Overview of Wireless Sensor Network (WSN) Security

Aparicio Carranza
NYC College of Technology
186 Jay Street – V626
Brooklyn, NY, USA
718 – 260 – 5897
acarranza@citytech.cuny.edu

Heesang Kim
NYC College of Technology
186 Jay Street
Brooklyn, NY, USA
703 – 232 – 2001
heesang.kim@mail.citytech.cuny.edu

Xiao Lin Chen
NYC College of Technology
186 Jay Street
Brooklyn NY, USA
917 – 285 - 5155
xiaolin,chen1@mail.citytech.cuny.edu

ABSTRACT
Our civilization has entered an era of big data networking. Massive amounts of data is being converted into bits for transmitting over the Internet and shared all over the world every second. The concept of "smart" devices has conquered preconceived notions of daily routines. Smartphones and wearable device technologies have made humans to be continuously interconnected, making distance a factor of no limitation. "Smart" systems are based on a single concept called the Internet of Things (IoT). IoT is a web of wirelessly networked devices embedded with electronic components and sensors that monitors physical and environmental conditions and acquires data to be shared with other systems. Wireless Sensor Network (WSN) is a combination of hardware and software network that uses a system of sensors to detect physical phenomena. WSN is being widely used in industrial and consumer applications. Privacy is a key factor of WSN, it is imperative to secure data that is used, transmitted, or stored. Wireless communication technologies, such as Bluetooth and WiFi, etc., exposes users to a plethora of potential security flaws. In this paper, security mechanisms such as Cryptography, Firewalls, Secure Routing Protocol and Secure Data Aggregation of WSN using the Raspberry Pi 3 model B, and tools embedded into Kali Linux is explored and reported**.**

Keywords
Cryptography, Data Aggregation, Firewall, IoT, Secure Routing, Wireless Sensor Network

1. INTRODUCTION

The sensor technology has rapidly been advancing. The main components of a Wireless Sensor Networks (WSN) are comprised of gateways, relay nodes, leaf nodes and sensors. The sensor nodes are wireless devices that have onboard sensing, processing and communication capabilities [1]. A gateway is a hardware and software system that enables communication between different protocols in a network. The gateway is responsible for properly translating the communication protocols on different networks. It also recognizes all layers of the OSI reference model and absorbs the communication networks of different transmission schemes, allowing the connection to be made between different models. A router is a good example of a gateway. There are two common Wireless network topologies commonly used in WSN systems, they are "relay nodes" and "leaf nodes". The relay network is a broad network topology where the source and destination are interconnected through some node. In these networks, the source and destination cannot communicate directly with each other because the distance between the source and destination is greater than the transmission range, due to this an intermediate node needs to relay. The advantage of using the relay nodes is that the information can travel long distances even when the caller and recipient are far away. It also speeds up data transmission by selecting the best path to move between the nodes to the receiver's computer. An example of a relay node is a laptop with wireless connection. A laptop can connect to a wireless network that sends and receives information over a network and can send information to another network until it reaches its destination. A leaf node or a tree is a data structure consisting of nodes/vertices and edges that have no cycles. A tree with no nodes is called a null or empty tree. A non-empty tree consists of many levels of root nodes and additional nodes that potentially form a hierarchy. Figure 1 shows structure of leaf nodes connection. The advantage of a leaf node is that it is easy to expand the network, easy to maintain, and detect errors.

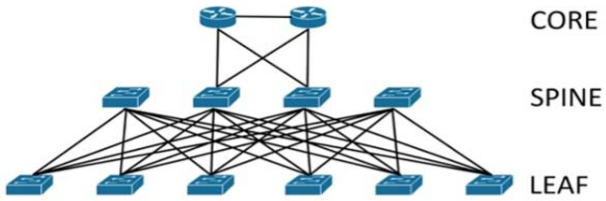

Figure 1: Structure of leaf nodes

The communication is divided into different phases. First, data is sensed by the sensors, and transferred to the Arduino Nano using a serial connection. Second, the data is sent from the Arduino Nano to the Arduino Uno by a Bluetooth connection. Third, USB connection from the Arduino Uno to a Raspberry Pi 3. Fourth, a Raspberry Pi 3

using a Wi-Fi connects to a Cloud Server. Last, a Cloud Server to a third-party client such as the mobile phone or personal laptop through the Internet.

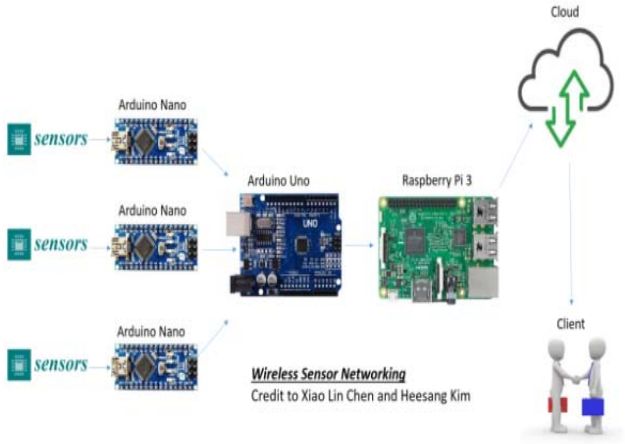

Figure 2: Concept of our WSN system

The background information for the Wireless Sensor Network was presented above, in section 2, we describe the Wireless Sensor Networking Hardware Components, in section 3, System Environment Setup is presented, followed by section 4, Communication Between Devices is described, Section 5, Security of Wireless Sensor Networks is covered and finally our Conclusion is noted in Section 6.

2. WIRELESS SENSOR NETWORKING HARDWARE COMPONENTS

Our WSN is composed of the following subsystems: The Arduino board (Uno & Nano), Raspberry PI model 3, Bluetooth Modules and Sensors.

A. Arduino Uno/Nano

Arduino is a single-board microcontroller based on open source technology referred to the boards and related development tools and environments. Various Arduino models are available – We will use the Arduino Uno/Nano. The Arduino Uno is based on the ATmega328 microcontroller. It has 14 input/output pins embedded on the board. The board uses 5V DC input as power source. The board supports serial connection via USB and Inter Integrated Circuit (I2C). The board can be controlled by programs using the Arduino IDE. The main Arduino board is not powerful enough to do multi-function taskS, but with extensible hardware and software, it is flexible enough to perform majority of projects/applications [2]. The Arduino Uno board is powerful enough to process all the data from the Arduino Nano. It is a low energy consumption board, which is smaller than the Arduino Uno. The Arduino Nano is based on the ATmega328p microcontroller. The Arduino Nano use alongside with the Arduino Uno. This will be connected to sensors and read data from a sensor and send it to The Arduino Uno. It can be serially connected using a USB cable or use I2C pins. Figure 3 shows the design of the two board and pinouts.

Figure 3: Arduino Uno and Arduino Nano Board

B. Raspberry Pi- model B

The Raspberry Pi Model B, shown in Figure 4, is a small Single-Board computer developed in the United Kingdom by the Raspberry Pi Foundation. It has 1.2GHz ARM Cortex-A53 processor, 1GB DDR2 RAM. It supports 100Mbps Ethernet, 2.4GHz 802.11n WiFi and Bluetooth 4.1. It also supports serial connection with a built-in USB port and uses the General-purpose input and output (GPIO) pins located on the board. Users can connect many types of sensors and other microcontroller boards such as The Arduino using the serial connection or wireless connection.

Figure 4: Raspberry pi 3 model B

The reason we chose a Raspberry PI is that it is small and simple embedded computer that users can obtain at a reasonable price and it has enough power to handle Sensor connections and communication. The Raspberry Pi uses the Linux Operating System (OS). Linux is an open source OS and many distributions of Linux are available from which

users can select. In this project, we will use Kali Linux as our OS and will later be explained in a separate section.

C. Bluetooth Modules

To connect the Arduino to the Raspberry PI via wireless communication, we need Bluetooth modules. Our implementation has made use of HC-05 and HC-06 modules shown in Figure 5. HC-05 is a Bluetooth Serial Port Protocol (SPP) module designed for wireless connection setup for The Arduino. It supports Bluetooth V2.0+EDR (*Enhanced Data Rate*) that uses 2.4GHz Frequency with 3Mbps modulation. The module has CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and AFH. HC-05 supports both master and slave. The HC-06 module has the same feature as HC-05 but only the slave mode is supported. A master/slave is a communication model of a hardware device in which one device provides one-way control to one or more devices. The master device can control one or more slave devices. The procedure of connection will be discussed later.

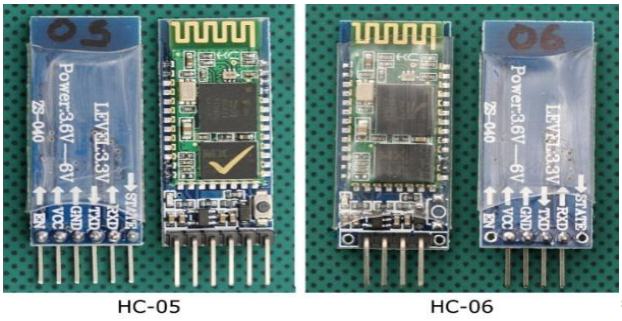

Figure 5: *HC-05 and HC-06 Module*

3. SYSTEM ENVIRONMENT SETUP

In the following subsections we will detail the setting of our environment.

A. Kali Linux Installation on Raspberry Pi 3

Kali Linux is based on the Debian Linux distribution and is mainly used for Penetration Testing and Computer Forensics. It has advanced networking features in contrast to other Linux distributions. To install in the Raspberry PI, an image file of Kali Linux for ARM processors needs to be downloaded from *www.Kali.org* and burn the system image into a micro SD card, Etcher software was used in our case and this software is available from *https://etcher.io/*. To perform first booting, a Raspberry PI needs Input/Output(I/O) devices. Users can setup optional Virtual Network Computing(VNC) or use same Input/Output source as before.

To perform an update of Kali Linux to the newest version, we set up a network connection manually. By default, Kali Linux uses a limited portion of partition. The user is recommended to expand partition size to maximize the storage space. On the terminal window type "***apt-get install gparted***." GParted is partition editing program that modifies partition size. The input command "***gparted***" launches the application. Set partition to maximum size then apply. Type "***apt-get install update***" and "***apt-get install upgrade***". These commands will download the newest updates and install. To download and install the full package of tools and programs for Kali Linux use command "***apt-get install kali-linux-full***".

B. ***Virtual Network Computing (VNC) Setup***

Virtual Network Computing (VNC) is a graphical desktop sharing system that is able to control the I/O of a Raspberry Pi remotely from another device such as a PC or smartphone using the network connection. To perform VNC connection, users should download VNC server on the Raspberry Pi (Host) and VNC viewer on desktop or smartphone (client). In the terminal window type "***apt-get install x11vnc***". This command will install x11vnc server on Kali Linux. On the client install RealVNC viewer. The user needs to configure the IP address of the Raspberry Pi. Type "***ifconfig***" command, this will show the network interfaces installed on a Raspberry Pi and its IP address. Take note of the IP address of wlan0. The IP address is required to connect a Raspberry Pi to the client. A command is required to setup the x11vnc. On terminal window type ***"x11vnc -cache 10 -auth guess -nap -forever-loop -repeat -rfbauth /root/.vnc/passwd -report 5900***". This command will enable the x11vnc server to run automatically when Kali Linux is booting up. It provides an automatic reliable connection between Host and client.

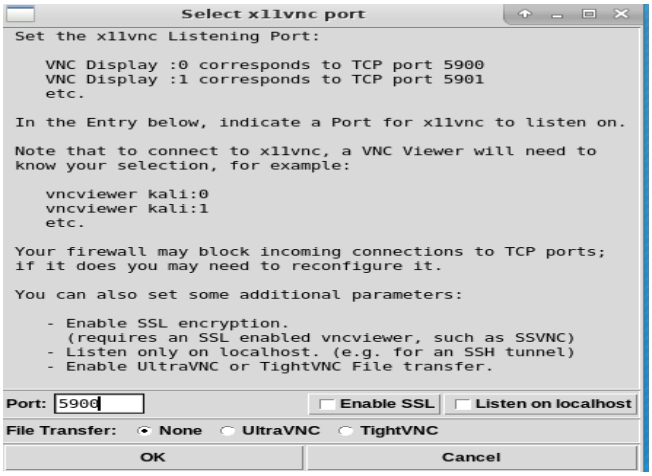

Figure 6: VNC port configuration

Launch the x11vnc server on Kali Linux, the default port is 5900 shown in Figure 6. On the x11vnc properties, follow instructions on the right. Launch VNC viewer from the client. Enter IP address found on a Raspberry Pi and enter with the port number at the end. The format should be same as shown in Figure 7. After successfully connection is

established, the desktop of Raspberry Pi should be displayed on the client's screen as seen in Figure 8.

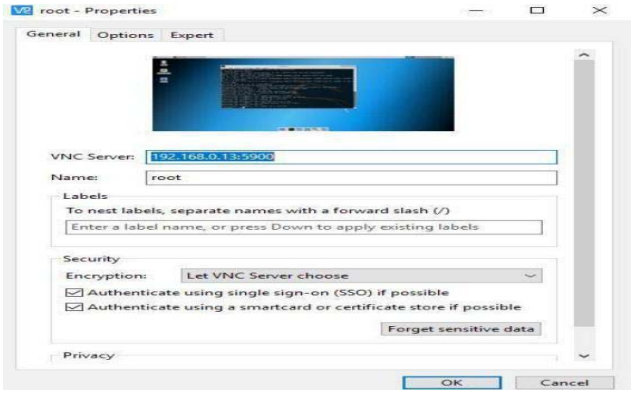

Figure 7: VNC viewer IP address & port

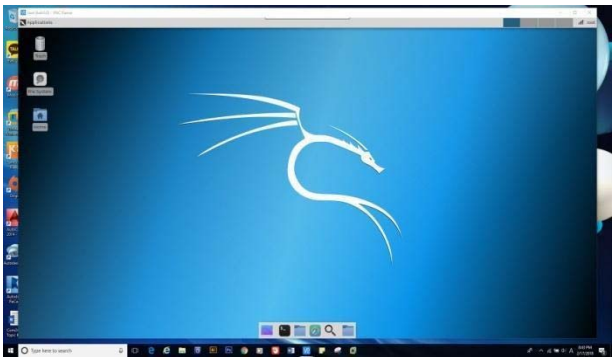

Figure 8: VNC viewer from laptop

C. *Arduino Uno and Nano Setup*

To set up the Arduino Uno/Nano boards, download and install the Arduino IDE. The Arduino IDE is the open source software that is required to run and compile the Arduino programs. The Arduino boards are connected to the Arduino IDE using USB serial connection. To establish any wireless connection between devices, extra modules are needed to communicate with each other. In this system, HC series Bluetooth modules (HC-05, HC-06) are used to establish a Bluetooth connection between the Arduino Uno and the Arduino Nano. The Bluetooth modules serve as transmitters only; the module did not analyze and store any data. The Arduino Nano in this WSN system serves as the nodes and all data is transferred to the Arduino Uno, and later transferred to the Raspberry Pi 3 for processing the data into the cloud.

D. *Setting up of Sensors*

Each sensor is serially connected to individual Arduino Nano in the system. The Arduino Nano has the program embedded in the board using Arduino IDE. The Arduino Nano serves as the controller of the sensors. In the system, we are using sensors from Adafruit [3]. The temperature and humidity sensor (*Si7021*), the light sensor (*TSL2591*), and the PIR motion sensor (*PIR189*). All sensors require library from Adafruit that is installed in the Arduino IDE library in order for the Arduino to detect the sensors.

4. COMMUNICATION BETWEEN DEVICES

A. Bluetooth Communication

The Bluetooth connection requires Bluetooth modules connect the devices. The status of the Bluetooth module needs to be configured before setting up the circuit to secure a successful connection. To configure the HC-05, enter the AT command mode in serial monitor in the Arduino IDE[4] as shown in Figure 9. A list of commands is recommended to configure the details about the module. After the main code is being compiled to the Arduino Nano, the connection between the Arduino Nano and the Arduino Uno is successfully established. In Figure 10, the connection between sensor and the Arduino is performed.

Commands for Bluetooth configure

AT – make sure communication is successful

AT+UART? - show the baud rate the devices communicate (default 38400 baud)

AT+NAME? -display the device name

AT+NAME:<name> -the name inside the less than and greater than sign is the new name you want to set (default HC-05)

AT+PSWD? -display the device password

AT+PSWD=<password> -the password inside the less than and greater than sign is the new password you want to set

(default 1234)

AT+ROLE? - check if the device is a master (1) or a slave (0)

AT+ADDR? - display the address of the device

Figure 9: AT commands

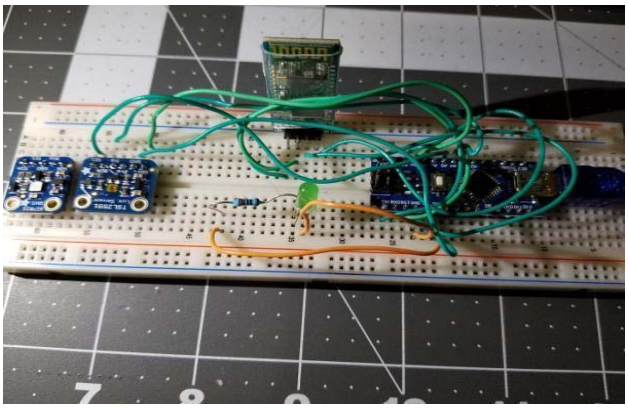

Figure 10: Bluetooth communication established with sensors attached

B. Serial Communication

To run the Arduino on a Raspberry Pi, the Arduino IDE is required to be installed. Type "***apt-get install Arduino***" in the terminal. Launch the Arduino IDE, click tools, board and choose Arduino Uno. The Arduino Uno and a Raspberry Pi can be connected via the serial connection using the General Purpose Input/Output (GPIO) from a Raspberry Pi to the Arduino Inter-Integrated Circuit (I2C) or using the USB cable. In this project, we chose the serial connection using the USB cable. The USB connection is safer and faster than GPIO to the I2C connection. Users just need to connect USB cable from a Raspberry Pi to the

Arduino Uno and launch the Arduino IDE to establish the connection. Figure 11, shows both GPIO and USB connections between a Raspberry Pi and the Arduino Uno. Figure 12, shows a successful connection between two devices with the Arduino IDE running the code.

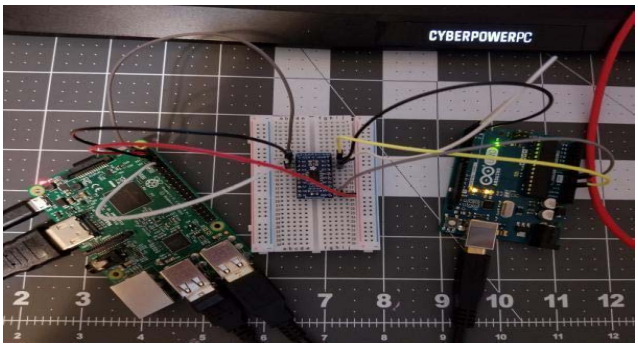

Figure 11: Serial connection

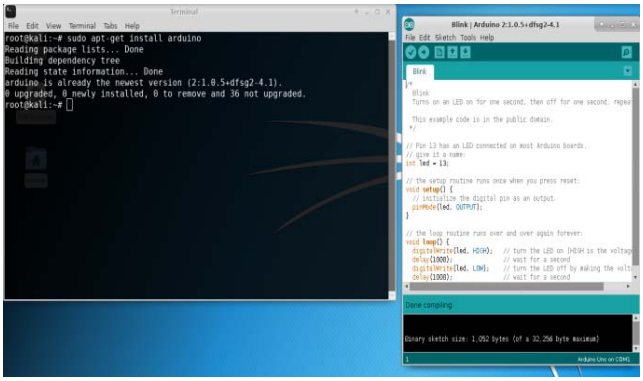

Figure 12: Arduino IDE in Raspberry Pi 3

5. Security of Wireless Sensor Networks

A. Types of Security Mechanisms

The security subsystem is the mechanism used to establish a secure communication between devices inside a network. The higher level of security mechanism such as Cryptography is used for protecting information from being hijacked by unauthorized users. "Cryptography has two basic operations: *encryption and decryption*" [5]. Data Encryption is a process where all the information is being encoded using encryption algorithm. Data Decryption is the process where authorized user receives data from the sender and decrypted into original information by using an encryption key. The encryption key system makes use of two kinds of keys: public key, and private key. The public key is used by the sender where all the data sent out from this user is encrypted using this key. The receivers use their own private key to decrypt the data to its original information. This system is ideal for preventing information from being hacked since only the receiver knows the private key, and the receiver can ensure the data is coming from the original sender. The drawback of this mechanism is that it takes time to encrypt and decrypt the data which increases the processing time. Another type of security mechanism we introduce next is a Firewall. Unlike Cryptography, "Firewall is a device that enforces security policies at the boundary between two or more networks"[6]. The function of the firewall is to control access to all the data from entering the computer system. Any malicious data will be detected and blocked.

B. Secure Shells (SSH)

Secure Shell (SSH) is a UNIX-based command interface and protocol for secure access to a remote computer. SSH is located in the application layer of OSI model. SSH refers to the application or protocol that allows the user to log in to another computer on the network. The user can execute commands on the remote system and copy files to another system. SSH is used by network administrators to remotely control various types of servers. SSH has strong authentication methods and the ability to communicate securely over unsecured networks. "With SSH, any kind of authentication, including password authentication, is completely encrypted. However, when password-based logins are allowed, malicious users can repeatedly attempt to access the server" [7]. Users can disable password-based authentication by setting SSH key authentication. SSH keys typically have more data bits than passwords, so there are far more combinations an attacker can execute." Most SSH key algorithms are considered to be inaccessible to modern computing hardware due to time-consuming to run all possible match"[7]. To start SSH in Kali Linux type "***service --status-all***" command on terminal. This checks installation of SSH on Kali Linux. type "***service sshstart***" and "***service ssh status***" to start SSH and check running status. Launch SSH and modify configuration files by direct to SSH directory by type "***cd /etc/ssh***" then type "***nanosshd_config***" on the following line. These commands display GNU sshd_config file shown in Figure 13.

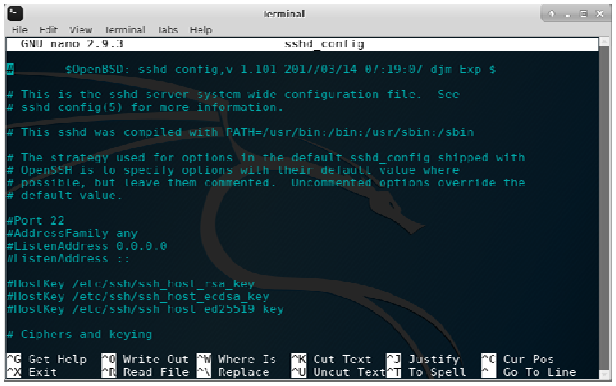

Figure13: Editing sshd_config

The user can modify port number in a range between 0 to 65535. The default port is set to 22 shown in Figure 14, but recommended to change the port to harden the security. Restart SSH by typing "***service ssh restart***" this will apply the settings and restart the SSH. RSA key pair is required

for SSH. type "***ssh-keygen-t rsa***" the terminal will display an option for the user to save a key in a specific location and passphrase users preference. The user can simply use the default values given and skip password. Once the setting is established, the user will see cryptography keys generated as shown in Figure 14.

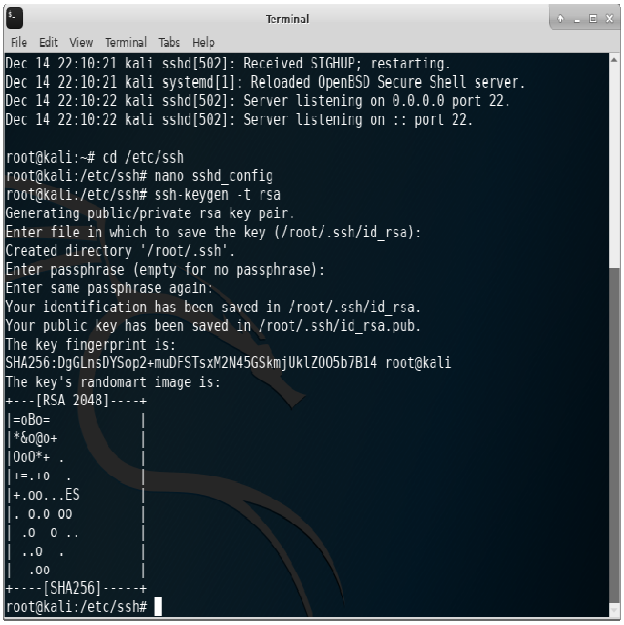

Figure 14: SSH keygen

C. Secure Data Aggregation

Data Aggregation is a technology used in the Wireless Sensor Network System for reducing the battery consumption by sending less redundant raw data gathered by sensors [8]. During data aggregation, the raw data gathered by sensors will be combined or eliminated to reduce the amount of repeated data. There will be less data processing through aggregation - this dramatically decreases the amount of processing and transmitting tasks for the central processor. The secure Data Aggregation is about adding a secure aspect during data aggregation. For example, in the WSN system, many nodes of sensor connected to the individual central controller. Each controller analyzes and processes the raw data. For hackers who try to hijack the data transferred through the network, it is easy of gaining control of one controller and sending false data to the main controller. As shown in Figure 15, the secure data aggregation concept is that only the central controller will store the information and communicate with all the nodes. For other controllers that attach to sensor directly, they can only send data to the central controller and have no privilege to store data and communicate to other nodes.

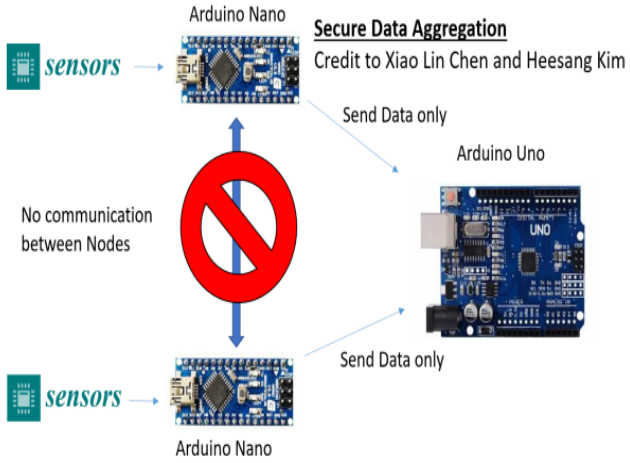

Figure 15: Secure Data Aggregation System

D. Secure Routing Protocol

Routing Protocol is the network connection between devices in the system. The idea of secure routing protocol needs to be considered when the security of data transmitting during the communication can be guaranteed. With secure routing protocol, the network connection is been protected from the various types of attack, such as mode capture, physical tampering, eavesdropping, denial of service etc [9]. In our system, the routing protocol will be established throughout the system since the Secure Routing Protocol is the foundation of secure data aggregation protocol. To apply the secure data aggregation, we need to develop a secure network connection between devices.

E. Connecting to server

Users will connect the Rraspberry Pi to the Server to send data to Server. We use google firebase as server demonstration. The process is to sign up for google firebase., add project on google firebase, name the project, in our case we named it as WSN house model - Select the origin to United States, then, choose database and click Realtime Database. Finaly, start the project with test mode.

F. House Model

In our wireless sensor network system, we establish a safe network by applying the concept of secure data communication. We proposed SSH to create a secure channel to encrypt the data through the unsecured network; this prevents unauthorized user to access the information. We have completed with the design layout, implemented and constructed the house model for our system. This smart house system is a complex multi-function WSN system that can monitor the status of the house for more convenient lifestyle. Figure 16, shows the layout design we created, Figure 17, shows the work in progress we did for the model; and Figure 18 and 19 renders our complete system.

Figure 16: Layout of the house

Figure 17: First prototype for house layout

Figure 18: prototype for living room demonstration

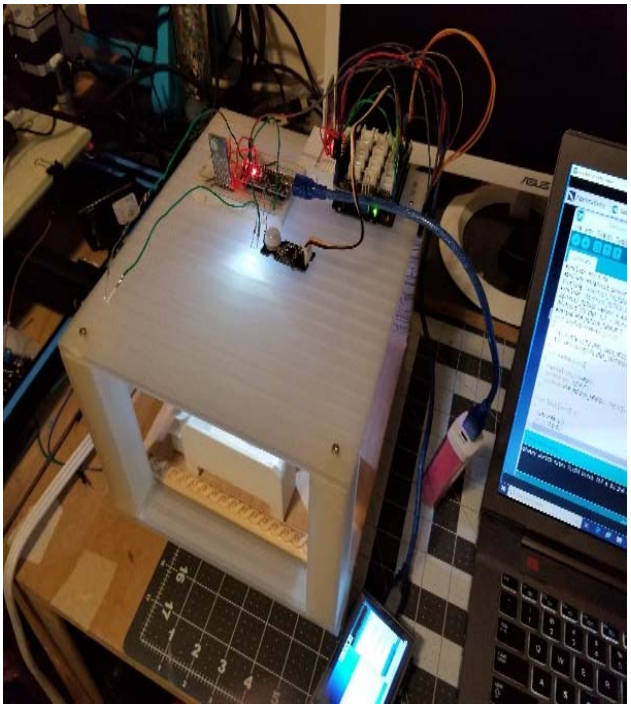

Figure 19: placement of electric circuit with components

6. CONCLUSION

Wireless Sensor Network (WSN) has been a significant advancement over the last decade. The concept of security mechanism has ensured the security in WSN by detecting, blocking, and eliminating the threats or attacks from hackers. We have been able to establish an actual WSN system with a physical House model. All devices in the system have successfully communicated wired or wirelessly. As shown in Figure 20, all the sensor data is being gathered and display in the Arduino IDE inside Kali Linux. The sensor data uploads to the database in Realtime with some minor errors, such as short delay time to display on the database. Due to the limited amount of time secure mechanisms such as secure data aggregation and secure routing protocol is about 80% finished. Cryptography and firewall are explored but requires future research effort and improvement to apply the concept to the WSN system. For future work, we will build a full smart house based on IoT and use AI for the main controlling and management system. For the security aspect, we will apply more in-depth concept of secure data aggregation, secure routing protocol, and cryptography-categories such as data encryption and data decryption.

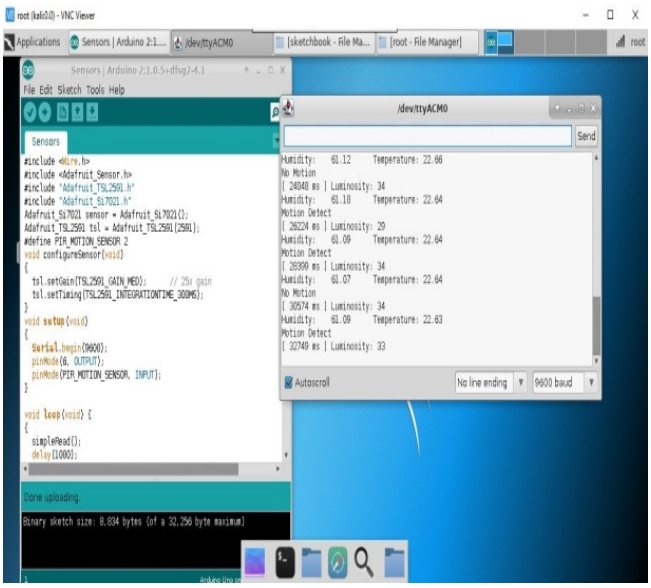

Figure 20: 3 sensor data are shown inside Arduino IDE in Kali Linux

REFERENCES

[1] Akyildiz, I. (2002). A survey on sensor networks. IEEE Communications Magazine, (pp. 102–114)

[2] Arduino.cc. (2018). Arduino - Introduction. https://www.arduino.cc/en/Guide/Introduction

[3] Adafruit (2018). Adafruit Industries, Unique & fun DIY electronics and kits. https://www.adafruit.com/

[4] Instructables.com. (2018). AT Command Mode of HC-05 and HC-06 Bluetooth Module. http://www.instructables.com/id/AT-command-mode-of-HC-05-Bluetooth-module/

[5] Horbaty, S. (2018). *Security in distributed systems* https://www.slideshare.net/EngHaitham/security-in-distributed-systems

[6] Oppenheimer, P. (2018). *Security Mechanisms Developing Network Security Strategies*, http://www.ciscopress.com/articles/article.asp?p=1626588&seqNum=2

[7] Ellingwood, J. (2018). *7 Security Measures to Protect Your Servers DigitalOcean.* https://www.digitalocean.com/community/tutorials/7-security-measures-to-protect-your-servers

[8] Alzaid, H., Foo, E. and Nieto, J. (2018). *Secure data aggregation in wireless sensor network: a survey*. https://dl.acm.org/citation.cfm?id=1385127

[9] Sen, J. and Ukil, A. (2018). A Secure Routing Protocol for Wireless Sensor Networks. https://link.springer.com/chapter/10.1007/978-3-642-12179-1_25 http://www.mwrf.com/systems/building-wireless-sensor-networks- wsn