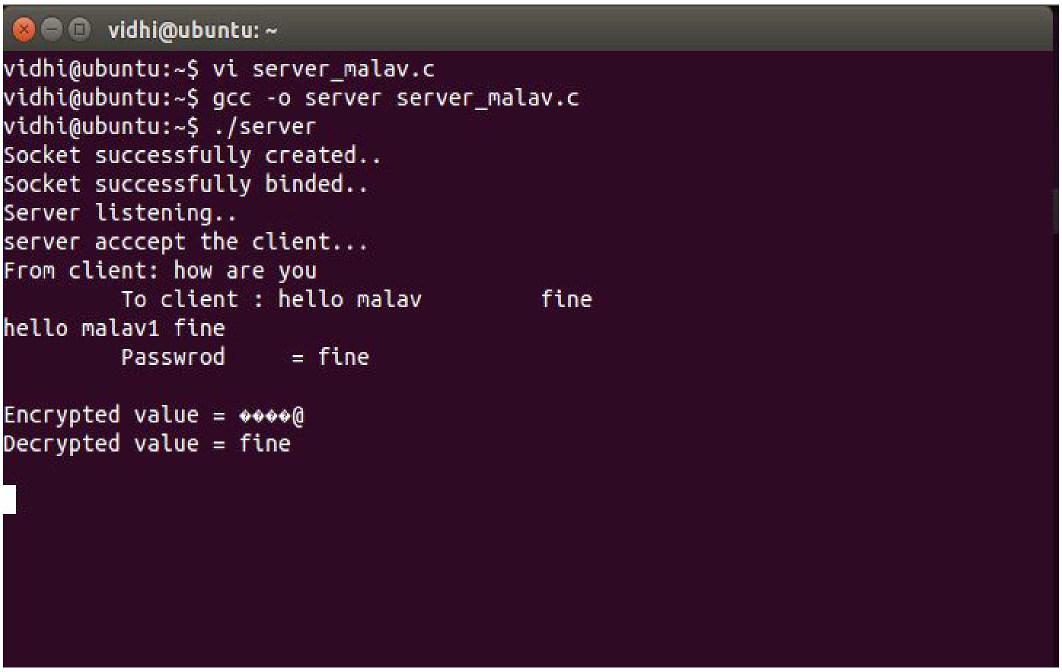


```
vidhi@ubuntu:~$ gcc -o server server_mala
^[[Avidhi@ubuntu./server
Socket successfully created..
Socket successfully binded..
Server listening..
server acccept the client...
hello
From client: hi
         To client : hello malav
         Passwrod      = hello

Encrypted value = �����@
Decrypted value = hello
```

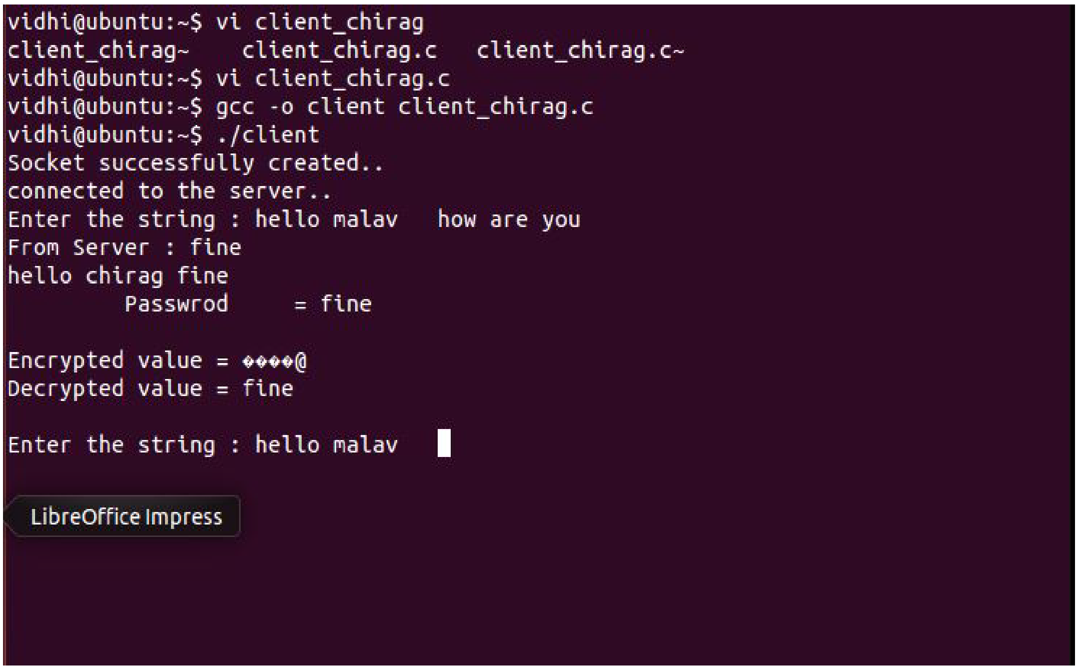
```
vidhi@ubuntu:~$ gcc -o client cli.c
vidhi@ubuntu:~$ ./client
Socket successfully created..
connected to the server..
Enter the string : hi
From Server : hello
Enter the string :
```

- Final Implementation for Wi-Fi data transfer security using encryption and decryption customize method.



```
vidhi@ubuntu:~$ vi server_malav.c
vidhi@ubuntu:~$ gcc -o server server_malav.c
vidhi@ubuntu:~$ ./server
Socket successfully created..
Socket successfully binded..
Server listening..
server acccept the client...
From client: how are you
         To client : hello malav         fine
hello malav1 fine
         Passwrod     = fine

Encrypted value = ����@
Decrypted value = fine
```



```
vidhi@ubuntu:~$ vi client_chirag
client_chirag~    client_chirag.c   client_chirag.c~
vidhi@ubuntu:~$ vi client_chirag.c
vidhi@ubuntu:~$ gcc -o client client_chirag.c
vidhi@ubuntu:~$ ./client
Socket successfully created..
connected to the server..
Enter the string : hello malav   how are you
From Server : fine
hello chirag fine
         Passwrod     = fine

Encrypted value = ����@
Decrypted value = fine

Enter the string : hello malav
```

CONCLUSTION

I learn the different paper and understand by ESP8266 modules working for Wi-Fi technology.ESP8266 Wi-Fi module using AT command for connection. We increase more security by encryption and decryption function in socket programming.

ACKNOWLEDGMENT

We take this opportunity to express our deepest gratitude and appreciation to all those who have helped us directly or indirectly towards the successful completion of this paper.

REFERENCES

[1] Surabhi Surendra Tambe in "WIRELESS TECHNOLOGY IN NETWORKS"International Journal of Scientific and Research Publications, Volume 5, Issue 7, July 2015

[2]Ruchir Bhatnagar, Vineet kumar birla in "WI-FI SECURITY: A LITERATURE REVIEW OFSECURITY IN WIRELESS NETWORK"International Journal of Research inEngineering & Technology (IMPACT:

IJRET)ISSN (E): 2321-8843; ISSN (P): 2347-4599Vol. 3, Issue 5, May 2015, 23-30

[3] NEIL KOLBAN, “Book of ESP8266”, USA, pp. 19-40, September 2015

[4] http://mcalabprogram.blogspot.in/2012/01/tcp-sockets-chat-applicationserver.html

[5] http://www.c-program-example.com/2012/04/c-program-to-encrypt-and-decrypt.html

[6] http://randomnerdtutorials.com/how-to-make-two-esp8266-talk/

[7] Vidhi B Patel, Chandresh Parekh, Malav Patel in “ESP 128266 exploration” International Journal of Advance Research in Engineering, Science &Technology,Volume 3, Issue 11, November-2016

[8] https://www.google.co.in/search?q=socket+prgramimg&oq=socket+prgramimg

[9] http://dominicm.com/esp8266-send-receive-data/