Improving Security of the Telemedicine System for the Rural People of Bangladesh

Toufik Ahmed Emon
Dept. of CSE
Jahangirnagar University
Savar, Dhaka 1212

Uzzal Kumar Prodhan
Dept. of CSE
Jatiya Kabi Kazi Nazrul Islam University
Trishal, Mymensingh

Mohammad Zahidur Rahman, Israt Jahan
Dept. of CSE
Jahangirnagar University
Savar, Dhaka 1212

***Abstract*—Telemedicine is a healthcare system where healthcare professionals have the capability to observe, diagnose, evaluate and treat the patient from a remote location and the patient have the ability to easily access the medical expertise quickly and efficiently. Increasing popularity of Telemedicine increase the security intimidations. In this paper, a security framework is implemented for the developed cost-effective Telemedicine system. The proposed security framework secure all the sections of the model following the recommendations of Health Level 7, First Healthcare Interoperability Resources and Health Insurance Portability and Accountability Act. Implementation of this security framework including authenticating the different types of user, secure connection between mobile and sensors through authentication, protect the mobile application from hackers, ensures data security through encryption, as well as secure server, using secured socket layer called SSL. Finally, we can say that the developed Telemedicine model is more secure and it can be implemented in any remote areas of developing countries as like as Bangladesh.**

***Keywords*—Telemedicine; security; encryption; hashing**

I. Introduction

About 400 million people around the world are deprived of the basic healthcare service. In Bangladesh, the number of physician per 10,000 people is only 3 and nurse per 10,000 is only 1.07 [1]. These data show a severe scarcity of healthcare services in Bangladesh. In this regards, we develop stored and forward telemedicine system. In our developed system, the expert and local doctors, the pharmacy admin and lab assistant need to register in the system. The system administrator does the registration. The local pharmacist registers the patients. The local doctors Login the mobile application which is connected to sensors to get sensors data. When the sensors data received by the mobile application, it shows in the mobile application. Then the data is encrypted as well as send to the remote server. These data decrypt in the server site and save on the server. Doctors examined all the data and send a prescription to the patients through proper channel. The expert doctors also registered in the system before they prescribe patients. The end user gets the prescription and gives the prescribed medicine as well as suggestions to the respective patients. To make the system reliable as well as trustworthy to the user, security becomes the main concern of this telemedicine system.

In this research, we improve the security and privacy features of the developed telemedicine system. The security framework is particularly designed for our developed telemedicine system. Our developed telemedicine includes several modules such as sensors, mobile application, web application as well as the web server. There are some other modules as like Bluetooth connected devices, wifi or data transmission through other media. Therefore, to secure a telemedicine , we need to secure all of its modules. We divide the telemedicine framework into five different sections to improve the security, such as authentication and application security, client layer security, patient data security, web server security as well as database security. Furthermore, ensures the security of each module following the recommendations of Health Level 7 or HL7 and Fast Healthcare Interoperability Resources or FHIR. Not only that we also several security models, such as Access control list or ACL, Multi-level Security or MLS as well as Role-based access model or RBAC.

The rest of the paper is organized as follows: In Section II, we briefly discuss different methods currently being used to secure the telemedicine system as well as their advantages and disadvantages. In Section III, we describe the main module of our telemedicine system followed by a brief description of security module with its security advantages in Section IV. Section V discusses the drawbacks and future work of this proposed module. Finally, in Section VI, we conclude with the summary.

II. Literature Review

Security of telemedicine is a growing concern as it becomes more complex day by day. There are several security models to secure the telemedicine system. M. Fahim Ferdous Khan and Ken Sakamura proposed a hybrid access control model for healthcare informatics considering the following issues such as network security, emergency access, the principle of minimum disclosure, user approval, access authorization, etc. Their authentication mechanism is based on the eTRON architecture [2]. On the other hand, Liu et al proposed a model named as Open and Trusted Health Information Systems or OTHIS targeting the Australian Health sector. Their proposed system is compatible with general Health Information System or HIS [3]. Prema T. Akkasaligar and Sumangala Biradar encrypt medical image using Chaos theory and DNA cryptography. They divided the image into odd and even DNA encoded image. Then they add both images to get the original encrypted image.

To decrypt the image they use reverse process [4]. M.J. Chang, J. K. Jung, M.W. Park and T.M. Chung find out the security holes in Telemedicine and suggest firewall to control

unauthorized access [5]. I. Chiuchisan, D.G. Balan, O. Geman and I. Chiuchisan and I. Gordin use signature verification, data encryption as well as secure network infrastructure to secure the telemedicine system [6]. C. Fu, Y. Lin, H. y. Jiang and H. f. Ma improves their encryption by extending the key length to 212 bits. They declared all the variables as 64-bit double precision type [7].

Basudev Halder and S.Mitra implement a watermarking based process in which they can recover the converted ECG images without any distortion [8]. C. Han, L. Sun, and Q. Du use fountain code and image segmentation to secure image transmission. They divide the image into two parts. The main advantages of their proposed method are lower complexity and lower cost [9]. R.M. Seepers, C. Strydis, I. Sourdis and C.I. De Zeeuw proposed a heartbeat based security module where they used IPI (Inter-Pulse-Interval) to generate the security code [10]. Uzzal Kumar Prodhan, Mohammad Zahidur Rahman, and Israt Jahan did a survey on Telemedicine status in Bangladesh and found most of the Bangladeshi people, doctors, and nurses, pharmacy, and hospital want telemedicine service [11]. A. Sudarsono, P. Kristalina, M.U.H. Al Rasyid and R. Hermawan encrypt 8-types of sensor data using AES128-bit and transmit these encrypt data using MD5 data sensor digest [12]. Ahmed Ibrahim et al. introduced a structure that permits the protected exchange of health information among different healthcare providers. Patients approve a particular type of medical information to be retrieved, which helps to prevent any undesired leakage of medical information [13].

Mamta Puppala et al. stated the METEOR framework which consists of two components: the enterprise data warehouse (EDW) and a software intelligence and analytics (SIA) layer which facilitates a wide range of clinical decision support (CDS) systems [14]. Role-based authorized method of access is proposed by T.W. Tseng, C.Y. Yang and C.T. Liu [15]. W.D. Yu, L. Davuluri, M. Radhakrishnan and M. Runiassy proposed a security-oriented design framework (SOD) which is a three-tier architecture. In their proposed system they use SHA1 algorithm to secure login data as well as use HTTPS secure web server[16]. Khan Zeb et al. introduced a U-Prove based security technique to authenticate Telemedicine users [17]. N. Jeyanthi et al. proposed a reputation based service where the users will be accepted by a proxy server which performs entry level authentication [18]. T. Vivas, A. Zambrano, and M. Huerta proposed a digital certificate based module to secure telemedicine [19]. Fatemeh Rezaeibagha and Yi Mu proposed a new protocol for telemedicine data security[20]. J. Singh and A.K. Patel proposed web late based watermarking for telemedicine security [21]. iMedic a four-tier based security model for telemedicine proposed by Amiya K. Maji et al. which includes an extra layer to make Telemedicine system more secure [22].

III. Developed Telemedicine System

We develop a low cost, portable and secured telemedicine system for the rural and deprived people of Bangladesh. To make it more user-friendly and flexible we divide our telemedicine system into four main module. These are Local Administrator in Pharmacy, Local Doctors, Expert Doctors and Health System Administrator. The following business process diagram shows working process of the four main module of our system (Fig. 1).

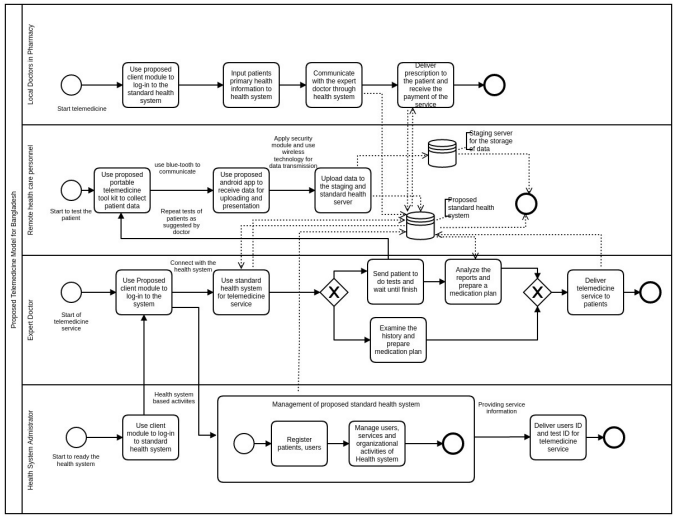

Fig. 1. Developed telemedicine system.

We have several components to complete the task for each module. The component diagram of our telemedicine module is given below (Fig. 2):

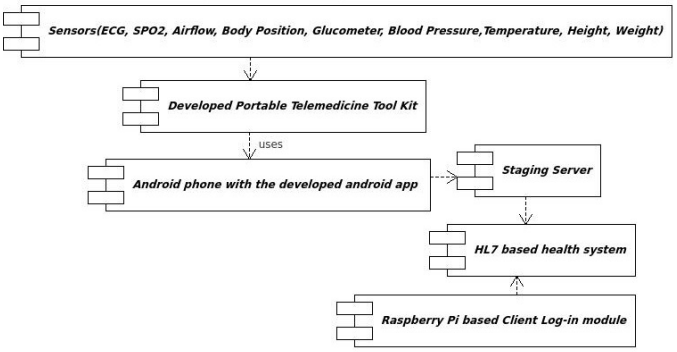

Fig. 2. Component diagram of developed telemedicine model.

In our developed Telemedicine system, every user must be registered. The health system administrator registers the expert doctors, the local pharmacy admin (a pharmacy admin is a person who works in the local pharmacy and responsible for all the local administrative work) as well as the local doctors. The pharmacy admin is responsible for the registration of the remote patients. The pharmacy admin also assigns a doctor for the patient at the time of registration. When the remote patients get registered, he/she have the patient id which is used for future correspondence. The completed registration in the system looks like Fig. 3:

When a patient needs Telemedicine service from the local pharmacy admin, he/she needs to describe his/her problems to the pharmacy admin. The pharmacy admin input all the patients data into the system using a Raspberry Pi based client login module against the respective patient ID. The expert doctor checks all the history of the patient. If needed then the expert doctor asked the pharmacy admin to do certain medical

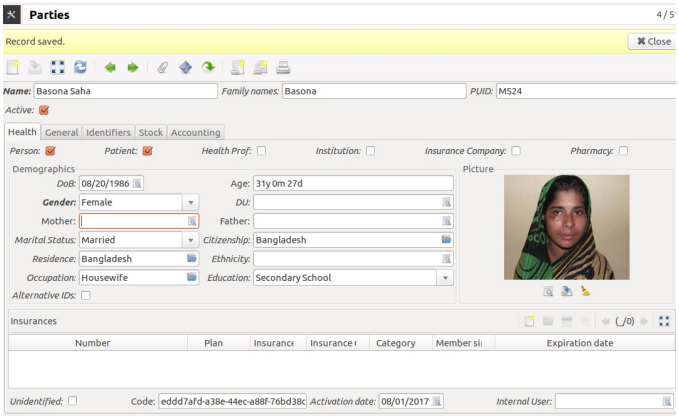

Fig. 3. Patients registration form.

checkup for the patient. After getting the request from doctors to do medical checkup for the patients, the pharmacy admin instructs local doctors to complete the prescribed task. The local doctors do the prescribed task using the develop portable Telemedicine toolkit. The toolkit includes nine types of sensors including ECG, SP02, Airflow, Body Position, Glucometer, Blood Pressure, Temperature, Height and Weight sensors. The local doctors do the checkup using a mobile application which is connected with the developed toolkit through a secured Bluetooth connection. The local doctors need to Login the Android mobile application to get the sensors data. When the sensors data received by the Android application, it showed on the application interface as follows (Fig. 4 and 5):

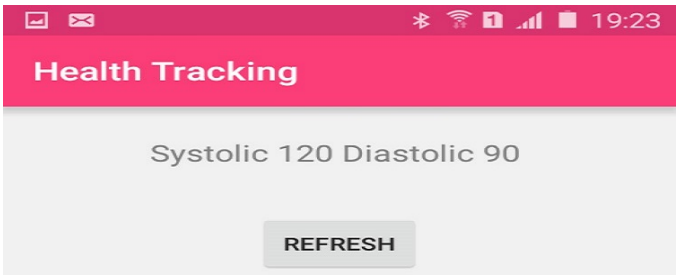

Fig. 4. Blood pressure data in mobile application interface.

The local doctors see the data and upload the data to the remote staging server against the respective patient id along with the test id. There needs to open the mobile internet connection to send the data. An encryption algorithm is implemented in the mobile application to encrypt the sensors data. The data sent to the staging server is originally an encrypted data. In the server site, there is a decryption algorithm to decrypt the data and store the original data. The expert doctors Login the HL7 based open health system called GNU health system and prescribe the patient by observing their medical test results. The remote pharmacy admin Login the system and get the prescription of the patient (Fig. 6).

After that the patients get the medicine or advice prescribed by the expert doctors from the pharmacy admin, giving a small amount of royalty fee for the prescription which is around 300 BDT. The prescription also stores in the system. Therefore, if needed then the patients get the prescription at any time. The pharmacy admin also generates invoice report, service report

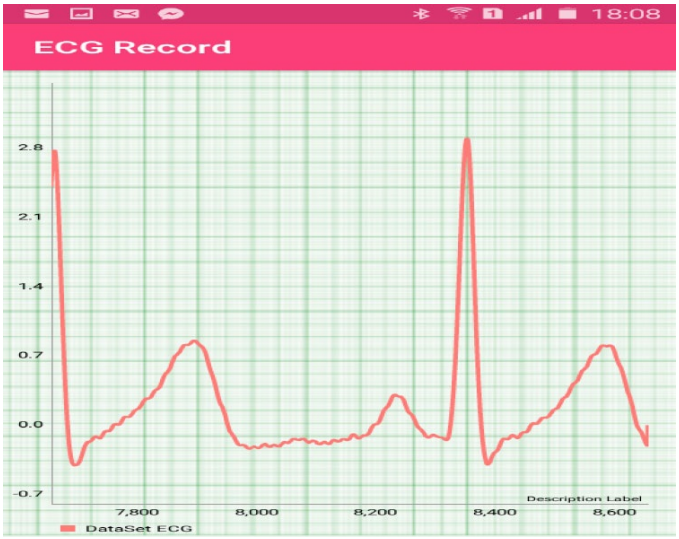

Fig. 5. ECG data in the mobile application interface.

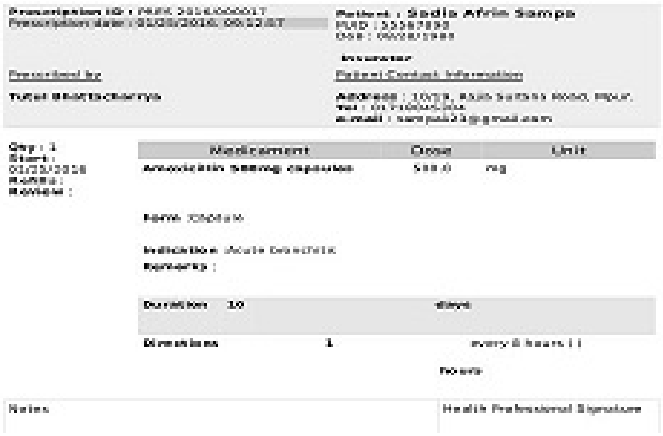

Fig. 6. Prescription of a patient.

as well as fix an appointment for the patients.

IV. Implemented Security Technique

We consider various security model to secure an authentication, authorization, transmission etc to secure our developed telemedicine system. Among them, Access Control List or ACL attach number permission to an object. Define the object accessibility as well as define which user access which section of the data [23]. Role-Based Access Control or RBAC works for a large number of people where ACL called the minimal RBAC model [24]. In Multi-Level Security, information flow between the authorized users only and only the privileged user can read the information. It also prevents unauthorized users to access the data. The most common Multi-Level Security model use for security purpose is called Bell-LaPadula model. Bell-LaPadula model checks the subject security model when a user tries to read or write on the subject as well as no object bypass any authorized users [25]. On the other hand, BIBA security model ensures the data security model by providing several access control rules. In BIBA security model, a lower level user is not permitted to request higher level user documents [26].

Not only security model but also some other organizations such as Health Insurance Portability and Accountability Act or HIPPA provide privacy, security, enforcement as well as breach notification rule to make a healthcare data to a Protected Health Information or PHI [27]. The National Authentication Service for Health or NASH provides PKI as well as Public Key Infrastructure certificate which helps users to know about their healthcare data authentication, integrity, non-repudiation as well as confidentiality [28].

Considering different security model as well different organizations security and privacy rules we divided our develop Telemedicine framework into five different sections such as application security, user authentication, web server security, database security as well as data security. The block diagram of our security measures are given below (Fig. 7):

Fig. 7. Block diagram of security framework.

A. Authentication and Application Security Layer

Authentication and Application Security Layer is designed to authenticate every user as well as provide security for the Android application. In our system, there are types of user, one is remote doctors or trained personnel, expert doctors and local pharmacy admin. There are three types of authentication needed to use this Telemedicine system.

Firstly, the Telemedicine system administrator registers the remote doctors, pharmacy admin as well as expert doctors. All these users get username and password to Login the system. The remote pharmacy admin registers the patients and assigned a doctor to their patient id.

Secondly, the remote doctors need to Login the mobile application with the same username and password which is provided when he/she register in the system before getting the patients data from sensors. **Thirdly**, a password is needed to establish the connection between toolkit Bluetooth module and mobile application. To get data from sensors we use HC-05 Bluetooth module which has a default password and username. To enrich security we set new password and username for this Bluetooth module.

To get the data from the sensors and send these data to the server we use an Android application. Therefore, reverse engineering made it possible for a hacker to find out the application data. To enrich security as well as prevent reverse engineering we will implement ProGuard. ProGuard compresses the application which saves a lot of space as well as encrypts the mobile application which makes the application code obfuscate to prevent any kind of security threats.

B. Client Layer

Client layer help users to interact with the system. It consists of both mobile application interface as well as web application interface. Mobile application layer helps remote healthcare personnel to get sensors data. On the hand, web interface help doctors to prescribe patients as well as remote doctors to get the expert doctors prescription.

- We use HL7 based open source health system called GNU [29] healthcare system for our Telemedicine system which provides a web interface for both specialized doctors and remote users, also for pharmacy admin. GNU healthcare system provides a user interface called Tryton to Login the system. Python language used to create Tryton web interface. Whereas, Tryton is a three-tier high-level customary design computer application principles. Fig. 8 describes the security measures we implement to secure the web application: We develop an Android-based mobile application to get data from the sensor. In order to get data from sensors, we implement different security measures to get data without any kind of data distortion. The end user needs to Login the Android application using username and password given at the time of registration. To secure the Login data, we implement Message Digest or MD algorithm. There are MD2, MD4, MD5, and MD6 hash algorithm. The MD2 has 18 rounds with 512 bits digest size. This hash algorithm is optimized for 8 bits computer. The MD4 has 3 rounds with 128-bits digest size and 512 bits block size. The MD5 hashing algorithm improves its security feather by adding one more round. Therefore, it has 4 round with 128-bits digest size with 512-bits block size. There are some vulnerabilities in message digest hashing algorithm but it has a little effect on MD5, even though the MD5 algorithm is faster than SHA algorithm [30]. Fig. 9 shows the block diagram of the MD5 hashing algorithm.

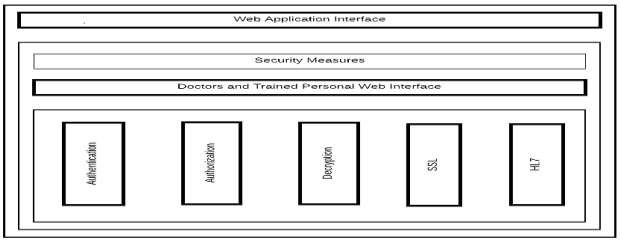

Fig. 8. Security steps implementation for web client.

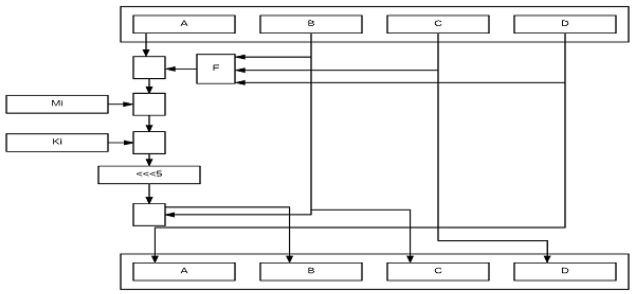

Fig. 9. MD5 hashing algorithm.

C. Data Security Layer

There are various types of encryption algorithm including symmetric, asymmetric, shared or cryptographic hash function to encrypt data. Among them, we will implement symmetric encryption algorithm named as Advanced Encryption Algorithm. There are AES-128 bits, AES-192 bits, and AES-256 bits. The AES-128 bit has 10 rounds, AES-192 has 12 rounds and AES-256 has 14 rounds, here more rounds mean more security against security attack. United States National Security Agency reviewed all AES algorithm and recommend AES-256 and AES-192 to secure classified documents secure [31]. Not only that Fast Healthcare Interoperability Resources or FHIR also recommend AES algorithm to secure data. Therefore we will implement AES-256 to encrypt our sensors data.

We implement AES-256 bit encryption in mobile application and encrypt data as well as send it to the remote web server. In the server site, we decrypt the data with the same algorithm and save data in the database.The following block diagram shows AES algorithm (Fig. 10).

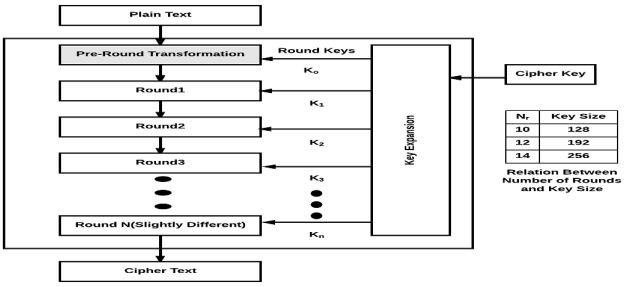

Fig. 10. AES algorithm block diagram.

D. Middle Layer

Apache Tomcat server is the main part of the middle layer where the GNU healthcare system is installed. GNU health provides an interface called Tryton. These server responses for HTTPS request from the mobile application as well as web clients.

E. Data Layer

The data layer consists of web database where the patient's data is stored. We use GNU healthcare which uses PostgreSQL database which an open source most secured database [32]. The remote healthcare personnel sends the data to the web server with a valid patients id. The patient's data store in the database with a valid and unique patients id with examination id.

There are several authentications procedure to get data from the sensors and send it to the server. Moreover, there are also some authentication steps to get the prescription from the doctors and make it available to the patients. The overall security steps with workflow from the Bluetooth connection between mobile application and toolkit to the data store in web server is represented in a flow chart as follows (Fig. 11):

After getting the data from remote doctors the expert doctors review the data. Therefore, he/she provide the prescription

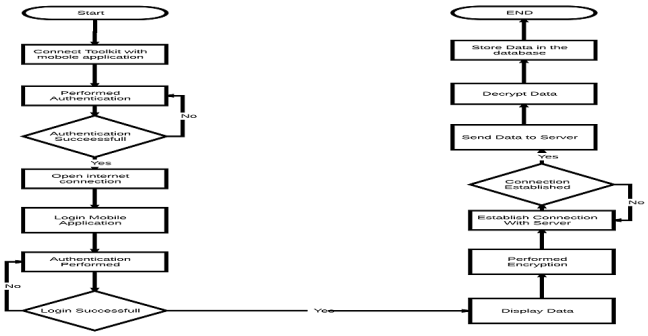

Fig. 11. Workflow from remote doctors to expert doctors.

for the respective patient. The remote doctor Login the system and find the prescription as well as provide medicine to the patients. The following Fig. 12 shows the complete workflow as well as proposed security steps.

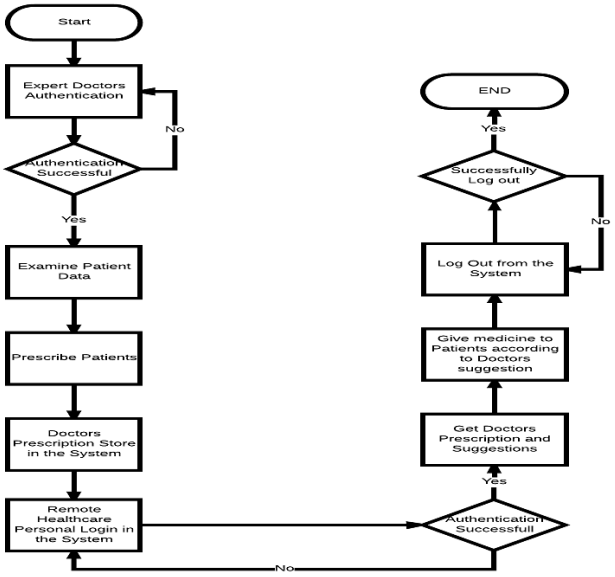

Fig. 12. Workflow from expert doctors to remote doctors.

V. Implementation and Result

The most common threats to data privacy and security include data theft, unauthorized access, improper disposal of data, data loss, hacking IT incidents and more. In this section, we have implemented the security measures to prevent unauthorized access, improper disposal of data and data loss. The list of section where the security measures are implemented are given below:

- Bluetooth Connection
- Mobile Application User Authentication
- Mobile Application Security
- Sensors Data Security
- Server Security

Therefore, Fig. 13 shows the sequence diagram of the Telemedicine system on which the security measures are implemented is as follows:

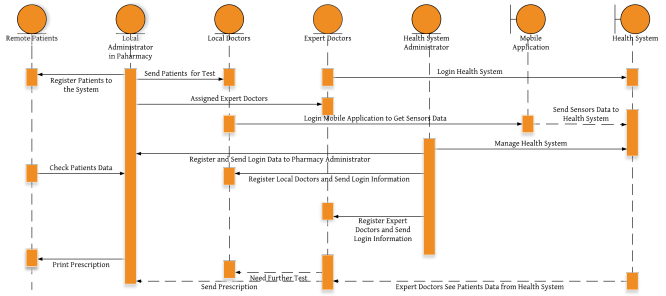

Fig. 13. Sequence diagram of the telemedicine system on which the security measures are implemented.

This chapter also describes the results of the implemented security technique. Fig. 14 shows the graphical representation of security implementation.

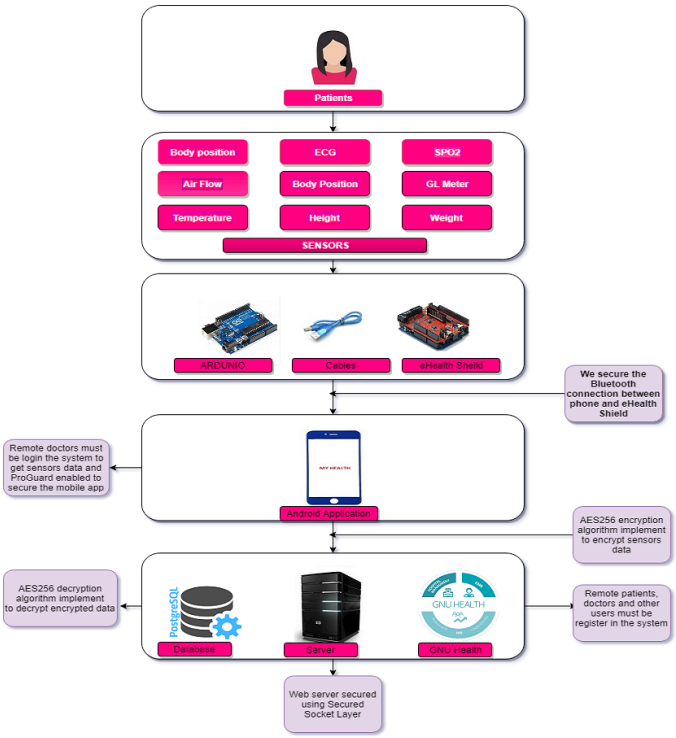

Fig. 14. Security techniques applied in different sections of the developed telemedicine system.

A. Enhancing Bluetooth connection security

HC-05 has a default password which is '0000' or '1234' and default Bluetooth name 'HC-05'. To improve security we change the default name to "My Telehealth" and change default password to a 15 digit password with a combination of numbers, digits, uppercase letter, lowercase letter and special characters using AT command. Here, AT command stands for Attention Command used to change default Bluetooth setting. For a 4 digit number password, there is 10,000 possible combination. Therefore, for a 15 digit password with a combination of number, character, special character, lowercase letter, and the uppercase letter, a total 70 character, the possible combination is 721480692460864. This password combination makes it unbreakable.

To set the password, first of all, we change the HC-05 mode to attention mode. Therefore, we press a enable button just on the opposite side of the HC- 05 until the lid shows the indication. The indication shows that it is enabled for attention mode. Then, we upload a blank code to the module. To set the Bluetooth name we write the following code in Arduino serial monitor.

AT+NAME= JU-TELEMEDICINE-CENTER

And set password we write:

AT+PSWD= Telemedicineju1

After successfully configure the Bluetooth, the authentication process looks like as follows where Fig. 15 shows the available Bluetooth device, Fig. 16 shows the Bluetooth connection asked for a password, Fig. 17 shows the entering the password for connection and the last Fig. 18 shows the successful connection.

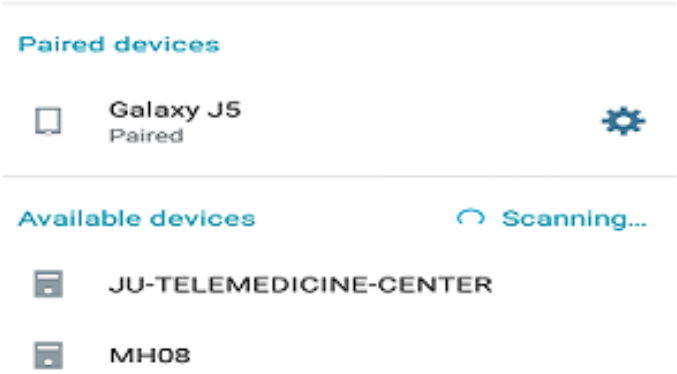

Fig. 15. Available Bluetooth devices.

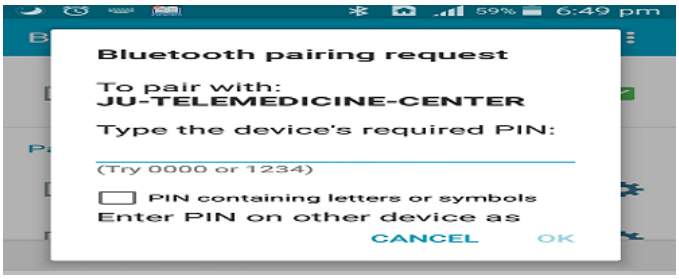

Fig. 16. Bluetooth connection asked for password.

B. Authenticating Remote Doctors in the Mobile Application

We create a Login module for the remote doctors in the mobile application. Every time when remote doctors open the mobile application to get the sensors data, he/she must be login the mobile application with their username and password that

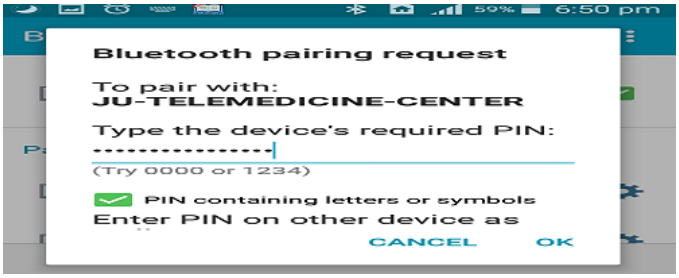

Fig. 17. Entering the password in the pop-up window.

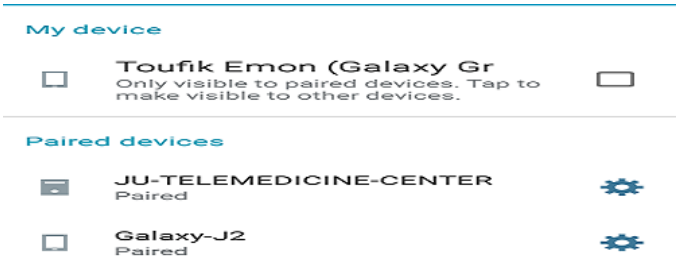

Fig. 18. Successful connection.

was given at the time of registration to use the app. Figures below shows the authentication process of the application (Fig. 19 to 21).

We also secure the username and password of the remote doctors by implementing MD5 hashing algorithm. There are several steps to implement the hashing algorithm. The implementation steps are:

- Step-1: Append Padding Bits

In this step, we extended the message so that the message length is similar to 448, modulo 512. The message is padded therefore it is just 64 bits which is a multiple of 512 bits. To complete the padding, first of all, a single "1" bit is added to the message, and then "0" bits are appended so that the length of bits of the full message becomes congruent to 448, modulo 512. At least one bit and at most 512 bits are appended.

- Step-2: Append Length

A 64-bit symbol of the message before the padding bits were added is appended to the result of the step-1. In the message, the bit is greater than 2^64, then only the low-order 64 bits of the message is used.

- Step-3: Initialize MD5 Buffer

This step includes the initialization of 34 bits four-word buffer A, B, C, D with the hexadecimal number where A= 01 23 45 67, B= 89 ab cd ef, C= fe dc ba 98 and D= 76 45 32 10.

Fig. 19. Authentication interface.

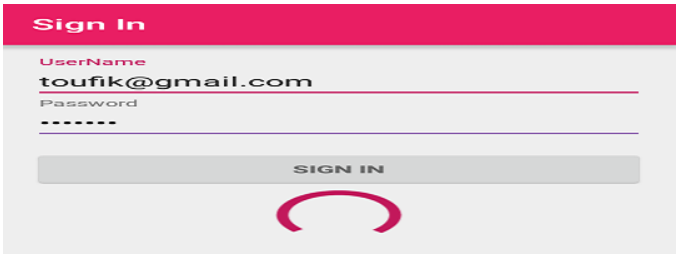

Fig. 20. Entering username and password.

Fig. 21. Successful login.

- Step-4: Process Message in 16-Word Blocks

This step calculates the MD5 hashing.

- Step-5: Output

This step includes the implementation of MD5 hashing.

Fig. 22 and 23 shows the results of the MD5 implementation.

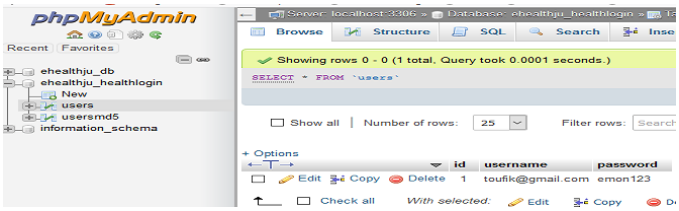

Fig. 22. Before implementation of MD5 hashing.

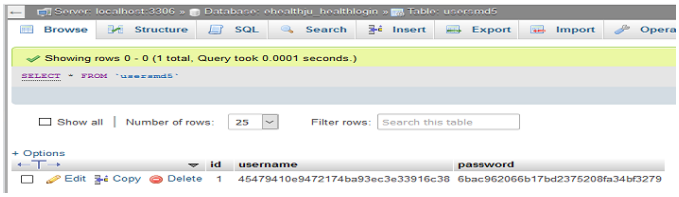

Fig. 23. After implementation of MD5 hashing.

C. Mobile Application Security

Securing the mobile application is one of the most important tasks. We secure our Android-based mobile application

from hackers as well as prevent to struct data from our mobile application by enabling ProGuard. To enable ProGuard we have to open our mobile application in Android Studio. Then, find the file named build.gradle(Module: app) and open the file. In the buildTypes section, we found minifyEnabled false. Now to enable ProGuard, we change the state of minifyEnabled to true state which looks like minifyEnabled true. Now we generate a signed APK for our application. After successful generation of signed APK, there is some change in our application file. Though enabling ProGuard changes variable names as well as the class name, therefore, we need to write the following code in "proguard-rules.pro" file to enable the application working properly.

-ignore warnings

-keep class * { public private *; }

D. Implementation of AES 256 bit Encryption Algorithm for Sensors Data Security

To secure the sensors data we implement AES256 bit encryption algorithm with our modified vector size and key size. AES256 bit encryption secure the sensors data during transmission. The implementation steps of AES-256 encryption are:

- Step-1: Round keys are originated from the cipher key using Rijndael's schedule.
- Step-2: First Round. Each byte of the state is consolidated with the round key using bitwise XOR in add round key step. Each byte is substituted with another according to a lookup table in a non-linear replacement step called Sub Byte. In this step, each row of the state is shifted cyclically a certain number of steps. This called Shift Rows step. Mix Columns is a Mixing operation in Mix Columns step where the columns of the state, combining the four bytes in each column. Round Key is added in this step.
- Step-3: 2nd round to 13th Round

The following steps are repeated in this step: Sub Bytes. Shift Rows. Mix Columns. Add Round Key.

- Step-4: Final Round

In the final step, the following steps are repeated: Shift Rows. Mix Columns. Add Round Key.

On the server side, we implement AES256 decryption algorithm. Therefore, in the server the original data stored. Fig. 24, 25 and 26 shows the result of the implementation result of AES256 bit encryption.

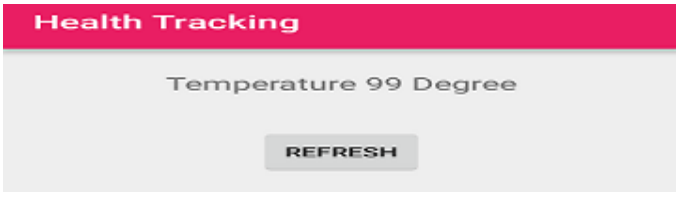

Fig. 24. Before AES 256 bit encryption.

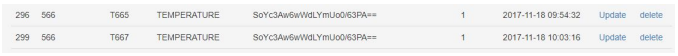

Fig. 25. Encrypted data in transmission.

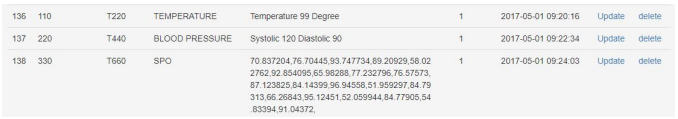

Fig. 26. Decrypted data in the server.

E. Server Security

All the sensors data stored in the remote staging server. Therefore, it is more important to secure the web server. Following the recommendation of Health Level 7, we implement the Secure Socket Layer or SSL-256 bit to secure our server. After implementation of SSL, our server URL link change from HTTP to https which means our server is secured. The implementation procedures of SSL is described below:

There are some prerequisites for SSL. These are certificates from the certificate authority (CA), registered domain name, web server (Apache HTTP, Nginx, HAProxy, or Varnish server).

There are three types of SSL server and these are single domain, wild card, and multiple domains.

1) Generating private key: For our system, we use single domain SSL certificate. To install SSL in our system, we buy SSL certificate from the certificate authority and get a.crt file bundles from them. After that, we generate a private key using OpenSSL which is called ehealthju.com.key and CSR file called ehealthju.com.csr. Therefore, we run the following command int the command line:

Openssl req -newkey rsa: 2048 -nodes -keyout ehealthju.com.key -out ehealthju.com.csr

After that, the following information is shown in the prompt. We should care about the common name field because common name field is the field that we put in our SSL certificate.

- Country Name (2 letter code) [AU]: BD
- State or Province Name (full name) Some-State: Dhaka
- Locality Name (eg, city) []: Dhaka
- Organization Name (eg, company) [Internet Widgits Pty Ltd]: ehealthju
- Organizational Unit Name (eg, section) []: Jahangirnagar University
- Common Name (e.g. server FQDN or YOUR name) []:ehealthju.com
- Email Address []:uzzalcseju@gmail.com

This will generate a .key and .csr file. The .key file is the private key and should be kept secure. The .csr file will send to the CA to request SSL certificate.By using the generated private key and CSR file, we send to request your CA's to

provide the SSL certificate. They will validate our domain by sending an email.

2) Certificate Installation: We made a backup of our configuration file by copying it using these commands:

- cd /etc/apache2/sites-available
- cp 000-default.conf 000-default.conf.orig

Then open the file for editing:

- sudo vi 000-default.conf

We change the <VirtualHost *:80> entry to <VirtualHost *:443> Then add the ServerName directive. ServerName ehealthju.com Then add the following lines to specify certificate and key paths:

- SSLEngine on
- SSLCertificateFile /home/user/ehealthju.com.crt
- SSLCertificateKeyFile /home/user/ehealthju.com.key

Then we add the following code at the top of the file and then save the file.

- VirtualHost *:80
- ServerName ehealthju.com
- Redirect permanent https://ehealthju.com/
- VirtualHost

After that, we enable the Apache SSL module by running this command:

- sudo a2enmod ssl

Then we restart Apache to load the new configuration and enable TLS/SSL over HTTPS using the following command

- sudo service apache2 restart

After restart the Apache server, it converted to HTTPS instead of HTTP and our server is secured. Fig. 27 shows status of our server before implementation of SSL and Fig. 28 shows the status of our server after SSL implementation.

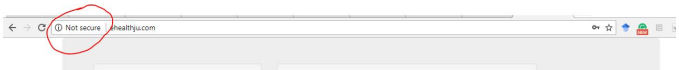

Fig. 27. Before SSL implementation.

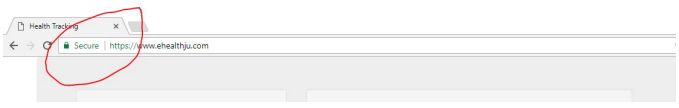

Fig. 28. After implementation of SSL.

VI. Security Analysis and Future Work

In this paper, we discussed the security framework for our telemedicine system which is developed for the unprivileged people of Bangladesh. To secure this telemedicine system we find five section. Therefore, the entire security of our developed telemedicine system depends on the security improvement of this five section. We have applied the specific solution of these five security holes. First of all, a security technique must be needed to establish a secure connection between toolkit and mobile application. In this paper, we have introduced an authentication system for Bluetooth connection which improves the security. After that, we implemented an authentication system to use the mobile application, so that no unauthorized user use the system. It makes the mobile application more secure and prevents any type of data breaches.

Healthcare data is the more vulnerable because the medical record is more lucrative to hackers than any other data as like as credit card numbers [33]. Therefore, we improve the healthcare data security of our system by encrypting data with the AES-256 bit which the most advanced encryption algorithm. We decrypt the data in the server site using the same algorithm. Moreover, we have introduced registration for expert doctors, remote healthcare personnel as well as patients so that no unauthorized user use the system. This protects the system from any kind of security vulnerability. All these steps are taken following the instruction of Health Level 7 and FHIR.

Although there are many measures to secure our telemedicine system there are still some security threats. In our developed telemedicine system, we receive sensor data using mobile application. Before we send data to remote server, the remote user has the opportunity to observe data. Moreover, the end user gets the patients prescription before the patients get the prescription. In both cases, there are possibilities to breach data and prescription. There are some other problems as like as there are third parties like lab assistant who investigate the data. Sometimes there are different lab tests, therefore, lab assistant may change. In this case, there are also possibilities of data breaches.

Considering all these issues, in future, we improve our security model so that patients can easily get their prescription as well develop a module to authenticate the lab assistant. As well as our security model will play a vital role in the widespread use of developed telemedicine service so that a secured telemedicine service can be given to the remote poor people of our country at low cost.

VII. Conclusion

In this paper, we implement a security framework to secure these principles and prevent the security breaches. The implemented security framework follows the recommendation of HL7 and FHIR, also consider the HIPPA and NEHTA recommendation for security and privacy of a health system. This paper also shows different advantages of our security framework. The implemented security framework is cost effective and efficient in performance. It can, therefore, be decided that the implemented security framework implements competent measures for real-time secure telemedicine data transmission.

ACKNOWLEDGMENT

The authors would like to thank all the faculty and employee of Computer Science and Engineering Department and ICT Division, Ministry of post, telecommunication, and information technology, Bangladesh for the full financial support granted to this research.

REFERENCES

[1] U. K. P. Mohammad Zahidur Rahman, Israt Jahan, "Ardunio based telemedicine system for bangladesh," *Jahangirnagar University Journal of Science, Volume 40, No: 2*, 2017.

[2] M. F. F. Khan and K. Sakamura, "Security in healthcare informatics: design and implementation of a robust authentication and a hybrid access control mechanism," in *Communications, Computers and Applications (MIC-CCA), 2012 Mosharaka International Conference on*. IEEE, 2012, pp. 159–164.

[3] V. Liu, W. Caelli, L. May, and T. Sahama, "Privacy and security in open and trusted health information systems," in *Proceedings of the Third Australasian Workshop on Health Informatics and Knowledge Management-Volume 97*. Australian Computer Society, Inc., 2009, pp. 25–30.

[4] P. T. Akkasaligar and S. Biradar, "Secure medical image encryption based on intensity level using chao's theory and dna cryptography," in *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Dec 2016, pp. 1–6.

[5] M. J. Chang, J. K. Jung, M. W. Park, and T. M. Chung, "Strategy to reinforce security in telemedicine services," in *2015 17th International Conference on Advanced Communication Technology (ICACT)*, July 2015, pp. 170–175.

[6] I. Chiuchisan, D. G. Balan, O. Geman, I. Chiuchisan, and I. Gordin, "A security approach for health care information systems," in *2017 E-Health and Bioengineering Conference (EHB)*, June 2017, pp. 721–724.

[7] C. Fu, Y. Lin, H. y. Jiang, and H. f. Ma, "Medical image protection using hyperchaos-based encryption," in *2015 9th International Symposium on Medical Information and Communication Technology (ISMICT)*, March 2015, pp. 103–107.

[8] B. Halder and S. Mitra, "Modified watermarked ecg signals by using adaptive normalization factor," in *2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS)*, July 2015, pp. 434–439.

[9] C. Han, L. Sun, and Q. Du, "Securing image transmissions via fountain coding and adaptive resource allocation," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, May 2016, pp. 1–5.

[10] R. M. Seepers, C. Strydis, I. Sourdis, and C. I. D. Zeeuw, "Enhancing heart-beat-based security for mhealth applications," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 254–262, Jan 2017.

[11] U. K. Prodhan, M. Z. Rahman, and I. Jahan, "Design and implementation of an advanced telemedicine model for the rural people of bangladesh," *Technology and Health Care*, no. Preprint, pp. 1–6, 2018.

[12] A. Sudarsono, P. Kristalina, M. U. H. A. Rasyid, and R. Hermawan, "An implementation of secure data sensor transmission in wireless sensor network for monitoring environmental health," in *2015 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, Oct 2015, pp. 93–98.

[13] A. Ibrahim, B. Mahmood, and M. Singhal, "A secure framework for medical information exchange (mi-x) between healthcare providers," in *Healthcare Informatics (ICHI), 2016 IEEE International Conference on*. IEEE, 2016, pp. 234–243.

[14] M. Puppala, T. He, X. Yu, S. Chen, R. Ogunti, and S. T. Wong, "Data security and privacy management in healthcare applications and clinical data warehouse environment," in *Biomedical and Health Informatics (BHI), 2016 IEEE-EMBS International Conference on*. IEEE, 2016, pp. 5–8.

[15] T. W. Tseng, C. Y. Yang, and C. T. Liu, "Designing privacy information protection of electronic medical records," in *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dec 2016, pp. 75–80.

[16] W. D. Yu, L. Davuluri, M. Radhakrishnan, and M. Runiassy, "A security oriented design (sod) framework for ehealth systems," in *2014 IEEE 38th International Computer Software and Applications Conference Workshops*, July 2014, pp. 122–127.

[17] K. Zeb, K. Saleem, J. Al Muhtadi, and C. Thuemmler, "U-prove based security framework for mobile device authentication in ehealth networks," in *e-Health Networking, Applications and Services (Healthcom), 2016 IEEE 18th International Conference on*. IEEE, 2016, pp. 1–6.

[18] N. Jeyanthi, R. Thandeeswaran, and H. Mcheick, "Sct: Secured cloud based telemedicine," in *The 2014 International Symposium on Networks, Computers and Communications*, June 2014, pp. 1–4.

[19] T. Vivas, A. Zambrano, and M. Huerta, "Mechanisms of security based on digital certificates applied in a telemedicine network," in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug 2008, pp. 1817–1820.

[20] F. Rezaeibagha and Y. Mu, "Practical and secure telemedicine systems for user mobility," *Journal of biomedical informatics*, 2017.

[21] J. Singh and A. K. Patel, "An effective telemedicine security using wavelet based watermarking," in *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Dec 2016, pp. 1–6.

[22] A. K. Maji, A. Mukhoty, A. K. Majumdar, J. Mukhopadhyay, S. Sural, S. Paul, and B. Majumdar, "Security analysis and implementation of web-based telemedicine services with a four-tier architecture," in *2008 Second International Conference on Pervasive Computing Technologies for Healthcare*, Jan 2008, pp. 46–54.

[23] K. C. Nelson and M. A. Noronha, "Facilitating ownership of access control lists by users or groups," Jul. 4 2017, uS Patent 9,697,373.

[24] S. Li and L. Da Xu, *Securing the Internet of Things*. Syngress, 2017.

[25] "Mls introduction — cryptosmith," https://cryptosmith.com/mls/intro/.

[26] O. N. Ely, "Secure computing system," Sep. 19 2017, uS Patent 9,767,297.

[27] L. M. Boyle and D. M. Mack, *HIPAA: a guide to health care privacy and security law*. Wolters Kluwer, 2017.

[28] K. K. Htat, P. A. Williams, and V. McCauley, "Security of eprescriptions: data in transit comparison using existing and mobile device services," in *Proceedings of the Australasian Computer Science Week Multiconference*. ACM, 2017, p. 56.

[29] L. F. Martín and G. Solidario, "Gnu health: A free/libre community-based health information system." in *OpenSym (Companion)*, 2016, pp. 11–1.

[30] S. P. Dwivedi, "Message digestion," 2017.

[31] D. Kraus and M. Welschenbach, "Advanced encryption standard," 2016.

[32] Z. Shan, "A study on altering postgresql from multi-processes structure to multi-threads structure," *arXiv preprint arXiv:1609.09062*, 2016.

[33] S. I. Khan and A. S. M. Latiful Hoque, "Digital health data: A comprehensive review of privacy and security risks and some recommendations." *Computer Science Journal of Moldova*, vol. 24, no. 2, 2016.