A Testbed for Implementing Lightweight Physical Layer Security in an IoT-based Health Monitoring System

Ahmed Mohamed Hussain*, Khalid Abualsaud*, Elias Yaacoub*, Tamer Khattab†,
Abdurrazzak Gehani‡, Mohsen Guizani*
*Computer Science and Engineering Department
†Electrical Engineering Department
Qatar University
‡Al-Ahli Hospital, Ahmed Bin Ali Street, P.O. Box 6401
Doha, Qatar
Email: ahmed.hussain@qu.edu.qa

***Abstract*—Telemedicine is a technique that allows patients to have health-related consultations without the need to be physically present in the hospital through phone and video calling technologies. In recent years, researchers have made many contributions to reform and facilitate better telemedicine services through the use of body area networks or wireless body area networks. This paper presents a testbed where we implement a lightweight physical layer security scheme, using gray code, on an IoT-based health monitoring system to secure transmitted patient readings while preserving its clinical features. We address several existing adversarial scenarios, where an adversary can eavesdrop on the packets and infer their content using some of the existing packet inspection techniques. We prove that the introduced physical layer security scheme effectively protects the patient transmitted data, even if read by an adversary.**

***Index Terms*—Telemedicine, eHealth, IoT, Physical Layer Security, Cyber-Physical Systems**

I. Introduction

Wireless Body Area Networks (WBANs) consists of several low-power sensor nodes, each of which acquires a specific physiological parameter from the patient. These nodes act as a bridge between the patient and the hospital. WBAN is designed to capture important aspects of the patient's health status and detect abnormalities. The use of these data is considered to lower mortality. Reliable WBAN requires sensors with the following characteristics: portable, lightweight, and low on power consumption [1]. Monitoring patients with long-term chronic disease, namely heart diseases, was (and still) significantly challenging during the COVID-19 pandemic. Due to the nature of the novel virus, many patients were not able to attend their scheduled meetings because of the imposed lockdown or fearing contracting this deadly airborne virus. Thus, hospitals conducted the majority of their appointments either online or through phone calls [2], [3]. However, for certain diseases, diagnosis and monitoring patients cannot be conducted unless the patient is physically present. Tracking and monitoring of patients are made possible by taking advantage of the latest Internet of Things (IoT) [4] and Body Area Network (BAN) technologies [5], namely wearable devices. Wearable devices include built-in sensors with dedicated processing power and dedicated features to track physiological changes, such as heartbeat, blood pressure, breathing, and body temperature. Wearable devices can also connect to other devices, such as computers, smartphones, and tablets, by leveraging embedded technologies such as Bluetooth, ZigBee, LTE, and WiFi [6]. They are commonly used in sports and to track human health [7]. Nevertheless, those devices are either power-constrained or not tailored to diagnose and monitor patients with long-term chronic diseases. Moreover, security issues concerning the user data always occur, where patients' data are left unprotected, leaked, or disseminated [8]. Thus, implementing a secure architecture against nowadays adversarial attacks is a significant challenge that requires considering the basic security principles: *confidentiality*, *integrity*, and *availability*.

Contribution. In this paper, we present a testbed implementing a secure, lightweight, and portable health monitoring system, with the focus on securing data sent from the patient to the hospital by applying the physical layer's security bit flipping technique namely gray code, while preserving the critical clinical features, and transmit the data based on the channel awareness. Besides, we address and adopt two existing adversarial models with two significant attack scenarios. The attacker can infer the link between the two communicating nodes while being located remotely in the Cloud or being physically located close to the patient. As well as being able to disclose the content of the packets sent by the patient. Finally, we prove that the proposed system with physical layer security (gray code) applied is capable of countermeasure these scenarios.

Paper Organization. The remainder of this paper is organized as follows: Section II highlights and summarizes the recent contributions in the field of secure eHealth and IoT, while Section III provides an overview of the general eHealth system architecture. Section IV present the adversarial models and their associated attack scenarios. Section V addresses the proposed solution that preserves patient privacy, as well as the experimental results. Finally, Section VI wraps the findings in this paper and draws some future directions.

II. Related Work

An electrocardiogram (ECG) records the electrical signals of the patient heart and detect if normal or abnormal [1], [9], [10]. Due to the essential technical flaws and limitations of the existing remote ECG monitoring systems, it becomes prone to security and privacy threats.

Ibaida and Khalil [11] presented a system where body sensor nodes are used to monitor the patient's physiological signals such as body temperature, electrocardiography (ECG) signal, blood pressure, and glucose level. Via Bluetooth, readings are transmitted to the patient's PDA, where steganography is applied to data, using wavelet transformation to obfuscate the patient's information. Steganography results in extensive computations [12], and thus, requires more computational power. This approach is not applicable when performing diagnosis or monitoring a patient due to the high computational power.

Basu et al. [13] proposed a smart heart monitoring system using Raspberry Pi 3B+. The system uses noninvasive sensors to monitor patients' vital parameters such as heart rate, SpO$_2$, and body temperature. The Pi is used for data acquisition, processing, and transmission over the WiFi to the hospital/physician using MathWorks cloud. The patient data are sent over WiFi, where Socket Secure Layer (SSL) and Transport Layer Security (TLS) is applied. The data is sent to a cloud storage service named ThingSpark. Relying only on SSL or TLS for securing transmitted data is not enough to prevent an adversary from disclosing the packet content [14].

Ghosh et al. [15] developed a real-time encoding scheme that performs iterative and approximation of wavelet coefficients for the sparse encoding the ECG signal. This method is used to reduce the bandwidth and energy consumption of the WBSN devices. The encoding scheme compresses the ECG signal while still preserving the essential clinical features. The experiment was conducted on a real-time microcontroller based IoT platform that serves as an end-to-end WBSN system. Experimental results show that at a system-level, the decrease in energy consumption is 96%, with an imperceptible impact of 2% on the ECG signal quality.

Zhao et al. [16] illustrated a portable system implementation based on a low power ECG acquisition system named CareON, which uses the ECG acquisition chip ADS1292R. CareON extracts the ECG signals steadily and accurately, then sends them to the Cloud for analysis using 4G mobile technology.

III. System Architecture

Figure 1 illustrates the system architecture, which consist of three main components: *Patient* (Client) Side, *Cloud*, *Hospital* (Server) Side. Each component has its own role described as following:

Patient side. Is an IoT device that has multiple biomedical sensors attached to it. As a client, it performs data collection, processing, and finally, transmission of the data.

Hospital side. A secure server that actively receives patients medical readings, stores them securely according to the industry standards, and share them with the physician whenever requested.

Cloud. Serves as a gateway for exchanging data between the patient device and hospital.

Since the presented architecture is aimed towards portability, the patient device is battery operated (in our case via power-bank) and connected to the Cloud through a WiFi link. At the same time, the hospital server is connected through a physical link. Visualizing the patient data machine is also assumed to be connected through the WiFi when requesting and visualizing the data.

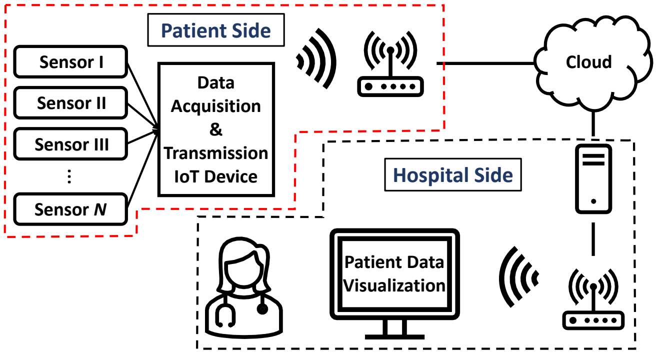
Patient Side
Sensor I
Sensor II
Sensor III
⋮
Sensor *N*
Data Acquisition & Transmission IoT Device
Cloud
Hospital Side
Patient Data Visualization

Fig. 1. Overview of a Portable Telemedicine System Architecture

IV. Adversarial Model(s)

In this section, we describe existing adversarial scenarios that could be set-up by an adversary. We focus on two separate adversarial scenarios, to disclosing the packet content of the patient data with the existing packet inspection techniques. We highlight that the adversary intention is to eavesdrop and visualize the content of the packet passively.

First, the adversary is located within the Cloud. Secondly, the adversary is physically within the perimeter and reasonable proximity of the patient. The adversary is assumed to be able to perform traffic collection and analysis in both scenarios.

A. Scenario I: Adversary in the Cloud

We consider that the Cloud is compromised by the adversary [17] and his actual location is unknown, i.e., anywhere within the Cloud. We assume that the adversary is aware of the patient's IP address, and ports used for communications. Given this information, we consider the adversary passive and perform a wide array of attacks, specifically: *Man in the Middle Attack*. By intercepting and capturing the flow of packets generated by the patient, the adversary can disclose the packet content using packet inspection techniques [14]. Figure 2 illustrates the adopted adversarial setup.

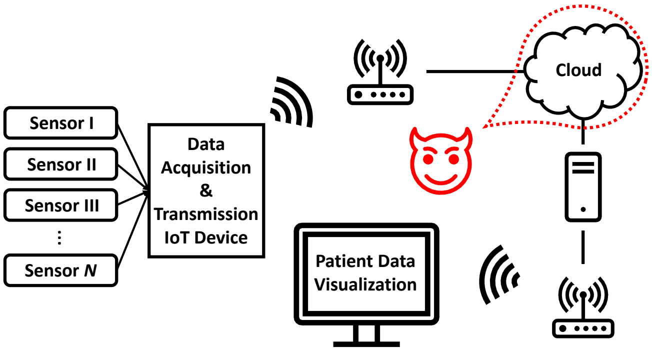
Sensor I
Sensor II
Sensor III
⋮
Sensor *N*
Data Acquisition & Transmission IoT Device
Patient Data Visualization
Cloud

Fig. 2. Adversarial Set-up in Scenario I, where the cloud is compromised by the adversary

B. Scenario II: Adversary within the Patient Proximity

This scenario is adopted from [18], and we consider that the adversary is at the patient location. Figure 3 illustrates the adversarial set-up when the adversary is positioned with a certain proximity of the patient (several meters away). The adversary is presumed to have obtained the IoT device's MAC address that acquires and transmits the patient readings. Additionally, the adversary knows which access point the patient device is connected to. We emphasize that the adversary is only eavesdropping on the WiFi link between the IoT device and the access point. Thus, the adversary is capable of reading the data exchanged between the IoT device and access point based on the adopted aforementioned set-up we discussed previously. With the appropriate tools and configurations, the adversary can either deploy an IoT based logger or use a laptop to log the transmitted data. The adversary will then be able to disclose the packet content and visualize it.

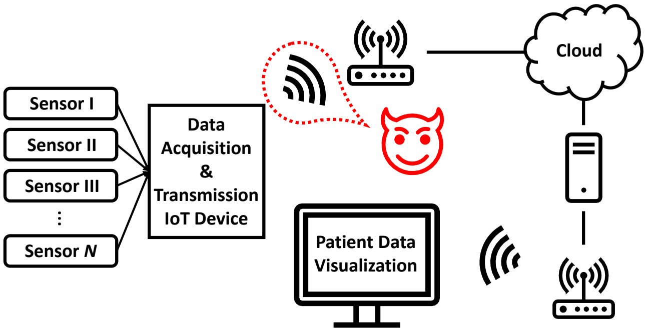
Sensor I
Sensor II
Sensor III
⋮
Sensor *N*
Data Acquisition & Transmission IoT Device
Patient Data Visualization
Cloud

Fig. 3. Adversarial Set-up in Scenario II, where the adversary is located near the patient

V. Proposed Solution

In this section, we illustrate and describe the proposed system architecture.

We propose a secure, lightweight, and portable solution that acquires several readings, applies physical layer security (gray code), and transmit the readings over-the-air. We considered the classical client–server architecture, where the client is a battery-powered IoT device held by the patient, and the hospital manages the Server. The acquired readings are transmitted from the patient network to the Cloud then to the hospital server, where the physician can view and visualize the readings. For the client-side, we adopted HealthyPi v4 shield [19] as it provides several biomedical readings and attached it to Raspberry Pi 4 Model B to acquire, secure (based on the channel quality), and transmit the readings. We consider any changes in the Received Signal Strength Indicator (RSSI) indicates that an adversary is eavesdropping on the WiFi link. Based on the channel quality, the Pi will either apply or not apply the bit-shifting technique, and finally, send the data. This information is included in the transmitted data to inform the receiver (Server) if physical layer security is applied to the data or not. The following subsections describe the exact functionality of these components.

A. HealthyPi

HealthyPi v4, shown in Figure 4, comes with three different biomedical sensors: ECG, SpO_2, and Temperature. The shield is equipped with an ESP32, a low power and cost microcontroller, for data collection and transmission over WiFi or Bluetooth. We used the shield serial pins to transmit the data when attached to the mini-computer to fit our proposed solution.

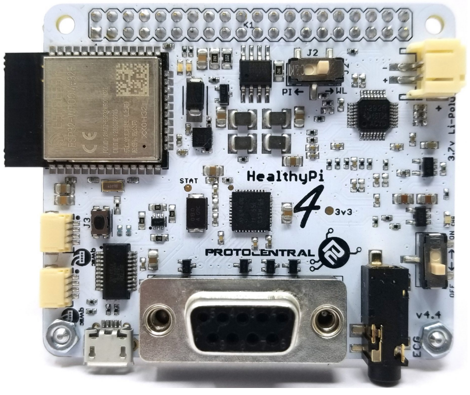

Fig. 4. HealthyPi–V4

B. Raspberry Pi

The Raspberry Pi (shown in Figure 5) acquires, applies physical layer security techniques, and transmits the data. Furthermore, it applies security based on the wireless channel quality by measuring the RSSI between itself and the patient Access Point. First, the data is continuously sent through a serial connection by the shield and read by a Java application running on the Pi. Secondly, before transmitting the data, the Raspberry Pi measures the RSSI between itself and the access point. The RSSI value is compared to a predefined unknown (to the adversary) threshold (α). In case the RSSI is less than

α, the data is transmitted without applying any security. If the RSSI is greater than α, security is applied to the data and then transmitted [20]. For the receiver to know if the security is applied or not, we include a field with the data named *encryption flag* to indicate any changes. Algorithm 1 illustrates the exact steps for data is transmitted. We define α, which represents a specific RSSI value.

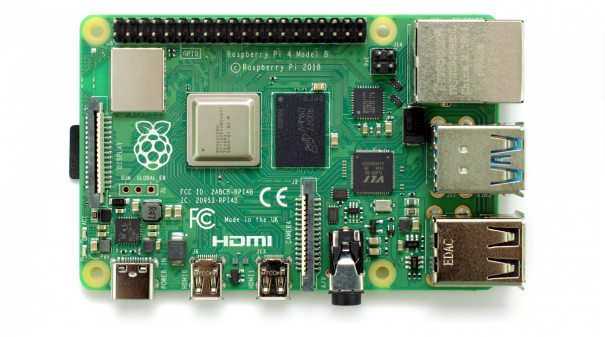

Fig. 5. Raspberry Pi 4

Algorithm 1: Client (Raspberry Pi)

```
RSSI = measureRSSI();
while true do
    acquireData();
    if RSSI < α then
        transmit();
    end
    if RSSI > α then
        secureData();
        transmit();
    end
end
```

Functions implemented on the Raspberry Pi.

Measure RSSI. As the name suggests, this function is used to measure the RSSI between the Raspberry Pi and the access point.

Acquire Data. This function is used to acquire the data read by the shield.

Secure Data. This function is used to apply the physical layer security, gray code, on the data acquired for the HealthyPi. Additionally, the *Encryption Flag* is set to 1 in case gray code is applied. Alternatively, left as its default value (0). Results show that this approach leads to strong confusion for a potential eavesdropper. In fact, when around 50% of the data gets flipped, the attacker becomes unaware of what is actual and what is fake data [20].

Figure 6 illustrate the data frame structure of the packet generated by the Raspberry Pi and sent to the Server.

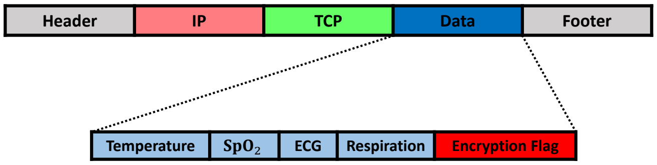

Fig. 6. Data frame fields within the transmitted packet

C. Hospital Server

The hospital server (receiver) runs a TCP Java socket application. Once a client is connected, the data is continuously received, and each value is set to the following corresponding field in the received object: Temperature, SpO_2, ECG, Respiration, and Encryption Flag.

The server checks if security is applied to the data by checking the *encryption* flag (recall Figure 6). If gray coding is applied (flag = 1), the data is then decrypted/constructed back to its actual values and then visualized. If gray coding is not applied (flag = 0), the data is directly visualized. Algorithm 2 describes the exact steps performed once the Server receives the data.

Algorithm 2: Receiver (Hospital Server)

```
while client.isConnected() do
    receive();
    if data.isEncrypted() then
        constructData();
        visualizeData();
    end
    if !data.isEncrypted() then
        visualizeData();
    end
end
```

Functions implemented on the Hospital Server.

Receive. This function is used to receive the data continuously.
is Encrypted. A function of the Data class, to check if encryption is applied or not, i.e., the flag is set to 1 or 0.
Construct Data. Also a function of the Data class, to construct/convert the actual data from the received gray code data.

Experimental Results. we implemented the aforementioned proposed system on a small scale test-bed using the following equipment:

- Laptop running Windows 10.
- D-Link DIR-822 WiFi Access Point.
- Raspberry Pi 4 Model B, running Debian OS.
- HealthyPiv4.

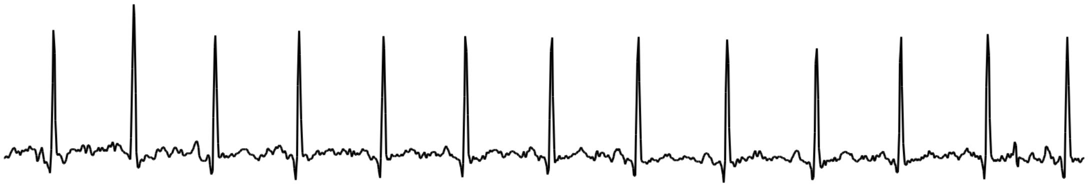

Fig. 7. ECG Signal acquired by the Raspberry Pi

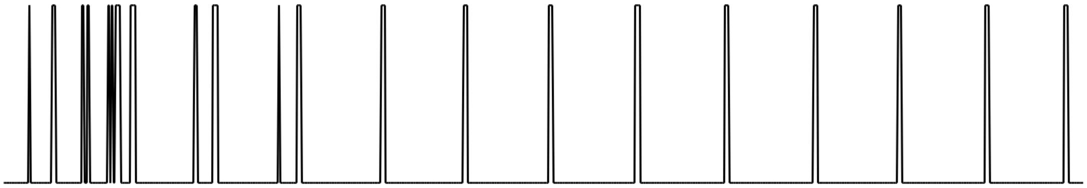

Fig. 8. ECG Signal transmitted by the Raspberry Pi

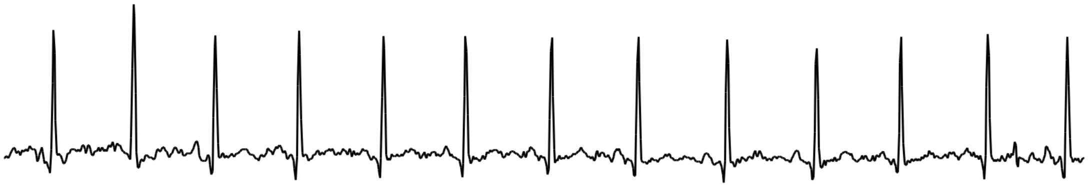

Fig. 9. ECG Signal received by the Hospital Server

Figure 10 illustrates the IoT based data acquisition and transmission devices, which has the HealthyPiv4 that combines all the sensors (SpO_2, Temperature, and ECG) attached to the Raspberry Pi and powered by a power-bank.

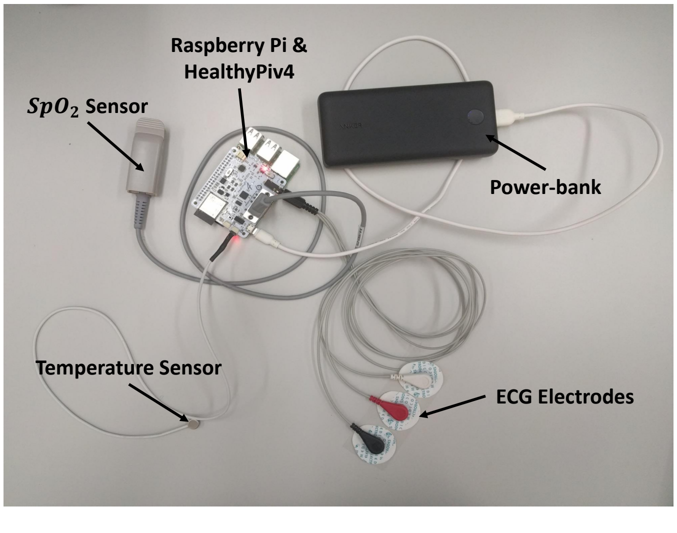

Fig. 10. Data acquisition and Transmission Set-up

On the Raspberry Pi, the α value is set to $-70dBm$. We used the laptop to emulate the hospital Server and performed data collection while connected through the WiFi. The results show that the proposed approach is proved to preserve the patient transmitted biomedical data. The data obtained by the adversary does not include any useful information when visualized. Figure 7 shows the actual signal acquired by the Raspberry Pi, while Figure 8 illustrates the transmitted signal by the Raspberry Pi. Clearly, an eavesdropper unaware of the encryption process will not be able to extract any meaningful information from this signal. Finally, Figure 9 shows the received and constructed signal by the Server, which clearly matches the initial signal shown in Figure 7. Hence, by comparing the aforementioned figures (7, 8, and 9), we see that the proposed approach has rendered the data, making it meaningless for an eavesdropper to infer on the data. The only entity that can recover the data to its original form is the hospital.

VI. Conclusion

In this paper, we presented an actual implementation and evaluation of a secure, lightweight, and portable IoT-based health monitoring system. Several existing adversarial scenarios were presented, where an attacker could intercept the traffic generated by the patient and infer on the data. Physical layer security technique is applied to the data, namely gray code, to preserve patient confidentiality and privacy. Furthermore, we utilized the channel quality metric to sense the presence of an eavesdropper before data transmission over WiFi. The system implementation is based on a HealthyPiv4 attached to a Raspberry Pi powered by a battery, namely power-bank. The readings are first acquired from the HealthyPi shield by the Raspberry Pi (through serial pins). Then the Raspberry Pi checks the RSSI between itself and the access point to check if the RSSI value is less or greater than a predefined α value. In case the RSSI is greater than α, which indicates that a third party is observing the channel, the gray code is applied to the readings and transmitted. Otherwise, if RSSI is less than α, then the readings are transmitted without applying gray code. We tested the proposed system implementation, and we were able to verify that it can countermeasure any type of attack on the user data and privacy.

Acknowledgment

This publication was made possible by NPRP grant NPRP10-1205-160012 from the Qatar National Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the authors.

References

[1] K. Abualsaud, M. Mahmuddin, and A. Mohamed, "Wbasn signal processing and communication framework: Survey on sensing communication technologies delivery and feedback," *Journal of Computer Science (JCS)*, vol. 8, no. 1, 2012.

[2] Reed Abelson. (2020, March) Doctors and patients turn to telemedicine in the coronavirus outbreak. Accessed January 2021. [Online]. Available: https://www.nytimes.com/2020/03/11/health/telemedicine-coronavirus.html

[3] Australian Government Department of Health. Accessing health services during coronavirus (COVID-19) restrictions. Accessed January 2021. [Online]. Available: https://www.health.gov.au/news/health-alerts/novel-coronavirus-2019-ncov-health-alert/ongoing-support-during-coronavirus-covid-19/accessing-health-services-during-coronavirus-covid-19-restrictions

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[5] M. Bazzani, D. Conzon, A. Scalera, M. A. Spirito, and C. I. Trainito, "Enabling the iot paradigm in e-health solutions through the virtus middleware," in *2012 IEEE 11th international conference on trust, security and privacy in computing and communications*. IEEE, 2012, pp. 1954–1959.

[6] A. I. Hussein, "Wearable computing: Challenges of implementation and its future," in *2015 12th Learning and Technology Conference*. IEEE, 2015, pp. 14–19.

[7] Y. Jin, "Low-cost and active control of radiation of wearable medical health device for wireless body area network," *Journal of medical systems*, vol. 43, no. 5, pp. 1–11, 2019.

[8] K. Abualsaud, M. E. Chowdhury, A. Gehani, E. Yaacoub, T. Khattab, and J. Hammad, "A new wearable ecg monitor evaluation and experimental analysis: Proof of concept," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2020, pp. 1885–1890.

[9] K. A. Al-Saud, H. M. Tahir, A. A. El-Zoghabi, and M. Saleh, "Performance evaluation of secured versus non-secured eigrp routing protocol."

[10] M. Elsersy, K. Abualsaud, T. Elfouly, M. Mahgoub, M. Ahmed, and M. Ibrahim, "Performance evaluation of experimental damage detection in structure health monitoring using acceleration," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016, pp. 529–534.

[11] A. Ibaida and I. Khalil, "Wavelet-based ecg steganography for protecting patient confidential information in point-of-care systems," *IEEE Transactions on biomedical engineering*, vol. 60, no. 12, pp. 3322–3330, 2013.

[12] B. J. Mohd, T. Hayajneh, and A. N. Quttoum, "Wavelet-transform steganography: Algorithm and hardware implementation," *International Journal of Electronic Security and Digital Forensics*, vol. 5, no. 3-4, pp. 241–256, 2013.

[13] S. Basu, M. Ghosh, and S. Barman, "Raspberry pi 3b+ based smart remote health monitoring system using iot platform," in *Proceedings of the 2nd International Conference on Communication, Devices and Computing*. Springer, 2020, pp. 473–484.

[14] R. B. Williams, P. Coccoli, G. L. Galloway, M. J. Kubilus, S. A. Mazur, J. K. Vossen *et al.*, "Intercepting, decrypting and inspecting traffic over an encrypted channel," May 1 2018, US Patent 9,961,103.

[15] A. Ghosh, A. Raha, and A. Mukherjee, "Energy-efficient iot-health monitoring system using approximate computing," *Internet of Things*, vol. 9, p. 100166, 2020.

[16] T. Zhao, X. Zhang, Q. Li, C. Qiu, C. Peng, S. Zhang *et al.*, "Design and implementation of a portable and remote ecg monitoring system for careon," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, vol. 1. IEEE, 2020, pp. 1094–1097.

[17] M. Abomhara and G. M. Køien, "Security and privacy in the internet of things: Current status and open issues," in *2014 international conference on privacy and security in mobile systems (PRISMS)*. IEEE, 2014, pp. 1–8.

[18] A. M. Hussain, G. Oligeri, and T. Voigt, "The Dark (and Bright) Side of IoT: Attacks and Countermeasures for Identifying Smart Home Devices and Services," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, G. Wang, B. Chen, W. Li, R. Di Pietro, X. Yan, and H. Han, Eds. Cham: Springer International Publishing, 2021, pp. 122–136.

[19] ProtoCentral. ProtoCentral HealthyPi v4. Accessed January 2021. [Online]. Available: https://healthypi.protocentral.com/

[20] E. Yaacoub, A. Chehab, M. Al-Husseini, K. Abualsaud, T. Khattab, and M. Guizani, "Joint security and energy efficiency in iot networks through clustering and bit flipping," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 1385–1390.