Wireless Sensor Networks in Support of E-Health Applications

Longsheng Yu

Submitted to the University of Hertfordshire
in partial fulfilment of the degree of
Master of Philosophy

November 2015

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing, which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This thesis contains fewer than 24,000 words including appendices, bibliography, footnotes, tables, figures, and equations.

Name:

Date:

Abstract

Nowadays, with the smart device developing and life quality improving, people’s requirement of real-time, fast, accurate and smart health service has been increased. As the technology advances, E-Health Care concept has been emerging in the last decades and received extensive attention. With the help of Internet and computing technologies, a lot of E-Health Systems have been proposed that change traditional medical treatment mode to remote or online medical treatment. Furthermore, due to the rapidly development of Internet and wireless network in recent years, many enhanced E-Health Systems based on Wireless Sensor Network have been proposed that open a new research field.

This research work has reviewed the E-Health Care System development and limitations in recent years and proposes a novel E-Health System based on Wireless Sensor Network by taking the advantage of the latest technologies. The proposed E-Health System is a wireless and portable system, which consists of the Wireless E-Health Gateway and Wireless E-Health Sensor Nodes. The system has been further enhanced by Smart Technology that combined the advantages of the smart phone. The proposed system has change the mechanisms of traditional medical care and provide real-time, portable, accurate and flexible medical care services to users.

With the E-Health System wieldy deployed, it requires powerful computing center to deal with the mass health record data. Cloud technology as an emerging technology has applied in the proposed system. This research has used Amazon Web Services (AWS) – Cloud Computing Services to develop a powerful, scalable and fast connection web service for proposed E-Health Management System.

The security issue is a common problem in the wireless network, and it is more important for E-Health System as the personal health data is private and should be safely transferred and storage. Hence, this research work also focused on the cryptographic algorithm to reinforce the security of E-Health System. Due to the limitations of embedded system resources, such as: lower computing, smaller battery, and less memory, which cannot support modem advance encryption standard. In this research, Rivest Cipher Version 5 (RC5) as the simple, security and software or hardware deployable encryption algorithm has been in-depth studied. As the Logistic map has good cryptographic algorithm properties, like unpredictable, random, and sensitive to the initial parameters it has been investigated. In this thesis, an enhanced RC5 cryptographic algorithm has been proposed that uses 1-D Logistic mapping in the random sub-key generation during each encryption round, which increases the unpredictability significantly. In addition, an effective cipher feedback model has been combined to further increase the cipher security. After in-depth research of the 1-D Logistic map, a 2-D Logistic map has been proposed that provides more complex chaotic behaviors than the 1-D Logistic map and further improves the security. Another novel RC5 cryptographic algorithm with 2-D Logistic map has been proposed in this thesis. The proposed algorithm uses a 2-D Logistic map to generate the sub-key and modified RC5 operations to encrypt data. Appropriate experiments have been carried out to evaluate the performance. The results show the proposed algorithms are better than standard RC5 or other modified RC5.

The contributions and innovation of this research project are summarized:

- Build up a Wireless E-Health Care System based on Wireless Sensor Network.
- Create the Cloud Management System for E-Health Care System.
- Proposed RC5 cryptographic algorithms based on Logistic Map to increase the randomness and security of cipher data.

Key Words: *E-Health, Wireless Sensor Network, Cloud, RC5, 1-D Logistic Map, 2-D Logistic Map.*

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Dr. Baochun Hou for the continuous supervise my MPhil study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my MPhil study.

Besides my advisor, I would like to thank the rest of my supervision team: Prof. Aladdin Ariyaeeinia and Mr. Johann Siau, for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

I thank my friends in the University of Hertfordshire for the stimulating discussions, for the sleepless nights we were working together, and for all the fun we have. In particular, I am grateful to Chenqi Yang and Zhongji Sun for enlightening me the first glance of research.

Last but not the least, I would like to thank my family: my parents for supporting me spiritually throughout writing this thesis and my life in general.

Longsheng Yu

Table of Contents

List of Figures

List of Tables

Glossary

ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AWS	Amazon Web Services
BPWESN	Blood Pressure Wireless E-Health Sensor Node
DAC	Digital to Analog Converter
DES	Data Encryption Standard
E-Health	Electronic Health
GPIO	General-Purpose Input/Output
I2C	Inter-Integrated Circuit
NHS	National Health Services
NIBP	non-invasive blood pressure
PFLOPS	Peta Floating-point Operations Per Second
PWM	Pulse Width Modulation
OSWESN	Oxygen Saturation Wireless E-Health Sensor Node
RC5	Rivest Cipher Version 5
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
WSN	Wireless Sensor Network
WESN	Wireless E-Health Sensor Node
1-D	One-Dimensional
2-D	Two-Dimensional
3G	Third Generation Mobile Network
4G	Fourth Generation Mobile Network

Chapter 1 Introduction

E-Health System is the Information and Communication Technology (ICT) based health system for the 21st century according to World Health Organization [1]. E-Health refers to computer technology, communications technology, multimedia technology, combined with medical diagnose and analysis technologies. E-Health System is the new medical service that aims to improve the diagnosis and medical care services; to reduce medical expenses; and to meet the demand of health care for increasing population.

In recent years, the aging population has drawn increasing public attention. In Britain, the aging problem is particularly serious. It was reported that the retired people occupied one fifth of the British population. The worse thing is that aged 85 or older people are more than one million and this number will double within the next 35 years [2]. Furthermore, these people will have a limited capacity for independent living, requiring continuous monitoring and daily assistance. The health service cost will be increased with the accelerated aging of the population. It has been forecasted, in 2050 the growth cost of health care will occupy 9.00% of Eurozone gross domestic product (GDP), which will be twice than in 2025 [3]. Information Communication Technology offers good opportunities into the solving the aging problems. ICT provides continuous monitoring and daily assistance services that is security, environment adaptable and optimizing health care [4].

The aging problem was a big challenge for National Health Services (NHS) in the United Kingdom as well. The NHS established in 1948 by British Government. This healthcare system was the largest publicly funded healthcare service when it has been built and its service has been continued more than 60 years until now. The principle of

NHS is to provide good and fair healthcare service for every resident in the UK. The general structure of the NHS healthcare system is shown in Figure1-1 as reported.

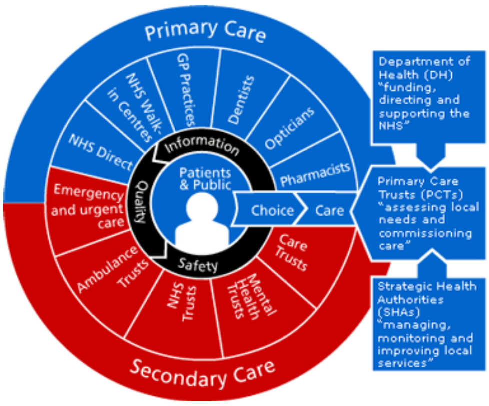

Primary Care
GP Practices
Dentists
Opticians
Pharmacists
NHS Walk-in Centres
NHS Direct
Information
Patients & Public
Quality
Safety
Choice
Care
Emergency and urgent care
Ambulance Trusts
NHS Trusts
Mental Health Trusts
Care Trusts
Secondary Care
Department of Health (DH) "funding, directing and supporting the NHS"
Primary Care Trusts (PCTs) "assessing local needs and commissioning care"
Strategic Health Authorities (SHAs) "managing, monitoring and improving local services"

Figure 1-1 NHS Healthcare System structure[5]

The NHS implements two different levels of healthcare. The blue part is the Primary Care that used as routine medical care. Another red part is the Secondary Care that mainly refers to the emergency or critical cares services; especially refers to the hospital services. Nowadays, it has already become an overburdened healthcare system because of the increasing population and the obsolescence of equipment. Under this situation, a lot of NHS system's shortages have been exposed. Unbearable long waiting time for medical diagnosis and treatment is the first and the important issue. The second problem is the lack of funds leading to the slow upgrades of medical equipment that cannot satisfy the modern medical requirements. The shortages of NHS show that this system should be reformed or upgraded to fit the demands of the modern health care.

With the advancement of technologies, especially the rapid development of communication technologies, the high speed and wide bandwidth of network have been significantly improved, which has laid a solid foundation for next generation of health care system. The E-Health System will be committed to personal digital health care based on Wireless Sensor Network (WSN) [6]. Recently, WSN has been extensively applied in the field of Electronic Health Care System. It provides portable, convenient, comfortable health care services now and it will offer more functions in the future.

With the fast speed evolution of wireless E-Health System, the security problem of E-Health System has been received more attention [7-12]. Encryption technique is one method to deal with security issue of E-Health System [13, 14]. A lot of encryption technologies have been used in Internet Communication, such as DES, Triple-DES, AES or RSA [15-20]. However, with the portability development of E-Health System, the system become wearable, smaller and light which has small battery, limited computing ability and memory resource that cannot support complex encryption algorithms [21]. This challenge requires a new kind of simple encryption algorithm to deal with. After reviewing lots papers of security method, RC5 (Rivest Cipher 5) and Chaos encryption algorithms have been chosen to enhance the proposed E-Health System security.

This research has investigated an E-Health care system based on Wireless Sensor Network to improve the modern health care service. The system will be developed in two parts; one is WSN-based hardware system that used to do health-care monitoring and data transmission. The other is Health Management System that used to provide health care service. Furthermore, the security issue of E-Health care service has been investigated and two enhance encryption algorithms have been proposed. The goal of this research is to offer a convenient, efficient and security health care service to users at anytime and anywhere.

The contributions and innovation of this research project are summarized:

- Build up a Wireless E-Health Care System based on Wireless Sensor Network.
- Create the Cloud computing service for E-Health Care System.
- Proposed RC5 cryptographic algorithms based on Chaotic theory to increase the randomness and security of cipher data.

Chapter 2 E-Health System based on Wireless Sensor Network

2.1 E-Health Care System

E-Health Care is a rising intersection of medical informatics and technologies, which relates to health services and information delivered or enhanced via the internet and relevant technologies[8, 12, 22]. E-Health is expected to provide and improve day-to-day healthcare [22].

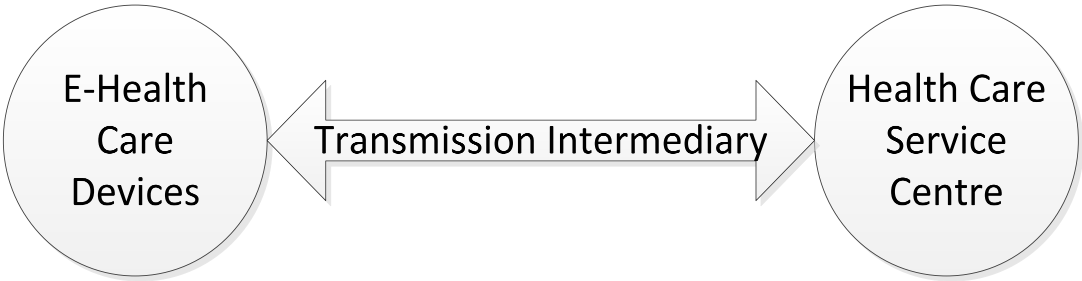


Figure 2-1 E-Health Care System Structure

As shown in Figure 2-1, E-Health Care System consists with E-Health Care Devices, Health Care Service Center and information transmission intermediary in generally. The E-Health Care Devices used for monitoring personal health status, and the Health Care Service Center responsible for data analysis and diagnosis of disease. The physiological data has been transmitted via transmission intermediary between the Devices and the Center. The transmission intermediary uses Information Communication Technology (ICT) or Wireless Technology.

In this system, both sides use digital information to communicate, which makes the information exchange easier. E-Health Care Devices normally are portable devices to monitor personal vital evidence. Nowadays, there are many kinds of personal health care devices, mainly divided into two categories, stationary health care devices and portable health care devices. E-Health should be convenient, efficient, enhancing quality, evidence based, empowerment, encouragement, education, enabling, extending, ethics and equity.

The most important advantage of using E-Health should be convenient, which can break through the bottleneck in modern health care system. Comparing to the traditional NHS system which is often complained by the patients because of the long waiting list, the E-Health System helps the users to save a lot of time. With the E-Health System, users do not need very often to go to hospital for checking and submitting the health data. Instead, users can stay at home and operate the E-Health care device to carry out regular health check, and then to send the health data to the health care center by communication technology. In the future, with the wireless sensor network integrated into the E-Health, this system can provide users with the mobility and portable experience. The users can do health monitoring in anywhere at any time.

The other advantage of using E-Health is efficiency. The reason is that the E-Health care will not affect the users' normal life and work since it can be used in users' free time. On the other hand, this system can send the health data to health care center via ICT which save time and money for users. Furthermore, health care staffs can get the health data in time, which can help them to make accurate diagnosis of disease. Therefore, E-Health provides the easy way and good experience to users.

2.2 Reviews of E-Health System

With the ICT advancement, the use of web has become more and more common. At early stage, web 1.0 protocol only allows users to view posted content via web browser. With the web 2.0 released, the web application has entered into the multimedia age, since the web 2.0 is an interactive web protocol that allows users to post and modify contents of the web applications [23]. Nowadays, the rapid development of the social communication network has opened up the ways to applications such as wellness and health care system.

The typical Web based health care system architecture is shown in Figure 2-2. It contains two parts, client part and server part [24].

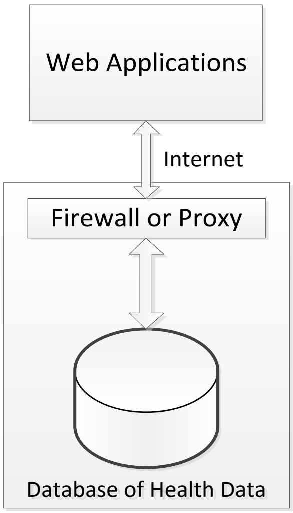


Figure 2-2 the architecture of Web based Health Care system

The Web Applications in the upper half part is called client part with user-oriented interface. Users can access to their health care record via PC that connects to Internet. When users want update their monitor data to remote health server, they can fill health record forms, and then upload them to the server through Internet.

The Web Server in the lower half part can be based in hospital used by doctors for disease condition analysis and processing. This part contains firewall that protects the server side, and provides health care services and databases that store users' personal information and health care records.

Between the clients and servers is the Internet transport which can exchange text, picture, video information etc. With the high speed of Internet connectivity, the communication between clients and server has become faster and more efficient.

The Web 2.0 based on health informatics system was introduced in [23]. The architecture of this system is very similar to the one shown in Figure 2-2. Users access to or update their health record in the database via client's application. And then the physician can exam the patient's condition by using the same database and gives them advices.

On the one hand, the advantage of this system is time-saving and costs-saving. Users can do simple health care checking at their home and upload results to health care center via Internet, rather than go to the hospital for checking.

On the other hand, the disadvantage is that it's not portable, as this system needs users to seat in the front of computer to submit their health care record. The most important is that the real-time monitoring mechanism is not available. Therefore, the system cannot give alert when users suffer emergencies.

In [25] a self-care system was proposed by adding telephone interface to web service. In this system, one software called Private Branch eXchange (PBX) was developed to cooperate with web service. Therefore the web services can be inputted or outputted through ordinary telephone device, and the users can talk with web service interactively. The system adds PSTN (Public Switched Telephone Network) into the Internet via VoIP provider, and it can provide web services to PSTN users by using software PBX. The process is shown in Figure 2-3.

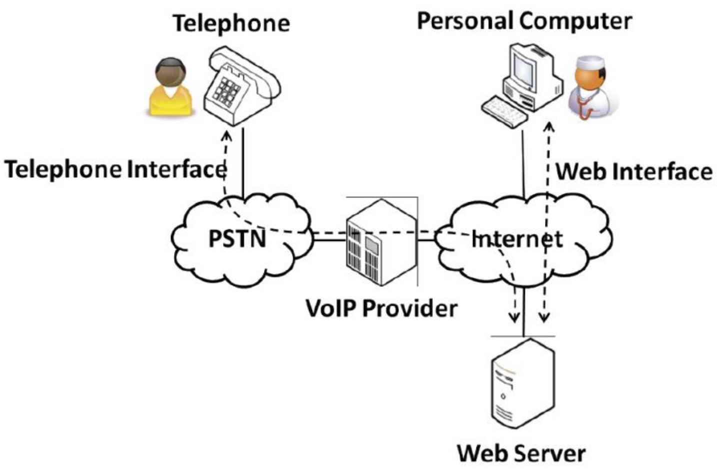


Figure 2-3 Adding Telephone interface to web service [26]

Followed by the voice guide, a patient can operate the telephone to enter health data and receive response to the web services. After a series of simple operations via telephone, the doctor can understand the patients' situation and be able to offer therapies to the patient.

The advantage of this system is that the system adds telephone interface to web service. The E-Health care system becomes more diversified by adding telephone interface. This method can help elderly or inexperienced people who are not familiar with operating the web interface.

However, this system suffers the same issue as previous system. Adding the telephone interface just add a way to communicate with health care center. The system still does not provide real-time measurement ability to help patients to monitor their health.

.

2.3 Wireless Sensor Network for E-Health

With the fast speed evolution of E-Health technology and wireless network technology, Wireless E-Health (WeHealth) System comes into the public's life [27]. The aim of Wireless E-Health System is to monitor patients' physical indicators, such as temperature, oxygen saturation and blood pressure. The rapid growth of chronic diseases led people to pay more and more attention on human's health in last decades, such as Hypertension (high or raised blood pressure), one of the most common cardiovascular and cerebrovascular chronic diseases in modern life. Because the Hypertension can cause a lot of complications, for example, stroke, myocardial infarction, heart failure and chronic kidney disease, therefore, it is called a "silent killer", and it became one of the global public health issues [28]. According to Lin, et al. [28]the best way to prevent and control hypertension is early detection and early treatment. Therefore, regular check of blood pressure is very necessary.

Wireless E-Health System uses wearable, wireless, reliable sensor to monitor patient's physical indicators and upload measurement data to health center. Most of the E-Health System can provide emergency mechanism. For example, one WeHealth system uses Alarm Access (AAC) scheme that use wasted bandwidth in contention free period (CFP) for emergency access in order to reduce transmission latency of alarming packets[27].

With constant upgrades in technological capability, Wireless Sensor Network technology has got great development. As the Wireless Sensor Network has integrated sensor technology, embedded computing technology, distributed information processing technology and communication technology, Wireless Sensor Network can be used to do real-time monitoring, sensing, acquisition and processing of various objects in the area of network with the form of collaboration. The acquired information can be transmitted to the user terminal by wireless network, in order to achieve the physical world, computing world and human society triple world connection.

The research of Wireless Sensor Network was started in the late 1990s. From 2000, a number of reports about Wireless Sensor Network have been published and this topic has been widespread in the recent years. Its application value has gradually been discovered with the rapid development of electronic technology and been applied into real life. There is a list of the application fields of Wireless Sensor Network.

1. Environment detection and protection: Wireless Sensor Networks can be used to monitoring environment parameters, like air, water and soil quality.

2. Industrial application: Wireless Sensor Networks can be deployed in a number of hazardous industrial environments, like mine, boilers, nuclear power plants, etc. rather than dispatch human to take risk to do these dangerous measurement tasks.

3. Smart Home Application: Family provides an excellent platform for Wireless Sensor Network to apply as well. A lot of furniture, appliances and other household devices by embedding the sensor nodes have been produced that connected to the Internet via wireless network. They provide people more comfortable, convenient and user-friendlier smart home environment.

4. Medical Care or Smart Health Care: As the medical care is a focus of world attention in recent years, health care based on Wireless Sensor Networks gain a great advance. Health care device used to detect a variety of physiological data of the human body. Wireless Sensor Networks provide a more convenient and efficient method for telemedicine.

2.4 The Architecture of Wireless Sensor Network

The typical wireless sensor network architecture as shown in Figure 2-4, was including distributed sensor nodes, the target node (sink), Internet and client.

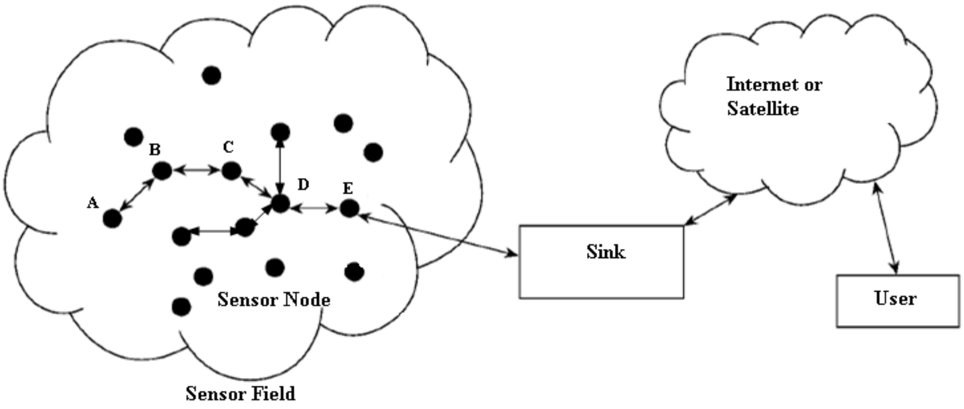


Figure 2-4 the Wireless Sensor Network Architecture[29]

Sensor Field consists of sensor nodes that were distributed as a group to cover a specific area for acquiring the local information. Sink, also called data center or base station, which has more powerful computing, storage and communication ability than the sensor node, which makes the sensor network and external network connectivity to realize the information conversion between communication protocol stacks. Each spread sensor node in the wireless network sends perceive data to sink through multi-hop routing approach, then the users can communicate with Sink via the internet or satellite. The common wireless network including mobile communication network, such as 3G, 4G network. Wireless Local Area Network (Wireless LAN), Bluetooth networks and Ad hoc networks. However, because of the particularity of Wireless Sensor Network, it has more features than these common wireless networks as following list.

1. Large-scale deployment: Due to the limitation of the power and performance, an individual sensor node has restricted sensing range. Therefore,

large-scale rang monitoring need dense, a huge number of sensors to be widely deployed.

2. Self-organization: a large number of Wireless Nodes be random deployed in the Wireless Network, each of them unknown their mutual neighbor before information transmission. Moreover there is no central control node to coordinate activities of the network nodes also. Hence, self-organization capability is very important for the Wireless Sensor Nodes. According to topology control mechanisms and hierarchical protocols, Wireless Sensor Nodes set and manage their configuration to build up a multi-hop wireless network system.

3. Limited hardware resources: As the sensor nodes have small size, however, each node required to be sustained long-term work with finite energy supply. The requirement of low power consumption become necessary, the sensors commonly uses low-power embedded processors to suit the power restriction. Hence, node energy, computing and storage ability was subject to limitation, the operating system and the protocol layer for the wireless node cannot be too complicated. A lot of complex security protocols and algorithms cannot be directly applied in Wireless Sensor Network as well.

4. Limited communications capacities: because of the limited hardware resources, sensor node communication bandwidth is very small, normally, it only a few hundred kbps. Besides, the maximum transmission power of wireless communication is only a few milliwatts, therefore, multi-hop relay communication mode was accept by the Wireless Sensor Network.

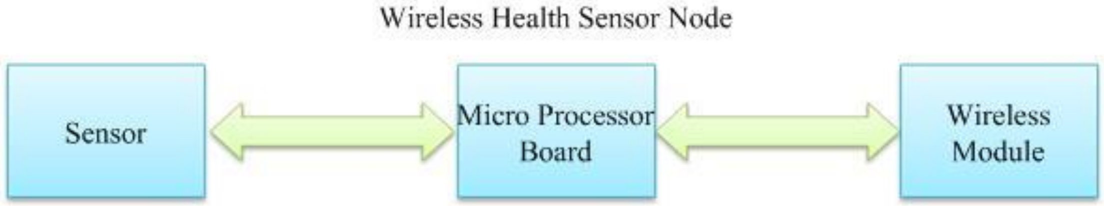


Figure 2-5 General Structure of Wireless Health Sensor Node

Wireless Health Sensor Node is the basic unit of E-Health Care system. Wireless Health Sensor Node can be worn on body and it can measure health status. The structure of general Wireless Health Sensor Node is illustrated in Figure 2-5; consisting of three parts: sensor, microprocessor and wireless module. The Microprocessor processes users' request, controls sensors and wireless modules. Sensor in this structure is responsible for monitor personal health status, such as blood pressure, blood oxygen saturation, temperature, etc. The wireless modules was used to transmit data or signal, like Bluetooth, Zigbee, Wi-Fi or GSM modules, these portable electronic medical devices can send data to Wireless E-Health Hub. Through hub, the monitoring results can be read by users. The health data can be forward transmitted to server in Cloud and saved in the database for future processing. At the same time, a lot of low power consumption wireless communication techniques have been developed, such as Zigbee and Bluetooth technologies.

Zigbee is a wireless communication technology that has feather of low complexity network, low power consumption and low data rate. These features are particularly suitable for automatic control, sensing, medical monitoring, remote control and other fields. Zigbee module can be embedded into various device, such as medical devices, smart devices and industrial equipments. XBee is one of the Zigbee modules that communicate in short distance, low complexity and low power consumption, low data rate. Zigbee is one of the best solutions for wireless medical devices.

Bluetooth is another excellent solution for wireless medical devices. Bluetooth technology is one of short-range wireless data communication technologies. The typical transmission speed of Bluetooth is 1 Mbps; the general transmission distance is about 10 meters. Bluetooth technology can effectively simplify the communication connection between devices. The Bluetooth system consists of four functional units, wireless unit, link control unit, link management unit and related software. Bluetooth uses 2.4GHz frequency band and time division duplex transmission scheme to achieve full-duplex transmission. The global 2.4GHz frequency band is the international free frequency band; therefore, Bluetooth devices do not have license issues from specialized management agency.

Zigbee and Bluetooth have many common features, for example, both of them are the short-range wireless communication technology and work at 2.4HGz ISM frequency band. On the other hand, they also have their own advantages in wireless medical device development. For example, Bluetooth is one kind of very mature wireless technology; a lot of mobile devices contain Bluetooth Module that used to Bluetooth communication, such as laptop, smartphone or PDA. Therefore, Bluetooth can be used in more areas than Zigbee.

2.5 Mobile communication techniques

In recent years, with the emergence of mobile phone and information technology, telemedicine entered the mobile digital age. The M-Health based on mobile has been built up and it is expected to replace the telephone medical gradually [30-35]. The reason is that the mobile phone provides a portable and real-time platform for health monitor. However, the low speed network of early mobile phone and its limited resources restricted the development of mobile health.

With the rapid development of mobile network and mobile device, high-speed mobile network, 3G and 4G (Long Term Evolution, LTE) have been implemented. [30] The 3G is the third generation mobile communication network with high-speed data transmission cellular mobile communication technology. One of the features of 3G is that it can simultaneously transmit voice (such as call) and data (such as e-mail, instant message, etc.) together in a high speed. 4G also called the Long Term Evolution (LTE) network, which is the evolution of the third generation wireless communication technology with higher bandwidth. On mobile phone side, with the high-speed wireless network development, the mobile phone becomes more and more powerful and intelligent. As a result, the smart phone emerged, such as iPhone, one of the popular smart phones, which are integrated, Internet browsing, mailing and social networking, etc. As the powerful platform of Smart Phone, M-Health stepped into Smart Health period [36-41]. The concept of Smart health can be explained as the public health care and medical practice supported by Smart Devices, like Smart Phone and Smart Sensors [33]. The Smart Phone has a powerful platform and rich resources that support complicated personal health management applications can be operated. Since 2009, a lot of Smart Phone Apps have been developed by u-Health Center of Asan Medical Center (AMC) in Korea [33]. All of these Apps have been used to collect patient’s health records and give them medical advises. The Smart Sensors have been integrated with high accuracy health sensors and various wireless modules that provide portable, reliable and effective health monitoring experience to users. Suzuki, et al. have described a patch type wearable vital monitoring devices that integrated with many vital sensors and a dual mode Bluetooth module [39]. The vital sensors include ECG, pulse and body surface temperature. The dual mode Bluetooth module was used to transfer sensor data to a coordinator and further forward to central server. Furthermore, combination of Smart Phone and Smart Devices is another rising development tendency. For example, the “S-Health App” which provided by Samsung for personal health management [42]. Through “S-Health App”, users can get more comprehensive health management services based on some special Galaxy S4 sensors. Such as the pedometer sensor embedded in the Galaxy 4 can be used to measure user's walk and run distances. After measurement, all the data will be stored in the "S-Health App". Moreover, daily dietary and exercise recommendations will be pushed by S Health App.

The advantage of Smart Health Care System is that it provides real-time personal medical services and allows users to fully understand of their health status. The chronic diseases can be detected early and users can be got effectively treatment as well.

2.6 The Cloud Computing

Cloud computing is a burgeoning technology, which is developed from Distributed Computing, Parallel Computing and Grid Computing. Cloud computing integrates large amount of data and processor resources in a large number of distributed computers, rather than stores them in local computers or remote servers [43]. With the technology of Virtualization, Utility Computing, Cloud computing provides Software as a Service (SaaS), Infrastructure as a Services (IaaS) and Platform as a Services (PaaS). Hardware and software resources can be seen as unlimited expansion in Cloud computing. Users can get accessibility, on-demand, scalable and "pay as go" services from Cloud computing rather than to configure the hardware or software by themselves. Cloud computing has all the features of distributed network, such as distributed computing and store, efficient management. Furthermore, Cloud computing has its own unique characteristics as following:

1. Virtualization

Cloud computing system uses a virtual platform to provide services. Cloud computing stores huge amounts of resources and runs applications in the Cloud somewhere; it is not a specific position or specific server.

2. Reliability

Cloud computing uses multiple copies of data fault tolerance mechanism and all the compute nodes are same institutions which are interchangeable. These two measures ensure the Cloud computing services not collapse because of one node error. Therefore, Cloud computing is more reliable than local server.

3. Scalability

Cloud computing has a large-scale shared pool, and its resources provided to customers as services form. Users' computing size can be dynamic expansion. The size of resources can be purchased by customer demand which can meet the needs of applications and customers' scale growth.

4. Economy

In Cloud computing, users only need to hire related services from Cloud service providers rather than to purchase a large amount of hardware. In addition, the automation and centralized management by Cloud services provider, which allows Cloud customer to avoid the high cost of data center management. Combined with the characteristics of scalable, resource utilization has been greatly improved; and the end users can fully enjoy the low cost advantage offered by the Cloud service provider.

5. Security

There is no necessary to worry about data loss, virus invasion trouble in Cloud computing environment, since there are lots of professional staffs can provide data maintenance and backup to ensure Cloud security. On the other hand, this mechanism helps users to reduce investment in security budget.

Currently, there are lots of IT companies developing their Cloud plans, such as Amazon, Google, Microsoft and IBM, etc.

Amazon Web Service (AWS) [44] is the Amazon Cloud platform which contains three main powerful Clouds. Amazon Elastic Compute Cloud (EC2) provides scalable, pay as you go computing capacity. Amazon Simple Storage Services (S3) provide fully redundant data storage infrastructure, then users can store and retrieve any data from S3. Amazon Relational Database Service (RDS) is a web service, which can easily set up, operate and extend relational database in the Cloud. AWS provide a flexible Cloud computing service to business though which users can build enterprise applications and personal applications.

Google App Engine (GAE) [45] is a SaaS Cloud computing platform developed by Google Company. It was designed for software developers and provided huge amount of computing power and storage space.

Windows Azure Platform [46] is a Cloud computing platform developed by Microsoft. This Cloud platform combined with Windows Azure and a set of platform services.

With the increasing demand of E-Health System, the personal health information will be rapidly increased. Since the Cloud Computing consists of lots of advantages, it is a suitable solution for future E-Health System. The scalability, for instance, Cloud server can support E-Health System with a powerful computing platform. The reliability, Cloud server can provide E-Health System a safety data management environment, as user’s personal measurement information can be stored in Cloud server that has mass storage space. Furthermore, with powerful computing technology and distributed network of Cloud service, the large traffic access can be coped. The security, Cloud server can provide multiple kinds of access limitation and powerful encryption methods, because the users' information is privacy and important. A health care system based on Cloud Computing has been proposed in [47]. The proposed system will be responsible for automate processing collected vital health data from connected sensors through network, and the processed data will be storage, processed and distribution to the medical center's "Cloud". In [48], an efficient Cloud framework for Health Care Monitoring System has been proposed. Through this system, the patients' health data will be collected automatically and be published to a Cloud information repository.

2.7 Conclusion

E-Health System has rapid development with the advancement of technologies. From the early stage, such as web-based health care system, telephone based health care system that provides long distance medical advice. Since the E-Health concept has been proposed in last decades, a lot of emerging E-Health Systems have been presented, like mobile-based M-Health care system, smart device based Smart-Health care system. This research work has reviewed these previous works and proposes a novel E-Health System based on Wireless Sensor Network.

Chapter 3 A Novel E-Health System based on Wireless Sensor Network

As previous chapter introduced, E-Health System has a lot of features that personal health management, disease prevention and control. With the wireless technology advance, fast transmission speed and low power consumption wireless technologies have been integrated into emerging E-Health Care System that produce the mobile, portable, wearable E-Health System. Based on the research of this area, this project proposed and implement a novel E-Health System based on Wireless Sensor Network. The innovation of Proposed E-Health System based on Wireless E-Health Gateway.

- Multiple Wireless Connection Choose
- Plug in and Play
- User Friendly Operation and Interface Design

This chapter describes the E-Health System based on Wireless E-Health Gateway at first section. Three main components, Wireless E-Health Gateway, Wireless E-Health Sensor Node and E-Health Management System based on Cloud Service have been introduced. Second part of this chapter presents the implementation of Wireless E-Health Gateway, Wireless E-Health Sensor Node and E-Health management System based on Cloud Service. Further more, two extension systems have been discuss and developed in third section, E-Health System based on Wireless Sensor Node and E-Health System based on smartphone. The experiments show that the proposed system has achieved the good results.

3.1 E-Health System based on Wireless E-Health Gateway

E-Health System based on Wireless E-Health Gateway is a portable, functional and powerful personal E-Health care system that used to measure and manage user’s vital signs. This powerful system integrated with Wireless E-Health Gateway, Wireless Sensor Nodes and E-Health Management System based on Cloud as Figure 3-1 shows.

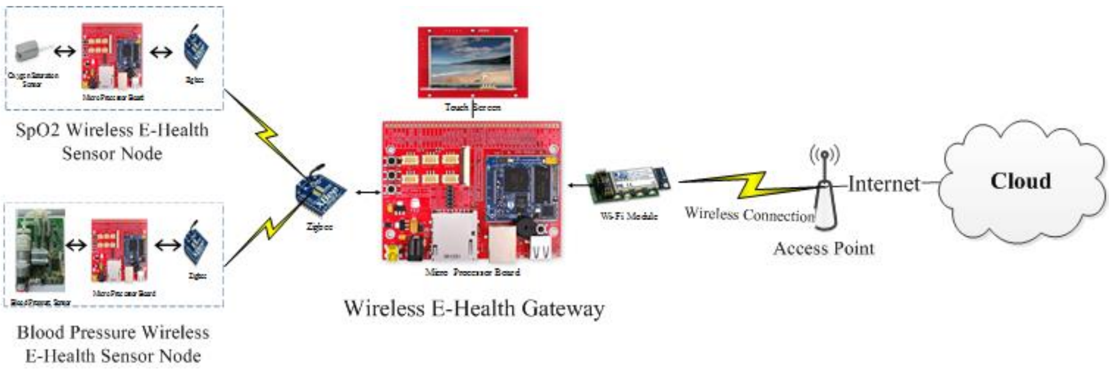


Figure 3-1 The architecture of proposed E-Health System based on Wireless E-Health Gateway.

Wireless E-Health Gateway as the center controller in the system used to communicate with different Wireless E-Health Sensor Nodes. At the same time, the Wireless E-Health Gateway can transmit personal health data between local and Cloud via Wi-Fi communication as the Figure 3-1 shown. Wireless E-Health Sensor Nodes are response for measurement of personal vital sign. The E-Health Management System has been setup in the Cloud platform, and the system used to manage, storage and analysis the personal health data. The following sections explicate each part of the proposed system in details.

3.1.1 Wireless E-Health Gateway

Wireless E-Health Gateway is one complex and powerful part of E-Health System. The main functions of this system are as follow.

- E-Health terminal, i.e. the Wireless E-Health Gateway can be used as a health care management center. Users can store and manage their personal information and health records in this system.

- Wireless E-Health Gateway, i.e. the gateway can connect with a lot of different wireless sensor nodes and build up wireless sensor network.

- Cloud terminal, i.e. the Wireless E-Health Gateway can communicate with Cloud Server. The health measurement data and services can be downloaded and uploaded from and to the Cloud Server.

3.1.2 Wireless E-Health Sensor Node (WESN)

Two Wireless E-Health Sensor Nodes (WESNs) have been proposed in this research, one is Blood Pressure Wireless E-Health Sensor Node (BPWESN) and another is Oxygen Saturation Wireless E-Health Sensor Node (OSWESN). The structure of Blood pressure sensor node and Oxygen Saturation sensor node are shown in Figure 3-2.

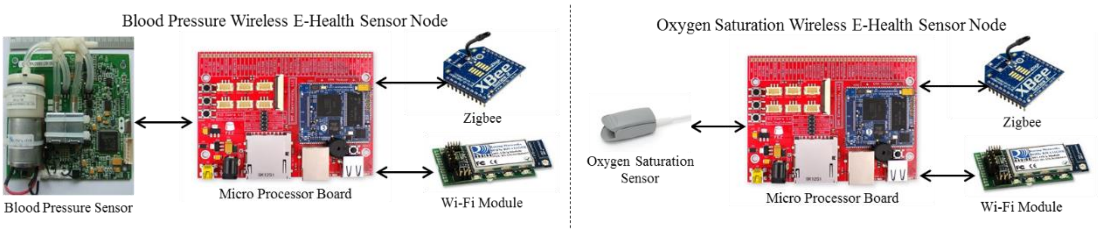


Figure 3-2 Blood Pressure Wireless E-Health Sensor Node (left) and Oxygen Saturation Wireless E-Health Sensor Node (right)

- **Blood Pressure Wireless E-Health Sensor Node**

Blood Pressure Wireless E-Health Sensor Node (BPWESN) is used to monitor users' blood pressure. The architecture of the proposed blood pressure sensor is shown in Figure 3-2 left, which includes four modules: one blood pressure sensor, microcontroller and two wireless modules (Zigbee module and Wi-Fi module).

- **Oxygen Saturation Wireless E-Health Sensor Node (OSWESN)**

Oxygen Saturation (SO2) means that the capacity percentage of Oxygenated Haemoglobin (HbO2) bonded with oxygen in the total capacity of Haemoglobin (Hb), which is also called the concentration of oxygen in the blood [49]. Oxygen Saturation is a very important parameter in respirator cycle of daily life and in clinical emergency treatment. Because many clinical diseases can cause a lack of oxygen supply, and directly affect the normal cell metabolism. What's worse, it will be a serious threat to human life when it is in very low concentration. In normally, the minimum value of oxygen saturation should be 94%. If the oxygen saturation is less than 94%, it means the user does not have enough oxygen and needs oxygen support.

Oxygen Saturation Wireless E-Health Sensor Node (OSWESN) is a continuous, non-invasive and portable blood oxygen saturation monitor which can be used to monitor users' oxygen saturation in real-time. The architecture of Oxygen Saturation Wireless E-Health Sensor Node is shown in Figure 3-2 right. The OSWESN consist of four parts: an oxygen saturation sensor, microcontroller and two wireless modules, Zigbee module and Wi-Fi module.

3.1.3 Cloud Service for E-Health

The Cloud service provides a powerful platform to management and storage E-Health data that transfer from E-Health Gateway shown in Figure 3-1. Cloud service promotes the process of medical informatization and greatly reduces the cost of construction of medical information system and help medical institutions to share information and resources, and furthers improve the level of medical services. In the hospital, the patient is the center of the entire medical procedure. There are mass user's data need to be processed and stored in the hospital. Therefore, hospitals need data center and computing center to handle and store these mass data. Cloud service can significantly reduce the upfront costs of hardware and software. Hospitals only need to purchase less service from Cloud service provider. In future, as the system expanded, the hospitals can use less money to purchase more services from Cloud service provider. Otherwise, the hospitals need to use lot money to upgrade equipment every time.

Utilizing the powerful computing capability and mass storage space, the Cloud can be a giant network server center to integrate the medical institutions and share medical resources. Moreover, Cloud service provides a personalized service to patients, where users can upload and store their personal information, medical records and medical requirement. Furthermore the Cloud service can be accessed by users via Internet from everywhere and anytime.

Cloud service provides a powerful computing ability for medical institutions to process health data. For example, Cloud can store and offer all of patient’s health records and drug using records to medical institutions to analyze. They can analyze the patients’ some physiological parameters, such as blood pressure, heart rate and oxygen saturation, etc. After timing analysis and processing, medical institutions can get the result and feedback to patients. Through Cloud computing, medical institutions can get the most powerful processing platform and then improve their efficiency of medical care.

3.2 E-Health System based on Wireless E-Health Gateway development

3.2.1 Wireless E-Health Gateway Management System development

The Figure 3-3 shows the construction of the Wireless E-Health Gateway. The proposed Wireless E-Health Gateway consists of four components, microprocessor, Zigbee Module, Wi-Fi module and the Touch Screen. The microprocessor is shown in the middle, used to control the entire Health Gateway system. In this project the microprocessor uses FEZ Cobra development board which is from GHI Electronics Company. This development board fully supports Micro .Net Framework 2.0 which is an Open-Source Embedded System platform. The touch screen is a 3.5 inch TFT display. This touch screen provides FEZ Cobra with the ability to display graphics with 16-bit colour depth. Touch screen connects with the FEZ Cobra board through a 5” flex cable.

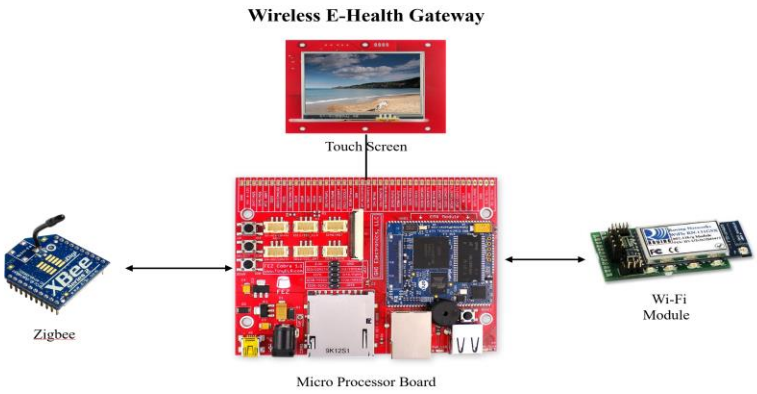


Figure 3-3 Wireless E-Health Gateway Structure

The Wi-Fi module uses WiFly GSX (RN-134) module which is a complete ultra-low power embedded TCP/IP solution. It supports 802.11b/g networking protocol and offers the ability to wake up, connect to a wireless network, send and return to sleep mode in less than 100 milliseconds. This module operates in the 2.4GHz ISM band. The communication between the microprocessor and the Wi-Fi module is through UART interface.

The XBee module is a Zigbee module based on IEEE 802.15.4 networking protocol from Digi Company. This module provides fast point-to multipoint or peer-to-peer networking. XBee modules are low-power, low-cost applications. The connection between the FEZ Cobra board and the XBee module is through the UART port.

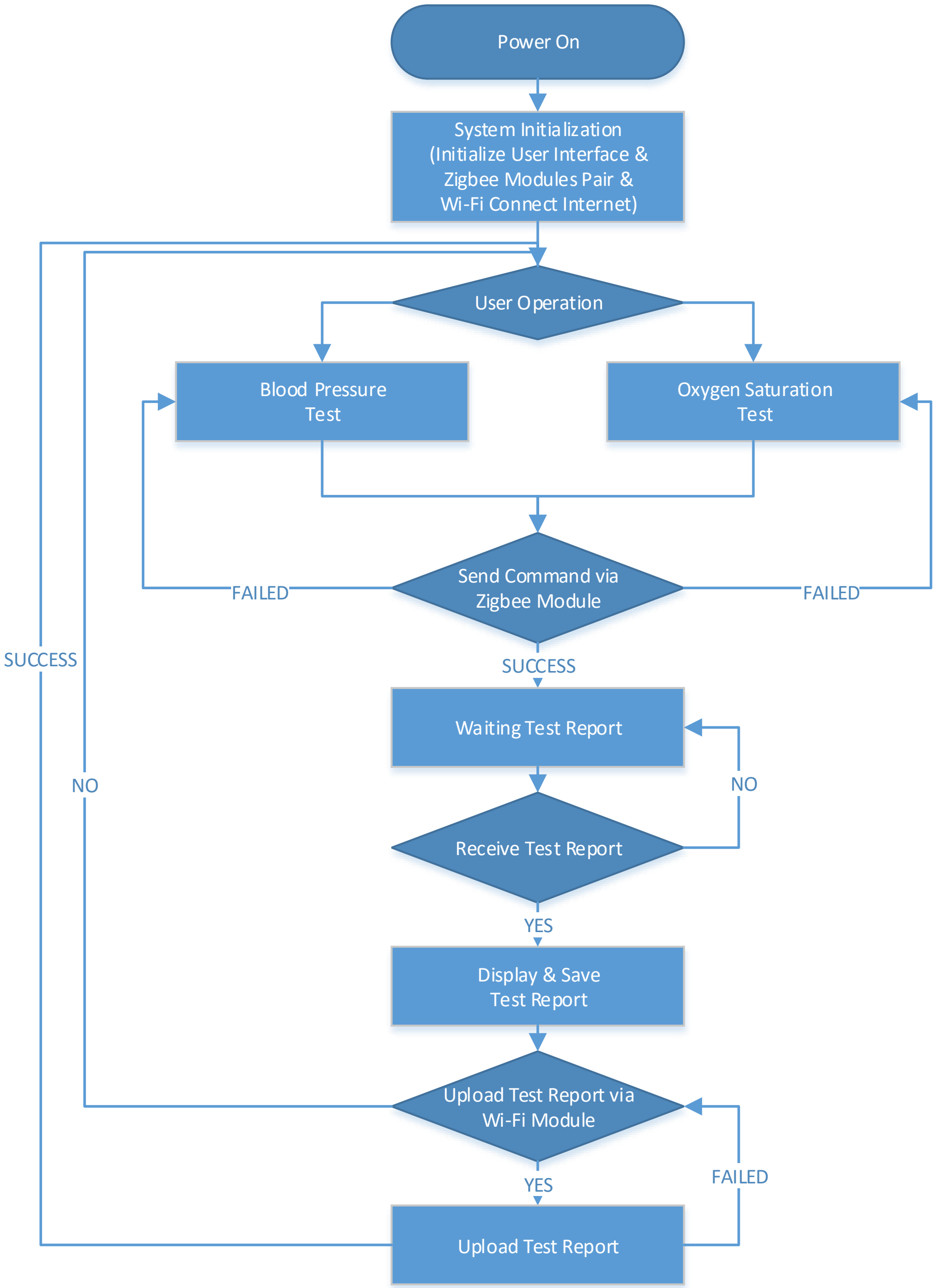

Power On
System Initialization (Initialize User Interface & Zigbee Modules Pair & Wi-Fi Connect Internet)
User Operation
Blood Pressure Test
Oxygen Saturation Test
Send Command via Zigbee Module
FAILED
FAILED
SUCCESS
SUCCESS
Waiting Test Report
NO
NO
Receive Test Report
YES
Display & Save Test Report
Upload Test Report via Wi-Fi Module
YES
FAILED
Upload Test Report

Figure 3-4 The flow chart of Wireless Health Gateway System

Figure 3-4 illustrates the flow chart of Wireless E-Health Gateway System. The Wireless E-Health Gateway starts initiation when the device power on. During the initialization process, the user interface will be initialized and displayed in the touch screen. Then the wireless modules will be configured, for example, the Wi-Fi module should be accessed with the Internet and the Zigbee modules should be connected as a pair. After health gateway initialization, users can use this device to control WESNs and carry out measurement. If user chooses blood pressure test, then the Gateway sends Blood Pressure Test command to BPWESN through Zigbee Module. The Oxygen Saturation measurement is similar as the Blood Pressure Test process; however, it sends different commands to OSWESN. After measurement, the WESN will return its result to the Wireless Health Gateway through paired Zigbee modules. Zigbee module driver code has show in Appendix III. Then the gateway will process data and save it to memory, or external storage, such as SD card. At last, the measurement data will be displayed on the touch screen in human friendly format. If users wish to send this result to health Cloud, the measurement data can be packaged and uploaded to Health Cloud through the Wi-Fi module. Wi-Fi driver code has been append in the Appendix II.

HDATA	**Systolic BP**	**Diastolic BP**	**Average BP**	**Pulse Rate**	**SPO2**

Figure 3-5 Wireless Health Gateway upload message format

As shown in Figure 3-5, the format of the upload message has been defined that contains six components. “HDATA” (Health Data) is defined as the message head, and it is transmitted from Wireless Health Gateway and it contains the measurement report. The following five parameters are systolic blood pressure values, diastolic blood pressure values, average blood pressure value, pulse rate and oxygen saturation value.

As previous chapter described, Wireless E-Health Gateway is a multifunctional E-Health component. In this project, a Wireless E-Health Gateway has been developed that integrated microprocessor, display, and varieties of wireless communication portable devices. The main function of Wireless E-Health Gateway is to control wireless sensor nodes to monitor personal health status. Health data will be collected from sensors and processed by microprocessor. Final, all the health data will be uploaded to Cloud server for storing and further analysis. The structure of the E-Health Gateway interface is shown in Figure 3-6. The system interface composed of some user windows. One main window, four function windows and one sub-window have shown in Figure 3-7. The main window called desktop, which contain four function buttons, SpO2 button, Blood Pressure button, Report button and Set button. Each button is corresponding to the respective function window. Appendix I describe the E-Health Gateway windows code.

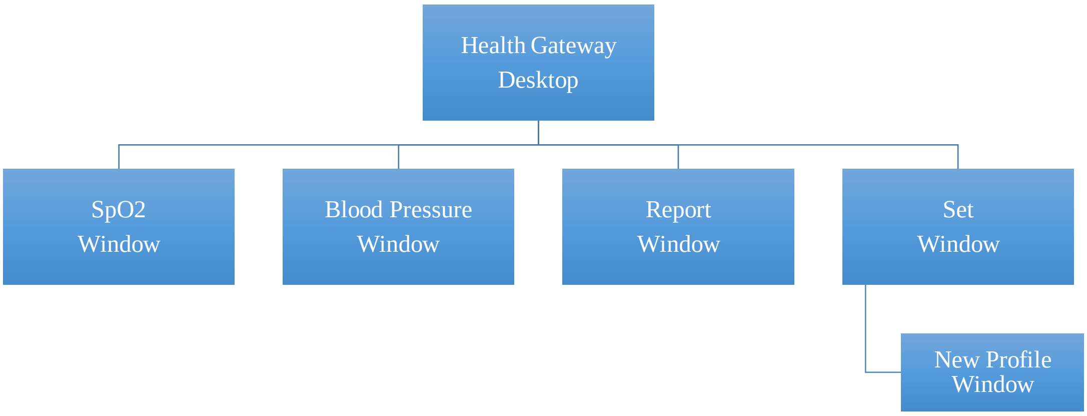
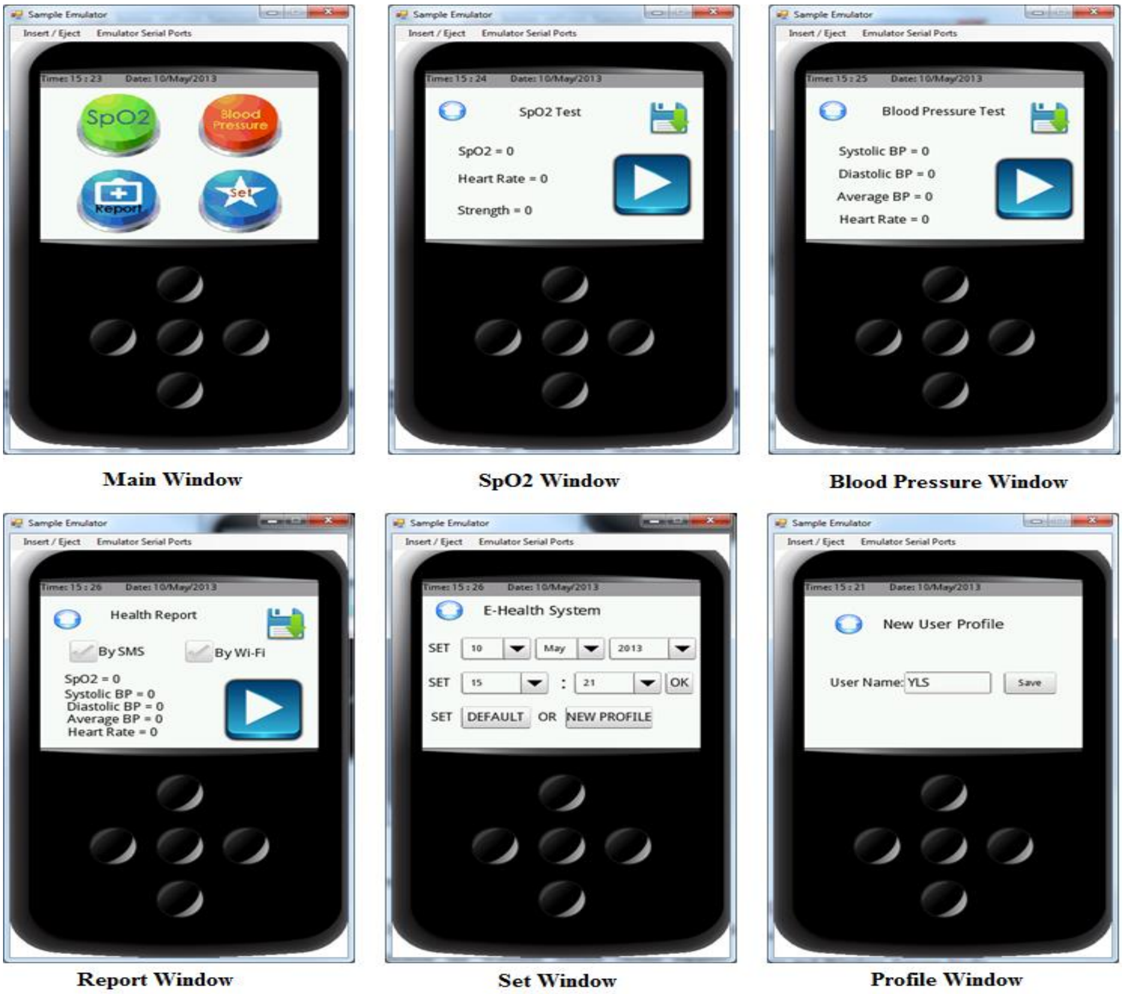


Figure 3-6 Structure of E-Health Gateway Interface



Time: 15 : 23 Date: 10/May/2013

Main Window

Time: 15 : 24 Date: 10/May/2013

SpO2 Test

SpO2 = 0

Heart Rate = 0

Strength = 0

SpO2 Window

Time: 15 : 25 Date: 10/May/2013

Blood Pressure Test

Systolic BP = 0

Diastolic BP = 0

Average BP = 0

Heart Rate = 0

Blood Pressure Window

Time: 15 : 26 Date: 10/May/2013

Health Report

□ By SMS □ By Wi-Fi

SpO2 = 0

Systolic BP = 0

Diastolic BP = 0

Average BP = 0

Heart Rate = 0

Report Window

Time: 15 : 26 Date: 10/May/2013

E-Health System

SET 10 May 2013

SET 15 : 21 OK

SET DEFAULT OR NEW PROFILE

Set Window

Time: 15 : 21 Date: 10/May/2013

New User Profile

User Name: YLS Save

Profile Window

Figure 3-7 Health Gateway Windows

The SpO2 button corresponds to the SpO2 window. In the SpO2 window, there is one Home button on the left corner that used to return to desktop. There are two buttons on the right side of the window, one is save button, and another is test button. The save button is on the left corner, which is used to save the measurement data into SD card. The test button under the save button is used to control OSWESN to start test person's oxygen saturation. After measurement, the test result will be return to the Wireless E-Health Gateway and display on the result panel that in the middle of the SpO2 window. The result contain three values, the SpO2 label shows the value of the oxygen saturation, the Heart Rate label shows the heart rate value and the Strength label used to inform user the strength of the signal.

The Blood Pressure button connects with the Blood Pressure (BP) Window. The layout and function of the Blood Pressure is same as the SpO2 window that contains three buttons (Home button, Save button and Test button) and a result panel. The result panel lists four blood pressure parameters, Systolic B, Diastolic BP used to, Average BP and Heart Rate, which used to display the value of corresponds' blood pressure.

The Report button is associated with the Report Window. The Report window is similar with above two windows, but there are two more option boxes in the middle of the Report Window. One option box called 'By SMS' means the report will be sent to patient's relatives or personal physician by short message, if this option box has been checked. Another option box called 'By Wi-Fi', which means the report will be uploaded by Wi-Fi module through Internet, if this option box has been checked. The result panel in Report Window incorporates all the test parameters, which means the report button will be enabled after Oxygen Saturation Measurement and Blood Pressure Measurement.

The forth button, Set Button used to set system time and user's profiles. In the Set Window, the first row drop down list used to set the system date, such as day, month and year. The second row of drop down list can set the system time, such as hour and minute. Wireless E-Health Gateway system uses 24 hours. The 'OK' button behind the minute drop list used to confirm and start system time. If users already have save the personal profile, then they can pressure the default button to use the default profile that saved in SD card. Otherwise, if a new user uses this system first time, the new user needs to save his profiles into the system. The new profile button was associated with the New Profile Window.

Through the Wireless E-Health Gateway management system, the patients can test their oxygen saturation and blood pressure anytime and anywhere. In the meantime, the health gateway can upload and report measurement data to Cloud server.

3.2.2 Proposed Wireless Sensor Node Development

- **Blood Pressure Wireless E-Health Sensor Node**

The proposed blood pressure node uses microprocessor board to coordinate the Blood Pressure Module closely with Wireless modules as Figure 3-2 left shown. The microprocessor board uses the FEZ Cobra Board that is produced by GHI Electronic Company. This development board fully supports Micro .Net Framework 2.0 which is an Open-Source Embedded System platform. This board contains an EMX Module which offers a robust foundation, such as 16MB of RAM, 4.5 MB of flash memory, Ethernet and Graphics Support. In addition, this board offers rich interfaces, such as UART (Universal Asynchronous Receiver/Transmitter), SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), GPIO (General-Purpose Input/Output), PWM (Pulse Width Modulation), ADC (Analog to Digital Converter), DAC (Digital to Analog Converter), etc. In the proposed blood pressure nodes, the development board connects with Blood Pressure module and Wireless Modules through UART interfaces. For example, the UART 1 connects with Blood Pressure, UART 2 connects with Zigbee module and the UART 3 connects with Wi-Fi module.

The Blood Pressure Module is a non-invasive measuring instrument that uses Oscillometric method to measure Blood Pressure. The principle of Oscillometric method is to use inflatable cuff to block artery and measure the pulse envelope sasser in the blood vessel wall during the slow deflating process. After deflating process, the blood pressure can be measured through calculating the pulse envelop sasser and arterial pressure.

In the proposed blood pressure node, there are two kinds of wireless modules, Zigbee and Wi-Fi. Zigbee wireless technology is used for communication between node and E-Health Gateway through peer-to-peer communication link. Wi-Fi wireless technology is used for communication between node and mobile devices through Ad-Hoc network.

The flow chart of proposed blood pressure node is shown in Figure 3-8. The node starts initialization when the power is on. During the initialization period, the Microprocessor will open the three UART ports and configure them. After initialization, two wireless modules start to receive commands from server. If the measurement command arrives, the blood pressure module starts measurement. Then the result of blood pressure will be pushed to wireless module after testing. However, if measurement fails, the module will restart or abort till next measurement command arriving. Finally, the wireless module will send health data to health gateway or mobile devices. After measurement, the wireless modules will turn to sleep mode automatically.

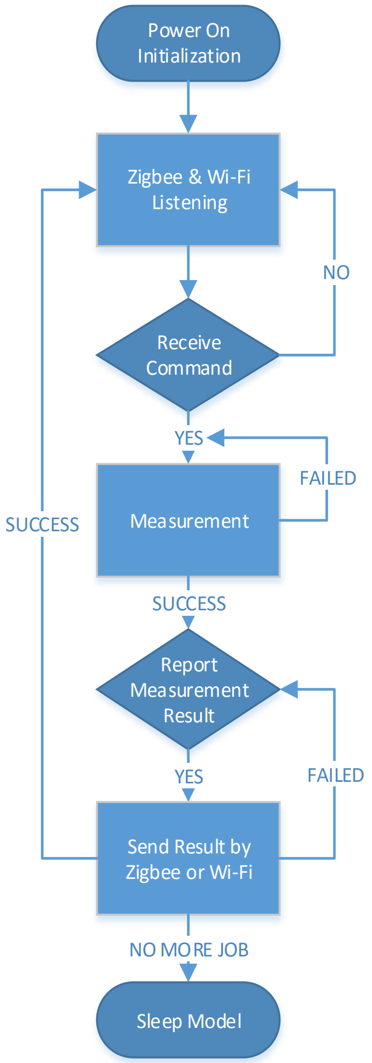


Figure 3-8 The proposed blood pressure Node flow chart

The measurement data include five parameters separated by delimiter “|”. The result message format is shown in Figure 3-9. The first parameter “NIBP” (Non-Invasive Blood Pressure) was defined as the blood pressure sign and it marks the message as the blood pressure measurement result. The second parameter was Systolic Blood Pressure data (Systolic BP), the third one was Diastolic Blood Pressure data (Diastolic BP), the forth one was Average Blood Pressure data (Average BP) and the last one was Pulse Rate data. For example, “|NIBP|120|86|97|78|” refers to a Blood Pressure measurement result, and the Systolic Blood Pressure value is 120, the Diastolic Blood Pressure value is 86, the Average Blood Pressure value is 97 and the Pulse Rate is 78 per minute.

NIBP	Systolic BP	Diastolic BP	Average BP	Pulse Rate

Figure 3-9 Result message format

The formula of the Average Blood Pressure is shown in the equation (1). The value of average blood pressure is systolic blood pressure value plus two times of diastolic blood pressure value and then the result divided by three.

$$AverageBP = (SystolicBP + 2 \times DiastolicBP)/\ 3 \qquad (1)$$

- **Oxygen Saturation Wireless E-Health Sensor Node (OSWESN)**

Oxygen Saturation Wireless E-Health Sensor Node (OSWESN) architecture has shown in Figure 3-2 (right). The OSWESN consist of four parts: an oxygen saturation sensor, microcontroller and two wireless modules, Zigbee module and Wi-Fi module.

The microprocessor in the OSWESN is used to manage the communication between Wireless Modules, E-Health Gateway and Smart devices, and it controls Oxygen Saturation Module to do measurement.

The fingertip photoelectric sensor is used in this project to measure users' oxygen saturation in blood. This is a continuous, non-invasive and portable blood oxygen saturation measure device. User puts the sensor on their finger and then starts test. The sensor uses a finger as a transparent container that is full of haemoglobin (Hb) and then it uses red light (wavelength of 660 nm) and near-infrared light (wavelength of 940nm) as the incident light. The oxygen saturation result can be calculated by determination of light transmission through the tissue blood intensity.

OSWESN flow chart is shown in Figure 3-10. The fingertip photoelectric sensor is a plug and play device. Therefore, the sensor will start to test oxygen saturation automatically when user's finger plugs into the probe. The first step of OSWESN is to initialize the wireless modules and sensor after power on. Zigbee module and Wi-Fi module will turn to sleep mode for power-saving and the SpO2 module will start to monitor the status of probe. If the probe attached, then microprocessor sends commands to wake up the Zigbee module and Wi-Fi module. At the same time, the SpO2 module begins to test oxygen saturation every second. The measurement data will be sent via wireless modules when they receive result request commands only. Then the SpO2 test process will be stopped till the sensor probe has detached.

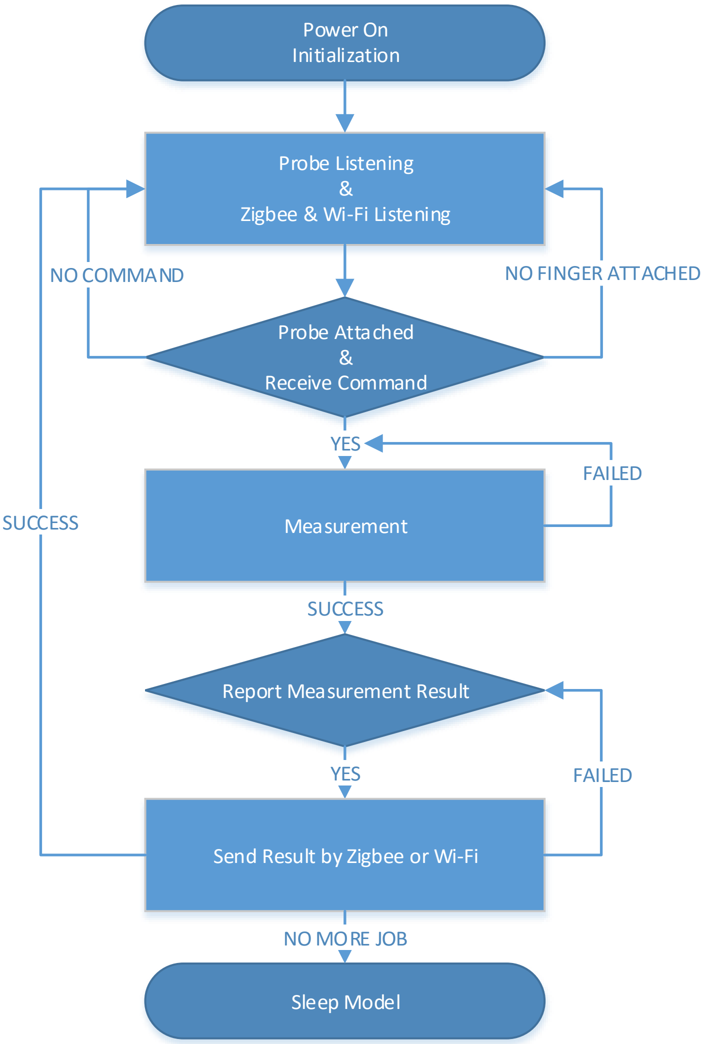


Figure 3-10 Proposed Oxygen Saturation Node Flow Chart

The measurement data include four parameters separated by delimiter “|”. The format of OSWESN message has been defined as Figure 3-11 shown. The first parameter “SPO2” is defined as the oxygen saturation sign that marks the message as the oxygen saturation measurement result. The second parameter is oxygen saturation, the third one was pulse rate and the forth one is the signal strength. For example, “|SPO2|98|78|4|” means the oxygen saturation measurement message and the oxygen saturation value is 98, the pulse rate is 78 per minute and the signal strength is 4.

SPO2	OXYGEN SATURATION	PULSE RATE	SIGNAL STRENGTH

Figure 3-11 OSWESN Result message format

3.2.1 Proposed Cloud E-Health Management System

In this research, a proposed Cloud E-Health Management System based on Amazon Web Service has been developed. The Cloud Health Management System holds in the Cloud Server and provides personal health services. The aim of the Cloud Health Management System is to provide user access and manage their personal health data. The structure of the web server is show in Figure 3-12. IIS (Internet Information Server) is one kind of Microsoft Web Server, which has specific ASP.NET processing engine and used to configure ASP.NET environment. When the asp.net request coming, the IIS will process the request and return the corresponding content. ASP.NET is a server-side scripting technology that can be embedded into web page and can be run in the Internet Server. The Database is a data collection unit to organize, store and mange user information. In this project, Microsoft SQL Server was used as Database to store and mange users’ information.

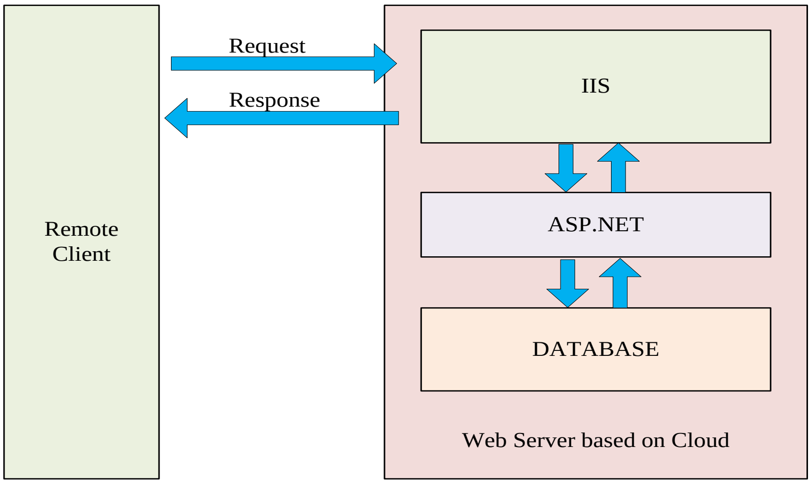
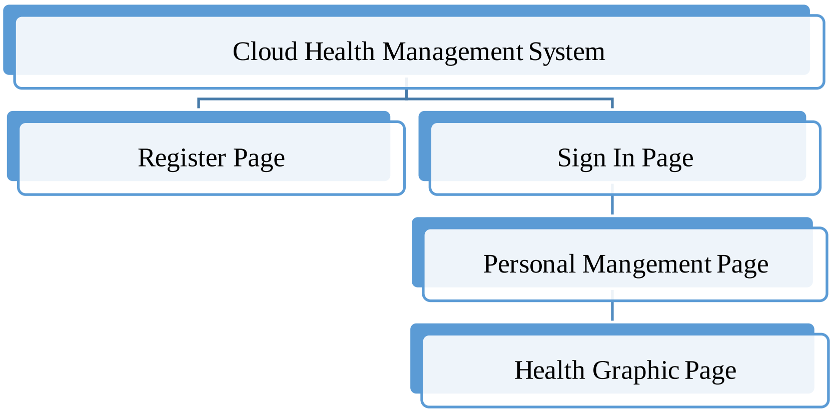


Figure 3-12 Basic Architecture of Web Server

The basic architecture of the web server based on Amazon Web Service as shown in Figure 3-12. The remote client send http request to web server based on Cloud for resources. Then the IIS components process the request and access the appropriate ASP.NET application to request resources. The corresponding ASP.NET application will process the request and response the IIS component with the required resources. If the ASP.NET response needs data from database, then it will send request to Database. After finding the resources, the Database responses the data. Then the ASP.NET application packages the data and sends back to IIS component. Finally, the IIS components will responses the remote client with the appropriate ASP.NET web pages.

Figure 3-13 shows the structure of the proposed Cloud E-Health Management System. The system has two branches; one is for new users who need to register, and another for users who already registered. In register page user need to fill up their personal information in the form and submit to server to complete register process. The 'Sign In Page' used for user to sign in the system. As shown in the diagram, under the 'Sign in Page' node, there are two management nodes. The first node is Personal Management Page that used to show personal information and his health data. The second node is Health Graphic Page that uses linear graph to present the user's health data.


Cloud Health Management System

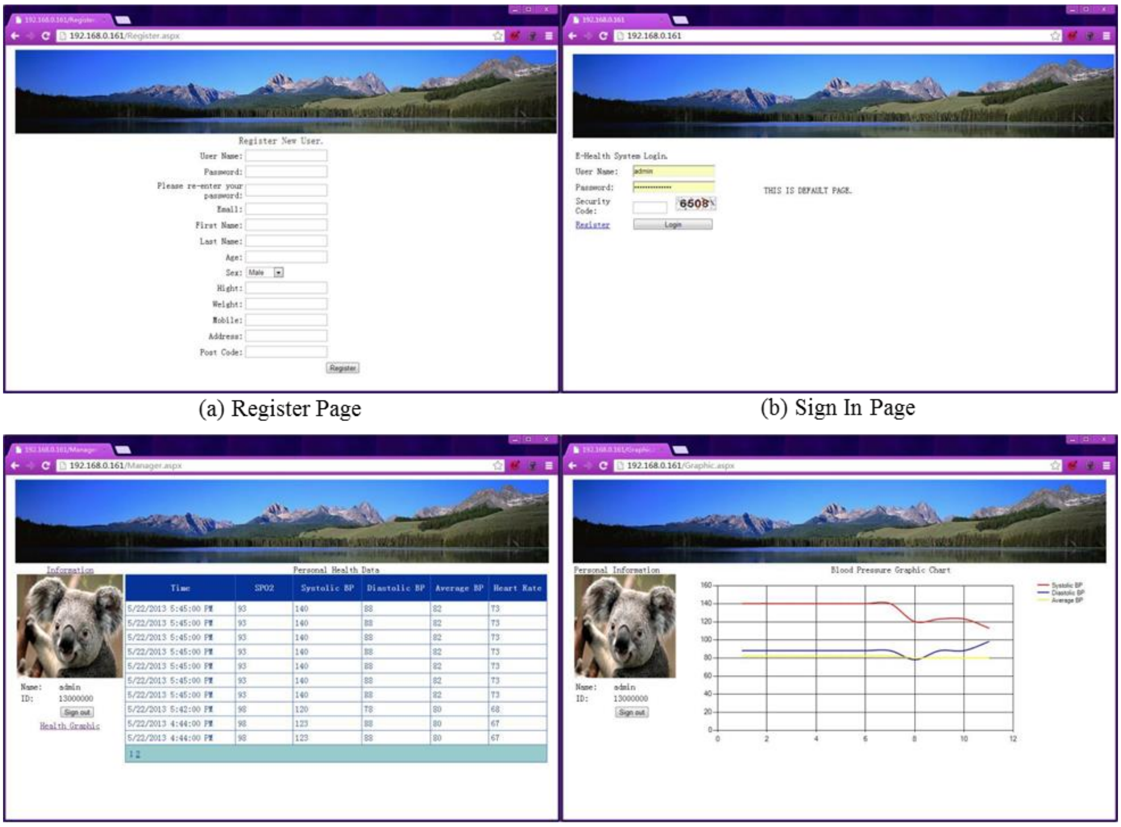
Register Page

Sign In Page

Personal Mangement Page

Health Graphic Page

Figure 3-13 Structure of the Cloud Health Management System


192.168.0.161/Register.aspx

Register New User.

User Name:

Password:

Please re-enter your password:

Email:

First Name:

Last Name:

Age:

Sex: Male

Hight:

Weight:

Mobile:

Address:

Post Code:

(a) Register Page

192.168.0.161

E-Health System Login.

User Name: admin

Password: •••••••••••••••

Security Code: 6508

THIS IS DEFAULT PAGE.

(b) Sign In Page

192.168.0.161/Manager.aspx

Name: admin

ID: 13000000

Personal Health Data

Time	SPO2	Systolic BP	Diastolic BP	Average BP	Heart Rate
5/22/2013 5:45:00 PM	93	140	88	82	73
5/22/2013 5:45:00 PM	93	140	88	82	73
5/22/2013 5:45:00 PM	93	140	88	82	73
5/22/2013 5:45:00 PM	93	140	88	82	73
5/22/2013 5:45:00 PM	93	140	88	82	73
5/22/2013 5:45:00 PM	93	140	88	82	73
5/22/2013 5:45:00 PM	93	140	88	82	73
5/22/2013 5:42:00 PM	98	120	78	80	68
5/22/2013 4:44:00 PM	98	123	88	80	67
5/22/2013 4:44:00 PM	98	123	88	80	67

1 2

(c) Table view of Health data

192.168.0.161/Graphic.aspx

Personal Information

Name: admin

ID: 13000000

(d) Line Chart view of Health data

Figure 3-14 Proposed Cloud E-Health Management System for patient

Figure 3-14 illustrates the proposed Cloud E-Health Management System for patient. (a) has shown the register page that used for new user registration. (b) has presented the sign in page that used for already user login. (C) and (d) were two kind of view model to display personal health data. With different view modes, user can more intuitive understand the health status change.

3.3 Implementation of E-Health System based on Wireless E-Health Sensor Node

The proposed Wireless E-Health System based on Wireless E-Health Sensor Node is the simplest, efficiency and smallest E-Health Systems. The system includes medical sensor integrated with Wi-Fi technology which can do measurement and transmission data through the Internet. The Wireless Sensor Node can be chosen as Blood Pressure Node, Oxygen Saturation Node, Temperature Node, etc. This system can access the Cloud database directly through the Internet. Wi-Fi is one of widely used wireless communication technologies at present, which can access to Internet seamless. The proposed architecture of the Wireless E-Health System is shown in the Figure 3-15.

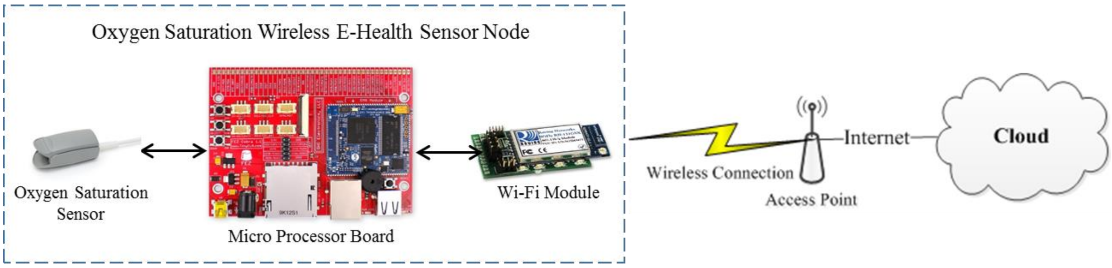


Figure 3-15 Wireless E-Health Sensor Node System Architecture

The proposed simplest Wireless E-Health System built up by three components: Wireless E-Health Sensor Node (WESN), Wi-Fi Access Point (AP) and the Cloud Server. This research project chooses BPWESN and OSWESN as WESN. The WESN connects to the specified Wi-Fi AP through the Wi-Fi module. After testing, the WESNs will access and upload data to the database in the Cloud through the Internet. The flow chart of this system operation is shown in Figure 3-16.

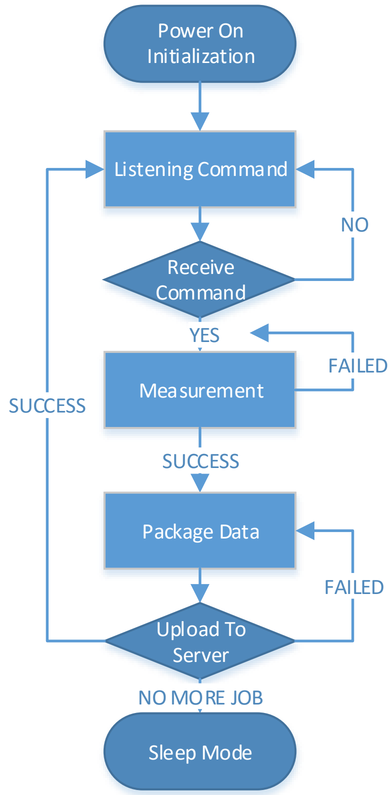


Figure 3-16 the Proposed Simplest Wireless E-Health System Flow Chart

The proposed simplest Wireless E-Health System initializes after power on and waits operation by user. After testing, the measurement data were packaged and uploaded to Cloud system. If uploading is successful, then the device will turn to sleep mode and wait next test operation. Otherwise, it will upload again, or abort this thread and report an error message to user. The Figure 3-17 shows the format of Wireless E-Health System uploading message.

WES Flag	Parameter 1	Parameter 2	………	Parameter n

Figure 3-17 Message Format of proposed Wireless E-Health System

As shown in Figure 3-17, the message of the proposed Wireless E-Health System contains the WES flag and its parameters. The WES flag means the role of the sensor, for example, “NIBP” flag is defined as BPWESN message and “SPO2” flag is defined as OSWESN message. The parameters are integer type or float type. The WES flag and parameters in the package are separated by delimiter “|” which is easy for server to read and store in the database. Through this system, the measurement data can be uploaded to cloud database directly without passing the transfer station. Therefore, portability, convenience and efficiency have been improved.

3.4 Development of E-Health System based on Smart Phone

With the development of Smart Phone, it provides a powerful and portable platform for E-Health care applications. The concept of Smart Health is to use Smart Phone Platform to provide health care services. In this project, two Smart Health Apps are developed, called Blood Pressure App and SpO2 App, which can be deployed in the proposed E-Health System.

The architecture of the Smart Health function based on Smart Phone is shown in Figure 3-18. The Smart Phone connects with WESNs through Wi-Fi link and controlled by corresponding app. For example, Blood Pressure App controls the BPWESN and SpO2 App controls OSWESN. After measurement, the results will be upload to the Cloud server through GSM/GPRS/3G network by Smart Phone.

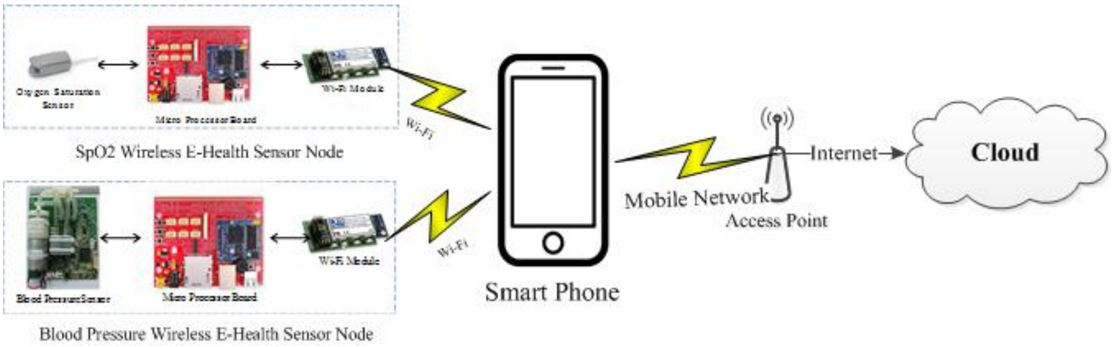


Figure 3-18 E-Health System based on Smart Phone

The workflow of the Smart Health App is shown in Figure 3-19. When WESN is power on, an Ad-Hoc wireless network will be set up to allow Smart Phone access. After Smart Phone joins into the Ad-Hoc network, the WESN will be connected with Smart Phone. The communication link between phone and the WESN is based on

Transmission Control Protocol/Internet Protocol (TCP/IP). The WESN opens a TCP port during its initialization and wait for request from clients. During the connection, the Smart Health app tries to connect the remote WESN TCP port and send connected message. Once the link is established, the smart app sends test command to WESN for measurement. If measurement is successful, the result will be sent back to Smart Phone from the WESN. Otherwise, the WESN will send error information to Smart Phone, and request future process or re-test. After result is processed, then the data will be stored in the memory and display on the screen. Furthermore, the health data can be uploaded to health Cloud server and the format is the same as the Wireless E-Health Gateway upload message format.

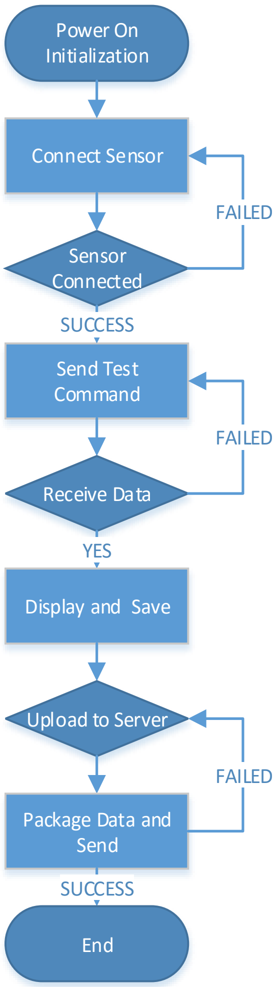
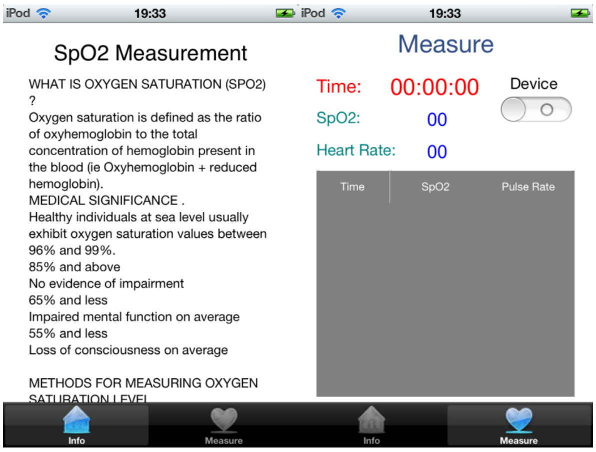


Figure 3-19 Smart Phone’s E-Health App Work Flow

3.4.1 E-Health App Development

In this project, two E-Health Apps based on iOS have been developed. Since the Wireless E-Health Sensor Node support Ad-Hoc network, therefore, the communication link between Apps and WESN is based on TCP/IP communication protocol. Before test, the Smart Phone needs access the WESN Ad-Hoc network, and then the Smart Phone and the WESN join the same network. After connection established, the App can control WESN to do measurement.

The SpO2 Measurement App as in Figure 3-20 is used to control OSWESN. There are two tabs (Info tab and Measure tab) in the app; the first tab is used to acquire the oxygen saturation information. The Second tab is used to control OSWESN. The “Device” switch in the App’s interface, which is used to connect to the OSWESN. After the connection established, the test command will be sent to the OSWESN automatically. At the same time, the “Time” control in red words will start timing also. Every ten seconds, the OSWESN will return a value of the Oxygen Saturation to the app and the data will be displayed in the data grid (the gray area in the Figure). The test will token one minute and six oxygen data will be sent to Smart Phone. Finally, SpO2 label and Heart Rate label will show the result of the test. The upload function will be added into this App in future development.



SpO2 Measurement

WHAT IS OXYGEN SATURATION (SPO2) ?
Oxygen saturation is defined as the ratio of oxyhemoglobin to the total concentration of hemoglobin present in the blood (ie Oxyhemoglobin + reduced hemoglobin).
MEDICAL SIGNIFICANCE .
Healthy individuals at sea level usually exhibit oxygen saturation values between 96% and 99%.
85% and above
No evidence of impairment
65% and less
Impaired mental function on average
55% and less
Loss of consciousness on average

METHODS FOR MEASURING OXYGEN SATURATION LEVEL

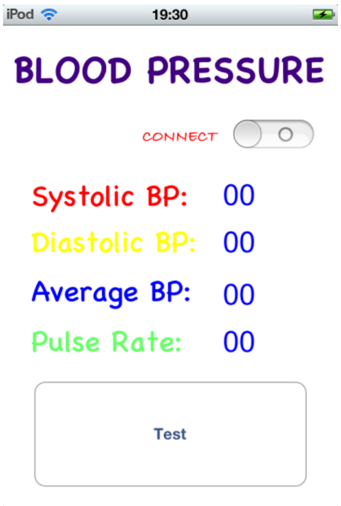
Measure

Time: 00:00:00
SpO2: 00
Heart Rate: 00

Time	SpO2	Pulse Rate

Figure 3-20 SpO2 Measurement APP

Another Smart Health App is the Blood Pressure Measurement App that is used to control Blood Pressure Wireless E-Health Node. This app can measurement user’s systolic blood pressure, diastolic blood pressure, average blood pressure and pulse rate. Figure 3-21 shows the interface of the Blood Pressure Measurement App that is very simple, friendly and easy to use. There is a switch called “Connect” and one button called “Test”. The colorful labels used to display the result of the measurement. The “Connect” switch response for establishing the connection between Wireless E-Health Sensor Node and Smart Phone. The “Test” button used to control BPWESN to start or stop measurement. After BPWESN measurement, the result will be return to the app and display on the screen. Furthermore, the upload function will be added into this app later.



Systolic BP: 00
Diastolic BP: 00
Average BP: 00
Pulse Rate: 00

Figure 3-21 Blood Pressure Measurement APP

3.5 Conclusion

In this chapter, a new E-Health System based on Wireless Sensor Network has been proposed. The Wireless Sensor Network was consisting of E-Health Gateway and Wireless E-Health Sensor Nodes. Blood Pressure Wireless E-Health Sensor Node and Oxygen Saturation Wireless E-Health Sensor Node have been developed. The BPWESN has been implemented to monitor patients’ blood pressure, and the OSWESN was used to monitor patients’ oxygen saturation. As the central connection point for WESNs, the E-Health Gateway has been developed in this research that was used to manage WESNs and transmission health data. Furthermore, a simplest Wireless E-Health System based on Wireless E-Health Sensor Node with Wi-Fi module has been proposed that can provide more portable E-Health care services. Moreover, Wireless E-Health System based on smart phone has been present, and two corresponding Smart Phone Apps, Blood Pressure Measurement App and SpO2 Measurement App have been developed. This proposed Wireless E-Health System has achieved the proposed aim that can provide patients with real-time monitoring and efficient health data management.

Cloud E-Health Management System has been presented to store and manage users' measurement data. Users can upload their measurement data through their E-Health System based on Wireless Sensor Network, which has been discussed in previous chapter.

In the Cloud E-Health Management System, different user groups, such as patients, medical institutions staff and system administrator can be supported. Each group has their own group policy. The patients can be allowed to register, access to system, manage their own information, upload health data, read and write feedbacks. The medical staff can access to system and process patients' medical records, summarize and send the results to patients. The administrator of the Cloud system has the highest rights; who can access to system, resolve system errors, modify or delete users and their records, etc.

Furthermore, with the patient number rapid the Cloud E-Health Management System can get unlimited computing and storage space from Amazon Web Services as very low price that is very suitable for model E-Health Management System.

Obviously, the Cloud E-Health Management System needs a security environment to keep user's records safe. Therefore, secure mechanisms are vital to E-Health System and will be further investigate in next chapter.

Chapter 4 Security Challenge of the Wireless E-Health System

4.1 Introduction

With the development of wireless technology, the Wireless E-Health System is entering into the public life [27]. Wearable, wireless and reliable sensors have been universally used in the Wireless E-Health System, which used to measure human body's physical indicators, record, and upload to personal health care centers by wireless transmission [50-53]. With the large-scale deployment of the wireless E-Health System, the security of the wireless system has become a challenge issue. Furthermore, personal health record is private and important, poor security of the open specification wireless transmission makes it very easy to be eavesdropped, destroyed or lost, which leading to many issues, such as the leaking of user information [54-57]. Consequently, a secure and safe wireless transmission is critical for the Wireless E-Health System.

Encryption is one of the technologies, which can be used to combat the security issue of wireless transmission. Currently, the commonly used encryption algorithms are DES (data encryption standard), Triple DES (triple data encryption algorithm), AES (advanced encryption standard)[18]. Due to the resources limitation of Wireless E-Health System, such as less computing ability, lower memory and battery capacity, it requires encryption techniques to be simple and lesser power consumption. In addition, the encryption algorithm should be able to deploy either in software or hardware for the embedded system. However, the above mentioned encryption algorithms require large amount of memory and powerful computing ability, therefore, they are not quite fitted to Wireless E-Health System based on Wireless Sensor Network.

The RC5 algorithm, introduced by Rivest [58], is a fast, simple, lesser memory required encryption method and suitable to be deployed in hardware or software in general. Hence, RC5 cryptosystem is quite suitable for Wireless E-Health System. The RC5 cryptographic algorithm has been used in Wireless Sensor Network [59-62]. Moreover, RC5 algorithm can be adjusted to trade-off security strength with power consumption and computational overhead. Kaliski, B. S., & Yin, Y. L [63] have proposed that RC5 cryptosystem, with 12 rounds and 64-bit block size, gives roughly the same security as the DES. Another enhanced RC5 algorithm with a changeable number of rounds to increase the randomness of cipher data has been proposed in [64], where the round is a random number that is computed by key and LFSR (Linear Feedback Shift Register) operation. However, this modification cannot control extreme situations, for instance, if the random round is too small then the cipher data is not secure enough, or there is a system overload if the random number is too large.

Furthermore, it is shown that RC5 encryption method within fewer rounds is not secure enough to defend against differential and linear cryptanalysis [65], which can be cracked as the RC5 algorithm is simply depending on the data rotation that lack of randomness. To overcome this disadvantage of RC5, a modified RC5 based on chaotic algorithm has been proposed in paper [66]. The modified RC5 algorithm uses Skew Tent Mapping to initial sub-key and mix with user’s secure key to enhance the secure key for encryption and decryption. With the help of Skew Tent Mapping, the randomness of the secure key of modified RC5 cryptosystem has been significantly increased. Moreover, Cai ke, et al. [60] propose another modified RC5 algorithm based on Chaos system. Cai’s algorithm uses integer chaotic map to add linear congruent generators to extend key space and combined with cipher text feedback mode, which achieved high performance. Since this modified RC5 has combined chaos system with linear congruent generator that make the algorithm more complex which not suitable for Wireless E-Health System. However, above two modified RC5 algorithms with Chaos system has significantly improve their randomness of cipher text and increase the system security. Furthermore, other Chaos based algorithms have been proposed due to its good randomness, unpredictability and sensitivity with initial value features [67] that can be used to enhance standard RC5 algorithm. Therefore, this research work further investigates on enhancing the RC5 encryption algorithm based on Chaos algorithm for Wireless E-Health System. This chapter has brief introduce the standard RC5 algorithm, and then two proposed enhanced RC5 algorithms based on Chaos system has been presented in the following two chapters.

4.2 RC5 Algorithm

The RC5 algorithm is a block and parameterized symmetric cipher [58]. This algorithm has a word-oriented rotation, $\mathrm{RC5-w/r/b}$, which ‘w’ is a variable word (block) size and the allowable word size are 32, 64 or 128 bits. ‘r’ is a variable amount of rounds and the range being from 0 to 255. ‘b’ is a variable key length and the length of the secret key, is between 0 and 255 bits.

The RC5 algorithm consists of three components: key expansion algorithm, encryption algorithm and decryption algorithm.

4.2.1 Key Expansion Algorithm

During the key expansion algorithm, the sub-keys ($S[n]$) will be generated by the expanded customer secret key ($K[n]$). There will be two “magic constants” and three steps to expand key: initialization, conversion and mix.

The two “magic constants” are two word-sized binary constants P_w and Q_w that defined for arbitrary w as equation (2).

$$\mathrm{P}_w = \mathrm{Odd}[(e-2)\ \times 2^w]$$
$$\mathrm{Q}_w = \mathrm{Odd}[(\emptyset-1)\ - 2^w]$$

(2) **[58]**

where e is Euler's number, which is the base of natural logarithms (approximately 2.7183); ϕ is Golden ratio (approximately 1.6180).

The hexadecimal format of P_w and Q_w are listed in Table 4-1 **[58]**.

Table 4-1 Hexadecimal Format of P and Q for RC5 Key Extension

w (word size)	16-bit	32-bit	64-bit
P_w	0xB7E1	0xB3E15163	0xB7E151628AED2A6B
Q_w	0x9E37	0x9E3779B9	0x9E3779B97F4A7C15

The first step is to convert user secret key from bytes to words format. As shown in the following pseudo code, the user secret key $K[0 \ldots b-1]$ will be copied into an array $L[0 \ldots c-1]\ of\ c = [b/u]$ words, where $u = w/8$ is the number of $bytes/word$. Key bytes of K will be filled up into L in low-order byte to high-order byte. Zero will be filled up in any unfilled byte position of L. Second procedure is to initializing the array S that key-independent pseudo-random bit pattern, using an arithmetic progression modulo 2^w determined by the "magic constants" P_w and Q_w [58].

```
$c = [max(b, 1)/u]$;
**for** $i = b - 1$ **downto** 0 **do**
        $L[i/u] = (L[i/u] \lll 8) + K[i]$;
```

[56]

```
$S[0] = P_w;$
$t = 2r + 2;$
for $i = 1$ to $t - 1$ do
    $S[i] = S[i-1] + Q_w;$
```

[56]

The third step is to mix the user’s secret key in three passes over the arrays S and L. The mathematical form of the mixing process is shown in following pseudo codes.

```
$i = j = 0;$
$X = Y = 0;$
do $3 * max\ (t, c)$ times
    $X = S\ [i] = (\ S[i] + X + Y)\ \rho \lll 3;$
    $Y = L\ [j] = (L[j] + X + Y\ ) \lll (X + Y)\ ;$
    $i = (i + 1)\ mod\ (t)\ ;$
    $j = (j + 1)\ mod\ (c)\ ;$
```

[56]

Finally, the Sub-Key array is generated and the length of the sub-key is $2(r + 1)$. Since the RC5 is the symmetric encryption algorithm, the sub-key will be used for both encryption and decryption

4.2.2 Encryption Algorithm

There are three basic operations used in the RC5 encryption process, as shown in the following list:

- Addition/subtraction of words modulo 2^w (w is the word size), denoted as $+/-$;
- Bit-wise exclusive-OR of words, XOR, denoted by $\oplus$;

- Rotation operation used to left-shift or right-shift the word by some bits. For example, word x left shift y bits can be denoted as $x \lll y$. The inverse operation is word *x* right shift *y* bits marked as $x \ggg y$.

During the encryption process, a *2w* bits plain text block will be entered. The iterative round number is r and the sub-key is $S\ [2r\ +\ 2]$. The output will be the same size of 2*w-bit* data. Assume the input plain text is stored in two *w*–bit registers A and B, the encryption process is shown as the follow pseudocode code:

```
$Input(A, B)$
$A\ = A\ + S[0]$
$B\ = B\ + S[1]$                                          [56]
For $i\ =\ 1$ to $r$ do
    $A_{i+1} =\ ((A_i \oplus B_i) \lll B_i)\ +\ S[2i]$
    $B_{i+1}\ =\ ((B_i \oplus A_{i+1}) \lll A_{i+1})\ +\ S[2i\ +\ 1]$
$Output\ (A_{i+1}, B_{i+1})$
```

At the beginning, A and B will be encoded by $S[0]$ and $S[1]$. Then in the first round encryption of *r* round, A_i will do XOR with B_i, the output will be left shift B_i bits and be encoded by $S[2i]$. This process also called half round encryption and the output marked as A_{i+1}. In another half round encryption, B_i will do XOR with A_{i+1}, the output will be left shift A_{i+1} bits and be encoded by $S[2i + 1]$, the output cipher is B_{i+1}. After r round encryption, the output A_{i+1} and B_{i+1} are the two *w-bit* cipher text. The flow chart of the RC5 encryption procedure is shown in Figure 4-1.

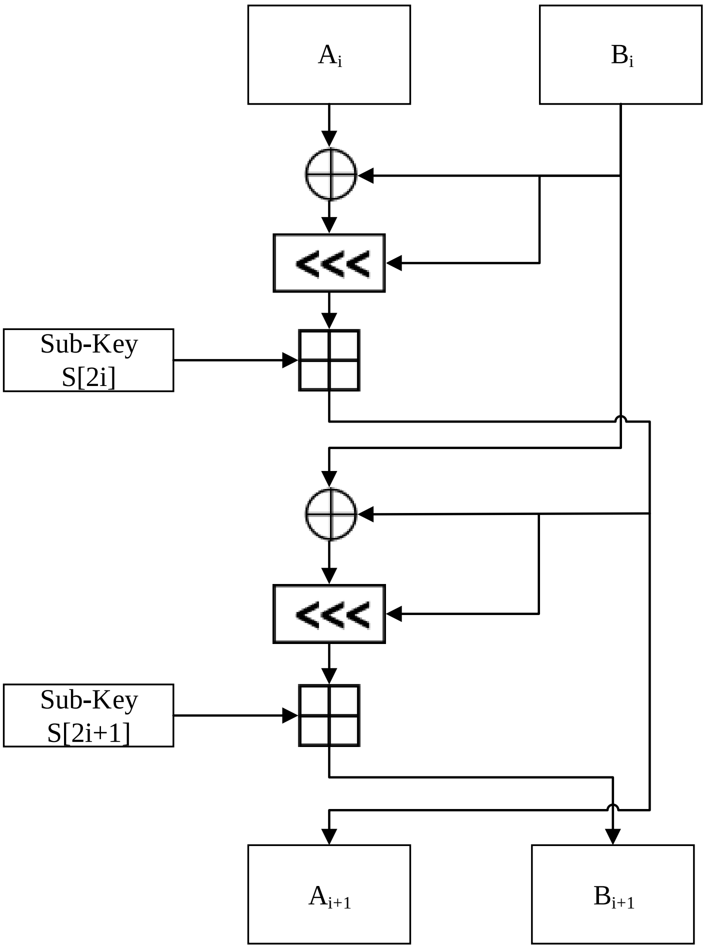


Figure 4-1 Flow chart of RC5 encryption

4.2.3 Decryption Algorithm

The decryption procedure of the RC5 is the reverse of encryption that is shown in following pseudo-code:

```
$Input(A, B)$
$for\ i\ = \boldsymbol{r}\ down\ to\ \mathbf{1}\ do$
    $B_{i-1}\ =\ \big((B_i - S[2i\ +\ 1]) \ggg A_i\big) \oplus A_i$
    $A_{i-1}\ =\ \big((A_i\ -\ S[2i])\big) \ggg B_{i-1}\ ) \oplus B_{i-1}$
$A\ =\ A\ -\ S[0]$
$B\ =\ B\ -\ S[1]$
$Output(A, B)$
```

[56]

where the operation of " $\ggg$ " is used for right shift and "is operation is used for the subtraction of the word modulo 2^w.

Clearly, the standard RC5 heavy depends on data-dependent rotations. Furthermore, the encryption and decryption operation are simple and low memory required. Therefore, RC5 is a very good candidate for Wireless E-Health application.

4.3 Conclusion

This chapter reviews the security challenge in Wireless E-Health System, as personal information is private that need secure transmission in wireless network. Encryption is a key technique to tackle the security issue. A lot of existing encryption technologies, such as DES, AES which are widely used in wireless communication system, but they are not suitable for Wireless E-Health System. Due to the limitation of Wireless E-Health System, such as lower computing capability, less memory and less battery capacity, it eliminates the deployment of complex encryption algorithms. RC5 is a simple, efficient and secure encryption algorithm, which can be deployed in both hardware and software. Therefore, this research work will focus on RC5 algorithm to make it more security and suitable for Wireless E-Health System. A brief review of standard RC5 algorithm has presented in this chapter. Recently, Chaos system has received more attention in encryption application. A brief review of Chaos theory has been presented in this chapter. Since the features: like non-periodic, indeterministic, sensitive with the initial parameters, of Chaos system makes them a good candidate for application.

Chapter 5 Enhanced RC5 Cryptographic Algorithm with 1-D Logistic Map

As discussed in the previous chapter, RC5 algorithm is a simple and secure encryption algorithm with variable parameters that can be easily deployed in a wireless sensor network. Since the Logistic map algorithm is one of the chaotic algorithms that are sensitive to initial values that a huge difference will be generated after long time computing with minor difference of initial value. In addition, the logistic map has a long-term unpredictability feature that can generate a large number of non-related, noise-like chaotic sequences. These chaotic sequences are hard to be reconstructed and forecast, which makes them very attractive for security applications. In this chapter an enhanced RC5 cryptographic with 1-D Logistic map has been proposed that concentrates on modifying the RC5 sub-key generation procedure, to ensure the security key is difficult to track and the cipher is hard to crack. The innovation of this research work lies in the fact that the sub-key will be encrypted by simple 1-D logistic mapping; furthermore, the 1-D Logistic map algorithm will be controlled by a cipher feedback model that can further increase the randomness in order to improve security.

5.1 1-D Logistic Map

1-D Logistic map is a simple one-dimensional discrete chaotic model, as described in equation (3) [68].

$$x_{n+1} = \lambda x_n(1 - x_n), \lambda \in [0,4], x_n \in [0,1] \qquad (3)$$

where, x_n is the system output at time n and λ is a parameter that in the range $[0,4]$.

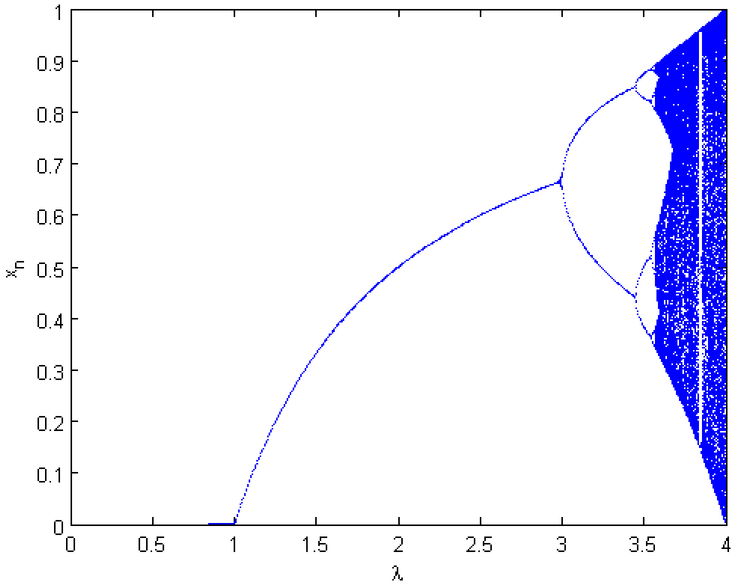


Figure 5-1 The bifurcation diagram of the Logistic mapping in $\boldsymbol{\lambda} \in [\mathbf{0}, \mathbf{4}]$

The bifurcations diagram of Logistic map with $\lambda \in [0,4]$ was shown in Figure 5-1. Appendix VI shows the code for plot the bifurcation of 1-D Logistic Map. As shown in the Figure 5-1, in the range of $\lambda \in [0,1]$, the system is very close to zero. The system enters into a stable increase region when $\lambda \in [1, 3]$, which only one exact value for one λ. However, after $\lambda > 3.0$, the system starts bifurcation. For example, when $3.0 < \lambda < 3.25$ the system has forked into two paths; and when $3.4 < \lambda < 3.55$, the system has forked into four paths, then the system forked into eight paths when $3.55 < \lambda < 3.57$ as shown in Figure 5-2.

The system reaches the period-doubling bifurcation after $\lambda > 3.57$ and the system can get any of the value. This is known as the chaotic status, which can't be predicted. As shown in Figure 5-2, the system back into 3 paths when the parameter λ near 3.82, however, in this range the system still unpredicted that can be regard as random. And then as the λ forward, the system restart bifurcates and chaos in the range $3.88 < \lambda < 4.0$. This feature can be used to randomize sub-key in each round of encryption.

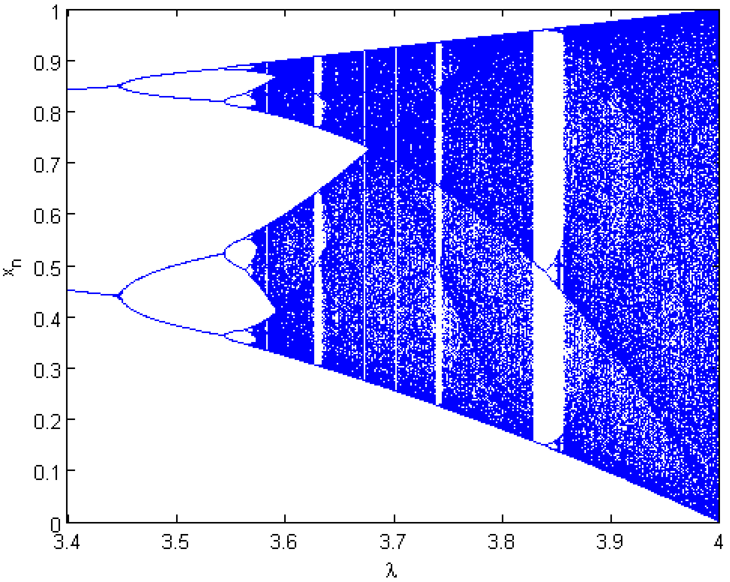


Figure 5-2 The bifurcation diagram of the Logistic mapping in $\boldsymbol{\lambda} \in [\mathbf{3.4}, \mathbf{4}]$

5.2 Proposed Enhanced RC5 Cryptographic with 1-D Logistic Map

RC5 is an attractive encryption algorithm for Wireless Sensor Network applications due to its simple and efficient implementation.1-D Logistic Map algorithm is a simple algorithm as well, and it has randomness features that can improve RC5 security level. In RC5 algorithm, the sub-key will be used in both the encryption and decryption process. Hence, security of sub-key is of great importance for RC5 algorithm. In this section, a randomized sub-key generation scheme in each round of encryption process is proposed by using 1-D Logistic Map to increase the unpredictability of standard RC5 algorithm. Meanwhile, a feedback mechanism was introduced in Logistic Map to control the initial value by using the cipher feedback text, which will further increase the randomness and security of RC5 algorithm. Figure 5-3 illustrates the structure of the proposed enhanced RC5 algorithm with 1-D Logistic map. The algorithm can be divided into three parts: sub-key generation using 1-D Logistic map, proposed encryption process and proposed decryption process, as described in the following sections.

The proposed RC5 algorithm can be denoted as RC5C-w/r/b/ λ, where *w*, *r* and *b* have the same meaning as in standard RC5 [58] and λ is a parameter of chaotic model that is in the range of [3.57, 4] [68]. Proposed enhanced RC5 algorithm with 1-D Logistic Map code has been show in the Appendix IV.

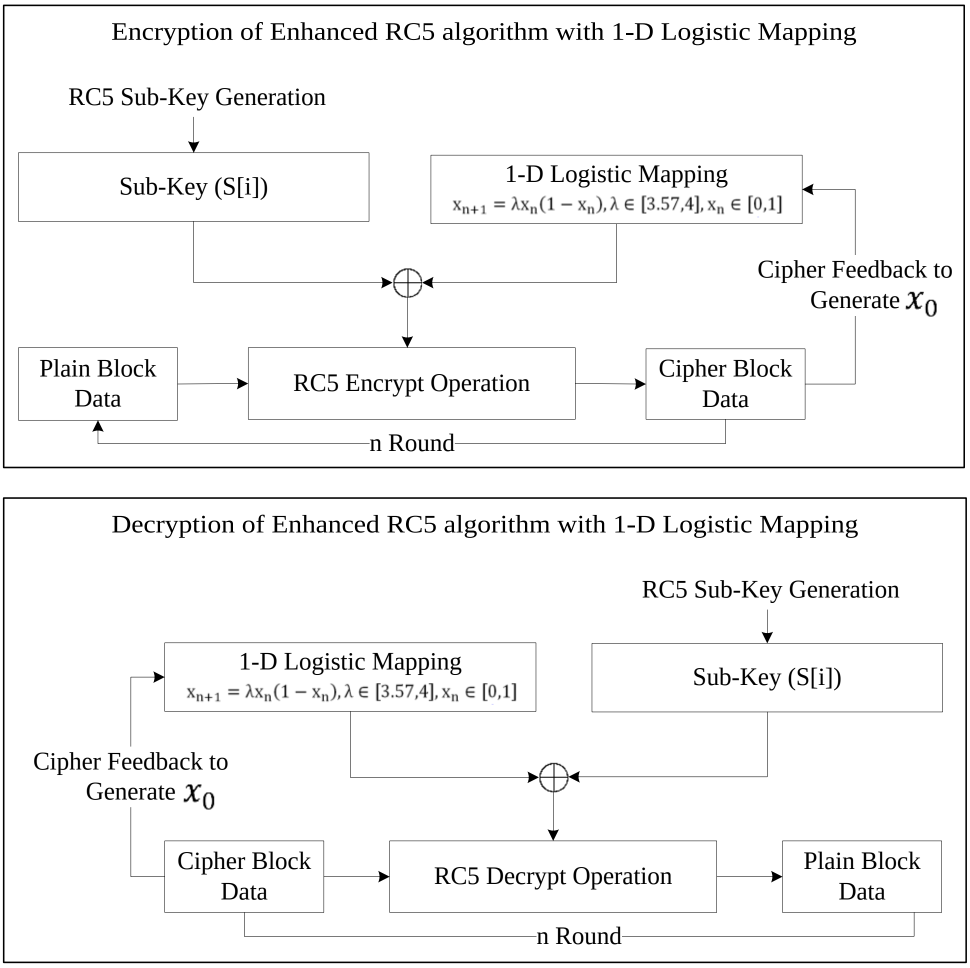

Encryption of Enhanced RC5 algorithm with 1-D Logistic Mapping
RC5 Sub-Key Generation
Sub-Key (S[i])
1-D Logistic Mapping
$x_{n+1} = \lambda x_n(1 - x_n), \lambda \in [3.57,4], x_n \in [0,1]$
Cipher Feedback to Generate x_0
Plain Block Data
RC5 Encrypt Operation
Cipher Block Data
n Round
Decryption of Enhanced RC5 algorithm with 1-D Logistic Mapping
RC5 Sub-Key Generation
1-D Logistic Mapping
$x_{n+1} = \lambda x_n(1 - x_n), \lambda \in [3.57,4], x_n \in [0,1]$
Sub-Key (S[i])
Cipher Feedback to Generate x_0
Cipher Block Data
RC5 Decrypt Operation
Plain Block Data
n Round

Figure 5-3 Structure of Enhanced RC5 algorithm with 1D Logistic Mapping

5.2.1 The Sub-Key Generation using 1-D Logistic Map

As shown in Figure 5-3, the initial sub-key will be taken from standard RC5 sub-key generation process and will be modified by 1D Logistic mapping to improve the randomness and sensitivity.

In each round of encryption, the sub-key $S[i]$ will be mapped by the 1-D Logistic Map with the initial value x_0. At the first half round of the proposed RC5 encryption process, the x_0 will be calculated from the second input block Bi that divided by the maximum 32-bit unsigned integer. In the next half round of the new RC5 encryption, the next x_0 will be provided by the feedback calculated from the last half round cipher block data $A_i^{'}$. After r rounds, the sub-key $S[i]$ will be completely changed from the initial sub-key generated by standard RC5 sub-key extension process. The new sub-key $S[i]$ will be used for next data encryption. Note that, in the new algorithm, the sub-key will be randomized in each round; therefore, the security is improved.

Since the decryption process is the inverse process of encryption, the sub-key generation for decryption is the inverse process of encryption sub-key generation. At the beginning of decryption, the two cipher blocks will be input and marked as $\left(A_i^{'}\ ,\ B_i^{'}\right)$. The sub-key $S[2r+1]$ for the first half round decryption can be computed by $A_i^{'}$ as it is generated by the $A_i^{'}$ feedback model during the encryption process. After the first half round decryption, the cipher block $B_i^{'}$ can be decoded to the middle cipher B_i. The middle cipher B_i is used to generate the initial value x_0 for $A_i^{'}$ decryption. In the forthcoming decryption round, the initial value of 1-D Logistic Map x_0 will be computed in the similar way.

5.2.2 Proposed Encryption Process

From Figure 5-3, in each round of encryption, the sub-key will be enhanced by 1-D Logistic Map and used as the security key during the encryption process. The encryption flow chart of the proposed RC5 algorithm is illustrated in Figure 5-4.

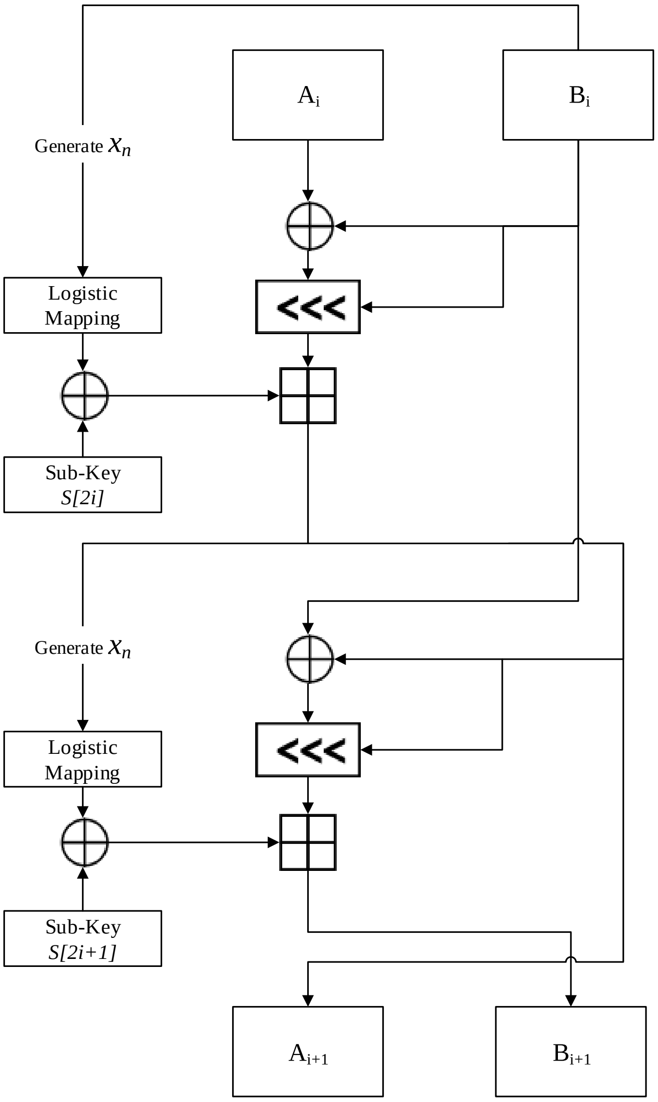


Figure 5-4 The Proposed RC5 Encryption Flow Chart

Assume a 64-bit (32 bitsgure $6 - 3$, plain data was taken into register Ai and Bi. With thain, Bi is used to compute the initial value of x_0 by dividing Bi with the largest 32-bit unsigned integer, as shown in Figure 5-4. There are two half rounds of the Feistel encryption processes. In the first half encryption, block Ai will do XOR operation between Ai and Bi, and left shifts Bi bits. Then the cipher block $A_{i+1}^{'}$ will add the enhanced sub-key S[2i]. In the second half Feistel encryption, $A_{i+1}^{'}$ will be used to generate the initial value of Logistic Map and encrypted using the sub-key S[2i+1]. The Bi will be mixed with $A_{i+1}^{'}$ by the XOR operation and left shift $A_{i+1}^{'}$

bits. Then the output will be added with the enhanced sub-key S[2i+1] to generate the B'_{i+1}. One encryption round will be completed after two half Feistel encryption processes. The entire encryption process will repeat these operations r times. The encryption pseudo code of proposed RC5 with 1-D Logistic Map is shown in following text box.

```
$x_0 = B_0 \ mod \ Max(uint)$
$S[0] = S[0] \oplus Chaos(x_0)$
$A_1 = A_0 + S[0]$
$x_0 = A_0 \ mod \ Max(uint)$
$S[1] = S[1] \oplus Chaos(x_0)$
$B_1 = B_0 + S[1]$
$For \ i = 1 \ to \ r \ do$ {
    $x_0 = B_i \ mod \ Max(uint)$
    $S_{2i} = S_{2i} \oplus Chaos(x_0)$
    $A_{i+1} = ROL(A_i \oplus B_i, B_i) + S[2i];$
    $x_0 = A_{i+1} \ mod \ Max(uint)$
    $S_{2i+1} = S_{2i+1} \oplus Chaos(x_0)$
    $B_{i+1} = ROL(A_{i+1} \oplus B_i, B_i) + S[2i+1];$
}
* $Chaos(x_0)$ is the Logistic Map Function.
* ROL means Rotate of Left
```

Note that the original sub-key S[n] will be generated by the RC5 sub-key extension process. As shown in the structure of proposed RC5 algorithm in Figure 5-3, the sub-key will be mapped by the 1-D logistic map and will use the cipher feedback to generate initial x_0 for the 1-D logistic map which makes cryptanalysis more difficult to guess. $Max(uint)$ is the maximum uint value, for example, $Max(uint) =$ 0xFFFFFFFF when $w = 32 - bit$.

5.2.3 Proposed Decryption Process

As mentioned before, decryption process is the inverse process of encryption, and again, in each round of decryption the sub-key will be changed by 1-D Logistic Map sequence. The decryption flow chart of the proposed RC5 algorithm is illustrated in Figure 5-5.

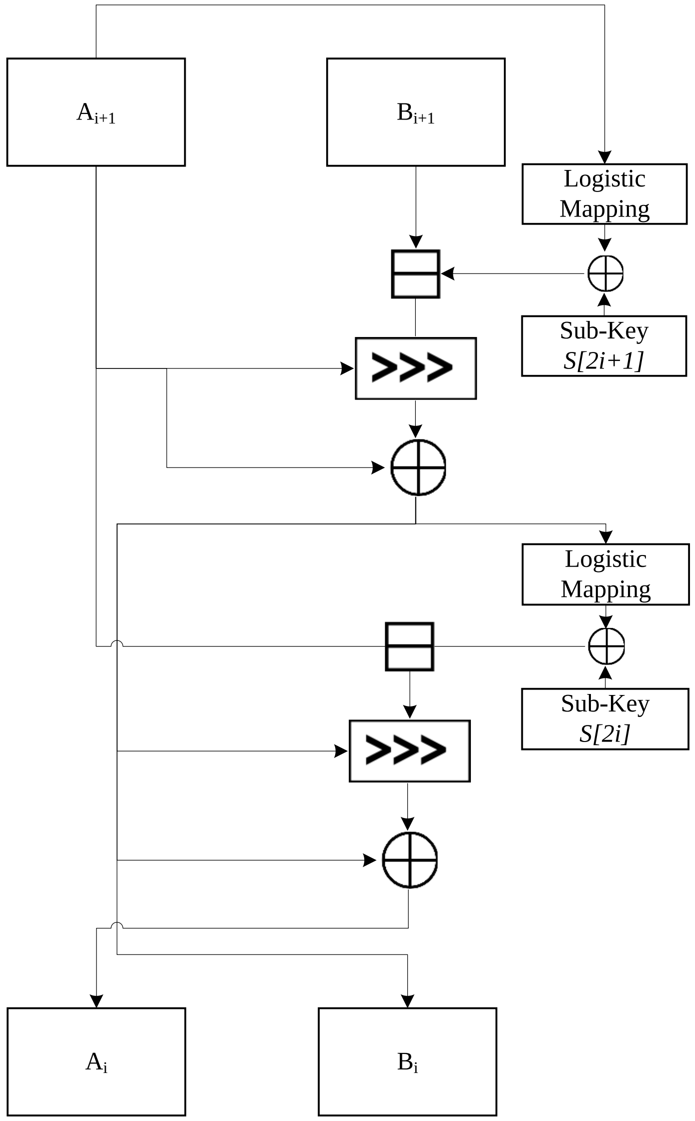


Figure 5-5 The Proposed RC5 Decryption Flow Chart

Ai is used to compute the initial value of x_{2i+1} by dividing Ai+1 with the largest 32-bit unsigned integer. Similar to the encryption, there are two half rounds of Feistel decryption process. In the first half Feistel decryption process, the cipher block Bi+1 needs to subtract the enhanced sub-key S[2i+1] that is computed from Ai+1. Then the output will be right shift Ai+1 bits. Finally, the cipher block B_i' will be generated by XOR operation between the last output and Ai+1. In the second half decryption, B_i' will be used to generate the initial value of Logistic Map and encrypt the sub-key S[2i]. The Ai will be subtracted from security key S[2i], then right shift B_i' bits. The output will be XORed with B_i' to generate A_i' . One round will be finished after two half Feistel decryption processes and the entire decryption process will repeat these operations r times. The encryption pseudo code of proposed RC5 with 1-D Logistic Map is shown in following text box.

```
$For\ i = r\ to\ 1\ do\{$
    $x_0 = A_i\ mod\ Max(uint)$
    $S_{2i+1} = S_{2i+1}\ \oplus Chaos(x_0)$
    $B_{i-1} = ROR(B_i - S[2i+1], A_i) \oplus A_i;$
    $x_0 = B_{i-1}\ mod\ Max(uint)$
    $S_{2i} = S_{2i}\ \oplus Chaos(x_0)$
    $A_{i-1} = ROR(A_i - S[2i], B_{i-1}) \oplus B_{i-1};$
$\}$
$x_0 = A_1\ mod\ Max(uint)$
$S[1] = S[1]\ \oplus Chaos(x_0)$
$B_0 =\ B_1 - S[1]$
$x_0 = B_0\ mod\ Max(uint)$
$S[0] = S[0] \oplus Chaos(x_0)$
$A_0 =\ A_1 - S[0]$
$*\ Chaos(x_0)\ is\ the\ Logistic\ Map\ Function.$
$*\ ROR\ means\ Rotate\ of\ Right$
```

Note that the original sub-key S[n] is generated by the RC5 sub-key extensions process.

5.3 Experiment Results and Security Analysis

Security analysis is a common method used to evaluate the performance of cryptosystem. The following security analysis and experiments have been performed to evaluate the satisfactory security improvement of the proposed RC5 algorithm by comparing with other encryption algorithms in terms of cipher image-only attack, known-plain image attack, chose-image attack, differential attack, and various brute-force attacks.

5.3.1 Key Space Analysis

In cryptology, the security key is critically important. Hence, the cipher key should be secure, sensitive and of large key space. The RC5 has a variable-length cryptographic key that ranges from 0 to 2040 bits. The exhaustive key search is one of the brute force methods that need 2^n times to find the correct key, where n is the key length in bits. For example, as suggested in [58], a 128-bit key size is strong enough to resist most cryptanalysis. Therefore, the key space of the RC5 is $2^{128} \approx 3.4082 \times 10^{38}$, which would need more than 1.07902831 * 10^{22} years [69] to analyses, by W. Stallings who has use a 20-year-old computer. If using the modern supercomputer Tianhe-2 (54.9 Peta floating-point operations per second (PFLOPS) at peak performance) [70] to brute force conduct, it would still take 1.9654*10^{14} years, as shown in the equation (4).

$$\frac{2^{128}}{54.9 \times 10^{15} \times 60 \times 60 \times 24 \times 365} \approx 1.9654 \times 10^{14} years \quad)$$

With the advance in computing ability, the standard RC5 key space will not satisfy the security requirement. The proposed enhanced RC5 algorithm with 1-D Logistic map has improved the security key scheme, which is changed in each round. Hence, the key space of the proposed RC5 is $2^{\mathrm{n}} \times \mathrm{r} \times \mathrm{length}$, where n is the key length in bits, r is the number of encryption rounds and length is the number of input blocks.

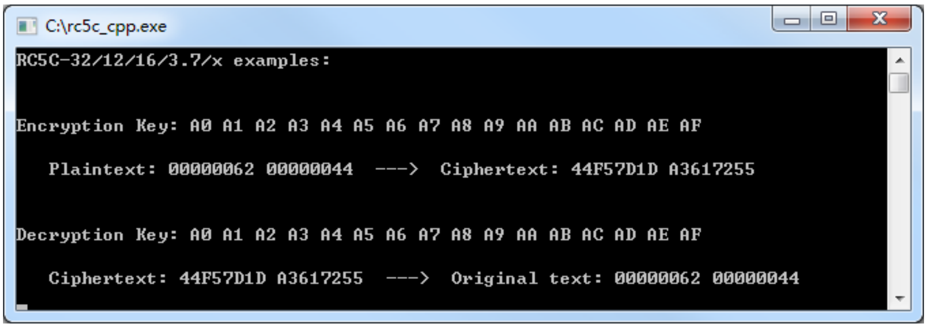
For example, with a key size of 128-bit, 12 rounds and a 128×128 image, then it will take $2^{128} \times 12 \times \frac{128\times128\times8}{32} \approx 1.6726 \times 10^{43}$ times computing, which is about $4.9\ \times 10^{4}$ times larger than standard RC5 key space and will increase as the plain text size increases.

5.3.2 Key Sensitive Test

Encryption technology requires the algorithm sensitive to the key, such that the correct information should not be decrypted by a tiny change in the security key [60]. This is particularly important for the Wireless E-Health System, where personal data will be transmitted cross the wireless network.

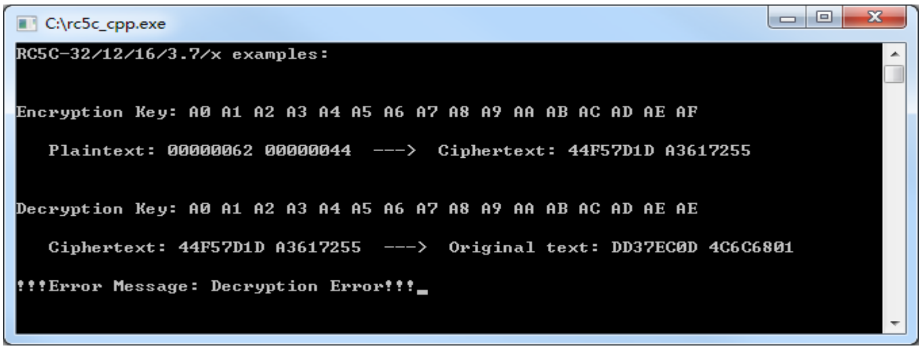
In the text message experiment, the proposed RC5 algorithm uses parameter RC5R-32/12/16/3.7 and the 128-bit security key: “0xA0, 0xA1, 0xA2, 0xA3, 0xA4, 0xA5, 0xA6, 0xA7, 0xA8, 0xA9, 0xAA, 0xAB, 0xAC, 0xAD, 0xAE, 0xAF”. The plain text is “98, 68”, which represents oxygen saturation is 98 percent and the pulse rate is 68 times per minute. The hexadecimal format of the plain text is “0x00000062 0x00000044”. After 12 rounds of the encryption, the cipher text is “44F57D1D A3617255”, which can only be decrypted by correct key, as shown in Figure 5-6.

Since the enhanced algorithm combined with the 1-D Logistic map algorithm and feedback mode, a tiny difference in the security key will cause a wrong decryption. For example, by using the very slightly changed key “0xA0, 0xA1, 0xA2, 0xA3, 0xA4, 0xA5, 0xA6, 0xA7, 0xA8, 0xA9, 0xAA, 0xAB, 0xAC, 0xAD, 0xAE, 0xAE” to decrypt the above cipher text, the recovered plain text became “DD37EC0D 4C6C6801”, which is not correct, as shown in Figure 5-7.



```
RC5C-32/12/16/3.7/x examples:

Encryption Key: A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF

   Plaintext: 00000062 00000044  ---> Ciphertext: 44F57D1D A3617255

Decryption Key: A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF

   Ciphertext: 44F57D1D A3617255  ---> Original text: 00000062 00000044
```

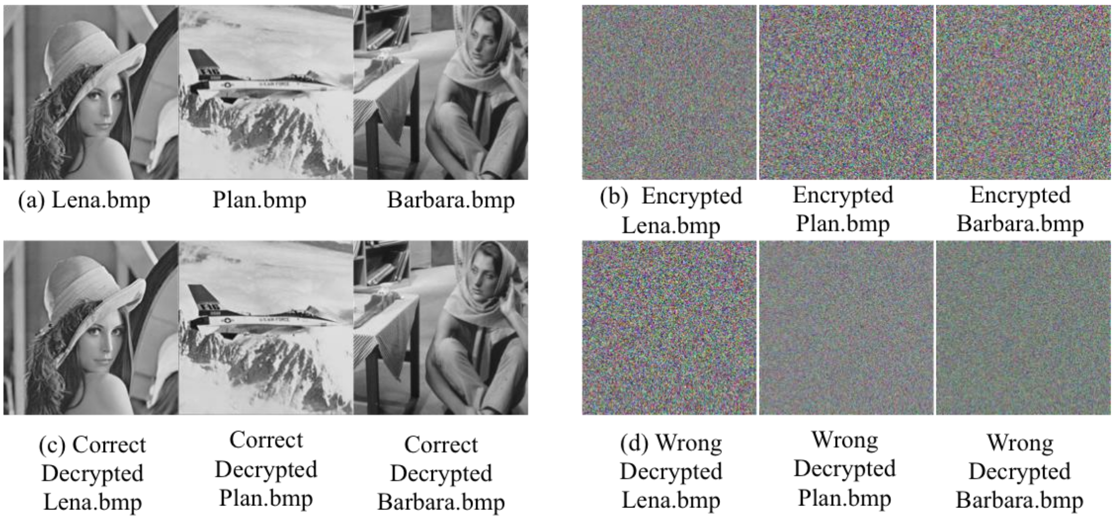
Figure 5-6 Hexadecimal Data Test of Enhancement RC5 algorithm



```
RC5C-32/12/16/3.7/x examples:

Encryption Key: A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF

   Plaintext: 00000062 00000044  ---> Ciphertext: 44F57D1D A3617255

Decryption Key: A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AE

   Ciphertext: 44F57D1D A3617255  ---> Original text: DD37EC0D 4C6C6801

!!!Error Message: Decryption Error!!!_
```

Figure 5-7 Hexadecimal Data Test of Enhancement RC5 algorithm with wrong decryption key

In the Wireless E-Health System, health data can be an image. In this image experiment, the enhanced RC5 algorithm will use the same parameters as the previous test, $RC5C - 32/12/16/3.7$ and the security key: “0xA0, 0xA1, 0xA2, 0xA3, 0xA4, 0xA5, 0xA6, 0xA7, 0xA8, 0xA9, 0xAA, 0xAB, 0xAC, 0xAD, 0xAE, 0xAF”. The experiment image is chosen the standard Lena image, Plan image and Barbara image, which is 512×512 grey scale bitmap is shown in Figure 5-8 (a). Figure 5-8 (b) is the cipher images after 12 rounds of the encryption, and the decryption images can be seen in Figure 5-8 (c), which have been decoded by correct key.



(a) Lena.bmp Plan.bmp Barbara.bmp

(b) Encrypted Lena.bmp Encrypted Plan.bmp Encrypted Barbara.bmp

(c) Correct Decrypted Lena.bmp Correct Decrypted Plan.bmp Correct Decrypted Barbara.bmp

(d) Wrong Decrypted Lena.bmp Wrong Decrypted Plan.bmp Wrong Decrypted Barbara.bmp

Figure 5-8 Images encrypted and decrypted by enhancement RC5 algorithm by correct key and wrong key.

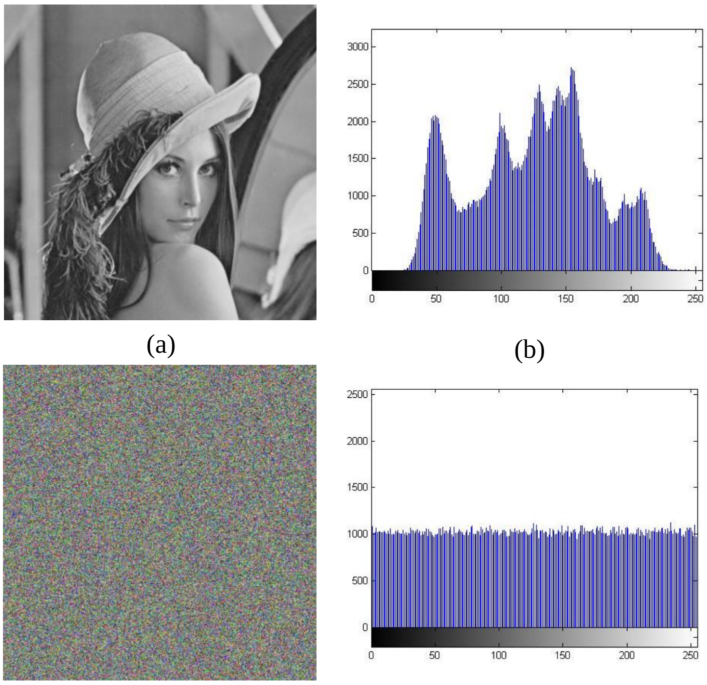
With a small change in the decryption key to “0xA0, 0xA1, 0xA2, 0xA3, 0xA4, 0xA5, 0xA6, 0xA7, 0xA8, 0xA9, 0xAA, 0xAB, 0xAC, 0xAD, 0xAE, 0xAE”, the decrypted images are shown in Figure 5-8 (d), which are noise-alike. As the text and image test results show, the proposed enhanced RC5 algorithm with 1-D Logistic map is very sensitive to the initial value and demonstrates a high level of security.

5.3.3 Statistical Analysis

To evaluate the capability of resistance to statistical attacks, the following tests have been carried out using the proposed RC5 algorithm.

a) Histograms Analysis

Image histogram is used to describe the image pixel distribution at each color intensity level [66]. A uniform histogram of cipher image should be generated from cryptosystem. Histogram analysis has been presented in Figure 5-9.



(a) (b)

(c) (d)

Figure 5-9 (a) Lena image, (b) Histogram for Lena image, (c) Lena Cipher image, (d) Histogram for Lena Cipher image

The two histograms (Figure 5-9 (b) and (c)) illustrating the original Lena image (Figure 5-9(a)) and the cipher image (Figure 5-9(c)) that using the proposed enhanced RC5. This experiment has been done in Matlab by the function of "imhist()". As Figure 5-9 shown, the original Lena image has the obvious feature in the histogram that almost all of the grey value is centered on the interval of [50,200]. After encryption using RC5R-32/12/16/3.7, the cipher image's grey value is random and its histogram uniformly distributed in the interval [0, 255], without any original image information. This result shows that the proposed algorithm has strong encryption and randomizing ability.

b) Correlation Analysis of Two Adjacent Pixels

Analysis of the correlation of two adjacent pixels is one of the useful evaluations of the encryption quality of an image cryptosystem [66]. The experiment plots the distribution of the horizontally, vertically and diagonally adjacent pixels in the plain image and its corresponding cipher image. The correlation coefficient between two adjacent pixels was defined and computed using the following equations (5) [66].

$$cov(x,y) = E\big(x - E(x)\big)\big(y - E(y)\big)$$
$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \qquad (5)\ \textbf{[66]}$$

where x and y are grey-scale values of two adjacent pixels in the image. In numerical computation, the following discrete version of equation (6) was used:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$$
$$\mathrm{D(x)} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i=1}}^{\mathrm{N}}(\mathrm{x_i} - \mathrm{E(x)})^2 \qquad (6)\ \textbf{[66]}$$
$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$$

The same parameters RC5R-32/12/16/3.7 are used as in histograms analysis and results of correlation coefficient of the adjacent pixels are listed in Table I. The experiment code has been list in Appendix VII. The horizontal, vertical and diagonal correlation distributions of plain image (left column) and cipher image (right column) are illustrated in Figure 5-10. The experiments used 1000 pairs of two horizontally adjacent pixels, two vertical adjacent pixels and two diagonal adjacent pixels in plain image and cipher image, respectively. The results of correlation analysis show that the proposed RC5 algorithm has strong capability to randomize the plain data.

Table 5-1 Correlation coefficient of the adjacent pixels results

Lena (512*512) grayscale bitmap	**Plain image**	**Ciphered image by RC5**	**Ciphered image using proposed enhanced RC5**
Horizontal	0.9710	0.0023	0.0015
Vertical	0.9855	0.0086	0.0015
Diagonal	0.9593	0.0048	0.0027

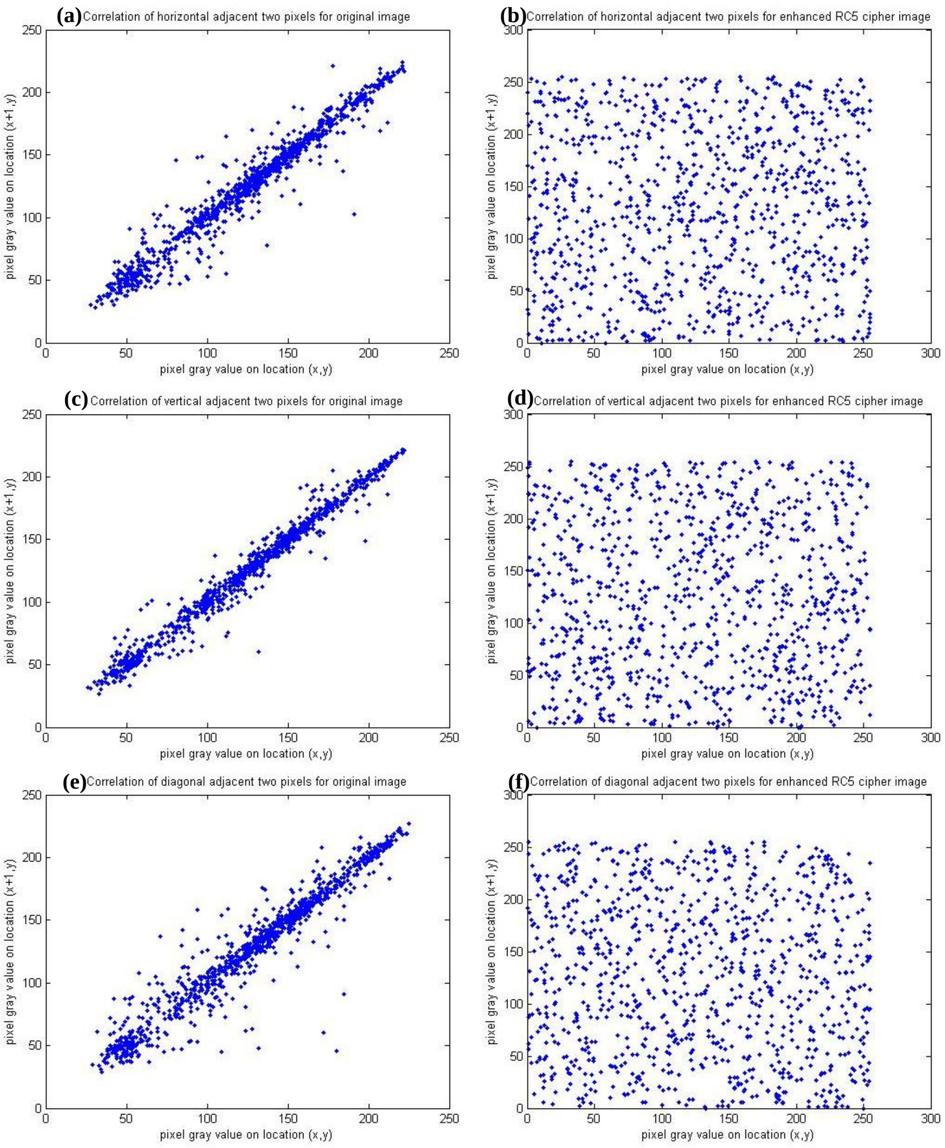


Figure 5-10 Correlation of two adjacent pixels for Lena image of size 512*512 (a) Horizontal direction of the plain image, (b) horizontal direction of the enhanced RC5 cipher image, (c) vertical direction of the plain image, (d) vertical direction of the enhanced RC5 cipher image, (e) diagonal direction of the plain test, (f) diagonal direction of the enhanced RC5.

c) Information Entropy Analysis

The amount of information can be described by entropy in information theory [71]. Entropy is defined in the below equation (7), and used to calculate the entropy H (m) of a source message m:

$$\mathrm{H(m)} = \sum_{i=0}^{\mathrm{M}-1} P(m_i) \log \frac{1}{P(m_i)} \; bits \qquad (7)\ \textbf{[69]}$$

where the total number of symbols M; the probability of occurrence of symbol m_i is represented as $P(m_i)$. Therefore, the entropy should be in bits. For example, the cipher message emits 2^8 symbols with equal probability, i.e. $m = \{m^1, m^2 \ldots m^{2^8}\}$, then the entropy is $H\ (m) = 8$. The cipher image from standard RC5, MRC5 and the proposed enhanced RC5 algorithm has been used to calculate the entropy and the results are listed in the table 5-2.

Table 5-2 Entropy test results

Encryption algorithm	**RC5**	**MRC5[66]**	**Enhanced RC5**
Entropy value	7.9730	7.9964	7.9993

The results in above table indicate that all three encryption algorithms are close to the theoretical value of 8. However, the test shows that the proposed enhanced RC5 gives best performance.

5.3.4 Differential Analysis

The concept of differential cryptanalysis is to analyze a pair of specially selected plain texts P and P*, which, with a certain difference $\mathrm{P}' = \mathrm{P} - \mathrm{P}^*$, are encrypted to generate two cipher texts C and C*, which can create a certain difference $\mathrm{C}' = \mathrm{C} - \mathrm{C}^*$ as well. Then the $(\mathrm{P}', \mathrm{C}')$ is called ‘characteristics’ and can be used to perform the differential attack analysis [65, 72]. The differential attack analysis is an important indicator for evaluating performance of the encryption algorithm. For image encryption application, two quantity methods, the number of changing pixels rate (NPCR) and the unified averaged changed intensity (UACI), are commonly used to evaluate the ability of the differential attack[73]. The NPCR and UACI Matlab code has been list in the Appendix VIII.

a) Number of Pixel Change Rate (NPCR)

Two ciphered images are denoted as C1 and C2, whose corresponding plain images are the same size with only one pixel value difference. $\mathrm{C_1(i,j)}$ and $\mathrm{C_2(i,j)}$ represent the grey-scale value of the pixels at $\mathrm{(i,j)}$. $\mathrm{D(i,j)}$ is defined as a bipolar array that is determined by $\mathrm{C_1(i,j)}$ and $\mathrm{C_2(i,j)}$. If $\mathrm{C_1(i,j) = C_2(i,j)}$ then $\mathrm{D(i,j) = 1}$; otherwise, $\mathrm{D(i,j) = 0}$. The NPCR is defined in equation (8) [66]:

$$NPCR = \frac{\sum_{\mathrm{i,j}} D(i,j)}{W \times H} \times 100\%$$

$$D(i,j) \begin{cases} 0, if\ C_1(i,j) \neq C_2(i,j) \\ 1, if\ C_1(i,j) = C_2(i,j) \end{cases} \tag{8}$$

(8) **[66]**

where W and H are the width and height of cipher image $\mathrm{C_1}$and $\mathrm{C_2}$ respectively. For better algorithm, the NPCR value should be high.

b) Unified Average Changing Intensity (UACI)

The UACI is defined in equation (9):

$$UACI = \frac{1}{w \times H}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\%.$$

(9) **[66]**

It is used to calculate the average intensity of differences between two cipher images. From the definition, the larger UACI value, better performance of encryption.

The NPCR and UACI experiments have been conducted using a 256 grey-scale Lena image with one random pixel difference. The results are listed in Table 5-3 with comparison to AES. The results show the proposed RC5 algorithm is of the same performance as AES in terms of resistance to differential attack but with less complexity in computation.

Table 5-3 The NPCR and UACI result

	RC5 [66]	**MRC5 [66]**	**Proposed RC5 with 1-D Logistic Map algorithm**	**AES**
NPCR	98.8%	98.8%	99.42%	99.42%
UACI	31.2%	31.2%	33.40%	33.41%

5.3.5 Randomness Test Using NIST SP800-22 Test Suit

The Nation Institute of Standards and Technology (NIST) published SP800-22 standard to test randomness and pseudo random number in 2003[74]. Since then, the standard is used worldwide.

The proposed RC5 algorithm has been tested using NIST with the same parameters, RC5C-32/12/16/3.7. The same image Lena is used as plain image, and 16 bytes, 32 bytes and 64 bytes security key are used, respectively. The results of the test are shown in Table 5-4.

Table 5-4 SP800-22 Test Result of $\boldsymbol{RC5C-32/12/16/3.7}$

SP800-22 Test List	Standard RC5 algorithm	Enhanced RC5 with 1-D Logistic map algorithm	Pass (>0.01)
Frequency	0.293751	0.512780	Yes
BlockFrequency	0.307767	0.478357	Yes
CumulativeSums	0.355076	0.537743	Yes
Runs	0.476054	0.547679	Yes
LongestRun	0.590430	0.367200	Yes
Rank	0.515548	0.651417	Yes
FFT	0.377607	0.442761	Yes
NonOverlappingTemplate	0.499440	0.498469	Yes
OverlappingTemplate	0.477481	0.480824	Yes
ApproximateEntropy	0.440865	0.468313	Yes
Serial	0.552267	0.578896	Yes
LinearComplexity	0.594117	0.655971	Yes

As shown in the Table 5-4, the proposed new algorithm has passed the entire test as recommended by NIST, which indicate the algorithm produced strong randomness of the cipher text.

5.3.6 Balance Analysis

Another useful performance evaluation for a cryptosystem is that the encryption algorithm should produce good balance of “0” and “1” statistically in the cipher text. That means the cipher text should consist of a substantially equal length of “0” and “1”. A balance analysis of the proposed RC5 algorithm is performed using Lena image with RC5C-32/12/16/3.7. The balance test results between different lengths of cipher text have been shown in Table 5-5.

Table 5-5 Balance Analysis result

Enhanced RC5 cipher text length (n)	**1000**	**10000**	**100000**	**1000000**
0 (***K1***) (bits)	484	4903	50064	499551
1 (***K2***) (bits)	516	5097	49936	500449
Proportion: $\|(\boldsymbol{K1}-\boldsymbol{K2})/\boldsymbol{n}\|$	0.032	0.0194	0.00128	0.000898

From Table 6-5, different length of cipher text contains almost same number of “0” and “1” statistically. In addition, with the cipher text length increasing, the ratio between 0’s and 1’s is closer. This indicates that the cipher text has better balance of “0” and “1”.

5.4 Conclusion

A new enhanced RC5 algorithm using 1-D Logistic Map combined with cipher feedback technique to improve the key security for data transfer over Wireless Network has been proposed in this chapter. It is particular of interest for E-Health care application based on Wireless Sensor Network, where the computation resources are very limited. The statistical analysis and experimental results have shown that the proposed new RC5 algorithm has superior performance in terms of strong resistance to variable type of attacks. Moreover, the proposed algorithm possesses low computing capability, less memory and more security.

Chapter 6 Enhanced RC5 Cryptographic Algorithm with 2-D Logistic Map

The statistical complexity of 1-D logistic mapping will be decreased with the increase of control parameters [75], since the key sensitivity of a one-dimensional logistic map relies on a single parameter λ and an initial value of x_0. Another demerit of 1-D logistic mapping is that it is weak in resisting differential attacks, such as chosen-plain text attacks. A 2-D Chaotic map has proposed to improve the effectivity and security of 1-D Chaotic map [75]. 2-D Logistic Mapping has been used in AES key expansion algorithm [76], which reduces the dependence between sub-keys and makes the secure key more robust. Wu, al etc. has proposed a new encryption that used two-dimensional Logistic chaotic map to generate random and sensitive sequence to encrypt data [77]. This chapter introduces an enhanced RC5 cryptographic algorithm with 2-D Logistic map.

6.1 2-D Logistic Map

The 2-D logistic map inherited the advantages and improves on the insufficiency of the 1-D logistic map. The 2-D logistic map can be used to expand the key space, since the chaotic map sequence becomes two dimensional and more initial conditions become $(\mathrm{x}_0, \mathrm{y}_0)$ available. In addition, it provides more complex chaotic behaviors than the 1-D logistic map. The 2-D logistic map can be defined as Eq. (10) [77].

$$\begin{cases} x_{n+1} = \lambda(3y_n + 1)x_n(1 - x_n) \\ y_{n+1} = \lambda(3x_{n+1} + 1)y_n(1 - y_n) \end{cases} \qquad)) \qquad \textbf{[75]}$$

where λ is the system parameter, which controls the mapping evolution from one kind of dynamics to another.

2-D Logistic mapping has chaotic behaviors when $\lambda \in [1.11, 1.19]$. $(\mathrm{x_n}, \mathrm{y_n})$ is the pair-wise point at the n^th iteration. Figure 6-1 illustrates the scatter plot of 50,000 points from the trajectory of the 2-D logistic map using the parameter λ and the initial value $(x_0, y_0) at$ $(0.8909, 0.3342)$ [78]. The trajectory $(\mathrm{x_n}, \mathrm{y_n})$ of the 2-D logistic map has stochastic distribution characteristics.

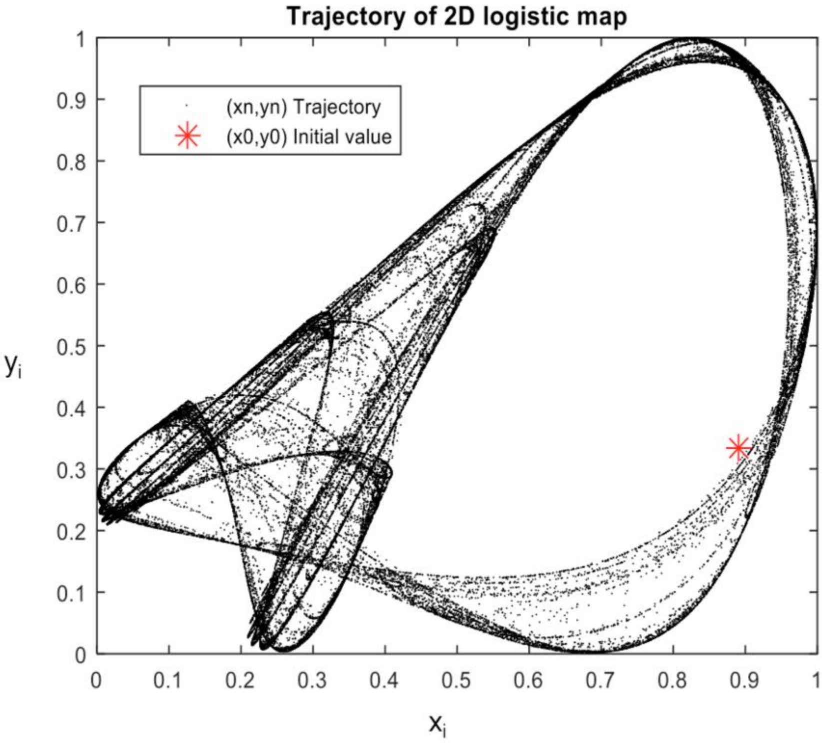


Figure 6-1 Trajectory of 2-D logistic map **[76]**

6.2 *Scheme of Enhanced RC5 Cryptographic Algorithm with 2-D Logistic Map*

Since the security key is the kernel of the encryption algorithm, the proposed RC5 algorithm with 2-D Logistic map is focused on modifying the original RC5 sub-key extension process by 2-D logistic mapping that increase the randomness and initial value sensitivity. Furthermore, the data rotation process of the original RC5 algorithm has been modified to improve the security.

The novel enhanced RC5 algorithm based on 2-D logistic map can be denoted as $RC5C-w/r/\lambda/(\mathrm{x}_0,\mathrm{y}_0)$. The w means the size of a word, such as 16-bit, 32-bit or 64-bit. r is the number of encryption or decryption rounds; λ is the 2-D logistic map control parameter in the interval of $[1.11, 1.19]$; and $(\mathrm{x}_0,\mathrm{y}_0)$ is the 2-D logistic map initial value, which is in the range of $\mathrm{x}_0 \in [0,1]\ and\ \mathrm{y}_0 \in [0,1]$. Figure 6-2 illustrates the structure of the proposed novel RC5 algorithm with a 2-D logistic map that consists of three processes: enhanced sub-key extension by 2-D Logistic map, modified encryption process and modified decryption process. The proposed algorithm code has been described in the Appendix V.

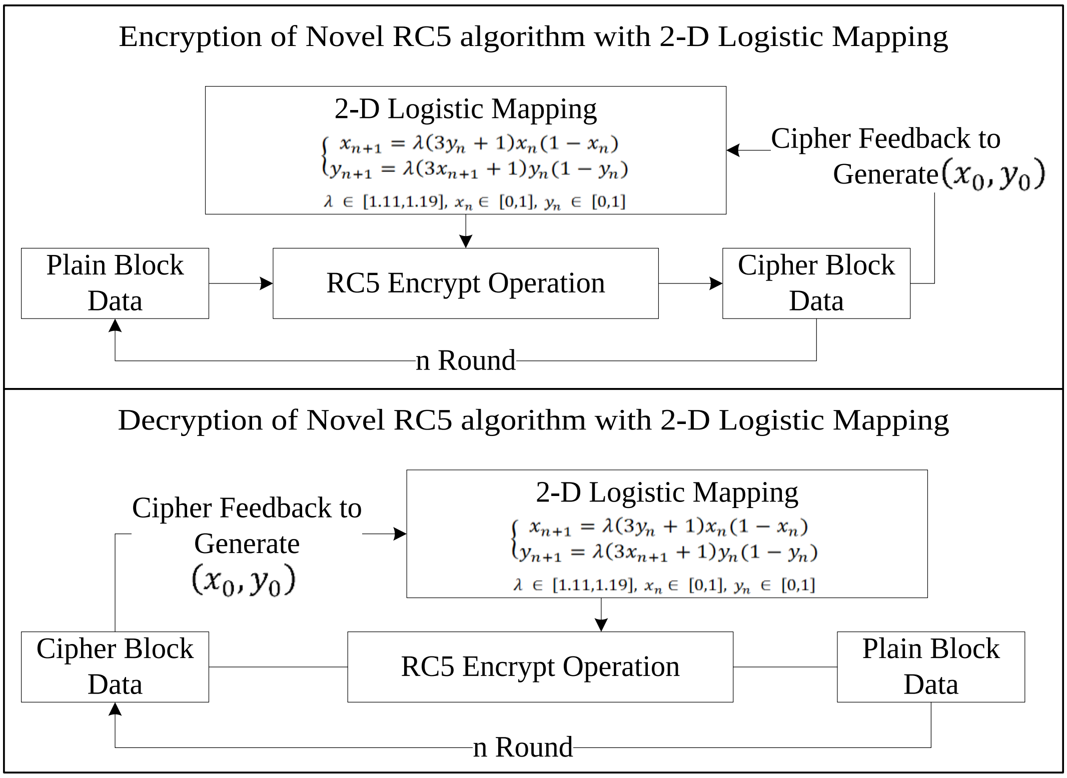

Encryption of Novel RC5 algorithm with 2-D Logistic Mapping
2-D Logistic Mapping
$\begin{cases} x_{n+1} = \lambda(3y_n + 1)x_n(1 - x_n) \\ y_{n+1} = \lambda(3x_{n+1} + 1)y_n(1 - y_n) \end{cases}$
$\lambda \in [1.11,1.19],\ x_n \in [0,1],\ y_n \in [0,1]$
Cipher Feedback to Generate(x_0, y_0)
Plain Block Data
RC5 Encrypt Operation
Cipher Block Data
n Round
Decryption of Novel RC5 algorithm with 2-D Logistic Mapping
Cipher Feedback to Generate (x_0, y_0)
2-D Logistic Mapping
$\begin{cases} x_{n+1} = \lambda(3y_n + 1)x_n(1 - x_n) \\ y_{n+1} = \lambda(3x_{n+1} + 1)y_n(1 - y_n) \end{cases}$
$\lambda \in [1.11,1.19],\ x_n \in [0,1],\ y_n \in [0,1]$
Cipher Block Data
RC5 Encrypt Operation
Plain Block Data
n Round

Figure 6-2 Enhanced RC5 algorithm with 2-D Logistic Map architecture

6.2.1 Enhanced Sub-key Extension by 2-D Logistic Map

In the enhanced sub-key extension stage, 2-D Logistic mapping has been used to generate the random chaotic sequence that is used as the sub-key. And the

λ and (x_0, y_0) are the security key of the proposed novel enhanced RC5 algorithm. As the sub-key length should be 2r+2, so the 2-D Logistic mapping will calculate r+1 times to generate the chaotic sequence $\{(x_0, y_0), (x_1, y_1) \cdots (x_r, y_r)\}$, to change to sub-key format, such as $\{x_0, y_0, x_1, y_1 \ldots x_r, y_r\}$.

The RC5 is a symmetric encryption algorithm that used the same security key for both encryption and decryption, therefore in the decryption process the decryption sub-key will be the same as the encryption process, which again uses λ and (x_0, y_0) as the initial value to generate the 2-D Logistic mapping sequence.

In this chapter, the 2-D Logistic map will use the chaos system control parameter λ=1.19 and initial value (x_0, y_0)at $(0.8909, 0.3342)$ for Eq. (11) that make the system in the chaos status and the mapping sequence will be random and unpredictable. After r+1 round iteration of 2-D logistic mapping, the generated sequence of $\{(x_0, y_0), (x_1, y_1) \cdots (x_r, y_r)\}$ will be used as the sub-key in the modified RC5 algorithm, denoted as sub-key S[n].

6.2.2 Modified Encryption Process

The encryption process of the novel RC5 algorithm with the 2-D Logistic map will use the sub-key from section 6.2.1 to encrypt plain block twice except for the first block. The flow chart of the novel RC5 algorithm with the 2-D Logistic map is shown in Figure 6-3. P_i and P_{i+1} are the two input blocks. At beginning, P_i is XOR with P_{i+1}, then left shift P_{i+1} bits, the output result will be added with sub-key *S[2r]* to product the cipher $P_i^{'}$ of the first half round of encryption. The P_{i+1} will do XOR operation with $P_i^{'}$ and left shift $P_i^{'}$ bits, then add sub-key *S[2r+1]* to get the second half round encryption cipher $P_{i+1}^{'}$. After r round operation, the final cipher blocks are $P_i^{'}$, $P_{i+1}^{'}$, and $P_i^{'}$ will be written into the cipher stream, but $P_{i+1}^{'}$ will be kept and used as part of the input plain block to do a second times encryption with the new input plain block. By such an approach, the following plain blocks will be encrypted, which better protects the cipher, with high sensitivity, unpredictability and security.

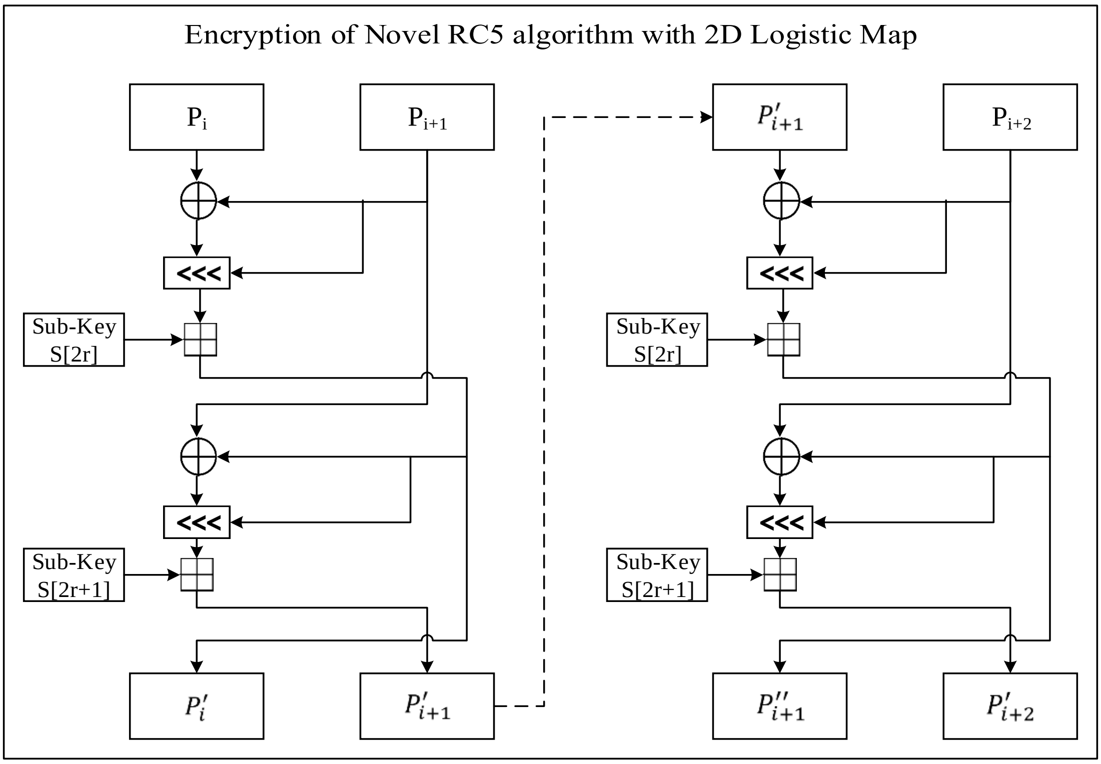


Figure 6-3 novel RC5 algorithm with 2-D Logistic map encryption flow chart

6.2.3 Modified Decryption Process

The decryption progress is the inverse operation of encryption. Since RC5 is a symmetric encryption algorithm, therefore the same sub-key will be used for both the encryption and the decryption process. $(\mathrm{x}_0, \mathrm{y}_0)$ $and\ \lambda$ will be used to compute 2-D Logistic mapping and generate the decryption sub-key. In the decryption process, the cipher stream will be read inversely, since all the cipher blocks (except the first and final block) have been encrypted twice with the previous and following block, therefor, decryption needs to start from the end of the cipher stream. Figure 6-4 describes the process of the proposed decryption process. The decryption sub-key will be generated from sub-key expansion by 2-D Logistic map. At the beginning of the decryption process, the last two cipher blocks C_i, C_{i-1} will be read into registers. C_i

subtracts the sub-key *S [2r+1]*, right shift C_{i-1} bits and do XOR operation with C_{i-1} . The output block $C_i^{'}$ is the first half round plain block. Then the C_{i-1} subtracts the sub-key *S[2r]*, right shift $C_i^{'}$ bits position and do XOR operation with $C_i^{'}$, getting the second half round plain block $C_{i-1}^{'}$. Then the $C_i^{'}$ and $C_{i-1}^{'}$ will do *r-1* round of decryption to recover the plain blocks. However, $C_{i-1}^{'}$ has been encrypted with C_{i-2} which needs decryption again to get the plain block as shown in Figure 6-4, where the $C_{i-1}^{'}$ will be used as input with C_{i-2} to do another r round of decryption.

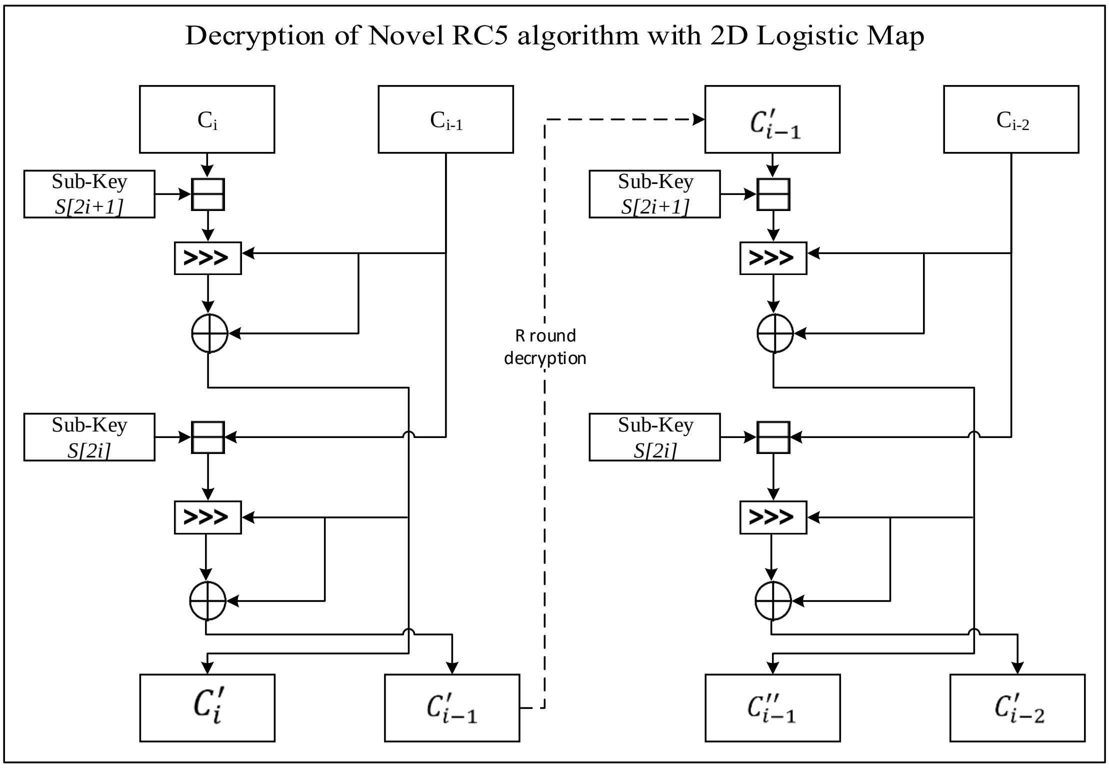


Figure 6-4 Flow chart of novel RC5 algorithm with 2-D Logistic map decryption

6.3 Experiment Results and Security Analysis

Security analysis is one of the best ways to evaluate the performance of a cryptosystem. The cryptosystem should provide the capabilities to resist all kinds of known attacks, for example, cipher image-only attack, known-plain image attack, chosen-image attack, differential attack, and various brute-force attacks. The proposed novel RC5 algorithm with 2-D Logistic map demonstrates a satisfactory security improvement compared with the standard RC5 as shown in the following security analysis.

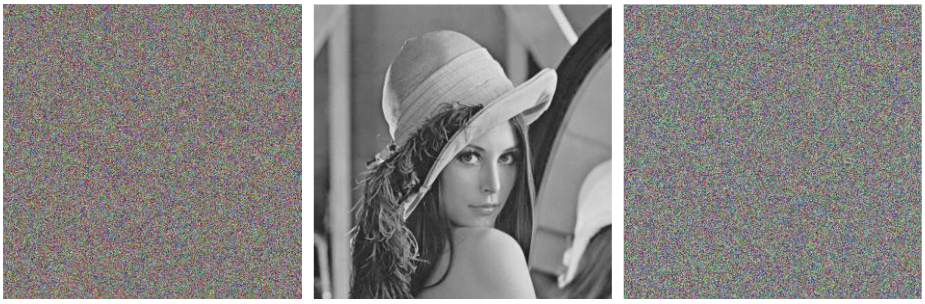
6.3.1 Key Space Analysis

In cryptology, the cipher key should be secure, sensitive and have a large key space. As the novel RC5 uses the 2-D logistic map, it has further extended sub-key space compared with the 1-D Logistic map. Since the 2-D logistic map has larger option range than the 1-D Logistic map, the proposed new algorithm can better against an exhaustive key search attack. Hence the novel RC5 algorithm with 2-D Logistic map has a larger key space compared to the standard RC5 and the RC5 based on the 1-D Logistic map.

6.3.2 Key Sensitive Test

Encryption technology requires an algorithm with a sensitive key, which means that it should not be possible for the correct information to be decrypted by making tiny changes to the security key.

In the key sensitive test, the Lena image (512×512 grey) is used. The proposed novel RC5 algorithm will use the *RC5C-32/12/1.19/ (0.8909, 0.3342)* parameter set-up. The cipher image was encrypted using the novel algorithm and decrypted image as shown in Figure 6-5 left and middle. After a tiny change applied to the initial value (x_0, y_0) from *(0.8909, 0.3342)* to *(0.8909, 0.3341)*, the decrypted image is shown in Figure 6-5 right, that is wrong. Which indicate the cipher image cannot be recovered after a tiny key changed and the wrong decrypted image without any of the original image information. This test proves the novel RC5 algorithm with 2-D Logistic map has very good key sensitivity to resist an attack.



Cipher Image

Decrypted by correct key

Decrypted by tiny changed key

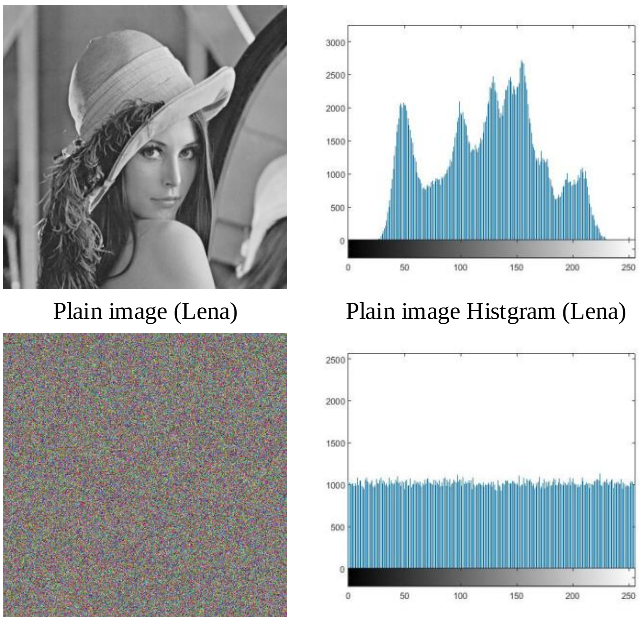
Figure 6-5 Key sensitive test

6.3.3 Statistical Analysis

The ability of the proposed RC5 to resist statistical attacks is analysed in this section. The tests show that the enhanced RC5 has improved confusion and diffusion properties.

a) Histograms Analysis

A uniform histogram of the cipher image should be generated from the plain image. Two histograms of plain image and cipher image were carried out in Figure 6-6.



Plain image (Lena)

Plain image Histgram (Lena)

The novel RC5 cipher image

The novel RC5 cipher image Histgram

Figure 6-6 Image histogram test

The two histograms, the original Lena image (512×512) and the novel RC5 algorithm with 2-D Logistic map’s cipher image in Figure 6-6. The original Lena image has the obvious feature in its histogram (upper right in Figure 6-6) that almost all the grey value is centerd on the interval of [50, 200]. After *RC5C-32/12/1.19/ (0.8909, 0.3342)* encryption processing, the grey value of cipher image histogram (bottom right) is uniformly distributed in the interval [0, 255] without any original image information. This result shows that the proposed algorithm has strong encryption and randomizing ability.

b) Correlation of Two Adjacent Pixels

As in the previous chapter, a correlation analysis of two adjacent pixels is conducted. This test was done by plot the distribution of the horizontally, vertically or diagonally adjacent pixels in the plain image and its corresponding cipher image. The test selected 1000 pairs of two horizontally adjacent pixels, 1000 pairs of two vertically adjacent pixels and 1000 pairs of two diagonally adjacent pixels in the plain image and the cipher image, respectively. The calculation of the correlation coefficient between two adjacent pixels is the same as in equation (6) and equation (7).

The result of the correlation coefficient of the adjacent pixels is presented in the Table 6-1. Figure 6-7 illustrates the horizontal, vertical and diagonal corresponding distribution of the plain image and the cipher image. It can be seen that the original image has strong correlation in all three directions, as shown in the left column of Figure 6-7; the cipher image indicated all three directions are randomly distributed.

Table 6-1 Correlation coefficient of the adjacent pixels result

Lena (512*512) grayscale bitmap	**Plain image**	**Ciphered image by RC5**	**Ciphered image by novel RC5 with 2-D Logistic Map**
Horizontal	0.9719	0.0015	0.0031
Vertical	0.9854	0.0110	0.0025
Diagonal	0.9602	0.0032	0.0022

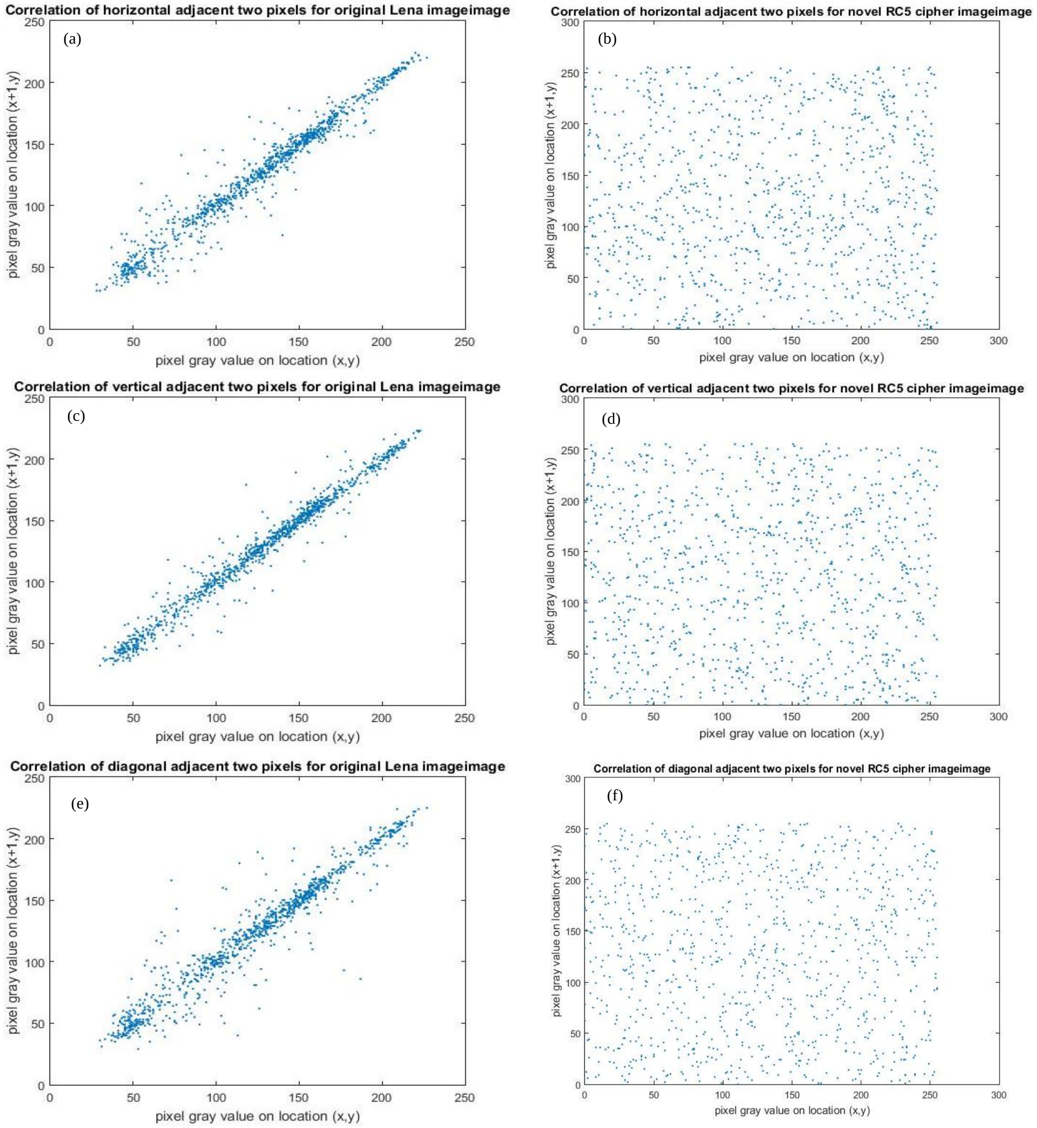


Figure 6-7 Correlation of two adjacent pixels for Lena image of size 512*512 (a) Horizontal direction of the plain image, (b) Horizontal direction of the enhanced RC5 cipher image, (c) Vertical corresponding distribution of plain image, (d) Vertical corresponding distribution of cipher image, (e) Diagonal corresponding distribution of plain image, (d) Diagonal corresponding distribution of cipher image.

c) Information Entropy Analysis

Similar to the information entropy analysis in section 6.3.3, three algorithms have been used for comparison. From the following table 7-2, the entropy values of the cipher image of these encryption algorithms are all close to the theoretical value of 8. However, the proposed algorithm produces a bigger value than others. This indicates that the proposed novel RC5 has been performs better.

Table 6-2 Results of the Cipher Image's Entropy Test

Encryption algorithm	**RC5**	**MRC5[66]**	**Novel RC5 with 2-D Logistic Map**
Entropy value	7.9730	7.9964	7.9993

6.3.4 Differential Attack

For differential cryptanalysis, two quantity measure methods – the number of changing pixel rate (NPCR) and the unified averaged changed intensity (UACI) – are used to evaluate the ability of the proposed novel RC5 algorithm with 2-D Logistic Map to resist differential attack.

The experiments were performed on the proposed algorithm, using the 256 grey-scale image of Lena. The results are shown in table 7-3, where the NPCR score is 99.42% and the UACI score is 33.41%. The results show that the proposed novel RC5 algorithm with 2-D logistic map has a better security level than the enhanced RC5 with 1-D Logistic and the AES. Furthermore, the novel RC5 is much simpler and faster than the enhanced RC5 with 1-D Logistic map and the AES algorithm in image encryption due to the simplified sub-key generation mechanism.

Table 6-3 The NPCR and UACI result

	RC5 [66]	**Enhanced RC5 with 1-D Logistic Map**	**Novel RC5 with 2-D Logistic Map**	**AES**
NPCR	98.8%	99.42%	99.42%	99.42%
UACI	31.2%	33.40%	33.41%	33.41%

6.3.5 Randomness Test Using NIST SP800-22 Test Suit

NIST SP800-22 test suit is a randomness test tool that introduced in provirus chapter. The proposed RC5 with 2-D Logistic map algorithm has been tested using NIST with the same parameters, *RC5C-32/12/1.19/ (0.8909, 0.3342)*. The same image Lena is used as plain image, and 16 bytes security key are used, respectively. The results of the test are shown in Table 7-4.

Table 6-4 SP800-22 Test Result of *RC5C-32/12/1.19/ (0.8909, 0.3342)*

SP800-22 Test List	**Standard RC5 algorithm**	**Proposed RC5 with 2-D Logistic map algorithm**	**Pass (>0.01)**
Frequency	0.293751	0.213309	Yes
BlockFrequency	0.307767	0.350485	Yes
CumulativeSums	0.355076	0.350485	Yes
Runs	0.476054	0.911413	Yes
LongestRun	0.590430	0.739918	Yes
Rank	0.515548	0.350485	Yes
FFT	0.377607	0.534146	Yes
NonOverlappingTemplate	0.499440	0.739918	Yes
OverlappingTemplate	0.477481	0.739918	Yes
ApproximateEntropy	0.440865	0.026528	Yes
Serial	0.552267	0.350485	Yes
LinearComplexity	0.594117	0.534146	Yes

As shown in the Table 7-4, the proposed novel RC5 with 2-D Logistic map algorithm has passed the entire test as recommended by NIST, which indicate the algorithm produced strong randomness of the cipher text.

6.3.6 Balance Analysis

Balance analysis has been introduced in the previous chapter. A balance analysis of the proposed RC5 cryptographic algorithm with 2-D Logistic map is performed using Lena image with *RC5C-32/12/1.19/ (0.8909, 0.3342)*. The balance test results between different lengths of cipher text have been shown in Table 7-5.

Table 6-5 Balance Analysis result

Novel RC5 algorithm with 2-D Logistic map cipher text length (n)	**1000**	**10000**	**100000**	**1000000**
0 (***K1***) (bits)	503	5003	50427	500355
1 (***K2***) (bits)	497	4997	49573	499345
Proportion: $\lvert(\boldsymbol{K1}-\boldsymbol{K2})/\boldsymbol{n}\rvert$	0.006	0.0006	0.00854	0.00101

From Table 7-5, different length of cipher text contains almost same number of "0" and "1" statistically. In addition, with the cipher text length increasing, the ratio between 0's and 1's is closer. This indicates that the cipher text has better balance of "0" and "1".

6.4 Conclusion

With increasing demand for security of wireless sensor networks, a simple, fast encryption algorithm based on the RC5 is proposed in this chapter. The proposed enhanced RC5-w/r/λ $/(x_0, y_0)$ algorithm uses a 2-D Logistic map to generate a sub-key to make it strong in resisting various attacks. Moreover, the modified RC5 algorithm incorporates sub-key and cipher data to further strengthen security. Experiments show the proposed algorithm satisfies the requirements of Wireless Sensor Network, such as low computing capability, less memory cost, power saving and increased security. It is of particular interest for Wireless E-Health application.

Chapter 7 Conclusions and Future Work

7.1 Conclusions

This research work has focused on the emerging E-Health System based on Wireless Sensor Network. With the improvement of living standards in last decades, personal health issue has received a great attention. Because of this reason, E-Health concept has been proposed that measure and record personal daily health status. E-Health concept breaks the traditional medical system architecture that provides real-time and accurate measurement for user. E-Health System is very useful for long time monitoring of chronic diseases, such as chronic heart disease, diabetes, and hypertension. This thesis has reviewed E-Health System in recent years, and focuses three aspects of new challenges in E-Health System: platform aspect, management and storage aspect, and security aspect.

In the platform aspect, a Wireless E-Health System based on Wireless Sensor Network has been developed. The proposed Wireless E-Health Sensor Node consists of microprocessor, sensor, and wireless module that used to measure users' physical signs. The Wireless E-Health System has been proposed using Wireless E-Health Sensor Node, which can be communicated with Cloud server without other middle platform, which increase the portability. The proposed system can also include the Wireless E-Health Gateway which is a middle platform using different wireless modules to communicate and manage Wireless E-Health Sensor Nodes. With the Gateway help, user can manage sensors and get feedback much more easily. Another feature in the proposed system was smart function that based on smart phone as the middle platform. In this thesis, two sensor APPs have been developed that can measure and record people's blood pressure and oxygen saturation.

In the aspect of management and storage, this research work introduces the novel Cloud concept to deal with the huge requirement of management and storage from E-Health System. Chapter 4 has presents the Cloud concept and describe the method to use Cloud for E-Health. In this project, an E-Health server has been set up in Amazon Cloud. Based on the advantage of Cloud technology, a scalable E-Health care web service has been developed. With this service, medical data can be stored, managed and analyzed by professional medical staff, which can give feedback to patient.

Considering the sensitivity of the medical data and personal privacy, security of E-Health System has attracted a lot of attention. This thesis has reviewed the Wireless Sensor Network security issue and related research work in recent years. RC5 is one of the simplest cryptographic algorithms and can be deployed in both hardware and software. Considering the feature of E-Health data and Wireless E-Health System resources limitation, RC5 is a good candidate for Wireless Sensor Network application. In order to increase security and randomness of RC5 sub-key generation, 1-D Logistic map has been integrated into the proposed enhanced algorithm. This high performance encryption algorithm will be used for transmit medical data between terminal and Cloud server that strengthen the security. With the deep research of 1-D Logistic map, 2-D Logistic map has been invoked that increase the control parameters and chaotic performance. A novel RC5 cryptographic with 2-D Logistic map has been proposed which use 2-D Logistic mapping to simplify and chaotic the process of sub-key generation. The experiments on both proposed new algorithms show that performance is better than the standard RC5 algorithm and even better than the other modify RC5 algorithm with low cost.

The contributions and innovation of this research project are summarized:

- Build up a Wireless E-Health Care System based on Wireless Sensor Network.
- Create the Cloud computing service for E-Health Care System.

- Proposed RC5 cryptographic algorithms based on Chaotic theory to increase the randomness and security of cipher data.

The proposed E-Health Care System based on Wireless Sensor Network achieves the smart health care and meet high security requirement.

7.2 Future Work

With the development of Cloud computing technology, the next step of this research is to further investigate the specific Cloud computing facilities for E-Health service apart from host E-Health System and database management. Big data is an emerging concept that focuses on use huge information to help people solve various problems [79]. At present, E-Health service based on Big data has been proposed, which apply Big data technology to process E-Health data and provide services [80]. Another path of this research area was using E-Health data to create a platform based on Cloud and Big data technology to improve E-Health service [81]. In further, the volume of health data will be significant amounts that can be used for Big data technology and provide powerful and accurately treatment. Also, integrated with the Big data technology, medical analysis, such as, ECG signal analysis or compression techniques could be incorporated into the system.

In the E-Health sensor aspect, the next step is to research the new hardware that use wireless module integrated with detect sensor only. Because the microprocessor of E-health sensor has use large power that greatly limits its service time. On the other hand, wireless module contains function interfaces that can communication with some simple sensor. Therefore, remove microprocessor will be one of the next steps. At the same time, more powerful and indigenized sensor will be researched and implemented to make the E-Health care system more functions. With the development of E-Health care system based on Wireless Sensor Network, there are increasing demand to develop a scalable hardware platform to enable user to plug and use different sensors or wireless connectivity to center hospital.

The proposed enhanced RC5 algorithm has achieved to deploy on embed system and the experiments show the algorithm has got the high security level, but the time cost was unacceptable. Therefore, optimization of the encryption algorithm is very urgent work. The further work will in-depth research of the microprocessor architecture, and propose optimized method to speed up algorithm that computing on embed system.

Security issue will not be stopped with the development of encryption technology, therefore, stronger encryption algorithm will be considered for Wireless E-Health System. Although RC5 has very good features, like simple operation, fast computing speed and security. As the super computer advance, RC5 will not stronger to suffer all kind attacks. Therefore, more advanced encryption algorithm should be involved. For example, RC6 is the upgraded version of RC5 that keeps the symmetric block cipher feature and increase the block size to meet the advance encryption algorithm requirement [82]. RC6 has good security ability to resist the various linear and differential attacks [83]. At the same time, Logistic map still has broad research areas. Further investigation of Logistic map can propose 3-D Logistic map to increase the chaotic behavior. Hence, RC6 algorithm integrated with 3-D Logistic map can provide more complex key produce procedure and increase the cipher randomness.

References

[1] C. S. Trevor Lewis, Gina Lagomarsino & Julian Schweitzer "E-health in low- and middle-income countries: findings from the Center for Health Market Innovations," *Bulletin of the World Health Organization,* vol. 90, pp. 332-340, 2012.

[2] K. W. e. al., "An Age Old Problem A review of the care received by elderly patients undergoing surgery," presented at the National Confidential Enquiry into Patient Outcome and Death, 2010.

[3] R. I. Muhamedyev, A. T. Mansharipova, R. Butin, E. Muhamedyeva, and N. Rakhimzhanova, "New ICT trends in solving the problem of accelerated population aging," in *15th International Conference on Control, Automation and Systems (ICCAS)*, 2015, pp. 1969-1973.

[4] N. Botezatu, R. Lupu, and A. Stan, "Energy-aware routing for e-health wireless sensor networks," in *E-Health and Bioengineering Conference (EHB)*, 2013, pp. 1-4.

[5] (April 15). *The NHS structure*. Available: http://www.nhs.uk/NHSEngland/thenhs/about/Pages/nhsstructure.aspx

[6] F. Pinciroli, M. Corso, A. Fuggetta, M. Masseroli, S. Bonacina, and S. Marceglia, "Telemedicine and E-Health," *IEEE Pulse,* vol. 2, pp. 62-70, May-June 2011.

[7] T. Sahama, L. Simpson, and B. Lane, "Security and Privacy in eHealth: Is it possible?," in *15th IEEE International Conference on e-Health Networking, Applications & Services (Healthcom)*, 2013, pp. 249-253.

[8] N. M. Shrestha, A. Alsadoon, P. W. C. Prasad, L. Hourany, and A. Elchouemi, "Enhanced e-health framework for security and privacy in healthcare system," in *6th International Conference on Digital Information Processing and Communications (ICDIPC)*, 2016, pp. 75-79.

[9] N. Thiranant, M. Sain, and H. J. Lee, "A design of security framework for data privacy in e-health system using web service," in *16th International Conference on Advanced Communication Technology*, 2014, pp. 40-43.

[10] A. Michalas, N. Paladi, and C. Gehrmann, "Security aspects of e-Health systems migration to the cloud," in *16th IEEE International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2014, pp. 212-218.

[11] A. Papalambrou, J. Gialelis, and D. Serpanos, "Increasing security in wireless e-health systems," in *IEEE International Symposium on Signal Processing and Information Technology*, 2013, pp. 000015-000020.

[12] H. R. Jara and E. Schafir, "e-Health: An introduction to the challenges of privacy and security," in *Central America and Panama Convention (CONCAPAN XXXIV)*, 2014, pp. 1-5.

[13] G. H. Zhang, C. C. Y. Poon, and Y. T. Zhang, "A review on body area networks security for healthcare," *CN*, vol. 2011, pp. 1-8, 2011.

[14] D. Shin, T. Sahama, and R. Gajanayake, "Secured e-health data retrieval in DaaS and Big Data," in *15th IEEE International Conference on e-Health Networking, Applications & Services (Healthcom)*, 2013, pp. 255-259.

[15] Z. Yingbing and L. Yongzhen, "The design and implementation of a symmetric encryption algorithm based on DES," in *5th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2014, pp. 517-520.

[16] W. Y. Zibideh and M. M. Matalgah, "Modified-DES encryption algorithm with improved BER performance in wireless communication," in *IEEE Radio and Wireless Symposium*, 2011, pp. 219-222.

[17] Y. A. Nasser, M. A. Bazzoun, and S. Abdul-Nabi, "AES algorithm implementation for a simple low cost portable 8-bit microcontroller," in *6th International Conference on Digital Information Processing and Communications (ICDIPC)*, 2016, pp. 203-207.

[18] B. Bhat, A. W. Ali, and A. Gupta, "DES and AES performance evaluation," in *International Conference on Computing, Communication & Automation (ICCCA)*, 2015, pp. 887-890.

[19] K. Balasubramanian, "Variants of RSA and their cryptanalysis," in *International Conference on Communication and Network Technologies (ICCNT)* 2014, pp. 145-149.

[20] S. A. Nagar and S. Alshamma, "High speed implementation of RSA algorithm with modified keys exchange," in *6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, 2012, pp. 639-642.

[21] A. A. Alsahli and H. U. Khan, "Security challenges of wireless sensors devices (MOTES)," in *World Congress on Computer Applications and Information Systems (WCCAIS)*, 2014, pp. 1-9.

[22] S. Soegijoko, I. M. Puspitasari, A. Aridarma, and I. D. Jani, "e-health for improving community healthcare: Encouraging clinical experience of simple e-prescription system and m-health system development for mother and childcare," in *13th IEEE International Conference on e-Health Networking Applications and Services (Healthcom)*, 2011, pp. 102-105.

[23] M. Weitzel, A. Smith, S. de Deugd, and R. Yates, "A Web 2.0 Model for Patient-Centered Health Informatics Applications," *Computer,* vol. 43, pp. 43-50, 2010.

[24] S. Subramoniam and A. H. M. Saifullah Sadi, "Healthcare 2.0," *IT Professional,* vol. 12, pp. 46-51, 2010.

[25] M. Kawarasaki, K. Konishi, M. Ohara, and T. Igarashi, "Adding telephone interface to web service Implication to the self-care support system for life-style diseases," in *13th IEEE International Conference on e-Health Networking Applications and Services (Healthcom)*, 2011, pp. 114-117.

[26] M. Kawarasaki, K. Konishi, M. Ohara, and T. Igarashi, "Adding telephone interface to web service Implication to the self-care support system for life-style diseases," in *e-Health Networking Applications and Services (Healthcom), 2011 13th IEEE International Conference on*, 2011, pp. 114-117.

[27] K. Guixia, "Wireless eHealth (WeHealth)-From concept to practice," in *14th IEEE International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2012, pp. 375-378.

[28] D. Lin, X. Zhang, F. Labeau, and G. Kang, "A hypertension monitoring system and its system accuracy evaluation," in *14th IEEE International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2012, pp. 132-137.

[29] A. I. Oluwaranti and T. D. Obasanya, "Mobility Conscious Medium Access Control Scheme for Wireless Sensor Networks: A Conceptual Approach," 2014.

[30] S. Adibi, "Biomedical Sensing Analyzer (BSA) for Mobile-Health (mHealth)-LTE," *IEEE Journal of Biomedical and Health Informatics,* vol. PP, pp. 1-1, 2013.

[31] S. Adibi, "Link Technologies and BlackBerry Mobile Health (mHealth) Solutions: A Review," *IEEE Transactions on Information Technology in Biomedicine,* vol. 16, pp. 586-597, 2012.

[32] C. Hao and J. Xueqin, "New requirements and trends of mHealth," in *14th IEEE International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2012, pp. 27-31.

[33] L. Jaeho, "Smart health: Concepts and status of ubiquitous health with smartphone," in *International Conference on ICT Convergence (ICTC)*, 2011, pp. 388-389.

[34] S. Kumar, W. Nilsen, M. Pavel, and M. Srivastava, "Mobile Health: Revolutionizing Healthcare Through Transdisciplinary Research," *Computer,* vol. 46, pp. 28-35, 2013.

[35] R. K. Lomotey, S. Jamal, and R. Deters, "SOPHRA: A Mobile Web Services Hosting Infrastructure in mHealth," in *1st IEEE International Conference on Mobile Services (MS)* 2012, pp. 88-95.

[36] N. Agoulmine, M. J. Deen, L. Jeong-Soo, and M. Meyyappan, "U-Health Smart Home," *Nanotechnology Magazine,* vol. 5, pp. 6-11, 2011.

[37] K. Jin, C. Hyeok-soo, W. Hui, N. Agoulmine, M. J. Deerv, and J. W. K. Hong, "POSTECH's U-Health Smart Home for elderly monitoring and support," in *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 2010, pp. 1-6.

[38] N. Noury, K. A. Quach, M. Berenguer, M. J. Bouzi, and H. Teyssier, "A feasibility study of using a smartphone to monitor mobility in elderly," in *14th IEEE International Conference on e-Health Networking, Applications and Services (Healthcom),* 2012, pp. 423-426.

[39] T. Suzuki, H. Tanaka, S. Minami, H. Yamada, and T. Miyata, "Wearable wireless vital monitoring technology for smart health care," in *7th International Symposium on Medical Information and Communication Technology (ISMICT),* 2013, pp. 1-4.

[40] H. Watanabe, M. Kawarasaki, A. Sato, and K. Yoshida, "Development of wearable heart disease monitoring and alerting system associated with smartphone," in *14th IEEE International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2012, pp. 292-297.

[41] Y. Yan, Z. Xuefeng, and O. Jinping, "A new idea: Mobile structural health monitoring using Smart phones," in *3rd International Conference on Intelligent Control and Information Processing (ICICIP)*, 2012, pp. 714-716.

[42] S. M. M. Rahman, M. M. Masud, C. Adams, K. El-Khatib, H. T. Mouftah, and E. Okamoto, "Cryptographic security models for eHealth P2P database management systems network," in *9th Annual International Conference on Privacy, Security and Trust (PST)*, 2011, pp. 164-173.

[43] Z. Wenwu, L. Chong, W. Jianfeng, and L. Shipeng, "Multimedia Cloud Computing," *IEEE Signal Processing Magazine*, vol. 28, pp. 59-69, 2011.

[44] K. R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf*, et al.*, "Performance analysis of high performance computing applications on the amazon web services cloud," in *2nd IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2010, pp. 159-168.

[45] "Google App Engine," in *Developing with Google App Engine*, ed: Apress, 2009, pp. 1-10.

[46] T. Redkar and T. Guidici, *Windows Azure Platform*: Springer, 2009.

[47] C. O. Rolim, F. L. Koch, C. B. Westphall, J. Werner, A. Fracalossi, and G. S. Salvador, "A cloud computing solution for patient's data collection in health care institutions," in *2nd International Conference on eHealth, Telemedicine, and Social Medicine*, 2010, pp. 95-99.

[48] B. E. Reddy, T. V. S. Kumar, and G. Ramu, "An Efficient Cloud Framework for Health Care Monitoring System," in *International Symposium on Cloud and Services Computing (ISCOS)*, 2012, pp. 113-117.

[49] Z. S. Hu, S. Y. Li, and D. Y. Li, "SpO2 Detecting Applied on Android Platform," *Applied Mechanics and Materials*, vol. 303, pp. 659-662, 2013.

[50] F. Zhen, Z. Zhan, S. Fangmin, C. Xianxiang, D. Lidong, L. Huaiyong*, et al.*, "The 3AHcare node: Health monitoring continuously," in *14th IEEE International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2012, pp. 365-366.

[51] M. Fengou, G. Mantas, D. Lymberopoulos, N. Komninos, S. Fengos, and N. Lazarou, "A New Framework Architecture for Next Generation e-Health Services," *IEEE Journal of Biomedical and Health Informatics,* vol. 17, pp. 9-18, 2013.

[52] R. G.K and K. Baskaran, "A Survey on Futuristic Health Care System: WBANs," *Procedia Engineering,* vol. 30, pp. 889-896, 2012.

[53] B. P. Lo, S. Thiemjarus, R. King, and G.-Z. Yang, *Body sensor network–a wireless sensor platform for pervasive healthcare monitoring*: na, 2005.

[54] M. T. Nkosi, F. Mekuria, and S. H. Gejibo, "Challenges in mobile bio-sensor based mHealth development," in *13th IEEE International Conference on e-Health Networking Applications and Services (Healthcom)*, 2011, pp. 21-27.

[55] A. Abdullah Alsahli and H. Ullah Khan, "Security challenges of wireless sensors devices (MOTES)," in *World Congress on Computer Applications and Information Systems (WCCAIS)*, 2014, pp. 1-9.

[56] M. Chan, D. Estève, J.-Y. Fourniols, C. Escriba, and E. Campo, "Smart wearable systems: Current status and future challenges," *Artificial intelligence in medicine,* vol. 56, pp. 137-156, 2012.

[57] S. Kumar and K. Ovsthus, "An Industrial Perspective on Wireless Sensor Networks—A Survey of Requirements, Protocols, and Challenges."

[58] R. L. Rivest, "The RC5 encryption algorithm," in *Fast Software Encryption*, 1995, pp. 86-96.

[59] S. K. Khuraijam and K. Radhika, "A Novel Symmetric Key Encryption Algorithm Based on RC5 in Wireless Sensor Network."

[60] C. Ke, Z. Xianzhang, and F. Yuqi, "Block EncryptionAlgorithmBasedon Chaos andRC5 in WSN," *Computer Measurement & Control,* pp. 2249-2252, 2009.

[61] O. Elkeelany and A. Olabisi, "Case study: integrated design of RC5 encryption," 2007, pp. 69-72.

[62] J. Kukkurainen, M. Soini, and L. Sydanheimo, "RC5-based security in wireless sensor networks: utilization and performance," *WSEAS TRANSACTIONS on COMPUTERS,* pp. 1109-2750, 2010.

[63] B. Kaliski and Y. L. Yin, "On the security of the RC5 encryption algorithm," RSA Laboratories Technical Report TR-602. To appear1998.

[64] N. Bajaj and A. Thakur, "Enhancement of RC5 for image encryption," in *International Conference on Image Information Processing (ICIIP)*, 2011, pp. 1-5.

[65] A. Biryukov and E. Kushilevitz, "Improved cryptanalysis of RC5," in *Advances in Cryptology*, ed: Springer, 1998, pp. 85-99.

[66] M. Amin and A. A. A. El-Latif, "Efficient modified RC5 based on chaos adapted to image encryption," *Journal of Electronic Imaging,* vol. 19, pp. 013012-013012-10, 2010.

[67] L. Yanbing and T. Simei, "Design and statistical analysis of a new chaos block cipher for WSN," in *IEEE International Conference on Information Theory and Information Security (ICITIS)*, 2010, pp. 327-330.

[68] Y. Sun and G. Wang, "An Image Encryption Scheme Based on Modified Logistic Map," in *4th International Workshop on Chaos-Fractals Theories and Applications (IWCFTA)*, 2011, pp. 179-182.

[69] W. Stallings, *Network and internetwork security: principles and practice* vol. 1: Prentice Hall Englewood Cliffs, 1995.

[70] X. Wei, Y. Chao, F. Haohuan, W. Xinliang, X. Yangtong, G. Lin, *et al.*, "Enabling and Scaling a Global Shallow-Water Atmospheric Model on Tianhe-2," in *28th IEEE International Parallel and Distributed Processing Symposium*, 2014, pp. 745-754.

[71] L. Abraham and N. Daniel, "Secure image encryption algorithms: A review," vol. 100, p. 2, 2013.

[72] B. S. Kaliski Jr and Y. L. Yin, "On differential and linear cryptanalysis of the RC5 encryption algorithm," in *Advances in Cryptology*, ed: Springer, 1995, pp. 171-184.

[73] H. Ahmed, H. M. Kalash, and O. Allah, "Implementation of rc5 block cipher algorithm for image cryptosystems," *International Journal of Information Technology*, vol. 3, pp. 245-250, 2007.

[74] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, *et al.*, "NIST special publication 800-22," *A statistical test suite for random and pseudorandom number generators for cryptographic applications,* 2001.

[75] K. Kumar Kabi, C. Pradhan, B. J. Saha, and A. Kumar Bisoi, "Comparative study of image encryption using 2D chaotic map," in *International Conference on Information Systems and Computer Networks (ISCON)*, 2014, pp. 105-108.

[76] C. Dongming, Q. Deding, and W. Dongqi, "AES Key Expansion Algorithm Based on 2D Logistic Mapping," in *5th International Workshop on Chaos-Fractals Theories and Applications (IWCFTA)*, 2012, pp. 207-211.

[77] Y. Wu, G. Yang, H. Jin, and J. P. Noonan, "Image encryption using the two-dimensional logistic chaotic map," *Journal of Electronic Imaging,* vol. 21, pp. 013014-1-013014-15, 2012.

[78] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D Sine Logistic modulation map for image encryption," *Information Sciences,* vol. 297, pp. 80-94, 2015.

[79] M. Díaz, G. Juan, O. Lucas, and A. Ryuga, "Big Data on the Internet of Things: An Example for the E-health," in *Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 2012, pp. 898-900.

[80] W. Liu and E. K. Park, "Big Data as an e-Health Service," in *International Conference on Computing, Networking and Communications (ICNC)*, 2014, pp. 982-988.

[81] S. Cha, A. Abusharekh, and S. S. Abidi, "Towards a 'Big' Health Data Analytics Platform," in *IEEE First International Conference on Big Data Computing Service and Applications (BigDataService)*, 2015, pp. 233-241.

[82] K. Aggarwal, "Comparison of RC6, Modified RC6 & Enhancement of RC6," in *International Conference on Advances in Computer Engineering and Applications (ICACEA)* 2015, pp. 444-449.

[83] K. Aggarwal and H. K. Verma, "Hash_RC6 - Variable Length Hash Algorithm using RC6," in *International Conference on Advances in Computer Engineering and Applications (ICACEA)*, 2015, pp. 450-456.

Appendixes

Appendix I. E-Health Gateway Window Design

1. Main Window Code

```
<Glide Version="1.0.6">
 <Window Name="Win2" Width="320" Height="240" BackColor="FFFFFF">
  <TextBlock Name="lab_Time" X="0" Y="0" Width="320" Height="20" Alpha="255" Text="Time"
TextAlign="Left" TextVerticalAlign="Top" Font="1" FontColor="0" BackColor="999999" ShowBackColor="True"/>
  <Image Name="imgBtn_0" X="40" Y="25" Width="100" Height="100" Alpha="255"/>
  <Image Name="imgBtn_1" X="180" Y="25" Width="100" Height="100" Alpha="255"/>
  <Image Name="imgBtn_2" X="40" Y="135" Width="100" Height="100" Alpha="255"/>
  <Image Name="imgBtn_3" X="180" Y="135" Width="100" Height="100" Alpha="255"/>
 </Window>
</Glide>
```

2. SpO2 Window Code

```
<Glide Version="1.0.6">
 <Window Name="Win3" Width="320" Height="240" BackColor="FFFFFF">
  <TextBlock Name="lab_Time" X="0" Y="0" Width="320" Height="20" Alpha="255" Text="Time"
TextAlign="Left" TextVerticalAlign="Top" Font="1" FontColor="0" BackColor="999999" ShowBackColor="True"/>
  <Image Name="imgHome" X="15" Y="38" Width="32" Height="32" Alpha="255"/>
  <Image Name="imgStart" X="213" Y="113" Width="96" Height="96" Alpha="255"/>
  <Image Name="imgSave" X="256" Y="41" Width="48" Height="48" Alpha="255"/>
  <TextBlock Name="labTitle" X="109" Y="46" Width="150" Height="32" Alpha="255" Text="SpO2 Test"
TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBlock Name="labSpo2" X="39" Y="103" Width="150" Height="32" Alpha="255" Text="SpO2 = 0"
TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBlock Name="labHr" X="38" Y="143" Width="150" Height="32" Alpha="255" Text="Heart Rate = 0"
TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBlock Name="labStrength" X="38" Y="189" Width="150" Height="32" Alpha="255" Text="Strength = 0"
TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
 </Window>
</Glide>
```

3. Blood Pressure Window Code

```
<Glide Version="1.0.6">
 <Window Name="Win4" Width="320" Height="240" BackColor="FFFFFF">
```

```
  <TextBlock Name="lab_Time" X="0" Y="0" Width="320" Height="20" Alpha="255" Text="Time"
TextAlign="Left" TextVerticalAlign="Top" Font="1" FontColor="0" BackColor="999999" ShowBackColor="True"/>
  <Image Name="imgHome" X="15" Y="38" Width="32" Height="32" Alpha="255"/>
  <Image Name="imgStart" X="215" Y="119" Width="96" Height="96" Alpha="255"/>
  <Image Name="imgSave" X="256" Y="41" Width="48" Height="48" Alpha="255"/>
  <TextBlock Name="labTitle" X="88" Y="45" Width="200" Height="32" Alpha="255" Text="Blood Pressure
Test" TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000"
ShowBackColor="False"/>
  <TextBlock Name="labSBP" X="39" Y="103" Width="150" Height="32" Alpha="255" Text="Systolic BP = 0"
TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBlock Name="labDBP" X="38" Y="136" Width="150" Height="32" Alpha="255" Text="Diastolic BP = 0"
TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBlock Name="labABP" X="36" Y="169" Width="150" Height="32" Alpha="255" Text="Average BP = 0"
TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBlock Name="labHR" X="39" Y="202" Width="150" Height="32" Alpha="255" Text="Heart Rate =  0"
TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
 </Window>
</Glide>
```

4. Report Window Code

```
<Glide Version="1.0.6">
 <Window Name="Win5" Width="320" Height="240" BackColor="FFFFFF">
  <TextBlock Name="lab_Time" X="0" Y="0" Width="320" Height="20" Alpha="255" Text="Time"
TextAlign="Left" TextVerticalAlign="Top" Font="1" FontColor="0" BackColor="999999" ShowBackColor="True"/>
  <Image Name="imgHome" X="15" Y="38" Width="32" Height="32" Alpha="255"/>
  <Image Name="imgReport" X="209" Y="137" Width="96" Height="96" Alpha="255"/>
  <Image Name="imgSave" X="258" Y="37" Width="48" Height="48" Alpha="255"/>
      <TextBlock Name="labSpO2" X="29" Y="131" Width="150" Height="32" Alpha="255" Text="SpO2 = 0"
TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000"
ShowBackColor="False"/>
  <TextBlock Name="labTitle" X="81" Y="38" Width="200" Height="32" Alpha="255" Text="Health Report"
TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000"
ShowBackColor="False"/>
  <TextBlock Name="labSBP" X="29" Y="153" Width="150" Height="32" Alpha="255" Text="Systolic BP = 0"
TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000"
ShowBackColor="False"/>
  <TextBlock Name="labDBP" X="31" Y="171" Width="150" Height="32" Alpha="255" Text="Diastolic BP = 0"
TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000"
ShowBackColor="False"/>
  <TextBlock Name="labABP" X="31" Y="189" Width="150" Height="32" Alpha="255" Text="Average BP = 0"
TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000"
ShowBackColor="False"/>
```

```
  <TextBlock Name="labHR" X="32" Y="208" Width="150" Height="32" Alpha="255" Text="Heart Rate = 0" TextAlign="Left" TextVerticalAlign="Top" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <CheckBox Name="checkBoxWIFI" X="167" Y="84" Width="32" Height="32" Alpha="255" Checked="False"/>
  <CheckBox Name="checkBoxSMS" X="34" Y="84" Width="32" Height="32" Alpha="255" Checked="False"/>
  <TextBlock Name="labBySMS" X="67" Y="85" Width="100" Height="32" Alpha="255" Text="By SMS" TextAlign="Left" TextVerticalAlign="Middle" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBlock Name="labByWIFI" X="201" Y="87" Width="100" Height="32" Alpha="255" Text="By Wi-Fi" TextAlign="Left" TextVerticalAlign="Middle" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
 </Window>
</Glide>
```

5. Set Window Code

```
<Glide Version="1.0.6">
 <Window Name="Win0" Width="320" Height="240" BackColor="FFFFFF">
  <TextBlock Name="lab_Time" X="0" Y="0" Width="320" Height="20" Alpha="255" Text="Time" TextAlign="Left" TextVerticalAlign="Top" Font="1" FontColor="0" BackColor="999999" ShowBackColor="True"/>
  <TextBlock Name="lab_title" X="70" Y="30" Width="200" Height="32" Alpha="255" Text="E-Health System" TextAlign="Left" TextVerticalAlign="Top" Font="5" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBlock Name="lab_Date" X="5" Y="80" Width="30" Height="32" Alpha="255" Text="SET DATE" TextAlign="Center" TextVerticalAlign="Middle" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBlock Name="lab_Time" X="5" Y="130" Width="30" Height="32" Alpha="255" Text="SET TIME" TextAlign="Center" TextVerticalAlign="Middle" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBlock Name="lab_Wifi" X="8" Y="180" Width="30" Height="32" Alpha="255" Text="SET WIFI" TextAlign="Center" TextVerticalAlign="Middle" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBlock Name="lab_Colon" X="155" Y="130" Width="15" Height="32" Alpha="255" Text=":" TextAlign="Center" TextVerticalAlign="Middle" Font="6" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBlock Name="lab_or" X="130" Y="180" Width="30" Height="32" Alpha="255" Text="OR" TextAlign="Center" TextVerticalAlign="Middle" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <Dropdown Name="dd_Day" X="45" Y="80" Width="80" Height="32" Alpha="255" Text="DAY" Font="2" FontColor="000000"/>
  <Dropdown Name="dd_Month" X="130" Y="80" Width="80" Height="32" Alpha="255" Text="MONTH" Font="2" FontColor="000000"/>
  <Dropdown Name="dd_Year" X="215" Y="80" Width="100" Height="32" Alpha="255" Text="YEAR" Font="2" FontColor="000000"/>
  <Dropdown Name="dd_Hour" X="45" Y="130" Width="100" Height="32" Alpha="255" Text="HOUR" Font="2" FontColor="000000"/>
  <Dropdown Name="dd_Minute" X="175" Y="130" Width="100" Height="32" Alpha="255" Text="MINUTE" Font="2" FontColor="000000"/>
      <Image Name="imgHome" X="15" Y="25" Width="32" Height="32" Alpha="255"/>
  <Button Name="btn_Default" X="45" Y="180" Width="80" Height="32" Alpha="255" Text="DEFAULT" Font="4" FontColor="000000" DisabledFontColor="808080" TintColor="000000" TintAmount="0"/>
  <Button Name="btn_newProf" X="165" Y="180" Width="100" Height="32" Alpha="255" Text="NEW PROFILE" Font="4" FontColor="000000" DisabledFontColor="808080" TintColor="000000" TintAmount="0"/>
  <Button Name="btn_timeOK" X="280" Y="130" Width="32" Height="32" Alpha="255" Text="OK" Font="4" FontColor="000000" DisabledFontColor="808080" TintColor="000000" TintAmount="0"/>
 </Window>
</Glide>
```

6. Profile Window Code

```
<Glide Version="1.0.6">
 <Window Name="Win1" Width="320" Height="240" BackColor="FFFFFF">
      <Image Name="imgHome" X="35" Y="45" Width="32" Height="32" Alpha="255"/>
  <TextBlock Name="lab_Time" X="0" Y="0" Width="320" Height="20" Alpha="255" Text="Time" TextAlign="Left" TextVerticalAlign="Top" Font="1" FontColor="0" BackColor="999999" ShowBackColor="True"/>
  <TextBlock Name="labTitle" X="90" Y="50" Width="200" Height="32" Alpha="255" Text="New User Profile" TextAlign="Left" TextVerticalAlign="Top" Font="5" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBlock Name="labUserName" X="30" Y="130" Width="100" Height="32" Alpha="255" Text="User Name:" TextAlign="Left" TextVerticalAlign="Middle" Font="4" FontColor="0" BackColor="000000" ShowBackColor="False"/>
  <TextBox Name="tb_UserName" X="115" Y="130" Width="100" Height="32" Alpha="255" Text="" TextAlign="Left" Font="4" FontColor="000000"/>
      <Button Name="btn_Save" X="230" Y="130" Width="60" Height="32" Alpha="255" Text="Save" Font="2" FontColor="000000" DisabledFontColor="808080" TintColor="000000" TintAmount="0"/>
 </Window>
</Glide>
```

Appendix II. Wi-Fi Driver Code – C#

```
public class Wifly
    {
        /// <summary>
        /// Wi-Fi Class Defination
        /// </summary>
        /// <param name="sokcetNumber">Socket Number ("COM1, COM2, COM3")</param>
        public Wifly(int socketNumber)
        {
            server_URL = "health4u.somee.com";
            server_Port = 80;
            services_fun_Login = "health4u.asmx/login";
            services_fun_Register = "health4u.asmx/register";
            services_fun_uploadSpo2 = "health4u.asmx/uploadSpo2 ";
            services_fun_uploadBP = "health4u.asmx/uploadBloodPressure ";
            services_Content_Type = "Content-Type: application/x-www-form-urlencoded\n";
            services_Connection = "Connection: close\n";
            LocalIP = "0.0.0.0";
            string t_Command_Init = "$$$";
            _Command_Init = System.Text.Encoding.UTF8.GetBytes(t_Command_Init);
            _Port_Name = "COM" + socketNumber;
            _wifly = new SerialPort(_Port_Name, 9600, Parity.None, 8, StopBits.One);
            Reset = new OutputPort((Cpu.Pin)GHI.Hardware.EMX.Pin.IO75, true);
            try
            {
                _wifly.Open();
                Debug.Print("Wi-Fi Open.");
            }
            catch (Exception err)
            {
                Debug.Print("Err:" + err.Message.ToString());
            }
        }

        #region "Private Data Types"
        //Class
        private Wifly_Settings _wifly_settings;
        //Strings
        private string _Port_Name = "";
        private string _Command_Mode_Response = "";
        private string _Serial_String_Buffer = "";
```

```
 //Ints
 private int _baud = 0;
 private int _Timeout = 5000;

 //Classes
 private SerialPort _wifly;
 //private SerialPort _debug_port;
 private static Thread listen = null;

 //Byte
 private byte[] _Command_Init = new byte[3];
private byte[] _Serial_Byte_Buffer = new byte[1] { 0x00 };

//Bools
private bool _internet_ready = false;
private bool _device_ready = false;
private bool _DHCP = false;
private bool _PostState = false;
private bool _Command_Mode_Response_Complete = true;
private bool _Command_Mode_Response_Okay = true;

//Pin declarations
public OutputPort Reset;
private OutputPort RTS;

//Others
private StreamMode _stream = StreamMode.NoStream;
/// <summary>
/// The current debug mode.
/// </summary>
public DebugMode _debug = DebugMode.NoDebug;
private DebugLevel _debug_level = DebugLevel.DebugErrors;
private DataReturnLevel _data_level = DataReturnLevel.ReturnIncomming;
private DateTime _TimeOutDate;

private RunMode _RunMode = RunMode.Normal;

#endregion

#region Define HTTP parameters

public string server_URL { get; set; }
public int server_Port { get; set; }
public string services_fun_Login { get; set; }
public string services_fun_Register { get; set; }
public string services_fun_uploadSpo2 { get; set; }
public string services_fun_uploadBP { get; set; }
public string services_Content_Type { get; set; }
public string services_Connection { get; set; }

#endregion

#region "Enumerations"

/// <summary>
/// Represents the debug mode.
/// </summary>
public enum DebugMode
{
  /// <summary>
  /// Use no debugging
  /// </summary>
  NoDebug,

  /// <summary>
  /// Report debugging to Visual Studio debug output
  /// </summary>
  StandardDebug,

  /// <summary>
  /// Re-direct debugging to a given serial port.
  /// Console Debugging
  /// </summary>
  SerialDebug
};

/// <summary>
/// Represents the debug level.
/// </summary>
public enum DebugLevel
{
  /// <summary>
  /// Only debug errors.
  /// </summary>
  DebugErrors,
  /// <summary>
```

```
    /// Debug everything.
    /// </summary>
    DebugAll
};

/// <summary>
/// Represents the level of data to return.
/// </summary>
public enum DataReturnLevel
{
    /// <summary>
    /// Returns all data.
    /// Command data and External Communications
    /// </summary>
    ReturnAll,

    /// <summary>
    /// Returns only External, non-command data
    /// </summary>
    ReturnIncomming
};

/// <summary>
/// Represents the stream mode.
/// </summary>
public enum StreamMode
{
    /// <summary>
    /// Idle
      /// </summary>
      NoStream,

      /// <summary>
      /// Command Data stream
      /// </summary>
      CommandStream,

      /// <summary>
      /// Stream for get reponses
      /// </summary>
      GetStream,

      /// <summary>
    /// Non-command data (external communications)
    /// </summary>
    DataStream
};

/// <summary>
/// Represents the socket protocal.
/// </summary>
public enum SocketProtocol
{
    /// <summary>
    /// UPD Mode: Connection-less protocol with no handshaking
    /// </summary>
    UDP = 1,

    /// <summary>
    /// TCP Server Mode: TCP Connection with handshaking (Client and Server)
    /// </summary>
    TCP_Server = 2,

    /// <summary>
    /// Secure Connection Mode: Only send to the stored host-ip
    /// </summary>
    Secure_Connection = 4,

    /// <summary>
    /// TCP Client Mode: TCP Connection with handshaking (Client Only)
    /// </summary>
    TCP_Client = 8
}

/// <summary>
/// Represents the encyrption mode to use.
/// </summary>
public enum WirelessEncryptionMode
{
    /// <summary>Open Authentication (No Passphrase required)</summary>
    Open = 0,
    /// <summary>128-bit Wired Equivalent Privacy (WEP)</summary>
    WEP_128 = 1,
    /// <summary>Wi-Fi Protected Access (WPA)</summary>
    WPA1 = 2,
    /// <summary>Mixed WPA1 &amp; WPA2-PSK</summary>
```

```
  MixedWPA1_WPA2 = 3,
  /// <summary>Wi-Fi Protected Access (WPA) II (uses preshared key)</summary>
  WPA2_PSK = 4
}

private enum RunMode
{
  Normal = 0,
  Update_Wait = 1, //Reserved for future use
  Update_Fail = 2, //Reserved for future use
  Update_Okay = 3, //Reserved for future use
  Boot = 4
}

#endregion

#region "Public Data Types"
/// <summary>
/// Is the device ready.
/// </summary>
public bool IsReady = false;

//strings
/// <summary>
/// The local IP of the device.
/// </summary>
public string LocalIP { get; protected set; }
/// <summary>
/// The local listening port of the device.
/// </summary>
public string LocalListenPort { get; protected set; }
#endregion

#region "Public Events"

//Delegates
/// <summary>
/// A delegate representing receipt of an HTTP request.
/// </summary>
/// <param name="request">The HTTP request.</param>
public delegate void HttpRequestReceivedHandler();
/// <summary>
/// A delegate representing data received.
/// </summary>
/// <param name="data">The data received.</param>
public delegate void DataReceivedHandler(string data);
/// <summary>
/// A delegate representing line received.
/// </summary>
/// <param name="line">The line received.</param>
public delegate void LineReceivedHandler(string line);
/// <summary>
/// A delegate representing connection opening.
/// </summary>
public delegate void ConnectionEstablishedHandler();
/// <summary>
/// A delegate representing connection closure.
/// </summary>
public delegate void ConnectionClosedHandler();
/// <summary>
///  A delegate representing connect to Wireless Router.
/// </summary>
public delegate void ConnectRouterHandler();
/// <summary>
/// A delegate representing connect to Wireless Router.
/// </summary>
public delegate void HttpBooleanHandler();

//Handlers
/// <summary>
/// Fired when a HTTP Request is received
/// </summary>
public event HttpRequestReceivedHandler HttpRequestReceived;

/// <summary>
/// Fired when any data is received
/// </summary>
public event DataReceivedHandler DataReceived;

/// <summary>
/// Fired when a complete line of data has been received
/// </summary>
public event LineReceivedHandler LineReceived;

/// <summary>
/// Fired when an connection has been establised to a
/// remote client.
/// </summary>
public event ConnectionEstablishedHandler ConnectionEstablished;

/// <summary>
/// Fired when an connection to a remote client
 /// has closed.
 /// </summary>
 public event ConnectionClosedHandler ConnectionClosed;

 /// <summary>
 /// Fired when an connection to a wireless router
 /// </summary>
 public event ConnectRouterHandler ConnectedRouter;

 /// <summary>
 /// Fired when an HttpBoolean Received.
 /// </summary>
 public event HttpBooleanHandler HttpBoolean;

 //Triggers

 /// <summary>
 /// Fired when an HTTP request is received.
 /// </summary>
 /// <param name="request">The request received.</param>
 protected virtual void OnHttpRequestReceived()
 {
   if (HttpRequestReceived != null)
   {
     //HttpStream stream = new HttpStream(request, _wifly);
     HttpRequestReceived();
   }
 }

 /// <summary>
 /// Fired when data is received.
 /// </summary>
 /// <param name="data">The line received.</param>
 protected virtual void OnDataReceived(string data)
 {
   if (DataReceived != null)
     DataReceived(data);
 }

 /// <summary>
 /// Fired when an a line is received.
 /// </summary>
 /// <param name="line">The HTTP request.</param>
 protected virtual void OnLineReceived(string line)
 {
   if (LineReceived != null)
     LineReceived(line);
 }

 /// <summary>
 /// Fired when the connection is opened.
 /// </summary>
 protected virtual void OnConnectionEstablished()
 {
   if (ConnectionEstablished != null)
     ConnectionEstablished();
 }

 /// <summary>
 /// Fired when the connection is closed.
 /// </summary>
 protected virtual void OnConnectionClosed()
 {
   if (ConnectionClosed != null)
     ConnectionClosed();
 }

 /// <summary>
 /// Fired when the connected a wireless router.
 /// </summary>
 protected virtual void OnConnectedRouter()
 {
   if (ConnectedRouter != null)
     ConnectedRouter();
 }
 #endregion

 #region "Construction And Initialization"
 /// <summary>
 /// Initialize Wi-Fi Http Client Command
 /// Commands: set ip proto 18 //enable http&tcp mode
 /// Commands: set ip host 0
 /// Commands: set dns name 0
 /// Commands: set comm remote 0
 /// Commands: set opt format 0
 /// Commands: set sys printlvl 2 //print only critical network access point connection level status.
 /// </summary>
 /// <returns></returns>
 public bool Initialize()
 {
   if (listen == null)
   {
     listen = new Thread(_Serial_Listen);
     listen.Start();
   }
   //_device_ready = false;

   //Reset the module
   Reset.Write(false);
   Thread.Sleep(100);
   Reset.Write(true);
   Thread.Sleep(250);

   return true;
 }

 public bool Reboot()
 {
   //Reset the module
   Reset.Write(false);
   Thread.Sleep(100);
   Reset.Write(true);
   Thread.Sleep(250);
   _stream = StreamMode.NoStream;
   Debug.Print("Reboot sending.");
   return true;
 }

 public bool SaveConf()
 {
   if (_Command_Mode_Start())
   {
     if (_Command_Mode_Save())
```

```
        {
            if (Reboot())
                return true;
            else
                return false;
        }
        else
            return false;
    }
    else
        return false;
}

public bool SoftReboot()
{
    if (_Command_Mode_Start())
    {
        _Command_Mode_Write("reboot\r\n");
        return true;
    }
    else
        return false;
}

public bool Ping()
{
    if (_Command_Mode_Start())
    {
        _Command_Mode_Write("ping www.google.co.uk 5\r\n");
        return true;
    }
    else
        return false;
}
#endregion

#region "Network Configuration"

/// <summary>
/// Attempt to join a wireless network with given parameters.
/// Function does not return until and IP address has been granted,
/// or a time-out has occured.
/// </summary>
```

```
    /// <param name="SSID"></param>
    /// <param name="Passphrase"></param>
    /// <param name="channel"></param>
    /// <param name="Authentication"></param>
    /// <returns>Whether or not it was successful</returns>
    public bool JoinWirelessNetwork(string SSID, string Passphrase, int channel = 0, WirelessEncryptionMode
Authentication = WirelessEncryptionMode.Open)
    {
      //Enter command mode
      if (!_Command_Mode_Start())
        return false;

      //Set DHCP off
      if (!_Command_Execute("set wlan ssid " + SSID))
        return false;

      //Set requested IP address
      if (!_Command_Execute("set wlan channel " + channel.ToString()))
        return false;

      //Set requested gateway
      if (!_Command_Execute("set wlan auth " + Authentication.ToString()))
        return false;

      //Set requested subnetmask
      if (!_Command_Execute("set wlan phrase " + Passphrase))
        return false;

      //Save the configuration settings to the config file
      if (!_Command_Execute("save"))
        return false;

      //Set requested DNS address
      if (!_Command_Execute("join"))
        return false;

      if (_DHCP)
      {
        if (!_Command_Execute("get ip"))
          return false;
      }

      //Exit command mode
    if (!_Command_Mode_Exit())
        return false;

    return true;
}

/// <summary>
/// Set Socket Protocol
/// </summary>
/// <param name="protocol">Desired Protocol</param>
/// <returns></returns>
public bool SetProtocol(SocketProtocol protocol)
{
    //Enter command mode
    if (!_Command_Mode_Start())
        return false;

    //Set the requested protocol
    if (!_Command_Execute("set ip protocol " + protocol.ToString()))
        return false;

    //Exit command mode
    if (!_Command_Mode_Exit())
        return false;

    return true;
}

/// <summary>
/// Set the device listen port to 80 and allow HTTP request parsing
/// </summary>
/// <returns></returns>
public bool EnableHttpServer()
{
    //Enter command mode
    if (!_Command_Mode_Start())
        return false;

    //Set the requested protocol
    if (!_Command_Execute("set ip local 80"))
        return false;

    //Exit command mode
    if (!_Command_Mode_Exit())
        return false;

    HttpEnabled = true;

    return true;
}

/// <summary>
/// Send data to the currently connected client
/// </summary>
/// <param name="data">Data</param>
public void Send(byte[] data)
{

    if (data.Length <= _wifly.BaudRate)
    {
        //Write-out directly
        _wifly.Write(data, 0, data.Length);
    }
    else
    {
        int baud_rate = _wifly.BaudRate;
        int iterations = data.Length / baud_rate;
        int exceeding = data.Length % baud_rate;
        int segment_start = 0;

        for (int i = 0; i < iterations; i++)
        {
            //Calculate segment range
            segment_start = i * baud_rate;

            //Write the current segment
            _wifly.Write(data, segment_start, baud_rate);

            //Write the remaining segment
            if (i == (iterations - 1) && exceeding > 0)
            {
                _wifly.Write(data, segment_start, exceeding);
            }
        }
    }
    return;
}

/// <summary>
/// Send data to the currently connected client
/// </summary>
/// <param name="data">Data</param>
public void Send(ref string data)
{
    //Convert string to byte array
    byte[] _data = System.Text.Encoding.UTF8.GetBytes(data);
    data = "";

    if (_data.Length <= _wifly.BaudRate)
    {
        //Write-out
        _wifly.Write(_data, 0, _data.Length);
    }
    else
    {
        int baud_rate = _wifly.BaudRate;
        int iterations = data.Length / baud_rate;
        int exceeding = data.Length % baud_rate;
        int segment_start = 0;

        for (int i = 0; i < iterations; i++)
        {
            //Calculate segment range
            segment_start = i * baud_rate;

            //Write the current segment
            _wifly.Write(_data, segment_start, baud_rate);

            //Write the remaining segment
            if (i == (iterations - 1) && exceeding > 0)
            {
                _wifly.Write(_data, segment_start, exceeding);
            }
        }
    }

    return;
}
```

```
/// <summary>
/// Send data to the currently connected client
/// </summary>
/// <param name="data"></param>
public void Send(string data)
{
  //Convert string to byte array
  byte[] _data = System.Text.Encoding.UTF8.GetBytes(data);

  if (_data.Length <= _wifly.BaudRate)
  {
    //Write-out
    _wifly.Write(_data, 0, _data.Length);
  }
  else
  {
    int baud_rate = _wifly.BaudRate;
    int iterations = data.Length / baud_rate;
    int exceeding = data.Length % baud_rate;
    int segment_start = 0;

    for (int i = 0; i < iterations; i++)
    {
      //Calculate segment range
      segment_start = i * baud_rate;

      //Write the current segment
      _wifly.Write(_data, segment_start, baud_rate);

      //Write the remaining segment
      if (i == (iterations - 1) && exceeding > 0)
      {
        _wifly.Write(_data, segment_start, exceeding);
      }
    }
  }
  data = "";
  return;
}

#endregion
```

```
#region "HTTP Parser"

private bool HttpEnabled = false;
private bool HttpStream = false;
private string HttpBuffer = "";
private int HttpBufferLength = 0;
//private HttpRequest current_request;
private bool bAwaitingPostData = false;
private bool bStartPostData = false;
private int ExitTimeOut = 4000;

private bool _Http_Boolean = false;
private int _Http_Int = -1;
private bool _Http_Int_Bool = false;

private bool _postMark_register = false;

private MemoryStream ms = new MemoryStream();
private XmlWriter xmlwriter;




/// <summary>
/// Login Post Function
/// </summary>
/// <param name="data">Post Data</param>
/// <returns>True: Login Suceess.</returns>
/// <returns>False: Faild Suceess.</returns>
public bool _Login(string data)
{

  _Http_Boolean = false;
  int timeCount = 0;
  while (!_Post(server_URL, server_Port, services_fun_Login, data))
  {
    if (timeCount++ > ExitTimeOut)
    {
      return false;
    }
  }
  timeCount = 0;
  while (!_Http_Boolean)
  {
        if (timeCount++ > ExitTimeOut)
        {
            return false;
        }
    }
    return true;
}

/// <summary>
/// Registration Post Function
/// </summary>
/// <param name="data">Post Data</param>
/// <returns>1: Registration Success.</returns>
/// <returns>2: Registration Faild.</returns>
/// <returns>3: Error.</returns>
/// <returns>4: Name alread exist.</returns>
/// <returns>5: Connection Error</returns>
public int _Register(string data)
{
    _Http_Boolean = false;
    _Http_Int = 0;
    _postMark_register = true;
    int timeCount = 0;
    while (!_Post(server_URL, server_Port, services_fun_Register, data))
    {
        if (timeCount++ > ExitTimeOut)
        {
            return 0;
        }
    }
    timeCount = 0;
    while (_Http_Int <= 0)
    {
        if (timeCount++ > ExitTimeOut)
        {
            return 0;
        }
    }
    return _Http_Int;
}

/// <summary>
/// SpO2 upload post function
```

```
    /// </summary>
    /// <param name="data">Post Data</param>
    /// <returns>True: upload success.</returns>
    /// <returns>False: upload Faild.</returns>
    public bool _Upload_Spo2(string data)
    {
        _Http_Boolean = false;
        if (!_Post(_wifly_settings.server_URL, _wifly_settings.server_Port, _wifly_settings.services_fun_uploadSpo2,
data))
        {
            int timeCount = 0;
            while (_Http_Int < 1)
            {
                if (timeCount++ > ExitTimeOut)
                {
                    return false;
                }
            }
            return _Http_Boolean;
        }
        else
        {
            return false;
        }
    }

    /// <summary>
    /// Blood Pressure upload post function
    /// </summary>
    /// <param name="data">Post Data</param>
    /// <returns>True: upload success.</returns>
    /// <returns>False: upload faild.</returns>
    public bool _Upload_BP(string data)
    {
        _Http_Boolean = false;
        if (!_Post(_wifly_settings.server_URL, _wifly_settings.server_Port, _wifly_settings.services_fun_uploadBP, data))
        {
            int timeCount = 0;
            while (_Http_Int < 1)
            {
                if (timeCount++ > ExitTimeOut)
                {
                    return false;
                }
            }
            return _Http_Boolean;
        }
        else
        {
            return false;
        }
    }

    /// <summary>
    /// Post Function to send Web Services data through HTTP post.
    /// </summary>
    /// <param name="serverURL">Server/Host URL</param>
    /// <param name="serverPort">Server Port</param>
    /// <param name="serverFunction">Services Function Name</param>
    /// <param name="data">Post Data</param>
    /// <returns>Ture: post success.</returns>
    /// <returns>False: post faild.</returns>
    private bool _Post(string serverURL, int serverPort, string serverFunction, string data)
    {
        _PostState = false;
        Debug.Print("<mark>");

        if (!_Command_Mode_Exit())
            return false;
        //return false;
        Thread.Sleep(500);

        if (!_Command_Mode_Start())
            return false;
        Thread.Sleep(500);

        Send("open " + serverURL + " " + serverPort + "\r\n");
        Thread.Sleep(500);

        Send("POST /" + serverFunction + " HTTP/1.1\n");
        Send("HOST: " + serverURL + "\n");
        Send("Content-Type: application/x-www-form-urlencoded\n");
        Send("Connection: close\n");
        Send("Content-Length: " + data.Length + "\r\n");
        Send("\r\n");
        Send(data + "\r\n");
```

```
        int timeCount = 0;
        while (!_PostState)
        {
            if (timeCount++ > _Timeout)
                return false;
            Thread.Sleep(1);
        }
        return true;
    }
    #endregion

    #region "Serial Port Communications Operation"
    /// <summary>
    /// Threaded listener for incomming data from serial port
    /// </summary>
    private void _Serial_Listen()
    {
        _Serial_Byte_Buffer = new byte[1024];
        while (true)
        {
            if (_wifly.BytesToRead > 0)
            {
                int i = _wifly.Read(_Serial_Byte_Buffer, 0, 1024);

                _Serial_Data_Received(_Serial_Byte_Buffer, i);
            }
            Thread.Sleep(50);
        }
    }

    //Data received
    private void _Serial_Data_Received(byte[] data, int size)
    {
        //Convert bytes into an indexable string
        string line = new string(System.Text.Encoding.UTF8.GetChars(data, 0, size));
        //Debug.Print("Line: " + line);

        if (line == null)
        {
            return;
        }
```

```
            _Serial_String_Buffer += line;

            //Handle Connection open request tags
            if (_Serial_String_Buffer.IndexOf("*OPEN*") >= 0)
            {
                _Serial_String_Buffer = StringReplace("*OPEN*", "", _Serial_String_Buffer);
                _stream = StreamMode.NoStream;
                //_postStat = true;
                OnConnectionEstablished();
                Debug.Print("Open");
            }

            //Handle Connection close request tags
            else if (_Serial_String_Buffer.IndexOf("*CLOS*") >= 0)
            {
                _Serial_String_Buffer = StringReplace("*CLOS*", "", _Serial_String_Buffer);
                _Serial_String_Buffer += "\r\n";
                _stream = StreamMode.NoStream;
                OnConnectionClosed();
                Debug.Print("Close" + ">>>" + _stream);
            }

            if (_Serial_String_Buffer.IndexOf("\r\n") >= 0)
            {
                int index = -1;
                while ((index = _Serial_String_Buffer.IndexOf("\r\n")) >= 0)
                {
                    string new_line = _Serial_String_Buffer.Substring(0, index);
                    _Serial_String_Buffer = _Serial_String_Buffer.Substring(index + 2);
                    new_line = new_line + "\r\n";
                    _Serial_Line_Received(new_line);
                }
            }

            ////End of post-data
            //if (current_request != null && bAwaitingPostData && _Serial_String_Buffer.Length >=
current_request.PostLength)
            //{
            //    _Serial_Line_Received(_Serial_String_Buffer);
            //}
            //Debug.Print("Rec: " + line);
            OnDataReceived(line);
        }
```

```
//Data received complete
private void _Serial_Line_Received(string line)
{
    Debug.Print("Wi-Fi Line: " + line);

    if (line.IndexOf("*READY*") >= 0)
        _device_ready = true;

    //Are we entering command mode?
    if (line.Length >= 3)
    {
        if (_stream != StreamMode.CommandStream && line.Substring(0, 3) == "CMD")
            _stream = StreamMode.CommandStream;
    }

    //Are we in command mode waiting for response?
    if (_stream == StreamMode.CommandStream)
    {
        //Append line to the response
        _Command_Mode_Response += line;

        //Are we leaving command mode?
        if (_Command_Mode_Response.IndexOf("EXIT") >= 0)
        {
            _stream = StreamMode.NoStream;

            //Report data to user-event
            if (_data_level == DataReturnLevel.ReturnAll)
                OnLineReceived(line);
            //This is all we need to do with this event
            return;
        }

        //Are we updating the firmware?
        if (_Command_Mode_Response.IndexOf("UPDATE OK") >= 0)
        {
            _Command_Mode_Response_Okay = true;
            _Command_Mode_Response_Complete = true;
        }

        //Are we updating the firmware?
        if (_Command_Mode_Response.IndexOf("Set Factory Defaults") >= 0)
```

```
{
    _Command_Mode_Response_Okay = true;
    _Command_Mode_Response_Complete = true;
}

//Did the command execute without error?
if (_Command_Mode_Response.IndexOf("AOK") >= 0)
{
    _Command_Mode_Response_Okay = true;
    _Command_Mode_Response_Complete = true;
}
//Did the command execute with an error?
else if (_Command_Mode_Response.IndexOf("ERR") >= 0)
{
    //_Print_Debug("ERROR! --- " + line);
    _Command_Mode_Exit();
    _Command_Mode_Response_Okay = false;
    _Command_Mode_Response_Complete = true;
    Reboot();
    return;
}

else if (_Command_Mode_Response.IndexOf("save my_config") > 0)
{
    _Command_Mode_Response_Okay = true;
    _Command_Mode_Response_Complete = true;
    return;
}

if (!_DHCP && _Command_Mode_Response.IndexOf("Associated!") >= 0)
{
    _Command_Mode_Response_Okay = true;
    _Command_Mode_Response_Complete = true;
    Thread.Sleep(1000);
    OnConnectedRouter();
}

if (!_DHCP && _Command_Mode_Response.IndexOf("*Reboot*") >= 0)
{
    _Command_Mode_Response_Okay = true;
    _Command_Mode_Response_Complete = true;
    _stream = StreamMode.NoStream;
}
```

```
    if (line.Length >= 3)
    {
        if (line.Substring(0, 3) == "IP=")
        {
            string line_buffer = line.Substring(3);
            string[] split_ip = line_buffer.Split(new char[] { ':' });
            LocalIP = split_ip[0];

            _Command_Mode_Response_Okay = true;
            _Command_Mode_Response_Complete = true;
        }

        //Check to see if the appropriate firmware is loaded
        if (line.Substring(0, 5) == "File=")
        {
            string line_buffer = line.Substring(5);
            string[] split_ip = line_buffer.Split(new char[] { ':' });
            LocalIP = split_ip[0];

            _Command_Mode_Response_Okay = true;
            _Command_Mode_Response_Complete = true;
        }
    }

    //Report data to user-event
    if (_data_level == DataReturnLevel.ReturnAll)
        OnLineReceived(line);
}

else
{
    if (!_DHCP & line.IndexOf("Associated!") >= 0)
    {
        Thread.Sleep(1000);
        OnConnectedRouter();
    }
    if (line.IndexOf("HTTP/1.1 200 OK") >= 0)
    {
        _stream = StreamMode.NoStream;
        _PostState = true;
        Debug.Print("Posted Successful.");
    }
```

```
        if (line.IndexOf("Content-Length:") >= 0)
        {
            string length = line.Substring(16);
            HttpBufferLength = Int32.Parse(length.TrimEnd());
            Debug.Print("Length: " + length + ">>>" + HttpBufferLength.ToString());
            bAwaitingPostData = true;
        }
        if (line.IndexOf("<?xml version=\"1.0\" encoding=\"utf-8\"?>") >= 0)
        {
            ms = new MemoryStream();
            xmlwriter = XmlWriter.Create(ms);
            xmlwriter.WriteProcessingInstruction("xml", "version=\"1.0\" encoding=\"utf-8\"");
            bStartPostData = true;
            return;
        }
        if (bAwaitingPostData && bStartPostData)
        {
            HttpBuffer += line;
            xmlwriter.WriteRaw(line.TrimEnd());
            if (HttpBuffer.Length >= HttpBufferLength - 38)
            {
                xmlwriter.Flush();
                xmlwriter.Close();
                processXML(ms);
                HttpBuffer = "";
                bAwaitingPostData = false;
                bStartPostData = false;
            }
        }
    }
}

public void processXML(MemoryStream xmlms)
{
    //////// display the XML data //////////
    byte[] byteArray = xmlms.ToArray();
    char[] cc = System.Text.UTF8Encoding.UTF8.GetChars(byteArray);
    string str = new string(cc);
    xmlms.Dispose();

    //////////read xml
    MemoryStream rms = new MemoryStream(byteArray);
    XmlReaderSettings ss = new XmlReaderSettings();
    ss.IgnoreWhitespace = true;
    ss.IgnoreComments = false;
    //XmlException.XmlExceptionErrorCode.
    XmlReader xmlr = XmlReader.Create(rms, ss);
    while (!xmlr.EOF)
    {
        xmlr.Read();
        if (xmlr.NodeType == XmlNodeType.Element)
        {
            if (xmlr.Name == "boolean")
            {
                xmlr.Read();
                if (xmlr.Value == "true")
                {
                    _Http_Boolean = true;
                    Debug.Print("POST successful.");
                }
                else if (xmlr.Value == "false")
                {
                    _Http_Boolean = false;
                }
            }
            if (xmlr.Name == "string" && _postMark_register)
            {
                xmlr.Read();
                _Http_Int = Int32.Parse(xmlr.Value);
                _postMark_register = false;
                Debug.Print("GET HTTP: " + _Http_Int.ToString() + ".");
            }
        }
    }
}

#endregion

#region "Logical Tools"
//This is needed because String.IndexOf is returning false randomly
int IndexIn(string needle, string haystack)
{
    int found_index = -1;
    int needle_length = needle.Length;

    for (int i = 0; i < haystack.Length; i++)
    {
        if ((i + needle_length) < haystack.Length)
        {
            if (haystack.Substring(i, needle_length) == needle)
            {
                found_index = i;
                break;
            }
        }
        else
        {
            break;
        }
    }

    return found_index;
}

//String replace method
string StringReplace(string token, string text, string haystack)
{
    string left = "";
    string right = "";
    string buffer = haystack;
    int index = buffer.IndexOf(token);

    while (index >= 0)
    {
        int other_index = index + token.Length;

        left = buffer.Substring(0, index);
        right = buffer.Substring(other_index);
        buffer = left + text + right;

        index = buffer.IndexOf(token);
    }

    return buffer;
}

//Qt style StartsWith function with PHP string function syntax
private bool StartsWith(string needle, string haystack)
{
    //Avoid Out of range exceptions
    if (needle.Length > haystack.Length)
        return false;

    //Grab the beginning of the string
    string buffer = haystack.Substring(0, needle.Length);

    //Does the substring match the needle?
    return buffer == needle ? true : false;
}

//Exception-less string to int method
private bool StringToInt(string str, out int integer)
{
    int total = 0;
    bool bOkay = true;

    for (int i = 0; i < str.Length; i++)
    {
        uint temp = str[i];
        temp = temp - 48;

        if (temp > 9 || temp < 0)
        {
            bOkay = false;
            break;
        }

        total = total * 10;
        total = total + (int)temp;
    }

    integer = bOkay ? total : 0;

    return bOkay;
}

#endregion

#region "Command Mode"

//Attempt to execute a command
/// <summary>
```

```
/// Executes the command.
/// </summary>
/// <param name="Command">The command to execute.</param>
/// <returns>Whether or not it was successful</returns>
public bool _Command_Execute(string Command)
{
  //Append the return
  if (Command.IndexOf("\r") < 0)
    Command += "\r";

  _Command_Mode_Write(Command);

  Thread.Sleep(10);

  //Wait until the device has responded to the last command
  int tempCount = 0;
  while (!_Command_Mode_Response_Complete)
  {
    tempCount++;
    if (tempCount > ExitTimeOut)
      return false;
    Thread.Sleep(1);
  }

  return _Command_Mode_Response_Okay;
}

//Initiate command mode
/// <summary>
/// Starts command mode.
/// </summary>
/// <returns>Whether or not it was successful</returns>
public bool _Command_Mode_Start()
{
  _Command_Mode_Write(_Command_Init);

  Thread.Sleep(300);
  int tempCount = 0;
  while (_stream != StreamMode.CommandStream)
  {
    tempCount++;
    Debug.Print("wifi start()>>>>>>>>" + tempCount);
    if (tempCount == ExitTimeOut)
```

```
            return false;
        Thread.Sleep(1);
    }

    return true;
}

//Exit command mode
/// <summary>
/// Exits command mode.
/// </summary>
/// <returns>Whether or not it was successful</returns>
public bool _Command_Mode_Exit()
{
    _Command_Mode_Write("exit\r");

    //Wait until we are out of command mode
    _TimeOutDate = DateTime.Now.AddMilliseconds((double)_Timeout);
    int tempCount = 0;
    while (_stream == StreamMode.CommandStream)
    {
        tempCount++;
        Debug.Print("wifi exit()>>>>>" + tempCount);
        if (tempCount == ExitTimeOut)
            return false;
        Thread.Sleep(1);
    }

    return true;
}

public bool _Command_Mode_Save()
{
    _Command_Mode_Write(_Command_Init);

    Thread.Sleep(300);
    int tempCount = 0;
    while (_stream != StreamMode.CommandStream)
    {
        tempCount++;
        if (tempCount == ExitTimeOut)
            return false;
        Thread.Sleep(1);
    }

    return true;
}

//Command write string method
/// <summary>
/// Write to the device.
/// </summary>
/// <param name="Command">The command to write.</param>
/// <returns>Whether or not it was successful</returns>
public void _Command_Mode_Write(string Command)
{
    //Await response
    _Command_Mode_Response = "";
    _Command_Mode_Response_Complete = false;

    //Convert string to byte array
    byte[] _Command = System.Text.Encoding.UTF8.GetBytes(Command);
    _wifly.Write(_Command, 0, _Command.Length);

    return;
}

//Command write bytes method
/// <summary>
/// Write to the device.
/// </summary>
/// <param name="Command">The command to write.</param>
/// <returns>Whether or not it was successful</returns>
public void _Command_Mode_Write(byte[] Command)
{
    //Await response
    _Command_Mode_Response = "";
    _Command_Mode_Response_Complete = false;

    _wifly.Write(Command, 0, Command.Length);
    return;
}

//Command get method
private void _Command_Mode_Get(string Command)
{
    //Enter command mode
    if (!_Command_Mode_Start())
        return;

    //Get the requested configuration
    if (!_Command_Execute("get " + Command))
        if (!_Command_Execute("get " + Command))
            return;

    //Exit command mode
    if (!_Command_Mode_Exit())
        return;
}

#endregion

#region "Timeout"
/// <summary>
/// Set the timeout for module communications
/// </summary>
/// <param name="Timeout">The amount of time in ms before timeout occurs</param>
public void SetTimeout(int Timeout = 3000)
{
    _Timeout = Timeout;
}
#endregion
 }
```

Appendix III. Zigbee Driver Code – C#

```
public class Zigbee
  {
     #region Define Parameters.
     /// <summary>
     /// Define the Zigbee Com Ports.
     /// </summary>
     private static SerialPort comXbee;

     /// <summary>
     /// Define String
     /// </summary>
     private static string _str_SCAN = "*SCAN*";
     private static string _str_TEST = "*TEST*";
     private static string _str_WSN = "*WSN*";
     private static string _str_DATA = "*DATA*";
     private static string _str_SPO2 = "*SPO2*";
     private static string _str_BP = "*BP*";
     private static string _str_ECG = "*ECG*";
     private static string _str_CONNECT = "*CONNECT*";
     private static string _str_OK = "*OK*";
     private static string _str_DISCONNECT = "*DISCONNECT*";

     private static bool _COMMAND_RESPONSE_OK = false;

     public bool _sensor_SPO2 = false;
     public bool _sensor_BP = false;

     public bool _ready_SPO2 = false;
     public bool _ready_BP = false;

     private int Timeout = 5000;
     private int Timeout_Test = 2000;

     /// <summary>
     /// Define the measurement data
     /// </summary>
     public int lastSpo2 = 0;
     public int lastHR = 0;
     public int lastSBP = 0;
     public int lastDBP = 0;
     public int lastABP = 0;
     public int lastPR = 0;
     #endregion

     #region Public Events
     //Delegate
     /// <summary>
     /// A delegate representing receipt of SPO2.
     /// </summary>
     public delegate void Scan_Sensor_SPO2_Handler();
     /// <summary>
     /// A delegate representing receipt of BP.
     /// </summary>
     public delegate void Scan_Sensor_BP_Handler();
     /// <summary>
     /// A delegate representing receipt of SPO2.
     /// </summary>
     public delegate void Sensor_SPO2_Handler();
     /// <summary>
     /// A delegate representing receipt of BP.
     /// </summary>
     public delegate void Sensor_BP_Handler();
     /// <summary>
     /// A delegate representing receipt of SPO2.
     /// </summary>
     public delegate void DataReceive_SPO2_Handler();
     /// <summary>
     /// A delegate representing receipt of BP.
     /// </summary>
     public delegate void DataReceive_BP_Handler();

     //Handlers
     /// <summary>
     /// Fired when a SPO2 data received
     /// </summary>
     public event Scan_Sensor_SPO2_Handler Scan_Sensor_SPO2;
     /// <summary>
     /// Fired when a SPO2 data received
     /// </summary>
     public event Scan_Sensor_BP_Handler Scan_Sensor_BP;
     /// <summary>
     /// Fired when a SPO2 data received
     /// </summary>
```

```
public event Sensor_SPO2_Handler Sensor_SPO2;
/// <summary>
/// Fired when a SPO2 data received
/// </summary>
public event Sensor_BP_Handler Sensor_BP;
/// <summary>
/// Fired when a SPO2 data received
/// </summary>
public event DataReceive_SPO2_Handler DataReceive_SPO2;
/// <summary>
/// Fired when a BP data received.
/// </summary>
public event DataReceive_BP_Handler DataReceive_BP;

//Triggers
/// <summary>
/// Fired when a spo2 DATA come
/// </summary>
protected virtual void OnScanSensor_SPO2()
{
  if (Scan_Sensor_SPO2 != null)
  {
    Scan_Sensor_SPO2();
  }
}
/// <summary>
/// Fired when a bp data come
/// </summary>
protected virtual void OnScanSensor_BP()
{
  if (Scan_Sensor_BP != null)
  {
    Scan_Sensor_BP();
  }
}
/// <summary>
/// Fired when a bp data come
/// </summary>
protected virtual void OnSensor_SPO2()
{
  if (Sensor_SPO2 != null)
  {
    Sensor_SPO2();
    }
}
/// <summary>
/// Fired when a bp data come
/// </summary>
protected virtual void OnSensor_BP()
{
    if (Sensor_BP != null)
    {
        Sensor_BP();
    }
}
/// <summary>
/// Fired when a spo2 DATA come
/// </summary>
protected virtual void OnDataReceived_SPO2()
{
    if (DataReceive_SPO2 != null)
    {
        DataReceive_SPO2();
    }
}
/// <summary>
/// Fired when a bp data come
/// </summary>
protected virtual void OnDataReceived_BP()
{
    if (DataReceive_BP != null)
    {
        DataReceive_BP();
    }
}
#endregion

public Zigbee(int Com, int baudRate)
{
    comXbee = new SerialPort("COM"+Com, baudRate, Parity.None, 8, StopBits.One);
    comXbee.DataReceived += new SerialDataReceivedEventHandler(comXbee_DataReceived);
    try
    {
        comXbee.Open();
        Debug.Print("XBee Com Open...");
    }
    catch (Exception err)
    {
        Debug.Print("ERROR: " + err.Message.ToString());
    }
}

#region Serial Port Operations
/// <summary>
/// Serial Received Data
/// </summary>
void comXbee_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    Thread.Sleep(500);
    int bytesToRead = comXbee.BytesToRead;
    byte[] byteArray = new byte[bytesToRead];
    comXbee.Read(byteArray, 0, bytesToRead);
    string response = new String(UTF8Encoding.UTF8.GetChars(byteArray));

    if (response.IndexOf("\r\n") >= 0)
    {
        int index = -1;
        while ((index = response.IndexOf("\r\n")) >= 0)
        {
            string new_line = response.Substring(0, index);
            response = response.Substring(index + 2);
            new_line = new_line + "\r\n";
            _Serial_Line_Received(new_line);
            Debug.Print(new_line);
        }
    }
}

/// <summary>
/// Serial Line Data Operation
/// </summary>
/// <param name="Command">Received Line Data.</param>
void _Serial_Line_Received(string line)
{
    if (line.IndexOf(_str_OK) > -1)
    {
        _COMMAND_RESPONSE_OK = true;
        string device = line.Substring(_str_OK.Length);
        switch (device)
        {
            case "*SPO2*": _ready_SPO2 = true; OnSensor_SPO2();
                break;
            case "*BP*": _ready_BP = true; OnSensor_BP();
                break;
        }
    }

    if (line.IndexOf(_str_WSN) > -1)
    {
        line = line.Substring(_str_WSN.Length);
        switch (line)
        {
            case "*SPO2*": _sensor_SPO2 = true; OnScanSensor_SPO2();
                break;
            case "*BP*": _sensor_BP = true; OnScanSensor_BP();
                break;
        }
    }

    if (line.IndexOf(_str_DISCONNECT) > -1)
    {
        line = line.Substring(_str_DISCONNECT.Length);
        switch (line)
        {
            case "*SPO2*": _ready_SPO2 = false;
                break;
            case "*BP*": _ready_BP = false;
                break;
        }
    }

    if (line.IndexOf(_str_DATA) > -1)
    {
        line = line.Substring(_str_DATA.Length);
        if (line.IndexOf(_str_SPO2) > -1)
        {
            line = line.Substring(_str_SPO2.Length);
            string[] data = line.Split(new char[] { '|' });
            if (data.Length > 0)
            {
                int length = Int32.Parse(data[0]);
                if (length == data.Length - 1 && length == 2)
```

```
                {
                    lastSpo2 = Int32.Parse(data[1]);
                    lastPR = Int32.Parse(data[2]);
                    OnDataReceived_SPO2();
                }
            }
        }
        else if (line.IndexOf(_str_BP) > -1)
        {
            line = line.Substring(_str_BP.Length);
            string[] data = line.Split(new char[] { '|' });
            if (data.Length > 0)
            {
                int length = Int32.Parse(data[0]);
                if (length == data.Length - 1 && length == 4)
                {
                    lastSBP = Int32.Parse(data[1]);
                    lastDBP = Int32.Parse(data[2]);
                    lastABP = Int32.Parse(data[3]);
                    lastHR = Int32.Parse(data[4]);
                    OnDataReceived_BP();
                }
            }
        }
    }
}

/// <summary>
/// Write to the device.
/// </summary>
/// <param name="Command">The command to write.</param>
public void _Command_Mode_Write(string Command)
{
    //Convert string to byte array
    byte[] _Command = System.Text.Encoding.UTF8.GetBytes(Command);
    comXbee.Write(_Command, 0, _Command.Length);
}
#endregion

#region Functions of Zigbee
/// <summary>
/// Scan wireless sensor
/// </summary>
/// <returns></returns>
public bool _SCAN()
{
    _Command_Mode_Write(_str_SCAN);

    int tempCount = 0;
    while (tempCount< Timeout)
    {
        //scan receive.
        Thread.Sleep(1);
    }
    return true;
}

/// <summary>
/// Connect to the Wireless Sensor Node
/// </summary>
/// <returns></returns>
public bool _CONNECT()
{
    _COMMAND_RESPONSE_OK = false;
    _Command_Mode_Write(_str_CONNECT);
    int tempCount = 0;
    while (tempCount < Timeout)
    {
        //scan receive.
        if (_COMMAND_RESPONSE_OK)
        {
            return true;
        }
        Thread.Sleep(1);
    }
    return true;
}

/// <summary>
/// Measurement Function
/// </summary>
/// <returns></returns>
public bool _TEST()
{
    _Command_Mode_Write(_str_TEST);
    int tempCount = 0;
```

```
        while (tempCount < Timeout_Test)
        {
            Thread.Sleep(1);
        }
        return true;
    }
    #endregion
     }
```

Appendix IV. Enhanced RC5 based on 1-D Logistic Map Code – C#

```
class RC5_Logistic
  {
    uint[] s;        //S array
    //uint[] temp_s;    //temp S that changed in each block.
    uint[] l;        //L array
    uint b, u, t, c;   //b for key length; u = w/8; w=32-> u=4; t is the max (c or u)
    byte[] key;      //for key array
    int rounds;      //number of round

    /// <summary>
    /// Chaos algorithem X[n+1] = U*X[n]*(1-X[n]);
    /// </summary>
    double chaosU;    //Chaos parameters u
    int chaosN;      //Chaos parameters n

    public RC5()
    {
      string str = "abcdefghijklmnop";
      key = GetKeyFromString(str);
      rounds = 16;
      b = (uint)key.Length;
      u = 4;
      t = (uint)(34);
      c = 12 / u;
      s = new uint[34];
      l = new uint[12];

      GenerateKey(key, rounds);

      ///Setup Chaos Parameters
      chaosU = 3.7;
      chaosN = 10;
    }

    public RC5(string password, int round)
    {
      key = GetKeyFromString(password);
      rounds = round;
      b = (uint)key.Length;
      u = 4;
```

```
            t = (uint)(2 * rounds + 2);
            c = Math.Max(b, 1) / u;
            s = new uint[2 * rounds + 2];
            l = new uint[key.Length];
            GenerateKey(key, rounds);

            ///Setup Chaos Parameters
            chaosU = 3.7;
            chaosN = 10;
        }

    public RC5(byte[] password, int round)
    {
      rounds = round;
      key = password;
      b = (uint)password.Length;
      u = 4;
      t = (uint)(2 * rounds + 2);
      c = Math.Max(b, 1) / u;

      s = new uint[2 * rounds + 2];
      l = new uint[password.Length];

            GenerateKey(key, rounds);

            ///Setup Chaos Parameters
            chaosU = 3.7;
            chaosN = 10;
        }
        //to circulate int left
        private uint leftRotate(uint x, int offset)
        {
            uint t1, t2;
            t1 = x >> (32 - offset);
            t2 = x << offset;
            return t1 | t2;
        }
        //to circulate int right
        private uint RightRotate(uint x, int offset)
        {
            uint t1, t2;
            t1 = x << (32 - offset);
            t2 = x >> offset;
            return t1 | t2;
        }
        //encryption operation on two block
        private void Encode(ref uint r1, ref uint r2, int rounds)
        {
            r1 = r1 + s[0];
            r2 = r2 + s[1];
            for (int i = 1; i <= rounds; i++)
            {
                r1 = leftRotate(r1 ^ r2, (int)r2) + s[2 * i];
                r2 = leftRotate(r2 ^ r1, (int)r1) + s[2 * i + 1];
            }
        }
        //decryption operation on two block
        private void Decode(ref uint r1, ref uint r2, int rounds)
        {
            for (int i = rounds; i >= 1; i--)
            {
                r2 = (RightRotate(r2 - s[2 * i + 1], (int)r1)) ^ r1;
                r1 = (RightRotate(r1 - s[2 * i], (int)r2)) ^ r2;
            }
            r2 = r2 - s[1];
            r1 = r1 - s[0];
        }

        /// <summary>
        /// Encode two blocks by RC5 with Chaos algorithm
        /// </summary>
        /// <param name="r1"></param>
        /// <param name="r2"></param>
        /// <param name="rounds"></param>
        private void EncodeWithChaos(ref uint r1, ref uint r2, int rounds)
        {
            double temp = r2;

            temp = (temp % 1000) / 1000;
            temp = logistic(chaosU, temp, chaosN);
            s[0] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ s[0];
            r1 = r1 + s[0];

            temp = r1;
            temp = (temp % 1000) / 1000;
            temp = logistic(chaosU, temp, chaosN);
            s[1] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ s[1];

            r2 = r2 + s[1];

            for (int i = 1; i <= rounds; i++)
            {
                temp = r2;
                temp = (temp % 1000) / 1000;
                temp = logistic(chaosU, temp, chaosN);
                s[2 * i] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ s[2 * i];

                r1 = leftRotate(r1 ^ r2, (int)r2) + s[2 * i];

                temp = r1;
                temp = (temp % 1000) / 1000;
                temp = logistic(chaosU, temp, chaosN);
                s[2 * i + 1] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ s[2 * i + 1];

                r2 = leftRotate(r2 ^ r1, (int)r1) + s[2 * i + 1];

            }
        }
        /// <summary>
        /// Decode two blocks by RC5 with Chaos algorithm
        /// </summary>
        /// <param name="r1"></param>
        /// <param name="r2"></param>
        /// <param name="rounds"></param>
        private void DecodeWithChaos(ref uint r1, ref uint r2, int rounds)
        {
            double temp;
            for (int i = rounds; i >= 1; i--)
            {
                temp = r1;
                temp = (temp % 1000) / 1000;
                temp = logistic(chaosU, temp, chaosN);
                s[2 * i + 1] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ s[2 * i + 1];
                r2 = (RightRotate(r2 - s[2 * i + 1], (int)r1)) ^ r1;

                temp = r2;
                temp = (temp % 1000) / 1000;
                temp = logistic(chaosU, temp, chaosN);
                s[2 * i] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ s[2 * i];
                r1 = (RightRotate(r1 - s[2 * i], (int)r2)) ^ r2;
            }
            temp = r1;
            temp = (temp % 1000) / 1000;
            temp = logistic(chaosU, temp, chaosN);
            s[1] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ s[1];
            r2 = r2 - s[1];

            temp = r2;
            temp = (temp % 1000) / 1000;
            temp = logistic(chaosU, temp, chaosN);
            s[0] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ s[0];
            r1 = r1 - s[0];
        }

        /// <summary>
        /// Encode two blocks by RC5 with Chaos algorithm
        /// </summary>
        /// <param name="r1"></param>
        /// <param name="r2"></param>
        /// <param name="rounds"></param>
        private void EncodeWithChaos_block(ref uint r1, ref uint r2, int rounds)
        {
            double temp = r2;
            uint[] temp_s = new uint[s.Length];

            for (int index = 0; index < s.Length; index++)
            {
                temp_s[index] = s[index];
            }

            temp = (temp % 1000) / 1000;
            temp = logistic(chaosU, temp, chaosN);
            temp_s[0] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ temp_s[0];
            r1 = r1 + temp_s[0];

            temp = r1;
            temp = (temp % 1000) / 1000;
            temp = logistic(chaosU, temp, chaosN);
            temp_s[1] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ temp_s[1];

            r2 = r2 + temp_s[1];
            for (int i = 1; i <= rounds; i++)
            {
                temp = r2;
                temp = (temp % 1000) / 1000;
                temp = logistic(chaosU, temp, chaosN);
                temp_s[2 * i] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ temp_s[2 * i];

                r1 = leftRotate(r1 ^ r2, (int)r2) + temp_s[2 * i];

                temp = r1;
                temp = (temp % 1000) / 1000;
                temp = logistic(chaosU, temp, chaosN);
                temp_s[2 * i + 1] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ temp_s[2 * i + 1];

                r2 = leftRotate(r2 ^ r1, (int)r1) + temp_s[2 * i + 1];

            }
            //foreach(uint xx in s)
            //    //Console.Write(xx.ToString());
            ////Console.WriteLine();
        }
        /// <summary>
        /// Decode two blocks by RC5 with Chaos algorithm
        /// </summary>
        /// <param name="r1"></param>
        /// <param name="r2"></param>
        /// <param name="rounds"></param>
        private void DecodeWithChaos_block(ref uint r1, ref uint r2, int rounds)
        {
            double temp;

            uint[] temp_s = new uint[s.Length];

            for (int index = 0; index < s.Length; index++)
            {
                temp_s[index] = s[index];
            }

            for (int i = rounds; i >= 1; i--)
            {
                temp = r1;
                temp = (temp % 1000) / 1000;
                temp = logistic(chaosU, temp, chaosN);
                temp_s[2 * i + 1] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ temp_s[2 * i + 1];
                r2 = (RightRotate(r2 - temp_s[2 * i + 1], (int)r1)) ^ r1;

                temp = r2;
                temp = (temp % 1000) / 1000;
                temp = logistic(chaosU, temp, chaosN);
                temp_s[2 * i] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ temp_s[2 * i];
                r1 = (RightRotate(r1 - temp_s[2 * i], (int)r2)) ^ r2;
            }
            temp = r1;
            temp = (temp % 1000) / 1000;
            temp = logistic(chaosU, temp, chaosN);
            temp_s[1] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ temp_s[1];
            r2 = r2 - temp_s[1];

            temp = r2;
            temp = (temp % 1000) / 1000;
            temp = logistic(chaosU, temp, chaosN);
            temp_s[0] = Convert.ToUInt32(Math.Floor(temp * 1000)) % 256 ^ temp_s[0];
            r1 = r1 - temp_s[0];

        }

        private void GenerateKey(byte[] key, int rounds)
        {
            uint P32 = uint.Parse("b7e15163", System.Globalization.NumberStyles.HexNumber);
            uint Q32 = uint.Parse("9e3779b9", System.Globalization.NumberStyles.HexNumber);

            int tempB = (int)b;
            int tempC= (int)(Math.Max(b, 1) / u);
            for(int i=tempB-1; i!=-1; i--)
            {
                l[tempC - 1] = 0;
                l[i/u] = (l[i/u]<<8)+key[i];
            }

            s[0] = P32;
            for (int i = 1; i <= t - 1; i++)
            {
                s[i] = s[i - 1] + Q32;
            }

            uint ii, jj;
            ii = jj = 0;
            uint x, y;
            x = y = 0;
            uint v = 3 * Math.Max(t, c);       //(26,8)

            for (int counter = 0; counter <= v; counter++)
            {
                x = s[ii] = leftRotate((s[ii] + x + y), 3);
                y = l[jj] = leftRotate((l[jj] + x + y), (int)(x + y));
                ii = (ii + 1) % t;           //2R+2=26
                jj = (jj + 1) % c;           //8
            }
            //Console.WriteLine("s[] " + s.Length+">>>>>>l[] "+l.Length);
        }

        //convert key from string to byte array
        private byte[] GetKeyFromString(string str)
        {
            //Console.WriteLine("Key: " + str);
            char[] mykeyinchar = str.ToCharArray();
            byte[] mykeyinbytes = new byte[mykeyinchar.Length];
            //Console.WriteLine("Key in char Length: " + mykeyinchar.Length);
            for (int i = 0; i < mykeyinchar.Length; i++)
            {
                mykeyinbytes[i] = (byte)mykeyinchar[i];
                //Console.WriteLine("key in byte: " + mykeyinbytes[i]);
            }
            //Console.WriteLine("Key in byte Length: " + mykeyinbytes.Length);
            return mykeyinbytes;
        }

        /// <summary>
        /// Stand RC5 encrypt
        /// </summary>
        /// <param name="streamreader"></param>
        /// <param name="streamwriter"></param>
        public void Encrypt(FileStream streamreader, FileStream streamwriter)
        {
            uint r1, r2;
            System.IO.BinaryReader br = new System.IO.BinaryReader(streamreader);
            System.IO.BinaryWriter bw = new System.IO.BinaryWriter(streamwriter);
            long filelength = streamreader.Length;
            //Console.WriteLine("File Length: " + filelength);
            while (filelength > 0)
            {
                try
                {
                    r1 = br.ReadUInt32();
                    try
                    {
                        r2 = br.ReadUInt32();
                    }
                    catch
                    {
                        r2 = 0;
                    }
                }
                catch
                {
                    r1 = r2 = 0;
                }
                ////Console.WriteLine("Length: "+r1.ToString());
                Encode(ref r1, ref r2, rounds);
                bw.Write(r1);
                bw.Write(r2);
                filelength -= 8;
            }
            streamreader.Close();
            streamwriter.Close();
        }

        /// <summary>
        /// Stand RC5 decrypt
        /// </summary>
        /// <param name="streamreader"></param>
        /// <param name="streamwriter"></param>
        public void Decrypt(FileStream streamreader, FileStream streamwriter)
        {
            uint r1, r2;

            System.IO.BinaryReader br = new System.IO.BinaryReader(streamreader);
            System.IO.BinaryWriter bw = new System.IO.BinaryWriter(streamwriter);
            long filelength = streamreader.Length;

            while (filelength > 0)
            {
                try
                {
                    r1 = br.ReadUInt32();
                    r2 = br.ReadUInt32();
                    Decode(ref r1, ref r2, rounds);

                    if (!(r1 == 0 && r2 == 0 && (filelength - 8 <= 0)))
                    {
                        bw.Write(r1);
                        bw.Write(r2);
                    }
                    if (r2 == 0 && (filelength - 8 <= 0))
                    {
                        bw.Write(r1);
                    }
                    filelength -= 8;
                }
                catch
                {

                    //Console.WriteLine("May be U try to decrypt an normal file (plain file)" + "Error");
                    return;
                }
            }

            streamreader.Close();
            streamwriter.Close();
        }

        public byte[] EncryptString(byte[] input, uint[] S)
        {
            byte[] output = new byte[8];
            uint A = 0;
            uint B = 0;
            for (int k = 0; k < 4; k++)
            {
                A += (uint)(input[k] & 0xFF) << (8 * k);
                B += (uint)(input[k + 4] & 0xFF) << (8 * k);
            }
            uint[] LE = new uint[13];
            uint[] RE = new uint[13];
            LE[0] = (A + S[0]);
            RE[0] = (B + S[1]);
            for (int i = 1; i <= 12; i++)
            {
                LE[i] = (leftRotate((uint)(LE[i - 1] ^ RE[i - 1]), (int)RE[i - 1]) + S[2 * i]);
                RE[i] = (leftRotate((uint)(RE[i - 1] ^ LE[i]), (int)LE[i]) + S[2 * i + 1]);
            }
            for (int k = 0; k < 4; k++)
            {
                output[k] = (byte)((LE[12] >> (8 * k)) & 0xFF);
            }
            for (int k = 0; k < 4; k++)
            {
                output[k + 4] = (byte)((RE[12] >> (8 * k)) & 0xFF);
            }
            //output[k] = (byte)((LE[12] >> (8 * k)) & 0xFF);
            return output;
        }

        public byte[] DecryptString(byte[] input, uint[] S)
        {
            byte[] output = new byte[8];
            uint[] LD = new uint[13];
            uint[] RD = new uint[13];
            for (int k = 0; k < 4; k++)
            {
                LD[12] += (uint)(input[k] & 0xFF) << (8 * k);
                RD[12] += (uint)(input[k + 4] & 0xFF) << (8 * k);
            }
            for (int i = 12; i > 0; i--)
            {
                RD[i - 1] = (RightRotate((uint)(RD[i] - S[2 * i + 1]), (int)LD[i]) ^ LD[i]);
                LD[i - 1] = (RightRotate((uint)(LD[i] - S[2 * i]), (int)RD[i - 1]) ^ RD[i - 1]);
            }
            uint A = 0;
            uint B = 0;
            A = LD[0] - S[0];
            B = RD[0] - S[1];
            for (int k = 0; k < 4; k++)
            {
                output[k] = (byte)((A >> (8 * k)) & 0xFF);
            }
            for (int k = 0; k < 4; k++)
            {
                output[k + 4] = (byte)((B >> (8 * k)) & 0xFF);
            }
            return output;
        }

        /// <summary>
        /// Stand RC5 encrypt for BMP files
        /// </summary>
        /// <param name="streamreader"></param>
        /// <param name="streamwriter"></param>
        public void EncryptBMP(FileStream streamreader, FileStream streamwriter)
        {
            uint r1, r2;
            System.IO.BinaryReader br = new System.IO.BinaryReader(streamreader);
            System.IO.BinaryWriter bw = new System.IO.BinaryWriter(streamwriter);

            byte[] header = new byte[54];
            header = br.ReadBytes(54);
            bw.Write(header);

            long filelength = streamreader.Length - 54;

            while (filelength > 0)
            {
                try
                {
                    r1 = br.ReadUInt32();
                    try
                    {
                        r2 = br.ReadUInt32();
                    }
                    catch
                    {
                        r2 = 0;
                    }
                }
                catch
                {
                    r1 = r2 = 0;
                }
                Encode(ref r1, ref r2, rounds);
                bw.Write(r1);
                bw.Write(r2);
                filelength -= 8;
            }
            streamreader.Close();
            streamwriter.Close();
        }

        /// <summary>
        /// Stand RC5 decrypt for BMP files
        /// </summary>
        /// <param name="streamreader"></param>
        /// <param name="streamwriter"></param>
        public void DecryptBMP(FileStream streamreader, FileStream streamwriter)
        {
            uint r1, r2;

            System.IO.BinaryReader br = new System.IO.BinaryReader(streamreader);
            System.IO.BinaryWriter bw = new System.IO.BinaryWriter(streamwriter);

            byte[] header = new byte[54];
            header = br.ReadBytes(54);
            bw.Write(header);
            long filelength = streamreader.Length - 54;

            while (filelength > 0)
            {
                try
                {
                    r1 = br.ReadUInt32();
                    r2 = br.ReadUInt32();
                    Decode(ref r1, ref r2, rounds);

                    if (!(r1 == 0 && r2 == 0 && (filelength - 8 <= 0)))
                    {
                        bw.Write(r1);
                        bw.Write(r2);
                    }
                    if (r2 == 0 && (filelength - 8 <= 0))
                    {
                        bw.Write(r1);
                    }
                    filelength -= 8;
                }
                catch
                {
```

```
      //Console.WriteLine("May be U try to decrypt an normal file (plain file)" + "Error");
      return;
    }
  }

  streamreader.Close();
  streamwriter.Close();
}

/// <summary>
/// Enhanced RC5 with Chaos encrypt BMP file
/// </summary>
/// <param name="streamreader"></param>
/// <param name="streamwriter"></param>
public void EncryptBMPWithChaos(FileStream streamreader, FileStream streamwriter)
{
  uint r1, r2;
  System.IO.BinaryReader br = new System.IO.BinaryReader(streamreader);
  System.IO.BinaryWriter bw = new System.IO.BinaryWriter(streamwriter);

  byte[] header = new byte[54];
  header = br.ReadBytes(54);
  bw.Write(header);

  long filelength = streamreader.Length - 54;
  while (filelength > 0)
  {
    try
    {
      r1 = br.ReadUInt32();
      try
      {
        r2 = br.ReadUInt32();
      }
      catch
      {
        r2 = 0;
      }
    }
    catch
    {
      r1 = r2 = 0;
    }
```

```
        EncodeWithChaos(ref r1, ref r2, rounds);
        bw.Write(r1);
        bw.Write(r2);
        filelength -= 8;
    }
    streamreader.Close();
    streamwriter.Close();
}

/// <summary>
/// Enhanced RC5 with Chaos decrypt BMP file
/// </summary>
/// <param name="streamreader"></param>
/// <param name="streamwriter"></param>
public void DecryptBMPWithChaos(FileStream streamreader, FileStream streamwriter)
{
    uint r1, r2;

    System.IO.BinaryReader br = new System.IO.BinaryReader(streamreader);
    System.IO.BinaryWriter bw = new System.IO.BinaryWriter(streamwriter);

    byte[] header = new byte[54];
    header = br.ReadBytes(54);
    bw.Write(header);
    long filelength = streamreader.Length - 54;

    while (filelength > 0)
    {
        try
        {
            r1 = br.ReadUInt32();
            r2 = br.ReadUInt32();
            DecodeWithChaos(ref r1, ref r2, rounds);

            if (!(r1 == 0 && r2 == 0 && (filelength - 8 <= 0)))
            {
                bw.Write(r1);
                bw.Write(r2);
            }
            if (r2 == 0 && (filelength - 8 <= 0))
            {
                bw.Write(r1);
            }
            filelength -= 8;
        }
        catch
        {
            //Console.WriteLine("May be U try to decrypt an normal file (plain file)" + "Error");
            return;
        }
    }

    streamreader.Close();
    streamwriter.Close();
}



/// <summary>
/// Enhanced RC5 with Chaos encrypt BMP file
/// </summary>
/// <param name="streamreader"></param>
/// <param name="streamwriter"></param>
public void EncryptBMPWithChaos_block(FileStream streamreader, FileStream streamwriter)
{
    uint r1, r2;
    System.IO.BinaryReader br = new System.IO.BinaryReader(streamreader);
    System.IO.BinaryWriter bw = new System.IO.BinaryWriter(streamwriter);




    byte[] header = new byte[54];
    header = br.ReadBytes(54);
    bw.Write(header);

    long filelength = streamreader.Length - 54;
    while (filelength > 0)
    {
        try
        {
            r1 = br.ReadUInt32();
            try
            {
                r2 = br.ReadUInt32();
            }
            catch
            {
                r2 = 0;
            }
        }
        catch
        {
            r1 = r2 = 0;
        }
        EncodeWithChaos_block(ref r1, ref r2, rounds);
        bw.Write(r1);
        bw.Write(r2);
        filelength -= 8;
    }
    streamreader.Close();
    streamwriter.Close();
}

/// <summary>
/// Enhanced RC5 with Chaos decrypt BMP file
/// </summary>
/// <param name="streamreader"></param>
/// <param name="streamwriter"></param>
public void DecryptBMPWithChaos_block(FileStream streamreader, FileStream streamwriter)
{
    uint r1, r2;

    System.IO.BinaryReader br = new System.IO.BinaryReader(streamreader);
    System.IO.BinaryWriter bw = new System.IO.BinaryWriter(streamwriter);

    byte[] header = new byte[54];
    header = br.ReadBytes(54);
    bw.Write(header);
    long filelength = streamreader.Length - 54;

    while (filelength > 0)
    {
        try
        {
            r1 = br.ReadUInt32();
            r2 = br.ReadUInt32();
            DecodeWithChaos_block(ref r1, ref r2, rounds);

            if (!(r1 == 0 && r2 == 0 && (filelength - 8 <= 0)))
            {
                bw.Write(r1);
                bw.Write(r2);
            }
            if (r2 == 0 && (filelength - 8 <= 0))
            {
                bw.Write(r1);
            }
            filelength -= 8;
        }
        catch
        {
            //Console.WriteLine("May be U try to decrypt an normal file (plain file)" + "Error");
            return;
        }
    }

    streamreader.Close();
    streamwriter.Close();
}



/// <summary>
/// Enhanced RC5 with Chaos algorithm encrypt document and used to analysis for NIST tool
/// </summary>
/// <param name="streamreader"></param>
/// <param name="streamwriter"></param>
public void EncryptWithChaos(FileStream streamreader, FileStream streamwriter)
{
    uint r1, r2;
    System.IO.BinaryReader br = new System.IO.BinaryReader(streamreader);
    //System.IO.BinaryWriter bw = new System.IO.BinaryWriter(streamwriter);

    long filelength = streamreader.Length;
    //Console.WriteLine("File Length: " + filelength);

    while (filelength > 0)
    {
        try
        {
            r1 = br.ReadUInt32();
            try
            {
                r2 = br.ReadUInt32();
            }
            catch
            {
                r2 = 0;
            }
        }
        catch
        {
            r1 = r2 = 0;
        }
        EncodeWithChaos(ref r1, ref r2, rounds);
        byte[] temp1 = Encoding.ASCII.GetBytes(Convert.ToString(r1,2));
        byte[] temp2 = Encoding.ASCII.GetBytes(Convert.ToString(r2, 2));

        streamwriter.Write(temp1, 0, temp1.Length);
        streamwriter.Write(temp2, 0, temp2.Length);

        filelength -= 8;
    }
    streamreader.Close();
    streamwriter.Close();
}

public void DecryptWithChaos(FileStream streamreader, FileStream streamwriter)
{
    uint r1, r2;

    System.IO.BinaryReader br = new System.IO.BinaryReader(streamreader);
    System.IO.BinaryWriter bw = new System.IO.BinaryWriter(streamwriter);

    long filelength = streamreader.Length;
    //Console.WriteLine("File Length: " + filelength);

    while (filelength > 0)
    {
        try
        {
            r1 = br.ReadUInt32();
            r2 = br.ReadUInt32();
            DecodeWithChaos(ref r1, ref r2, rounds);

            if (!(r1 == 0 && r2 == 0 && (filelength - 8 <= 0)))
            {
                    bw.Write(r1);
                    bw.Write(r2);
                }
                if (r2 == 0 && (filelength - 8 <= 0))
                {
                    bw.Write(r1);
                }
                filelength -= 8;
            }
            catch
            {
                return;
            }
        }

        streamreader.Close();
        streamwriter.Close();
    }


    ///Logistic Model：X_n+1=u*Xn(1-Xn)
    /// <summary>
    /// Logistic Map Chaotic Encryption and Decryption model
    /// </summary>
    /// <param name="u">in interval [3.57,4]</param>
    /// <param name="x0">in interval (0,1)</param>
    /// <returns></returns>
    private static double logistic(double u, double x, int n)
    {
        for (int i = 0; i < n; i++)
        {
            x = u * x * (1 - x);
        }
        return x;
    }
}
```

Appendix V. Enhanced RC5 based on 2-D Logistic Map Code – C#

```
class RC5_logisitc2d
    {
        uint[] s;
        int rounds;

        public logisitc2d(int round, double x0, double y0, double logistic2d_r)
        {
            rounds = round;
            s = new uint[2*rounds+2];

            double[,] randomkey = logistic2d(rounds, x0, y0, logistic2d_r);
            for (int i = 0, j = 0; i < 2*rounds + 1; i = i + 2, j++)
            {
                s[i] = Convert.ToByte(Convert.ToInt32(Math.Floor(randomkey[0, j] * 10000) % 256));
                s[i + 1] = Convert.ToByte(Convert.ToInt32(Math.Floor(randomkey[1, j] * 10000) % 256));
            }

        }

        #region Rotate Operation
        //to circulate int left
        private uint leftRotate(uint x, int offset)
        {
            uint t1, t2;
            t1 = x >> (32 - offset);
            t2 = x << offset;
            return t1 | t2;
        }
        //to circulate int right
        private uint RightRotate(uint x, int offset)
        {
            uint t1, t2;
            t1 = x << (32 - offset);
            t2 = x >> offset;
            return t1 | t2;
        }
        #endregion

        #region Encoding and Decoding
        //encryption operation on two block
        private void Encode(ref uint r1, ref uint r2, int rounds)
        {
            r1 = r1 + s[0];
            r2 = r2 + s[1];
            for (int i = 1; i <= rounds; i++)
            {
                r1 = leftRotate(r1 ^ r2, (int)r2) + s[2 * i];
                r2 = leftRotate(r2 ^ r1, (int)r1) + s[2 * i + 1];
            }
        }
        //decryption operation on two block
    private void Decode(ref uint r1, ref uint r2, int rounds)
    {
      for (int i = rounds; i >= 1; i--)
      {
          r2 = (RightRotate(r2 - s[2 * i + 1], (int)r1)) ^ r1;
                r1 = (RightRotate(r1 - s[2 * i], (int)r2)) ^ r2;
            }
            r2 = r2 - s[1];
            r1 = r1 - s[0];
        }
        #endregion

        #region Encryption with 2d Logisitic
        /// <summary>
        /// Enhanced RC5 with Chaos encrypt BMP file
        /// </summary>
        /// <param name="streamreader"></param>
        /// <param name="streamwriter"></param>
        public void EncryptBMPWith2dLogisitic(FileStream streamreader, FileStream streamwriter)
        {
            uint r1, r2;
            System.IO.BinaryReader br = new System.IO.BinaryReader(streamreader);
            System.IO.BinaryWriter bw = new System.IO.BinaryWriter(streamwriter);

            byte[] header = new byte[54];
            header = br.ReadBytes(54);
            bw.Write(header);

            long filelength = streamreader.Length - 54;
            while (filelength > 0)
            {
                try
            {
                r1 = br.ReadUInt32();
                try
                {
                    r2 = br.ReadUInt32();
                }
                catch
                {
                    r2 = 0;
                }
            }
            catch
            {
                r1 = r2 = 0;
            }
            Encode(ref r1, ref r2, rounds);
            bw.Write(r1);
            bw.Write(r2);
            filelength -= 8;
        }
        streamreader.Close();
        streamwriter.Close();
    }

    /// <summary>
    /// Enhanced RC5 with Chaos decrypt BMP file
    /// </summary>
    /// <param name="streamreader"></param>
    /// <param name="streamwriter"></param>
    public void DecryptBMPWith2dLogisitic(FileStream streamreader, FileStream streamwriter)
    {
        uint r1, r2;

        System.IO.BinaryReader br = new System.IO.BinaryReader(streamreader);
        System.IO.BinaryWriter bw = new System.IO.BinaryWriter(streamwriter);

        byte[] header = new byte[54];
        header = br.ReadBytes(54);
        bw.Write(header);
        long filelength = streamreader.Length - 54;

        while (filelength > 0)
           {
```

```
    try
    {
      r1 = br.ReadUInt32();
      r2 = br.ReadUInt32();
      Decode(ref r1, ref r2, rounds);

      if (!(r1 == 0 && r2 == 0 && (filelength - 8 <= 0)))
      {
        bw.Write(r1);
        bw.Write(r2);
      }
      if (r2 == 0 && (filelength - 8 <= 0))
      {
        bw.Write(r1);
      }
      filelength -= 8;
    }
    catch
    {
      //Console.WriteLine("May be U try to decrypt an normal file (plain file)" + "Error");
      return;
    }
  }

  streamreader.Close();
  streamwriter.Close();
}

#endregion

///Logistic 2d Model：
/// x[i+1] = r*(3*y[i]+1)*x[i]*(1-x[i]);
/// y[i+1] = r*(3*x[i+1]+1)*y[i]*(1-y[i]);
/// <summary>
/// Logistic Map Chaotic Encryption and Decryption model
/// </summary>
/// <param name="u">in interval [1.11,1.19]</param>
/// <returns></returns>
private double[,] logistic2d(int rounds, double logistic2d_x0, double logistic2d_y0, double logistic2d_r)
{
  double[,] logistic2d_arr = new double[2, (rounds + 1)];
  logistic2d_arr[0, 0] = logistic2d_x0;
  logistic2d_arr[1, 0] = logistic2d_y0;
```

```
            for (int i = 0; i < rounds; i++)
            {
                logistic2d_arr[0, i + 1] = logistic2d_r * (3 * logistic2d_arr[1, i] + 1) * logistic2d_arr[0, i] * (1 -
logistic2d_arr[0, i]);
                logistic2d_arr[1, i + 1] = logistic2d_r * (3 * logistic2d_arr[0, i + 1] + 1) * logistic2d_arr[1, i] * (1 -
logistic2d_arr[1, i]);
            }

            return logistic2d_arr;
        }
    }
```

Appendix VI. Logistic Map Bifurcation Plot Code

1. 1-D Logistic Map bifurcation plot code

```
%-----------------------------------------------------------------------
% LOGISTIC MAP
%-----------------------------------------------------------------------
n=64;
key=0.512;
an=linspace(0.0, 3.99,400);
figure;
hold on;box on;axis([min(an),max(an),0,1]);

N=n^2;
xn=zeros(1,N);
for a=an;
   x=key;
   for k=1:20;
      x=a*x*(1-x);
   end;
   for k=1:N;
      x=a*x*(1-x);
      xn(k)=x;
      b(k,1)=x;
   end;
   plot(a*ones(1,N),xn,'k.','markersize',1);
end;
xlabel('{\lambda}');
ylabel('{\chi}');
title('\bf Bifurcation plot of LOGISTIC map');
```

2. 2-D Logistic Map bifurcation plot code

```
x(1) = 0.8909;
y(1) = 0.3342;
r = 1.19;

for i = 2:20000
   x(i) = r*(3*y(i-1)+1)*x(i-1)*(1-x(i-1));
   if x(i)>1
      x(i) = x(i) - 1;
   end
```

```
  y(i) = r*(3*x(i)+1)*y(i-1)*(1-y(i-1));
  if y(i)>1
     y(i) = y(i) - 1;
  end
end
figure;
plot(x,y, '.', 'Markersize',3);
xlabel('{\lambda}');
ylabel('{\chi}');
title('Trajectory of 2D logistic map');
```

Appendix VII. Correlation Analysis of Two Adjacent Pixels Matlab code.

```
function [ result ] = corr2adjacent( img, str )
%%
% Selecting subsets of paired pixels: Let's start with the set of unique horizontal pairings of pixels. If I select the pixels in the first column of A and place them in the subset x, then the horizontally adjacent pixels will be those in the second column of A, and these will be placed in the subset y. I can also add the pixels in the second column to the subset x, and the horizontally adjacent pixels in the third column would then be placed in the subset y. Repeating this for all columns in A, we can see that the pixels in columns 1 through 255 will be in subset x, and the pixels in columns 2 through 256 will be in the subset y. The matrix indexing would therefore look like this:
%
% x = A(:,1:end-1,1);  %# All rows and columns 1 through 255 from red plane
% y = A(:,2:end,1);    %# All rows and columns 2 through 256 from red plane
% Following similar logic as above, you can construct the entire set of unique vertical pairings of pixels in this fashion:
%
% x = A(1:end-1,:,1);  %# Rows 1 through 255 and all columns from red plane
% y = A(2:end,:,1);    %# Rows 2 through 256 and all columns from red plane
% And likewise for the set of unique diagonal pairings of pixels, where "diagonal" runs from top left to bottom right in the matrix:
%
% x = A(1:end-1,1:end-1,1);  %# All but the last row and column
% y = A(2:end,2:end,1);      %# All but the first row and column
% Or for "anti-diagonals", where "diagonal" runs from bottom left to top right in the matrix:
%
% x = A(2:end,1:end-1,1);  %# All but the first row and last column
% y = A(1:end-1,2:end,1);  %# All but the last row and first column
% Now, you can choose any one of these sets of x and y data to perform the statistical calculations you want for the red color plane. You can repeat the above substituting 2 or 3 for the last index in each line to get the calculation for the green and blue color planes, respectively.
%
% Performing the statistical tests: This part is simple. There is already a built-in function CORRCOEF for computing the correlation coefficient in MATLAB. You may have to reshape the subsets of pixel values x and y into column vectors first using single-colon indexing:
%
% r_xy = corrcoef(x(:),y(:));
% Functions also exist for the other formulae as well: MEAN for E(x), VAR for D(x), and COV for cov(x,y).
%
% In regard to your second question, you can first create x and y as I did above for all unique pairs of horizontally adjacent pixels, then create a vector with a random permutation of the integer indices into x and y using the function RANDPERM. Selecting the first 5000 entries of those randomly permuted indices will give you 5000 random indices into x and y:
%
```

```
% randIndex = randperm(numel(x));  %# A random permutation of the integers
%                          %#   from 1 to numel(x)
% randIndex = randIndex(1:5000);   %# Pick the first 5000 indices
% xRand = x(randIndex);           %# 5000 random values from x
% yRand = y(randIndex);           %# The corresponding 5000 values from y
% This will give you your 5000 pairs of horizontally adjacent pixel values in x and y. However, it is unclear what you mean by "plot the distribution". I'm guessing you will either end up using the function HIST or perhaps the function SCATTER for this purpose.
%%

%img = imread('lena.bmp');

x = img(:,1:end-1);
y = img(:,2:end);

randIndex = randperm(numel(x));
randIndex = randIndex(1:1000);

xR = x(randIndex);
yR = y(randIndex);
h_result = corrcoef(double(xR(:)),double(yR(:)));

% figure('name','Horizontal');
% plot(xR(:),yR(:),'.', 'MarkerSize', 5);
% title(['Correlation of horizontal adjacent two pixels for ',str ,'image']);
% xlabel('pixel gray value on location (x,y)');
% ylabel('pixel gray value on location (x+1,y)');

x = img(1:end-1,:);
y = img(2:end,:);
randIndex = randperm(numel(x));
randIndex = randIndex(1:1000);
xR = x(randIndex);
yR = y(randIndex);
v_result = corrcoef(double(xR(:)),double(yR(:)));

% figure('name','Vertical');
% plot(xR(:),yR(:),'.', 'MarkerSize', 5);
% title(['Correlation of vertical adjacent two pixels for ',str ,'image']);
% xlabel('pixel gray value on location (x,y)');
% ylabel('pixel gray value on location (x+1,y)');

x = img(1:end-1,1:end-1);
```

```
y = img(2:end,2:end);
randIndex = randperm(numel(x));
randIndex = randIndex(1:1000);
xR = x(randIndex);
yR = y(randIndex);
d_result = corrcoef(double(xR(:)),double(yR(:)));

% figure('name','Diagonal');
% plot(xR(:),yR(:),'.', 'MarkerSize', 5);
% title(['Correlation of diagonal adjacent two pixels for ',str ,'image']);
% xlabel('pixel gray value on location (x,y)');
% ylabel('pixel gray value on location (x+1,y)');

result = [h_result(1,2), v_result(1,2), d_result(1,2)];

end
```

Appendix VIII. NPCR and UACI Matlab Code.

```
function results = NPCR_and_UACI( img_a, img_b, need_display, largest_allowed_val )
%
=======================================================================
====
% FUNCTION:
%       gives NPCR & UACI quantitative and qualitative scores for
%       the strength against possible differential attacks of image
%       ciphers
%
=======================================================================
====
% INPUT:
%       img_a, img_b: two encrypted images of same size and type
%       need_display: on/off option to show outputs (default: on)
%       largest_allowed_val: is the value of the largest theoretical
%               allowed value in encrypted image. If it is not
%               provided, algorithm will automatically choose one.
%
=======================================================================
====
% OUTPUT:
%       results.npcr_score: quantitative NPCR score (larger is better)
%       results.npcr_pVal : qualitative NPCR score  (larger is better)
%       results.npcr_dist : theoretical NPCR normal dist. (mean +\- var)
%       results.uaci_score: quantitative UACI score (larger is NOT better)
%       results.uaci_pVal : qualitative UACI score  (larger is better)
%       results.uaci_dist : theoretical UACI normal dist. (mean +\- var)
%
=======================================================================
====
% DEMO:
%       % Demo 1: simple use
%       % generate two 256x256 8-bit random-imges
%       img_a = randi(256,256,256) - 1;
%       img_b = randi(256,256,256) - 1;
%       % get NPCR and UACI scores
%       results = NPCR_and_UACI( img_a, img_b, 1, 255 );
%
%       %% Demo 2: why we need qualitative pVals besides quantitative scores
%       img_a = imread('cameraman.tif');
%       img_b = uint8( randi(256,256,256)-1 );
%       tmp = NPCR_and_UACI( img_a, flipud( fliplr (img_a) ) );
%       score.camman = [ tmp.npcr_score, tmp.uaci_score ]; pVals.camman = [ tmp.npcr_pVal, tmp.uaci_pVal ];
%       tmp = NPCR_and_UACI( img_b, flipud( fliplr (img_b) ) );
%       score.rand = [ tmp.npcr_score, tmp.uaci_score ]; pVals.rand = [ tmp.npcr_pVal, tmp.uaci_pVal ];
%       tmp = NPCR_and_UACI( img_a, img_b );
%       score.cam_and_rand = [ tmp.npcr_score, tmp.uaci_score ]; pVals.cam_and_rand = [ tmp.npcr_pVal, tmp.uaci_pVal ];
%       display( ['Can you easily figure out random-like image pairs from their scores?']), display(score)
%       display( ['These scores are often less imformative than you thought, arnt they?'])
%       display( ['-------------------------------------------------------------------'])
%       display( ['Can you easily figure out random-like images from pure pVals (the smaller a pVal is, the less likely a test image is random-like)?']), display(pVals)
%       display( 'Do you see that UACI pVals for the camman image pair and the camman and rand image pair are extremely small?')
%       display( 'These pVals indicating that these two pairs are distinguishable from truely random ones.')
%       display( ['-------------------------------------------------------------------'])
%       %% Demo 3: NPCR and UACI distributions for random images
%       nsample = 10000;
%       npcr_score = zeros(1,nsample); npcr_pVal = zeros(1,nsample);
%       uaci_score = zeros(1,nsample); uaci_pVal = zeros(1,nsample);
%       for i = 1:nsample
%           img_a = randi(256,256,256) - 1;
%           img_b = randi(256,256,256) - 1;
%           this_result = NPCR_and_UACI( img_a, img_b, 0 , 255);
%           npcr_score(i) = this_result.npcr_score;
%           uaci_score(i) = this_result.uaci_score;
%           npcr_pVal(i) = this_result.npcr_pVal;
%           uaci_pVal(i) = this_result.uaci_pVal;
%       end
%       % npcr pdf distribution
%       [ hist_npcr, val_npcr ] = hist( npcr_score, [0:(1/65536):1] );
%       figure,subplot(221),bar( val_npcr, hist_npcr / sum(hist_npcr) ), title( 'NCPR pdf' );
%       theoretical_hist_npcr = normcdf( val_npcr, this_result.npcr_dist(1), sqrt( this_result.npcr_dist(2) ) ) ...
%            - normcdf( [0,val_npcr(1:end-1)], this_result.npcr_dist(1), sqrt( this_result.npcr_dist(2) ) );
%       hold on, plot( val_npcr, theoretical_hist_npcr, 'r--','LineWidth', 2 ), xlim( [ this_result.npcr_dist(1)+4*sqrt( this_result.npcr_dist(2) )*[-1,1] ])
%       legend( 'sample distribution', 'theoretical distribution', 4); axis square;
%       % npcr cdf distribution
%       subplot(223),bar( val_npcr, cumsum(hist_npcr / sum(hist_npcr)) ), title( 'NCPR cdf' );
%       hold on, plot( val_npcr, cumsum(theoretical_hist_npcr), 'g--','LineWidth', 2 ), xlim( [ this_result.npcr_dist(1)+4*sqrt( this_result.npcr_dist(2) )*[-1,1] ])
%       legend( 'sample distribution', 'theoretical distribution', 4); axis square;
%       % uaci pdf distribution
%       [ hist_uaci, val_uaci ] = hist( uaci_score, [0:(1/65536):1] );
%       subplot(222),bar( val_uaci, hist_uaci / sum(hist_uaci) ), title( 'UACI pdf' );
%       theoretical_hist_uaci = normcdf( val_uaci, this_result.uaci_dist(1), sqrt( this_result.uaci_dist(2) ) ) ...
%            - normcdf( [0,val_uaci(1:end-1)], this_result.uaci_dist(1), sqrt( this_result.uaci_dist(2) ) );
%       hold on, plot( val_uaci, theoretical_hist_uaci, 'r--','LineWidth', 2 ), xlim( [ this_result.uaci_dist(1)+4*sqrt(
this_result.uaci_dist(2) )*[-1,1] ])
%       legend( 'sample distribution', 'theoretical distribution', 4 ); axis square;
%       % uaci cdf distribution
%       subplot(224),bar( val_uaci, cumsum(hist_uaci / sum(hist_uaci)) ), title( 'UACI cdf' );
%       hold on, plot( val_uaci, cumsum(theoretical_hist_uaci), 'g--','LineWidth', 2 ),  xlim( [ this_result.uaci_dist(1)+4*sqrt(
this_result.uaci_dist(2) )*[-1,1] ])
%       legend( 'sample distribution', 'theoretical distribution', 4 ); axis square;
%
==========================================================================
====
% SCORE INTERPRETATION:
%       if your cipher is abled to encrypted images that indistinguishable
%       from random images under the NPCR and UACI measures, pVals simply
%       represent the possibility that your tested images are indeed random
%       -like, and thus a larger pVal is preferred. On the other hand,
%       pVals are random variables, and could be very small (say 0.0001)
%       even though test images are truely random-like. Therefore, it is
%       meaningless to make any conclusive claim for a small test data set.
%       However, if you observe that out of 100 tested image pairs, 5 of
%       them fail to achieve pVals greater than 0.01 (or 1%), then this is
%       a clear indicator that this image cipher fail to generated
%       random-like outputs, because if we use 100 truely random-like
%       image pairs, we will only observe 1 out 100 with pVal less than
%       0.01 in theory.
%
==========================================================================
====
% NOTE:
%       1. This code is only free-of-use for research and acadmic use.
%       2. Whenever the proposed code is used in scitific research,
%         please kindly cite the related article(s).
%       3. Achieving a good randomness P-vals does not guarantee a
%         cipher is secure. The only thing that is safe to claim is that
%         "a cipher is able to generate random-like data indistinguishable
%         from those truely random-like under XXX measure"
%       4. One may find UACI and NPCR are defined differently in
%         literature. This implementation adopts the definitions given in
%         the paper below.
```

```
%
======================================================================
====
% PAPER INFORMATION:
%      Wu, Y., Noonan, J. P., & Agaian, S.
%      NPCR and UACI randomness tests for image encryption.
%      on Journal of Selected Areas in Telecommunications (JSAT), 31-38.
%      2011. (http://www.cyberjournals.com/Papers/Apr2011/05.pdf)
%
======================================================================
====
% CONTACT:
%       Name: Dr. Yue Wu
%      Email: ywu03@ece.tufts.edu.
%
======================================================================
====

%% 1. input_check
[ height_a, width_a, depth_a ] = size( img_a );
[ height_b, width_b, depth_b ] = size( img_b );
if ( ( height_a ~= height_b ) ...
  || (  width_a ~=  width_b ) ...
  || (  depth_a ~=  depth_b ) )
   error( 'input images have to be of same dimensions' );
end
class_a = class( img_a );
class_b = class( img_b );
if ( ~strcmp( class_a, class_b) )
   error( 'input images have to be of same data type');
end

%% 2. measure preparations
if ( ~exist( 'largest_allowed_val', 'var') )
   switch  class_a
      case 'uint16'
         largest_allowed_val = 65535;
      case 'uint8'
         largest_allowed_val = 255;
      case 'logical'
         largest_allowed_val = 2;
      otherwise
         largest_allowed_val = max ( max( img_a(:), img_b(:) ) );
   end
end
if ( ~exist( 'need_display', 'var' ) )
   need_display = 1;
end
img_a = double( img_a );
img_b = double( img_b );
num_of_pix = numel( img_a );

%% 3. NCPR score and p_value
results.npcr_score = sum( double( img_a(:) ~= img_b(:) ) ) / num_of_pix;
npcr_mu  = ( largest_allowed_val ) / ( largest_allowed_val+ 1 );
npcr_var =  ( ( largest_allowed_val) / ( largest_allowed_val+ 1 )^2 ) / num_of_pix;
results.npcr_pVal = normcdf( results.npcr_score, npcr_mu, sqrt( npcr_var ) );
results.npcr_dist = [ npcr_mu, npcr_var ];

%% 4. UACI score and p_value
results.uaci_score = sum( abs( img_a(:) - img_b(:) ) ) / num_of_pix / largest_allowed_val;
uaci_mu  = ( largest_allowed_val+2 ) / ( largest_allowed_val*3+3 );
uaci_var = ( ( largest_allowed_val+2 ) * ( largest_allowed_val^2 + 2*largest_allowed_val+ 3 ) /18 / (
largest_allowed_val+ 1 )^2 / largest_allowed_val) / num_of_pix;
p_vals = normcdf( results.uaci_score, uaci_mu, sqrt( uaci_var ) );
p_vals( p_vals > 0.5 ) = 1 - p_vals( p_vals > 0.5 );
results.uaci_pVal = 2 * p_vals;
results.uaci_dist = [ uaci_mu, uaci_var ];

%% 5. optional output
if ( need_display )
  format long;
  display( results );
end
```