









Security Enhanced Emergency Situation Detection System for Ambient Assisted Living

PLACIDE SHABISHA ¹, CHAMARA SANDEEPA ² (Student Member, IEEE),
CHARUKA MOREMADA ² (Student Member, IEEE), NADEEKA DISSANAYAKA ² (Student Member, IEEE),
THARINDU GAMAGE ², AN BRAEKEN ¹, KRIS STEENHAUT ¹ (Member, IEEE),
AND MADHUSANKA LIYANAGE ^{3,4} (Senior Member, IEEE)

¹ Engineering Technology Department (INDI) and the Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel (VUB), Brussel 1050, Belgium

² Department of Electrical and Information Engineering, University of Ruhuna, Galle 80042, Sri Lanka

³ Center for Wireless Communications, University of Oulu, 90570 Oulu D04 V1W8, Finland

⁴ School of Computer Science, University College Dublin, D04 V1W8 Dublin, Ireland

CORRESPONDING AUTHOR: MADHUSANKA LIYANAGE (e-mail: madhusanka.liyanage@oulu.fi)

This work was supported in part by the European Union in RESPONSE 5G under Grant 789658, in part by Academy of Finland in 6Genesis under Grant 318927, and in part by VLIR-UOS in IUC 2017 UB Phase3 project.

ABSTRACT Typical wearable devices use a dedicated mobile phone as relay node to transfer the collected sensor data to a server. However, such relay nodes can be faulty or inactive due to various reasons, leading to interruptions of the communication link. To mitigate this challenge, we propose a novel security-enhanced emergency situation detection system, where 3rd party unknown mobile relays are used instead of dedicated gateways as opposed to many existing solutions for IoT healthcare applications. The proposed underlying key agreement and authentication scheme ensures anonymity and untraceability for both sensors (wearable devices) and relay nodes, and relies on symmetric key-based operations to function under resource-constrained environments. We have also developed a prototype of the system using commercial off-the-shelf devices to verify the proposed method's validity and evaluate the performance advantage over existing approaches. Bluetooth Low Energy (BLE) communication technology is used to connect sensor nodes (wearable devices) and mobile relays. After sending medical data to the cloud server, the relay node is responsible for emergency detection and alert generation.

INDEX TERMS Internet of Things, Bluetooth Low Energy, mobile relays, emergency detection, mobility, symmetric key, key agreement, Rubin-logic.

I. INTRODUCTION

Ambient Assisted Living (AAL) is a novel concept in which Information and Communication Technology (ICT) is used to ensure the safety and comfort of elderly people to enable them to make their lives self-dependent to some extent [1]. AAL systems usually consist of IoT-based wearable devices, medical sensors, intelligent communication devices, wireless network technologies, and other supporting software and mobile applications. These technologies make aging in place possible.

IoT-based wearable devices are also becoming very popular among the young and middle-aged population. These devices

and body sensors are typically used to monitor fitness levels, track physical exercises (i.e., step count, travel distance), track location with GPS and navigate. Furthermore, advanced wearable devices can play video and music, view text messages and emails, which allow users to be active without the need to take their devices out of their pockets. These IoT-based wearable devices play a vital role in realizing AAL systems. For instance, patients with chronic disease conditions and older adults can get support from Bluetooth Low Energy (BLE) wearable sensors to keep track of their health [2].

Usually, these IoT devices, including wearable devices, use a relay-based communication model. Due to the limited

resources, IoT devices use energy-efficient short-range communication technology such as Zigbee, Near Field Communication (NFC), or Advanced and Adaptive Network Technology (ANT) to communicate with a relay or gateway node. This relay/gateway node supports long-range communication technologies such as 4 G, 5 G, Wi-Fi and Wi-Max.

Generally, IoT-based wearable devices use a dedicated mobile phone as relay node. BLE, also named Bluetooth Smart, is a superior option to be used over alternative technologies such as Zigbee, ANT, or NFC for mobile phones [3]. Since it is available as short-range communication means on the vast majority of mobile phones. The work in [3] shows that BLE provides many features that get wide adoption by the mobile manufacturers to include in the daily life of mobile devices. BLE is a low energy version of Bluetooth specified in version 4.0, [4]. It is a modern Bluetooth technology developed by Bluetooth Special Interest Group (SIG), intended to support short-range communications [5]. Nowadays, BLE is a widely used technology, applicable in a variety of use cases such as healthcare, consumer electronics, and smart energy, and is expected to be used in billions of devices in the near future [6].

At present, these BLE sensors facilitate relay-based communication with the user's mobile phones. It is vital to continuously transmit the data from the sensor to the mobile, to keep track of their health conditions. However, elderly people may often forget to keep their mobile phones always charged or even carry their mobile phone when they move around. Also, during an emergency situation (e.g., a car accident, falling while walking/running), the user's dedicated mobile phone might get turned off or damaged. When this happens, the IoT devices which only use dedicated gateways cannot transmit the data. In these needful situations, those wearable devices fail to alert the caretaker. The elderly person might even lose his/her life in this kind of helpless situation. The continuous monitoring can also get interrupted due to connection loss when the wearable devices get out of the gateway's or dedicated device's range. Moreover, a dedicated mobile might not have an active Internet connection due to suspension of Internet services caused by issues such as late payments.

To resolve this issue, 3rd party mobile relays can be used in addition to the dedicated devices to provide the required long-range communication link for the IoT-based wearable devices. Such 3rd party mobile relays can provide connectivity when the primary dedicated device is not available. In this method, the patient monitoring device can establish a connection to a 3rd party mobile device through BLE to eliminate the issues associated with having only one dedicated mobile relay.

However, there is a risk of sending medical records through a 3rd party mobile device since those records are confidential. Some attacks like theft of data, bluebugging, bluesnarfing, phone hijacking, protocol-based denial-of-service attacks, and battery draining can cause adverse effects for confidentiality, integrity, and availability as well [7]. With the above facts, it is evident that there is a requirement for a secure relay-based communication method for transferring health device data through BLE technology. Therefore, a specific security

scheme has also been proposed and implemented along with an emergency situation detection system. The security integration has mainly addressed four features being confidentiality, mutual authentication, anonymity, and unlinkability. For limiting the 3rd party mobile relays' computation and communication costs, we have designed a symmetric key-based security scheme restricted to use XOR and one-way cryptographic hash functions as described in the upcoming sections.

A. OUR CONTRIBUTION

To mitigate the above mentioned connectivity and security challenges, we propose a novel security-enhanced emergency situation detection system. The proposed approach utilizes 3rd party unknown mobile relays instead of dedicated gateways as opposed to many existing solutions for IoT wearable devices. Furthermore, the proposed system uses BLE communication technology to connect sensor nodes (wearable devices) and mobile relays. In addition, we offer a highly efficient security scheme and implement it along with an emergency situation detection system to enable mainly four security features, being confidentiality, mutual authentication, anonymity, and unlinkability.

Therefore, the main contributions of the paper are the following:

- Addresses issues associated with real-time and stored data transmission from IoT devices to the cloud server by establishing secure communication via third-party mobile relays.
- Proposes a novel secure symmetric key agreement that can establish a shared common key between end devices to protect against security threats and to ensure confidentiality, mutual authentication, anonymity, and unlinkability.
- Implements a prototype of the proposed solution with off-the-shelf devices to analyze the performance and illustrate the viability of proposed features.
- Provides formal verification of the security strengths of the proposed scheme using Rubin logic and informally analyses protection against security threats, i.e., node capturing attack, impersonation attack, man-in-the-middle attack, replay attack, online/offline dictionary attack.

The remainder of this paper is organized as follows: Section II provides the related work. Section III provides a detailed description of the proposed architecture, whereas information on the proposed communication protocol are provided in Section IV. It acts as a core development within the overall project. Moreover, Section V presents details on the developed security protocol and includes several subsections for the clarity of the content. Section VI shows the overall system implementation details, and Section VII discusses performed experiments and their associated details. Informal and formal security evaluations under Section VIII provide a detailed security verification. The paper is finalized with a discussion in Section IX and a conclusion in Section X.

II. RELATED WORK

Many associated works based on IoT systems for health-care are already available, and some use mobile phones for connection establishment. One such piece of work is in [8], [9], which introduces a mobile-based relay assistance system for establishing a secure end-to-end (E2E) connection between low-power IoT sensors and cloud servers without using any specific and dedicated gateway. It proposes a basic prototype to accomplish communication tasks and E2E connection establishment through a secure AES-CCM encryption technique. Another work that utilizes BLE is available in [10], where authors introduce an open BLE platform (custom-designed beacon platform nRF24Cheep) and open source development of a BLE physical and Medium Access Control (MAC) layer. This development provides capabilities to adapt the communication stack. Their work is in very abstract form, which includes guidance on establishing a BLE connection for prototyping. In [11], a model named iConfig presents an approach for managing IoT devices in smart cities via BLE. The system has an edge-driven platform that has addressed the three major issues in current IoT management: registration, configuration, and maintenance. But their work does not establish an E2E connection of IoT sensors with a backend.

When considering technologies other than BLE, the solution in [12] contains the central server and several other servers which are acting as gateways. It describes the IoT sensor network's middle-ware to perform sensor data translations. However, as the system is not a cost-effective solution and is poorly scalable, it is not a feasible solution for IoT applications. In [13], a scheme named Collaborative on Demand Wi-Fi Sharing (COWS) is introduced with the purpose of enabling Wi-Fi roaming facilities for users. But this system is not fully compatible with resource-constrained devices such as those that have power limitations. Besides, [8], all the other related works are using a dedicated gateway for data transmission. Moreover, the proposed solution in [8] supports a single relay for real-time data transmission only. This approach implies that it can fail when there is no relay nearby during an incident since critical data is dropped or is not available on time.

The work related to dedicated gateways presented in [14] describes the implementation of a smart e-health gateway (named UT-GATE) at the edge of healthcare IoT in clinical environments. In addition to cloud processing, they suggest local data processing through smart gateways. This step helps to decrease latency, but it may be vulnerable to security problems such as the possibility of implementing malicious gateways that could eavesdrop on patient's data. Moreover, this system may not support mobility-related aspects due to cost and difficulty to provide universal connectivity (due to interoperability issues) in external environments with the proposed method. By giving more attention to mobility and security, in [15], the authors propose an end-to-end security scheme for mobility enabled healthcare IoT. Their strategy has three main characteristics, (i) Secure and efficient end-user authentication and authorization architecture based on

certificate-based Datagram Transport Layer Security (DTLS) handshake, (ii) Secure end-to-end communication developed on session resumption, and (iii) Robust mobility implemented through interconnected smart gateways. But, instead of specific smart gateways, we use third-party mobile devices, which increase mobility further. Thus a more trustworthy system is mandatory in such a situation, considering the security aspect.

As a consequence, a dedicated security protocol is required. In literature, this refers to so-called tripartite schemes, where three entities need to agree on a common key. We can distinguish several mutual authentication and key agreement schemes. Some of the techniques developed are based on symmetric key mechanisms, using a pre-shared common key [16]–[19]. In particular, Gong *et al.* [16], [19] study the minimum number of communication rounds and messages needed to establish mutual authentication among three different parties, taking into account various assumptions. The disadvantage in these schemes is that the session key is only constructed by the authentication server and the other two entities do not participate in the construction of it, making these schemes vulnerable for key control resilience attacks [20]. Moreover, none of the schemes provide anonymity, which was even considered impossible [21] to be reached by symmetric key-based schemes. Instead, we will provide in this paper a counterexample, showing that this statement is not correct. Our system relies on the basic structure of [22], but is extended to a tripartite architecture and corrects the weakness of the occurrence of an offline dictionary attack.

To further complete the state of the art concerning authentication schemes for tripartite architecture, we also mention several public key-based mechanisms for relay-based architectures. In [23], the authors propose an example of a key agreement scheme for a healthcare application, in which anonymity of the end device towards the mobile relay is obtained. The system is an improvement of [24] in which the derived key was static and thus not able to establish perfect forward secrecy. However, it is limited to devices possessing a smart card-based entry and it uses compute-intensive pairing operations. Another secure identity-based tripartite scheme is presented in [25], which is in particular designed for mobile distributed computing environments. However, this scheme does not provide anonymity to participants and consists of a pairing operation at the device's side. In addition, it is also not able to compute pairwise keys using the available key material at the end of the protocol. Finally, there is the scheme of [26], representing an identity-based, mutual authenticated key agreement protocol, in which the sensor device and the relay node can establish a secure communication without leakage of their identities. Only the cloud server can control the identities of the sensor device and relay node. It has been formally proven that the session keys are also protected in the Canetti-Krawczyk security model. In which adversaries have access to session state-specific information, previous session keys, or long-term private keys. The scheme is efficient compared to [23], [25] as it only utilizes elliptic curve operations and basic symmetric key operations.

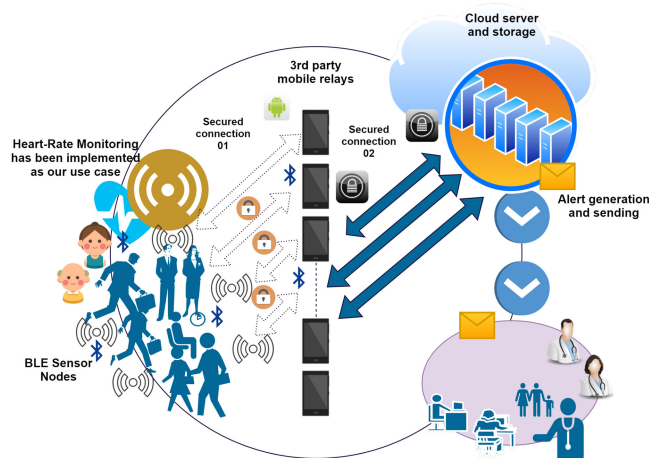


FIGURE 1. Proposed system architecture.

III. PROPOSED ARCHITECTURE

We design and implement an IoT-based remote patient monitoring and caring system which offers maximum mobility and flexibility to its users. The proposed method is similar to a fog computing approach [27], with third-party mobile relays. Fig. 1 shows an overview of the system architecture.

As indicated in Fig. 1, the whole architecture consists of four main components. At one end of this architecture, the network consists of BLE-based sensor nodes with low power consumption. This part of the system is responsible for gathering the required patient data. For use case scenarios, we select heart rate and fall detection.

Data generated from sensors get forwarded into a 3rd party unknown mobile relay. In this case, the BLE sensor node selects a specific mobile relay that is in range and available before sending information. The communication protocol in Section III describes the selection procedure in detail. Furthermore, each 3rd party mobile relay has a mobile application that enables and controls its connectivity with the network. There is no data processing or storing at mobile stations, but the mobile can attach its location information with the data that it transmits to approximate patient location.

In the next step, the mobile relay sends data to a cloud server via its internet connection. Secure socket communication can be established between mobile relay and cloud server.

The server performs data processing, data storing, and emergency situation detection. After detecting any emergency, the server sends notifications to registered carers of the patients through Short Message Service (SMS) and emails. Other than alerting functionality, the system also supports real-time data monitoring and location tracking services for patient's carers. Patient data is available for carers via a web application.

Our proposed architecture contains two key protocols, i.e. *communication protocol* and *security protocol*. The communication protocol allows to establish communication between the different entities of the proposed system. It extends the existing solution with new capabilities such as multi-connect,

automatic handover, storage, forwarding of data, and load balancing. Further details are discussed in Section IV. The proposed security protocol provides essential security features such as confidentiality, mutual authentication, unlinkability, and anonymity. It also offers protection against several types of security threats, i.e., node capturing attack, impersonation attack, man-in-the-middle attack, replay attack, and online/offline dictionary attack. Further details can be found in Section V.

IV. COMMUNICATION PROTOCOL

A. SINGLE MOBILE RELAY NODE BLE CONNECTION

The Fig. 2 depicts message flow of protocol for a single mobile in the relay.

- 1) **Phase 1:** The donor mobile relay node connects with the Cloud Server (CS) via an HTTPS connection request sent by the mobile app. In this case, the registration phase initiates, and both BLE sensor node and mobile relay have to be registered with the cloud server as described in Section V-D. A trusted mobile relay, like a person's own smartphone needs to be utilized to perform this initial registration. This registration consists of the establishment of the security key material. After successful registration with the CS, the IoT device can function with any other preregistered mobile relay as indicated in Fig. 2. On the other hand, a mobile can register itself as a donor with the CS through an inbuilt process. Upon successful authentication, the CS issues a dynamic value a_f for the donor mobile, as discussed in Section V-D. This value is considered the advertisement ID, and mobile phone uses it for advertising its presence via BLE.
- 2) **Phase 2:** The mobile relay node starts advertising with the received advertisement ID. Meanwhile, this mobile also scans for an advertisement from a wearable device that is advertising with the same ID. A wearable device that scans for mobile relays can get an advertisement ID from the mobile relay node and start advertising itself with that same ID. The correctness and authenticity of these data is evaluated after a successful run of the key agreement protocol. Therefore, a key agreement phase needs to be performed as described in Section V-E. In the case of multiple mobile relays in proximity, the wearable device can select the one with the best Received Signal Strength Indicator (RSSI) value. Then, the mobile app can establish a connection with a wearable device after finding a match with its ID.
- 3) **Phase 3:** After connection establishment with the mobile relay node, the wearable device can initiate a connection request with the CS. The mobile relay would forward this request to the CS. The server can then validate this request and approve the connection.
- 4) **Phase 4:** After connection approval, the wearable device initiates the transmission of data to the mobile relay. These data packets may contain a timestamp, and

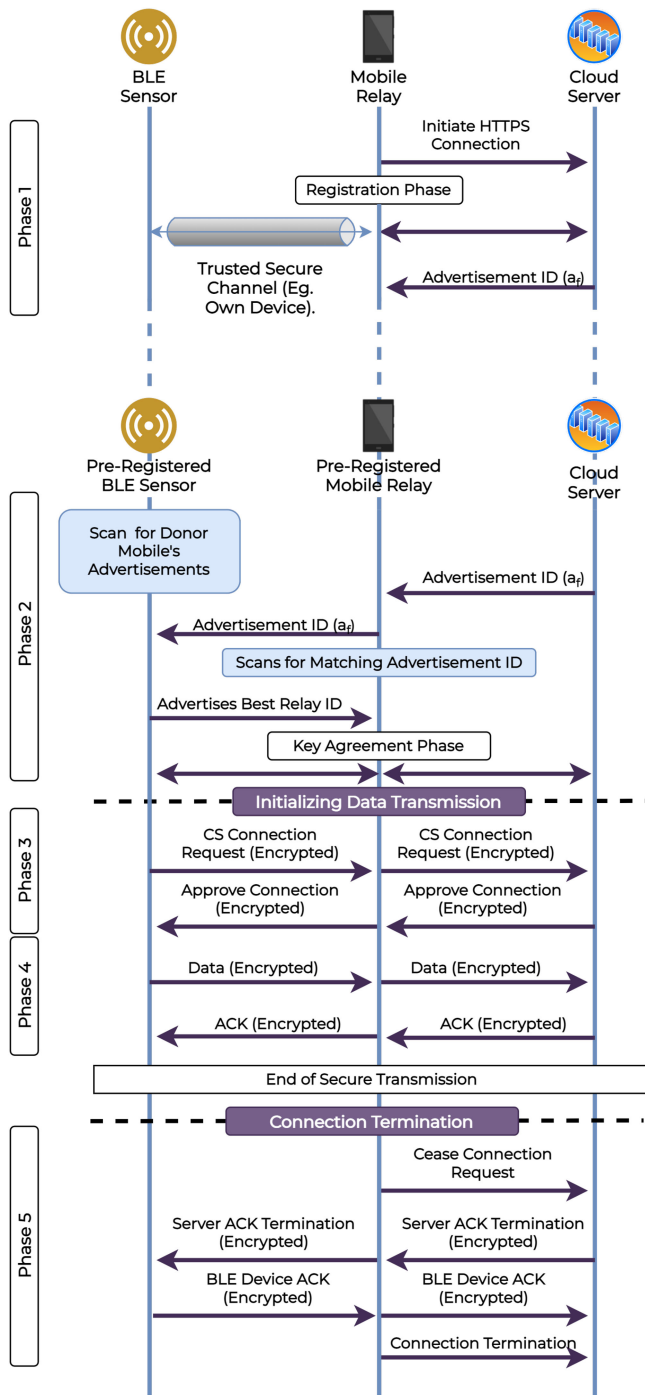


FIGURE 2. Message flow of the proposed protocol.

data is encrypted so that mobile relays cannot eavesdrop on user's data. After sending a fixed amount of information, the wearable device expects an encrypted acknowledgment from the CS. If an acknowledgment is received, data transmission is resumed. Otherwise, the wearable device terminates the connection with the relay and reports this session to the CS in subsequent successful data transmission.

5) **Phase 5:** The donor mobile can set a maximum threshold for the number of data packets that a wearable device can send. Once the threshold is reached, it can request to cease the connection from the CS. The CS then sends the last acknowledgment message to the wearable device and terminates the connection with the mobile relay. In this case, the wearable device discards the session information as this is a legitimate session termination. Then, the wearable device can restart from "Phase 2" and start scanning for other nearby mobile relays for sending more data.

B. WEARABLE DEVICE MULTI CONNECT

The transmission of both real-time and previously stored data via a single mobile relay node may cause problems such as latency when sharing the same connection. Here, stored data is the data that is generated by a wearable device when it is not connected to a nearby mobile relay. Therefore, the paper proposes expanding the same protocol to transmit real-time and stored data separately, using two mobile relays. Note that the current prototype implementation is not yet included this concept. It will be considered in future work. Here, the wearable device follows a procedure such that one mobile relay allocates for transmitting real-time data and another one for transmitting stored data.

There should be at least two donor mobile devices for achieving this. Real-time data has a higher priority, and hence the selection of mobile relay devices for real-time data transmission is based on the best RSSI value. The wearable device selects the second-best mobile to send the stored data.

C. BLE HANDOVER

To improve connectivity duration and QoS of the 3rd party relay-based communication channel, we propose a handover mechanism. It helps maintaining channel conditions at an acceptable level for a longer duration than a system without handover. Handover can also avoid unnecessary data losses or delays within a dynamic environment. This extra feature will be an added benefit of the proposed system. However, handover can only be performed in dense environments where multiple mobile relays are available.

In our system, handover can be triggered mainly under two situations.

- 1) When the currently connected relay decides to terminate the connection with the connected mobile relay.
- 2) When the patient is traveling away from the connected mobile relay (i.e., if RSSI of the patient's device reaches a predefined threshold, handover can be performed).

Therefore, handover can help to keep a continuous data flow between IoT devices and the cloud server. It can also help to support real-time data monitoring and emergency alert generation processes online.

With the currently developed handover mechanism, if a handovering instance triggers, the IoT device scans for another mobile relay to connect. After detecting possible mobile relays, the IoT device selects the mobile relay device

that provides the best RSSI value. Then, it establishes a new connection by following steps we describe in Sections IV and V.

Besides the scenarios mentioned earlier, several other ways, like environmental interference, hardware failures in mobile and BLE devices, and malfunctioning BLE radios, can lead to connection termination. In the absence of a mobile relay to transmit the data or to perform handover, the BLE device stores the generated data in the integrated micro SD card, in order to minimize the impact of data loss.

V. SECURITY PROTOCOL

The proposed security scheme consists of three main entities: BLE sensor nodes N , relay nodes F and the cloud server CS as illustrated in Fig. 1. Both sensor nodes and relay nodes need to register with the CS and receive key material, which should be securely stored on the device. We assume that storage is tamper-proof, which is common on state of the art devices. Also, the server needs to store its master secret key in tamper-resistant hardware.

If the sensor wants to communicate with the CS , it sends a request to the nearby relay, which forwards the message to the CS after adding additional information. Based on the received data, the CS verifies the authenticity of the request, and if it is validated, it generates the necessary key material, allowing relay and sensor node to forge a common shared key with the server; hence all three can derive a common shared key. The relay node should not be able to derive the sensor's identity throughout the process and vice versa.

A. REQUIRED SECURITY FEATURES

To summarize, the following security features should be established.

- **Confidentiality:** Only the involved entities should be able to derive the key material.
- **Mutual authentication:** The common shared key should involve key material coming from all entities able to derive the corresponding key.
- **Anonymity:** No outsider, not even the relay node or other sensor nodes can derive the identity of the sensor node. Also, the sensor node is not able to derive the identity of the relay node.
- **Unlinkability:** No outsider, not even the relay node or other sensor nodes can link messages coming from the same device. Also, the sensor node is not able to deduce a relation between messages coming from the same relay node.

B. ATTACK MODEL

In the attack model, we assume that the adversary can eavesdrop on the channel or actively manipulate the transmitted messages, i.e., insert, change, reply messages. These activities are typically applied when trying impersonation, man-in-the-middle (MITM) attacks, replay attacks, and online/offline dictionary attacks. An attacker can also capture a node or relay node and derive the key material stored in the tamper-proof

part of the memory. In this case, it is essential to keep the impact of the attack local to the tampered device.

C. PROPOSED SCHEME

To establish the above security features, we design the security scheme for the proposed architecture. The scheme consists of two phases: *registration phase* and *key agreement phase*.

The operations in the scheme are limited to XOR \oplus and a one-way cryptographic hash function H (e.g., SHA2 or SHA3). Since the proposed scheme uses very low-cost cryptographic operations, it is efficient in terms of computation. Furthermore, the concatenation of two messages is denoted by $m_1 || m_2$.

D. REGISTRATION PHASE

In the registration phase, both wearable devices and mobile relay need to register with the CS . During this operation, the CS makes use of its master key k_m . The process for both is indicated in Fig. 3.

The CS chooses a temporary key k_i and derives the following parameters for the entity with identity id_i . Here $i = n$ in case of the wearable device N and $i = f$ in case of the Relay F .

$$a_i = id_i \oplus H(k_m || k_i)$$

$$b_i = a_i \oplus k_m \oplus k_i$$

$$c_i = H(id_i || k_m)$$

The parameters (id_n, a_n, b_n, c_n) and (id_f, a_f, b_f, c_f) are sent over a secure channel to the wearable device and mobile relay respectively. Note that the temporary keys k_n and k_f , are not stored, neither by the wearable devices, nor by the mobile relay, nor by the CS . The parameters id_i, c_i , corresponding with the static identity, require secure storage in tamper-resistant hardware at the wearable device and the mobile relay. These are fixed parameters and do not vary over time. The parameters $\{a_i, b_i\}$ represent the dynamic identity, are publicly available as they appear in the message sent by the device, and are updated in each communication phase.

Note that no specific storage is required for each entity on the CS side, except the master key k_m . If the CS still wants to keep track of which devices are registered, it should not store the identities but the hashed values of the identities. The identities must be under no circumstances leaked. Otherwise, the security of the corresponding device is not guaranteed anymore, as the device is sufficiently weakened but not broken. The temporary keys k_n, k_f should not be stored and must be deleted by the CS .

E. KEY AGREEMENT PHASE

In this phase, we distinguish five different steps, where the first four steps correspond with a message sent over the channel. Fig. 4 shows the key agreement phase in detail.

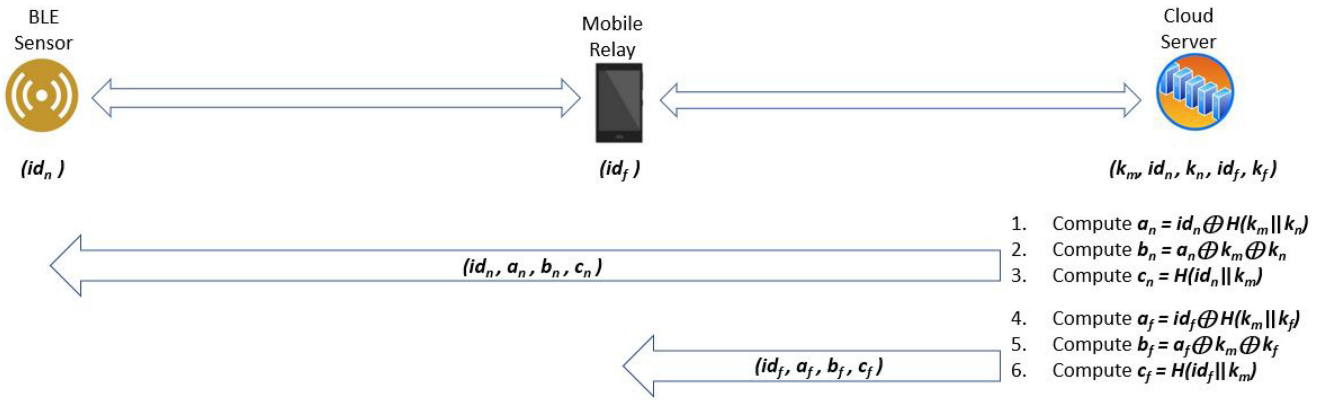


FIGURE 3. The registration phase where the CS shares the temporary identity with the mobile relay and the wearable device.

1) WEARABLE DEVICE REQUEST

In this step, the wearable device broadcasts a key agreement request. Therefore, it needs to choose a random value r_n and computes, using its stored parameters (id_n, a_n, b_n, c_n) and the current timestamp t_n , the following values:

$$\begin{aligned} x_n &= a_n \oplus id_n \\ y_n &= x_n \oplus r_n \\ tid_n &= H(id_n || t_n || c_n || r_n) \end{aligned}$$

The message $m_1 = \{tid_n, y_n, a_n, b_n, t_n\}$ is sent by the wearable device.

2) MOBILE RELAY REQUEST

The mobile relay, which picks up this message, also derives in the same way the following values using its stored parameters (id_f, a_f, b_f, c_f) , the received timestamp t_n and a randomly chosen parameter r_f :

$$\begin{aligned} x_f &= a_f \oplus id_f \\ y_f &= x_f \oplus r_f \\ tid_f &= H(id_f || tid_n || t_n || c_f || r_f) \end{aligned}$$

Next, the message $m_2 = \{tid_n, y_n, a_n, b_n, t_n, tid_f, y_f, a_f, b_f\}$ is sent to CS.

3) CLOUD SERVER CHECK AND KEY ESTABLISHMENT

Upon the arrival of this message, CS verifies the correctness of the message by first deriving the identities of the wearable device and the relay and then checking if the message is well-formed. This results in the following operations:

$$\begin{aligned} k_f^* &= k_m \oplus a_f \oplus b_f \\ x_f^* &= H(k_m || k_f^*) \\ id_f^* &= x_f^* \oplus a_f \\ r_f^* &= x_f^* \oplus y_f \\ c_f^* &= H(id_f^* || k_m) \\ \text{Verify: } & H(id_f^* || tid_n || t_n || c_f^* || r_f^*) == tid_f \end{aligned}$$

$$k_n^* = k_m \oplus a_n \oplus b_n$$

$$x_n^* = H(k_m || k_n^*)$$

$$id_n^* = x_n^* \oplus a_n$$

$$r_n^* = x_n^* \oplus y_n$$

$$c_n^* = H(id_n^* || k_m)$$

$$\text{Verify: } H(id_n^* || t_n || c_n^* || r_n^*) == tid_n$$

At this stage, the correctness of the identities of the mobile relay and the wearable device is verified. CS now creates new dynamic identities for the mobile relay and the wearable device and then derives a session key, using the random values chosen by the mobile relay and the wearable device. Therefore, it chooses two random values r_n^s, k_n^+ related to the wearable device and two random values r_f^s, k_f^+ related to the mobile relay. Next, it computes the new identity related material a_i^+, b_i^+ and session key K_i with $i = \{n, f\}$.

$$a_i^+ = id_i^* \oplus H(k_m || k_i^+)$$

$$b_i^+ = a_i^+ \oplus k_m \oplus k_i^+$$

$$\eta_i = H(r_i^s || id_i^*) \oplus a_i^+$$

$$\mu_i = H(id_i^* || r_i^s) \oplus b_i^+$$

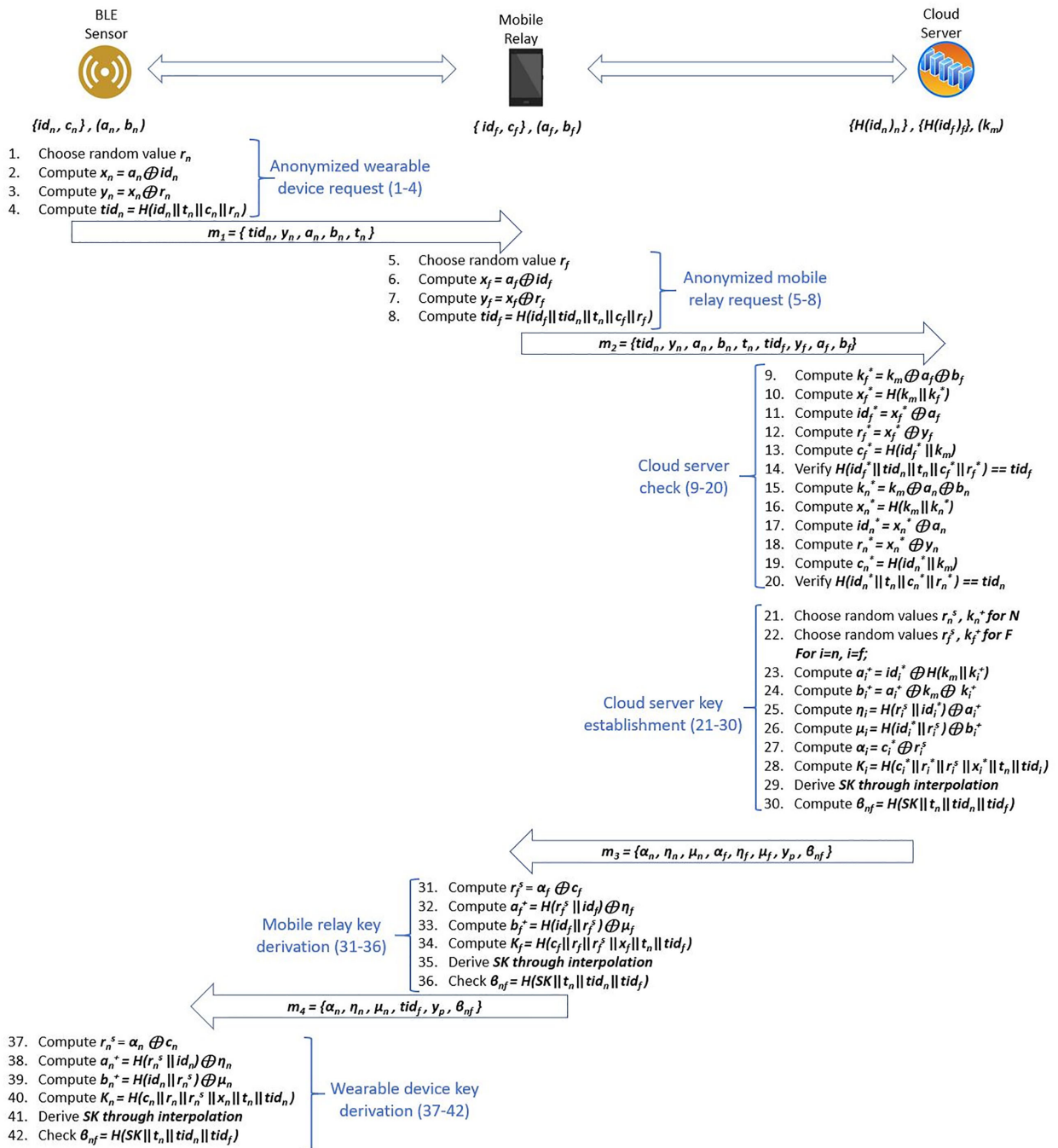
$$\alpha_i = c_i^* \oplus r_i^s$$

$$K_i = H(c_i^* || r_i^* || r_i^s || x_i^* || t_n || tid_i)$$

CS determines the line containing the points $(1, K_n)$, $(2, K_f)$ to derive a common shared key between the three entities. The intersection with the Y-axis, i.e., $x = 0$, results in the secret key SK . The Cloud server also derives the point on the line for $x = 3$, resulting in $y = y_p$. Then, to allow the mobile relay and the wearable device to derive the correctness of the message, the following parameter is also computed:

$$\beta_{nf} = H(SK || t_n || tid_n || tid_f)$$

The message $m_3 = \{\alpha_n, \eta_n, \mu_n, \alpha_f, \eta_f, \mu_f, y_p, \beta_{nf}\}$ is sent to the mobile relay.


FIGURE 4. Key agreement phase.

4) MOBILE RELAY KEY DERIVATION

The mobile relay considers the last five parts of the received message. Using α_f , it can find r_f^s . As a consequence, the new dynamic identity (a_f^s, b_f^s) is derived using this value r_f^s and the received parameters η_f, μ_f . Next, the common shared key K_f with CS can be computed. To compute the common shared key between all three entities, it derives the line through the points $(3, y_p)$, $(2, K_f)$ and computes the intersection with the Y-axis to find SK . Finally, the correctness is verified by checking the hash value to derive β_{nf} . If this last check is

successful, the mobile relay forwards the message to the wearable device.

$$m_4 = \{\alpha_n, \eta_n, \mu_n, tid_f, y_p, \beta_{nf}\}$$

5) WEARABLE DEVICE KEY DERIVATION

The wearable device can perform similar steps as the mobile relay to derive r_n^s , its new dynamic identity (a_n^s, b_n^s) and its common shared key K_n with CS. In the same way, the common shared key SK is determined, and its validity is checked through β_{nf} .

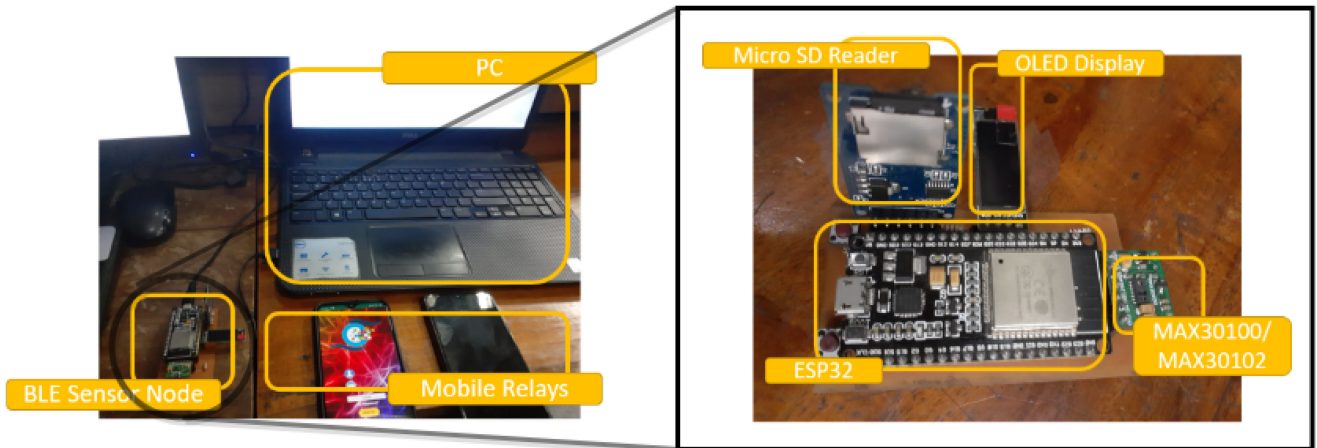


FIGURE 5. Experimental setup.

TABLE I Configuration Settings for ESP32

Attribute	Configured values
Transmission power	-21dBm
Number of BLE services	1
Number of BLE characteristics	2
Maximum packet size	244 bytes
Maximum app memory	3.5 MB

VI. IMPLEMENTATION

To implement this protocol, we consider emergency situation detection of patients who suffer from heart issues as the basic use-case scenario. This scenario involves sensors that can detect heart problems, a BLE device that can transmit the data, a relay mobile that supports BLE, and a cloud server that performs operations such as user registration, authentication, storage, detection of emergencies, and generation of alerts. The overall system is named “The Healer”.

In order to perform some of the experiments mentioned in Section VII, we prepare a specific setup as shown on Fig. 5.

We use a micro-controller unit named ESP32 (with BLE 4.2 capabilities) to establish the communication pathway. Table 1 provides details on the ESP32 unit. The mobile phones utilize for the implementation are Samsung Galaxy M20 and Samsung Galaxy A20 with Android 9 Pie system, and OPPO A37 running Android 5.1 Lollipop. We use Android Studio 3 libraries to develop the mobile application and the Java Spring Boot framework for server-side implementations. PostgreSQL acts as database management system of our development with its TimescaleDB extension to handle time-series data storage. A sensor model named MAX30100 records the heart rate of a patient.

During a communication failure or while waiting for the BLE sensor to connect with the mobile relay, sensors capture the data and store them in an external flash memory connected to the ESP32. This flash memory device stores the data during the handover process (time interval between the current connection termination and new connection establishment).

Moreover, we need an external flash for the ESP32 since ESP32 overall flash memory is 4 MB, and a considerable amount of this space (3.5 MB) is for the storage of the running application. The remaining amount of space may not be sufficient to store information generated over a long period.

The ESP32 supports FreeRTOS. Parallel operation of multiple threads is possible from its dual-core CPU having an Xtensa LX6 microprocessor. We use this to establish the parallel operation of transferring real-time data from one thread and transferring stored data from another thread. It is more feasible than using a single thread as the real-time data should prioritize old stored data.

Before starting data transmission, the device needs registration by the user. For that, the registration phase described in Section V-D should be executed. It is best to utilize a trusted mobile relay (E.g., personal mobile) for IoT device registration. In this case, the remote server issues an IoT device ID to the patient that the BLE device stores in its memory. The server can distinguish each device and associate the patient with it for all data when attached with this saved device ID.

After that, ESP32 continuously scans for mobile relays within proximity of the sensor. Meanwhile, it saves the data generated by the heart rate sensors in flash memory. It connects with the mobile relay after discovering a single mobile relay device by following the key agreement phase of the security protocol as indicated in Section V-E and starts transferring data. In the presence of multiple mobile relay nodes, it selects the best mobile relay node based on the RSSI value. Before connecting, it repeats scanning numerous times to verify the availability of the best relay node.

The components mentioned above and techniques are put together into a single prototype device indicates in Fig. 6. Besides these sensors and micro-controllers, we integrate a body temperature sensor (MAX30205) with the prototype device for two purposes.

- We use human body temperature as an additional biomedical measurement to detect abnormalities of the patient.



FIGURE 6. Prototype IoT device.

- If the body temperature becomes lower than usual for a certain period and no heart rate data is generating, carers of the patient can identify that the patient is not wearing the device at the moment.

The device should be placed near the patient's wrist, and therefore, they can wear the device all the time during their daily activities.

Registration of this handheld device with the server requires a specific methodology. This device comes with two buttons built into it. The user is asked to keep pressing both of these buttons for 3 to 5 seconds to enter the device's menu window. In this window, the user has to select the 'register' option to enable the registration mode. This registration process can process via any trusted mobile relay by following the security protocol mentioned earlier. During the registration process, the server automatically grants a device ID for a specific device after receiving a request from it (when registration mode is on and the trusted mobile is nearby, the IoT device automatically sends the request). After the registration process, the IoT device can make a connection with any other 3rd party mobile relay which is running the "The Healer" mobile application.

After the Key Exchange phase, the data transmission phase initiates as the next step of the communication protocol. During this phase, the relay directs these data into the server via a mobile application. Moreover, Fig. 7 indicates the essential functionalities associated with the mobile application. The mobile relay device also appends location

details to the transmitting data stream (as an anonymous data field) to detect the approximate location of the patient instead of implementing the location tracing service within the IoT device itself. This approach is a kind of energy-saving strategy as more energy will drain to operate a dedicated GPS sensor along with the BLE device.

A specific web application satisfies the data visualization and user registration requirements of the system (Fig. 8). The front-end web user interface utilizes the Angular Framework for more sophisticated front-end development and efficient request handling with the backend. A primary requirement for the web application is to perform user registration before data transmission. We implement it as follows. A person should first undergo initial registration as a general user, and they can select their role as a patient, donor, or carer according to the requirements. The donors are the third-party mobile users who contribute as mobile relay. The carer is someone who has the privilege to receive notifications about the patient's health status. Carers can receive these notifications and view real-time health information about their patient after the patient has permitted to do so. A user can perform any of these roles, including all three. The platform can render real-time data from the patient in a graph format visible for both patient and carer.

We use Java Spring Boot as the technology for remote server backend as a Representational State Transfer (REST) Application Program Interfaces (APIs).

We make use of web sockets to transmit data from mobile app to servers. This data continuously gets stored in the database. For each user session, the system constantly monitors patient status based on various criteria. The monitoring criterion can vary according to the patient's health-care device sensors. First, we select heart rate detection as the use-case for the implementation and introduce a specific criterion based on upper and lower threshold heart Beats Per Minute (BPM). The lower threshold for a non-athlete individual is 60 BPM, and the upper is 100 BPM [28]. To test another use case, we utilize a device with an accelerometer sensor ADXL345 to detect falls. New devices can integrate different sensors that need any other criteria, according to the requirements of the patients or carers. If the emergency situation detection module detects an emergency, it sends a request to the notification API to forward email and SMS notifications to patients and pre-registered carers. The CS and some other online services are responsible for controlling and triggering the system notifications according to the requirements. To send SMS notifications, we use a cloud communication platform named 'Twilio' and the notification API within the CS.

VII. EXPERIMENTS AND RESULTS

To address several issues associate with the IoT device and BLE while designing the system, we conduct two experiments and adapt the system according to the results.

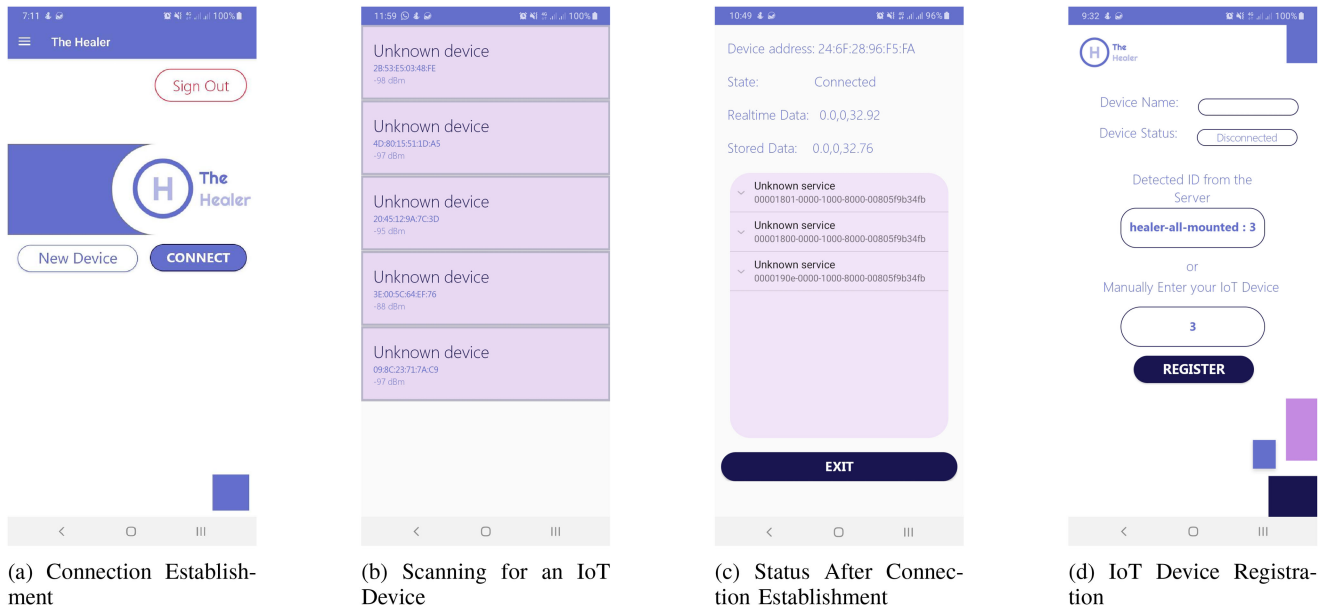


FIGURE 7. Interfaces and functionalities of mobile application.

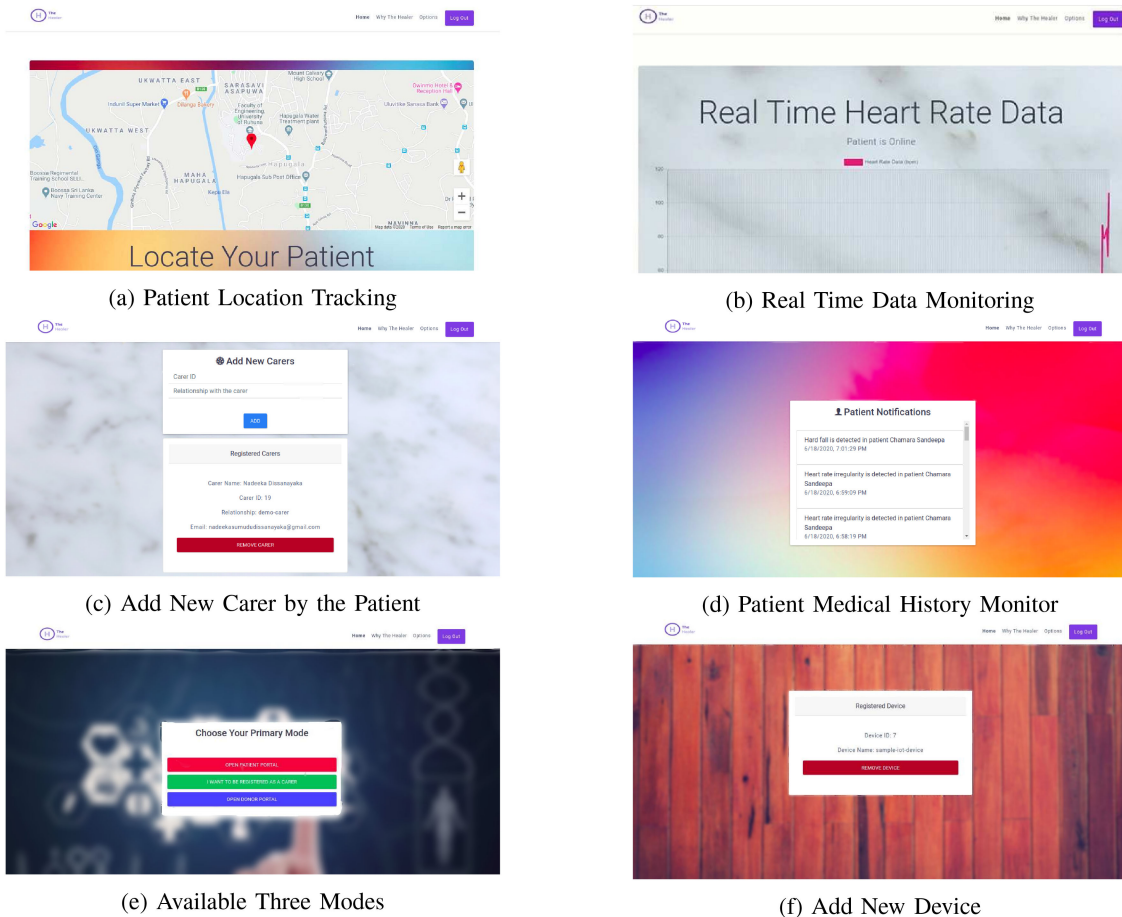


FIGURE 8. Interfaces and functionalities of Web application.

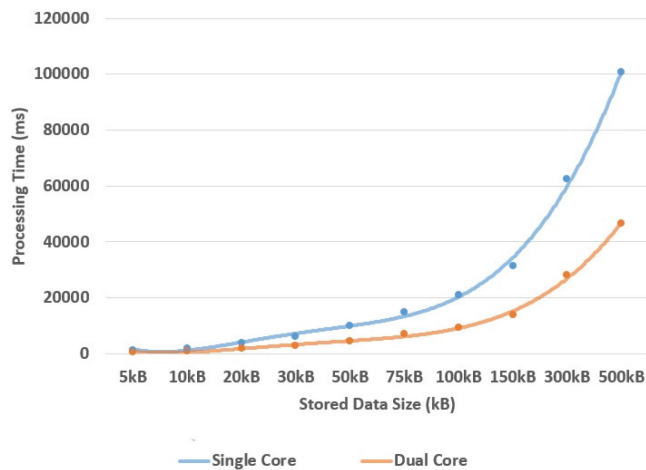


FIGURE 9. Stored data size vs processing time of data in a single core and dual cores.

TABLE II Processing Time Improvement Under Multi-Connect

Sample Size	Processing Time Improvement (%)
5kB	53.33 ± 2.36
10kB	55.93 ± 1.52
20kB	55.34 ± 1.77
30kB	54.91 ± 2.44
50kB	53.67 ± 3.02
75kB	53.01 ± 2.98
100kB	55.36 ± 1.27
150kB	55.68 ± 2.69
300kB	55.65 ± 6.42
500kB	53.59 ± 1.46

A. DATA PROCESSING TIME UNDER SINGLE CORE AND DUAL CORE PROCESSING

In this experiment, we observe data processing times under single and dual-core conditions of the ESP32 module. Under the single-core mode, both the stored data and real-time data process through a single-core and transmits. In contrast, two cores split the real-time data processing and stored data processing under the dual-core model. For the experiment, we select 10 data samples with sizes of 5 kB, 10 kB, 20 kB, 30 kB, 50 kB, 75 kB, 100 kB, 150 kB, 300 kB and 500 kB. Then each data sample processes 30 times in the ESP32 module and calculates the average processing times accordingly. We observe a significant improvement in processing time with the utilization of two cores for the data processing function in ESP32, as indicates in the following Fig. 9.

Moreover, the Table 2 presents summary of the improvement. As observable in the Table 2, we can achieve a decrease of more than 50% of processing time delay in each case through the utilization of two cores for the data processing ESP32.

B. DATA TRANSMISSION LOSSES WITH DISTANCE

This experiment intends to detect transmission losses related to transmission distance. In this case, ESP32 sends data to

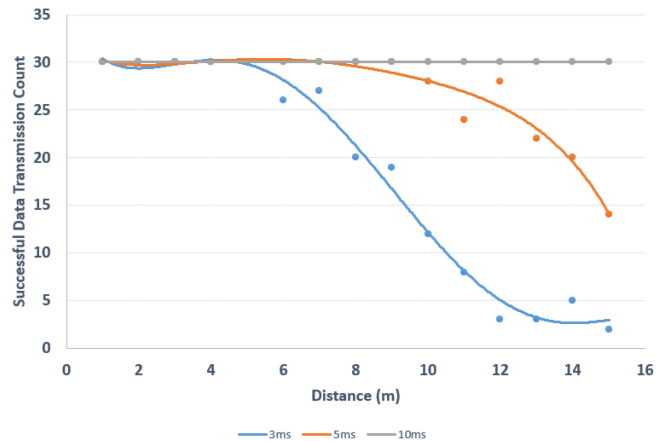


FIGURE 10. Distance vs successful data transmission count.

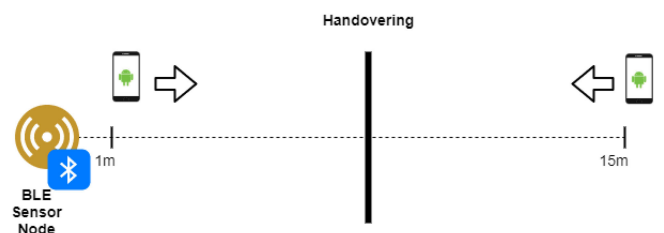


FIGURE 11. Handover of mobile relays at a threshold distance.

the mobile phone continuously, and for every 1000 points of data, ESP32 sends an acknowledgment to the mobile. Meanwhile, the mobile also keeps track of the number of packets received. Due to higher inter-frame delays, the mobile phone can synchronize this acknowledgment with its count exactly. But when the delay is lower, we observe that counts do not match all time. This observation shows in Fig. 10, and it is clear that the transmission disparity increases with increasing transmission distance. It shows a drop in transmission success rate when the mobile moves away from BLE sensor ESP32. We also observe that more dissimilarities occur when the inter-frame delay is decreased. Therefore, we can expect more reliable data transmission under sufficiently large inter-frame delays.

Thus, under our setup, with multiple relays nearby, the best mobile donor should be selected (the one having the best RSSI). It shows the importance of implementing a handover mechanism to mitigate data transmission loss when the previously selected mobile moves away from the BLE sensor. Multipath fading can also cause data transmission loss. In such a situation, the IoT sensor scans and connects to the preferred mobile as indicated under the protocol description.

The handover mechanism is illustrated in Fig. 11. The BLE sensor can perform handover from one mobile to another if the current host donor mobile moves further away from the sensor than a threshold distance. According to our experiments, it is halfway to 15 m distance under this scenario. From Fig. 12, it can be observed that data transmission success increases

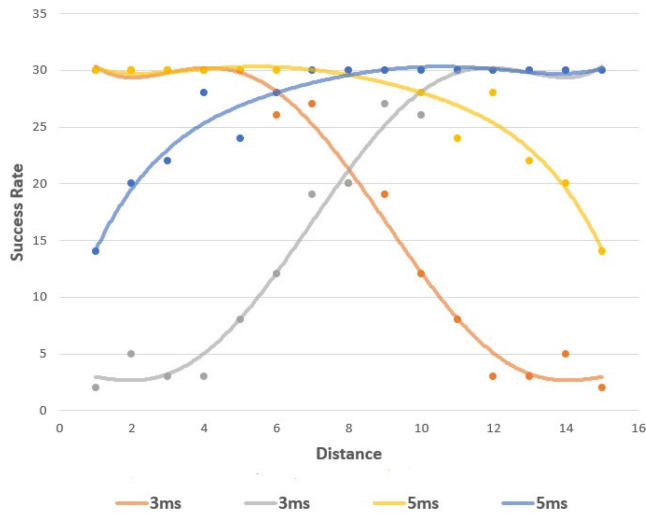


FIGURE 12. Distance vs successful data transmission count when handovering under 3 ms and 5 ms inter frame gaps.

TABLE III Time Taken for the Security Protocol Establishment

Task	Average time taken (ms)	
	Time for 128 bits (ms)	Time for 256 bits (ms)
Connection Establishment without security	13671 ± 232	
Security Protocol in IoT Device	1962 ± 1	1997 ± 1
Security Protocol in mobile relay device	500 ± 2	560 ± 2
Security Protocol in cloud server	286 ± 28	299 ± 29
Time taken for the security key agreement	2248 ± 28	2297 ± 29
Total connection establishment delay with Security	15919 ± 232	15968 ± 232
Percentage delay due to security protocol (%)	14.15 ± 0.25	14.40 ± 0.26

significantly with mobile handover because it limits the distance between the BLE sensor and the mobile.

1) IMPACT OF SECURITY

We measure the impact of the proposed security schemes on the system performance via several experiments as follows. These security features integrate with three main entities: the cloud server, the BLE sensor device, and the mobile relay device. We perform each experiment 30 times and obtain the average values.

C. CONNECTION ESTABLISHMENT DELAY

We measure the E2E connection establishment delay between the IoT node and the Cloud Server and show the results in the following Table 3.

From Table 3, it can be seen that the implementation of the proposing security protocol does not consume much processing time, compared with the total time it takes to establish the communication without the security.

TABLE IV AES 128 Bits Encryption Time Taken for Multiple File Sizes

File Size	Time taken without Encryption (ms)	Total Time taken with Encryption (AES 128bits) (ms)	Additional delay due to encryption in %
5 kB	1428 ± 21	1512 ± 21	5.58 ± 0.20
10 kB	2020 ± 19	2130 ± 19	5.19 ± 0.12
20 kB	3986 ± 43	4207 ± 43	5.26 ± 0.14
30 kB	6123 ± 92	6465 ± 92	5.30 ± 0.24
50 kB	10092 ± 188	10673 ± 188	5.46 ± 0.25
75 kB	14883 ± 277	15754 ± 277	5.54 ± 0.26
100 kB	20983 ± 163	22145 ± 163	5.25 ± 0.10
150 kB	31258 ± 508	32983 ± 508	5.24 ± 0.21
300 kB	62410 ± 933	66912 ± 933	5.24 ± 0.19
500 kB	100623 ± 911	106570 ± 911	5.58 ± 0.12

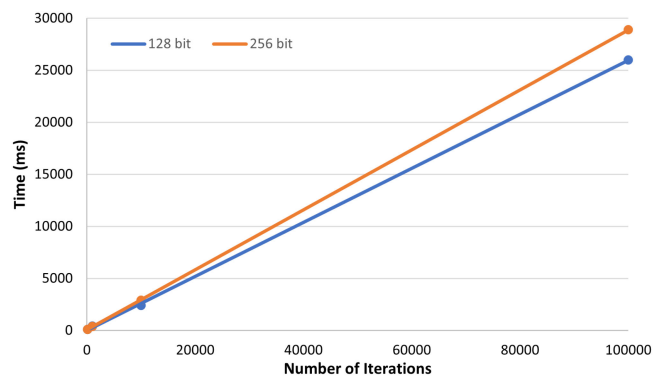


FIGURE 13. Server Side data processing time with number of iterations for 128 and 256 bits master key length.

1) IMPACT ON E2E LATENCY

In the next experiment, we measure the impact of security on E2E latency. We evaluate the time taken for AES (Advanced Encryption Standard) encryption by the IoT device for the same file sizes we used in Table 2, ranging from 5 kB to 500 kB. The Table 4 presents the results of the experiments with a confidence interval of 95%.

From Table 4, we can observe that the encryption only takes a small fraction of time when comparing it with the time taking for sending data without encryption. The delay percentage is almost the same for all file sizes (around 5%) but the actual time takes for encryption is low for small files (i.e., 84 ms for 5 kB) and high for bigger files (i.e., 5947 ms for 500 kB). Therefore, the impact of encryption on delay is lower for small amounts of data transmitting by the IoT device.

2) SCALABILITY (SERVER-SIDE)

In the next experiment, we measure the scalability of the proposing system by increasing the number of concurrent requests. The change of processing times with the number of server-side operations in security protocol shows in Fig. 13. This experiment utilizes a Linux computer with 4 GB RAM, Intel Core i5-4200 U CPU.

We measure the E2E connection establishment delay between the IoT node and Cloud. The results are displayed in Table 3.

For the CS, processing time increases linearly with the number of operations it handles. Therefore, the server can scale up to be ready for the increasing demand from the users of the system.

VIII. ANALYSIS OF SECURITY PROTOCOL

This section contains the security analysis of the proposed scheme. We use existing methods and tools to verify the security properties of our protocol. We will provide both a formal security analysis as well as an informal security analysis.

A. SECURITY ANALYSIS

1) FORMAL SECURITY ANALYSIS

We choose to use Rubin-logic [29] to perform the verification of the protocol. Rubin-logic is a method used to do security protocols analysis by several authentications and key agreement protocols [30]–[33]. This method is based on well-defined specifications and uses Global sets, local sets, secret sets and actions.

The protocol specifications are detailed below. The proposed scheme is executed by a group comprised of wearable devices (N) controlled by a mobile relay (F) connected to a CS.

Global Set:

- 1) Principal Set: N,F,S. N is the initiator of the protocol.
- 2) Rule Set: The inference rve new statements from existing assertions.
- 3) Secret Set: $\{k_m, id_n, c_n, id_f, c_f\}$
- 4) Observer Set:
 - Observer(k_m): {S}
 - Observer(id_f, c_f): {F}
 - Observer(id_n, c_n): {N}

Local Set: This set is defined for each principal, i.e., N, F, and S, respectively. As the key agreement process is being initiated by N, we start as follows:

• Principal N

- POSS(N): $\{id_n, a_n, b_n, c_n\}$
 BEL(N): $\{id_n, a_n, b_n, c_n\}$
 BL(N) =
- N1: Generate random value r_n
 - N2: $x_n \leftarrow a_n \oplus id_n$
 - N3: $y_n \leftarrow x_n \oplus r_n$
 - N4: $tid_n \leftarrow H(id_n \| t_n \| c_n \| r_n)$
 - N5: Send(F, $tid_n, y_n, a_n, b_n, t_n$)
 - N6: Receive(F, $\alpha_n, \eta_n, \mu_n, tid_f, y_p, \beta_{nf}$)
 - N7: $r_n^s \leftarrow \alpha_n \oplus c_n$
 - N8: $a_n^+ \leftarrow H(r_n^s \| id_n) \oplus \eta_n$
 - N9: $b_n^+ \leftarrow H(id_n \| r_n^s) \oplus \mu_n$
 - N10: $K_n \leftarrow H(c_n \| r_n \| r_n^s \| x_n \| t_n \| tid_n)$
 - N11: Derive through interpolation the shared key SK
 - N12: Verify ($H(SK \| t_n \| tid_n \| tid_f), \beta_{nf}$)

• Principal S

- POSS(S): $\{k_m, k_f, k_n, id_n, id_f\}$
 BEL(S): $\{k_f, k_n\}$
 BL(S) = S1: $k_f^* \leftarrow k_m \oplus a_f \oplus b_f$
 S2: $x_f^* \leftarrow H(k_m \| k_f^*)$
 S3: $id_f^* \leftarrow x_f^* \oplus a_f$
 S4: $r_f^* \leftarrow x_f^* \oplus y_f$
 S5: $c_f^* \leftarrow H(id_f^* \| k_m)$
 S6: Verify($H(id_f^* \| tid_n \| t_n \| c_f^* \| r_f^*), tid_f$)
 S7: $k_n^* \leftarrow k_m \oplus a_n \oplus b_n$
 S8: $x_n^* \leftarrow H(k_m \| k_n^*)$
 S9: $id_n^* \leftarrow x_n^* \oplus a_n$
 S10: $r_n^* \leftarrow x_n^* \oplus y_n$
 S11: $c_n^* \leftarrow H(id_n^* \| k_m)$
 S12: Verify($H(id_n^* \| t_n \| c_n^* \| r_n^*), tid_n$)
 S13: Generate random values r_n^s, k_n^+ for N
 S14: Generate random values r_f^s, k_f^+ for F
 For i=f and i=n do the following
 S15: $a_i^+ \leftarrow id_i^* \oplus H(k_m \| k_i^+)$
 S16: $b_i^+ \leftarrow a_i^+ \oplus k_m \oplus k_i^+$
 S17: $\eta_i \leftarrow H(r_i^s \| id_i^*) \oplus a_i^+$
 S18: $\mu_i \leftarrow H(id_i^* \| r_i^s) \oplus b_i^+$
 S19: $\alpha_i \leftarrow c_i^* \oplus r_i^s$
 S20: $c_n^* \leftarrow H(id_n^* \| k_m)$
 S21: $K_i \leftarrow H(c_i^* \| r_i^s \| r_i^s \| x_i^* \| t_n \| tid_i)$
 S22: Derive through interpolation the shared key SK
 S23: $\beta_{nf} \leftarrow H(SK \| t_n \| tid_n \| tid_f)$
 S24: Send(F, $\alpha_n, \eta_n, \mu_n, \alpha_f, \eta_f, \mu_f, y_p, \beta_{nf}$)

• Principal F

- POSS(F): $\{id_f, a_f, b_f, c_f\}$
 BEL(F): $\{id_f, a_f, b_f, c_f\}$
 BL(F) =
- F1: Receive(N, $tid_n, y_n, a_n, b_n, t_n$)
 - F2: Generate random value r_f
 - F3: $x_f \leftarrow a_f \oplus id_f$
 - F4: $y_f \leftarrow x_f \oplus r_f$
 - F5: $tid_f \leftarrow H(id_f \| tid_n \| t_n \| c_f \| r_f)$
 - F6: Send(S, $tid_n, y_n, a_n, b_n, t_n, tid_f, y_f, a_f, b_f$)
 - F7: Receive(S, $\alpha_n, \eta_n, \mu_n, \alpha_f, \eta_f, \mu_f, y_p, \beta_{nf}$)
 - F8: $r_f^s \leftarrow \alpha_f \oplus c_f$
 - F9: $a_f^+ \leftarrow H(r_f^s \| id_f) \oplus \eta_f$
 - F10: $b_f^+ \leftarrow H(id_f \| r_f^s) \oplus \mu_f$
 - F11: $K_f \leftarrow H(c_f \| r_f \| r_f^s \| x_f \| t_n \| tid_f)$
 - F12: Derive through interpolation the shared key SK
 - F13: Verify ($H(SK \| t_n \| tid_n \| tid_f), \beta_{nf}$)
 - F14: Send(N, $\alpha_n, \eta_n, \mu_n, tid_f, y_p, \beta_{nf}$)

Below we proceed with the protocol verification. The verification process starts with execution of the actions in BL(N). In actions from N1 – N5, N computes x_n, y_n, tid_n and sends $m_1 = tid_n, y_n, a_n, b_n, t_n$ to F.

Hence, the local sets of N are changed as follows:

- POSS(N) = $\{id_n, a_n, b_n, c_n, tid_n, y_n, t_n, r_n, x_n\}$
- BEL(N) = $\{id_n, a_n, b_n, c_n, tid_n, y_n, t_n, r_n, x_n\}$

The global sets are updated as follows:

- Secret set: $\{r_n, x_n\}$

- Observer sets:

$$\text{Observer}(r_n, x_n): \{N\}$$

Upon receiving m_1 , F in actions (F2) – (F6), generates a random value r_f , computes x_f, y_f, tid_f and $m_2 = \{tid_n, y_n, a_n, b_n, t_n, tid_f, y_f, a_f, b_f\}$ to S.

Now, the local sets at F change as shown below

- $\text{POSS}(F) = \{id_n, a_n, b_n, tid_n, y_n, t_n, tid_f, y_f, a_f, b_f, x_f, r_f\}$
- $\text{BEL}(F) = \{id_n, a_n, b_n, tid_n, y_n, t_n, tid_f, y_f, a_f, b_f, x_f, r_f\}$

The global sets at F change as follows:

- Secret set: $\{r_f, x_f\}$
- Observer sets:

$$\text{Observer}(r_f, x_f): \{F\}$$

After completing the action in (F6), the actions (S1)–(S12) in BL(S) are performed where S checks the received values from F. At the end of these steps, S will have successfully verified tid_n and tid_f , the identities of N and F, respectively. In case the verification fails, the protocol execution should be aborted. The actions in (S13) – (S24) are performed after which the shared key SK is derived through interpolation.

Then, the local sets of S are changed as follows.

- $\text{POSS}(S) = \{K_f, K_n, SK\}$
- $\text{BEL}(S) = \{K_f, K_n, SK\}$

Now the global sets of S are updated as follows:

- Secret set: $\{K_f, K_n, SK\}$
- Observer sets:
 - $\text{Observer}(K_f): \{F, S\}$
 - $\text{Observer}(K_n): \{N, S\}$
 - $\text{Observer}(SK): \{S\}$

After receiving message m_3 in F7, F executes actions in (F8) – (F14) thus deriving the shared key SK and the session key K_f . F then sends message $m_4 = \{\alpha_n, \eta_n, \mu_n, tid_f, y_p, \beta_{nf}\}$ to N. The local sets of F are finally changed as follows.

- $\text{POSS}(F) = \{SK, K_f\}$
- $\text{BEL}(F) = \{SK, K_f\}$

The global sets are updated as follows:

- Secret set: $\{SK, K_f\}$
- Observer sets:
 - $\text{Observer}(SK, K_f): \{F, S\}$

Upon receiving the message m_4 in N6, N executes the actions in (N7) – (N12) and derives the shared key SK and the session key K_n . The local sets of N are finally changed as follows.

- $\text{POSS}(N) = \{SK, K_n\}$
- $\text{BEL}(N) = \{SK, K_n\}$

The global sets are updated as follows:

- Secret set: $\{SK, K_n\}$
- Observer sets:
 - $\text{Observer}(SK, K_n): \{N, S\}$

This result implies that:

- r_f, r_n as well as k_f, k_n are fresh for each session, and are known by the legitimate F, N and S.
- Only the legitimate entities N, F, and S, are able to derive the common shared key SK .
- S can verify the identities of N and F.

This result proves that our scheme resists against the attack model described in Section V-B.

2) INFORMAL SECURITY ANALYSIS

Let us now discuss the strength of the proposed protocol with respect to the obtained security features and the resistance against the attacks considered in the required security features in Section V-A.

- **Confidentiality:** The transmitted messages m_1, m_2, m_3, m_4 have no meaning without knowledge of the secret key material.
- **Mutual authentication:** The common shared keys between CS and the mobile relay K_f on the one hand and CS and the wearable device K_n , on the other hand, contain a random variable derived by each of the involved entities. As the common key SK is based on these keys K_f, K_n , each entity has contributed to its construction.
- **Anonymity:** This feature is guaranteed by the fact that the dynamic identity of the wearable device (a_n, b_n) and the mobile relay (a_f, b_f), sent in messages m_1, m_2 is updated independently in each request. Without knowing the key material stored in the tamper-proof part of the memory, it is impossible to derive the relation with the real identity.
- **Unlinkability:** To derive the updated dynamic identity, knowledge of the parameters c_i, id_i is required; these parameters are securely stored by the wearable device $i = n$ and mobile relay $i = f$.
- **Node capture attack:** Suppose the node is captured and even the key material stored in the tamper-resistant memory is leaked, the security for that node is completely broken. However, the attack remains local as it has no impact on other devices in the system since each device has its specific construction. The master key cannot be retrieved from the stored values without breaking the hash function.
- **Impersonation and MITM attacks:** These attacks are not possible due to the mutual authentication feature. However, special care should be given to a malicious mobile relay node willing to take over the request sent to another mobile relay. For the computation of K_n , the server also includes the temporary identity of the mobile relay tid_f in the hash, which is also added in clear text by the mobile relay to the message m_4 . As a consequence, it is not possible for another mobile relay to change the identity of the mobile relay without CS being aware of it.
The verification parameter β_{nf} involves the temporary identities of the entities, both mobile relay and wearable device, and therefore, no impersonation attacks can be performed. In addition, mobile relay and the wearable device are ensured in this way that CS has legitimated them.
- **Replay attacks:** are avoided by using both timestamps and random values in the protocol.

TABLE V The Cryptographic Operations That Device, Relay Node and CS Need to Perform for a Key Agreement Scheme. Note That T_b = Time for Bilinear Pairing, T_{mp} = Time for Point Multiplication, T_{ap} = Time for Point Addition, T_s = Time for Symmetric encryption/decryption, T_H = Time for Map to Point, T_h = Time for Hash Operation, T_x = Time for XOR Operation

Scheme	Cost for wearable device	Cost for mobile relay	Cost for CS
[23]	$2T_{mp} + 1T_b + 6T_h$	$2T_{mp} + 1T_b + 4T_h$	$3T_{mp} + 1T_b + 11T_h$
[25]	$T_H + 5T_{mp} + 1T_b + 3T_{ap} + 4T_h$	$4T_H + 13T_{mp} + 7T_b + 7T_{ap} + 8T_h$	$T_H + 6T_{mp} + 3T_b + 4T_{ap} + 5T_h$
[26]	$7T_{mp} + 2T_{ap} + 12T_h + 2T_s$	$7T_{mp} + 2T_{ap} + 13T_h + 2T_s$	$9T_{mp} + 4T_{ap} + 13T_h + 4T_s$
Ours	$5T_x + 5T_h$	$5T_x + 5T_h$	$14T_x + 11T_h$

TABLE VI Comparison of Proposed Solution With the Existing Pertinent Works

Characteristic	Ref. [14]	Ref. [10]	Ref. [15]	Ref. [11]	Ref. [8]	Ours
BLE Support	-	✓	-	✓	✓	✓
Support for Third party Relay	-	-	-	✓	✓	✓
E2E Encryption	✓	✓	✓	✓	✓	✓
Automatic Relay Handover	✓	-	✓	-	-	✓
Multi Connect	-	✓	-	-	-	✓
Transmission of Real-time data	✓	✓	✓	✓	✓	✓
Store and Forwarding of data	✓	✓	✓	-	-	✓
Support for Load Balancing	✓	-	-	-	-	✓
Confidentiality Protection	✓	✓	✓	-	✓	✓
Mutual authentication	✓	✓	✓	✓	✓	✓
Anonymity	-	-	-	-	-	✓
Unlinkability	-	-	-	-	-	✓

- **Online/offline dictionary attack:** Guessing the master key is useless as another temporary key needs to be determined as well. This key changes in each communication request.

Guessing the identity of the node results in deriving x_n, r_n , but will not lead to finding the random value r_n^s chosen by CS as that requires knowledge of c_n . Both values need to be known to derive the key material with the wearable device. Moreover, to check the validity by using the hash value β_{nf} , and the identity material of the relay id_f, c_f should be guessed. Consequently, in order to make a successful dictionary attack, four different parameters id_f, c_f, id_n, c_n should be guessed.

B. PERFORMANCE ANALYSIS

The proposed scheme is very efficient in terms of computation and communication. For the resource-constrained wearable devices, only 5 XOR and 5 hash operations are executed. Table 5 shows a comparison between our scheme and related works in the literature. By using hash functions and negligible XOR operations, we avoid compute-intensive pairing operations as observed in many cryptographic algorithms [23], [25], [26] thus offering a secure exchange of data at a very low cost.

IX. DISCUSSION

A. COMPARISON WITH EXISTING WORK

Table 6 shows the added features of our proposed architecture compared with existing solutions. From this table, it is

clear that our proposed system is a unique solution and addresses many problems existing with similar proposals and implementations.

Our system proposes and implements a mobile relay-based procedure, and it provides anonymity and unlinkability with our security protocol. It also ensures the confidentiality of all parties involved in establishing communication. Comparing with similar works, we can state that most of the solutions do not provide those features. In addition, our solution proposes multi-connect, load balancing features, and we implement the automatic relay handover, which is not available in many related works.

B. LIMITATIONS

1) LIMITATIONS IN COMMUNICATION PROTOCOL

Though we have proposed and implemented a highly sophisticated communication protocol, there are a few limitations associated with this design. In this case, the protocol is limited to the usage of smartphones that support BLE. That means it supports Android versions from Android 4.3 (API Level 18) [34] and it does not work with legacy Bluetooth versions. In addition to that, the relay mobile should be powered with both BLE transmission and reception capabilities. Moreover, as indicated from Section VII-B, there can be issues associated with small inter-frame gaps when the distance between mobile relay and IoT device increases. Therefore, it is evident that we define somewhat higher inter-frame intervals such as 10 ms. Furthermore, we limit the associated implementations of the proposed protocol for connection between an IoT handheld device to a single mobile; however, it can expand multiple connections as described in Section IV-B.

2) LIMITATIONS IN SECURITY

The proposed protocol also has a few limitations related to security. First, secure key storage is required in wearable devices and mobile relay. Also, the centralized CS is a single point of attack since the whole system security relies on the master key $\{k_m\}$.

3) LIMITATIONS IN MOBILE APP

In our proposed system, the relay user has to install the mobile app. He should also agree to share the location data for BLE discovery and Internet connection to upload the received IoT data. Since this is a burden for relay users, a subscription model must be developed to motivate relay users. For

instance, the IoT device owners should pay an annual subscription for the services, and mobile relay owners receive a fee for their service. However, the development of such a business model is beyond the scope of this paper.

The proposed system works for any smartphone which has BLE connectivity. In the current implementation, we have developed an Android app, but it can be implemented to other mobile OSs (Operating Systems) such as iOS and Windows. Since the underlying concept of the proposed system is independent of the mobile OS, it is technically feasible to implement the proposed system with any mobile OS.

The relay service periodically scans nearby relay devices and establishes an automatic connection with them in the proposed system. Since mobile OSs do not provide such service by default, it is necessary to install a dedicated mobile app. Although BLE is available for many mobile phones, the proposed solutions can not be implemented with low-end mobile phones which do not support third-party application installation.

C. RELIABILITY OF THIRD PARTY BLE RELAY

The proposed mobile relay solution works as a secondary option to establish the connection. The user owning the dedicated mobile phone acts as the primary relay device for any IoT device. The third-party mobile relays are used as an additional enhancement to maintain the availability of the connection when the owner's mobile device (primary relay) goes offline or gets damaged. In such a case, if there is no mobile device available nearby, data will be saved locally in the sensor node, which can later be submitted to the server whenever a relay device is available. In addition, transmitted IoT data via third-party BLE relays remain in the local storage until the cloud responds with a positive acknowledgment.

D. USEFULNESS OF EXTRA SESSION KEY

The proposed key exchange mechanism is distributing different keys between the entities. In the current version of the proposed architecture, the mobile relay just acts as a relay and forwards the received data from sensor nodes. Thus, wearable devices currently only use the key K_n for encryption since we support E2E encryption. However, our key agreement phase supports another session key (i.e., SK) shared between all three entities. If the mobile relay can do some preprocessing for data at the relay itself, wearable devices can use the key SK for encryption. That way, the mobile relay can decrypt and process the data. The preprocessing of data at mobile relay will be carried out in future research.

E. CHOICE OF BLE FOR COMMUNICATION LINK ESTABLISHMENT

We use BLE as the short-range communication establishment method as most mobile phones already support BLE. Furthermore, the work in [3] shows that BLE provides many features that get wide adoption by the mobile manufacturers to be added in the daily life of mobile devices. Therefore, compared with 6LoWPAN, we select BLE, as our use-case is based on

mobile relay devices. The work in [3] also shows that BLE is a superior option to be used over many alternative technologies such as Zigbee, ANT, or NFC for u-Healthcare applications.

X. CONCLUSION

This paper proposed a secure BLE relay-based emergency detection system for AAL. The proposed solution extends the features of existing solutions by adding new capabilities such as multi-connect, automatic handover, storage, forwarding data, load balancing, security features like confidentiality, mutual authentication, unlinkability, and anonymity. We developed a prototype of the proposed solution and performed several experiments to get insights into the proposed system's performance. It was observed that multi-core processing of real-time and stored data separately is a better solution than processing both together sequentially. Furthermore, the experiments revealed that limiting the distance of the mobile from the sensor is vital for data communication reliability. Therefore, the BLE sensor's handover mechanism to change from a distant mobile relay to a closer one is an important asset.

To enhance the security of the system, we also proposed a novel secure symmetric key agreement scheme. The proposed method is secure as building a shared common key is based on freshly generated parameters. Moreover, our scheme is efficient in terms of computation since it uses very low-cost cryptographic operations (i.e., concatenation, XOR, and hash functions). We have used Rubin logic to provide this scheme's security strengths. We have also done an informal analysis of the scheme for several security threats i.e. node capturing attack, impersonation attack, man-in-the-middle attack, replay attack, online/offline dictionary attack.

We intend to extend the work to machine learning algorithms to detect emergencies and anomalies in both health and security-related data in future work.

REFERENCES

- [1] H. Sun, V. De Florio, N. Gui, and C. Blondia, "Promises and challenges of ambient assisted living systems," in *Proc. 6th Int. Conf. Inf. Technol.: New Generations*, 2009, pp. 1201–1207.
- [2] B. Farahani, F. Firouzi, and K. Chakrabarty, "Healthcare IoT," in *Proc. Intell. Internet Things*, 2020, pp. 515–545.
- [3] F. T. R. Tabish, A. B. Mnaouer and A. M. Ghaleb, "A comparative analysis of BLE and 6LoWPAN for U-HealthCare applications," in *Proc. 7th IEEE GCC Conf. Exhib.*, 2013, pp. 1–5.
- [4] "Bluetooth[®] Technology Website, 2021. Specifications | Bluetooth[®] Technology website," [Accessed 8 April 2021]. [Online]. Available: <https://www.bluetooth.com/specifications/archived-specifications>
- [5] K. Townsend *et al.* *Getting Started With Bluetooth Low Energy: Tools and Techniques for Low-Power Networking*, ser. *EBSCOhost ebooks online*. O'Reilly Media, 2014, [Online]. Available: <https://books.google.lk/books?id=24N7AwAAQBA>.
- [6] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of Bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11734–11753, 2012.
- [7] D. Dagon, T. Martin, and T. Starner, "Mobile phones as computing devices: The viruses are coming!" *IEEE Pervasive Comput.*, vol. 3, no. 4, pp. 11–15, Oct.–Dec. 2004.
- [8] P. Porambage, A. Manzoor, M. Liyanage, A. Gurtov, and M. Ylianttila, "Managing mobile relays for secure E2E connectivity of low-power IoT devices," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf.*, 2019, pp. 1–7.

- [9] A. Manzoor, P. Porabage, M. Liyanage, M. Ylianttila, and A. Gurtov, "Mobile relay architecture for low-power IoT devices," in *Proc. IEEE 19th Int. Symp. A World Wireless, Mobile Multimedia Netw.*, 2018, pp. 14–16.
- [10] S. Raza, P. Misra, Z. He, and T. Voigt, "Building the Internet of Things with Bluetooth smart," *Ad Hoc Netw.*, vol. 57, pp. 19–31, 2017.
- [11] M. Haus, A. Y. Ding, and J. Ott, "Managing iot at the edge: The case for BLE beacons," in *Proc. 3rd Workshop Experiences with Des. Implementation Smart Objects*, 2017, pp. 41–46.
- [12] J.-W. Yoon, Y.-k. Ku, C.-S. Nam, and D.-R. Shin, "Sensor network middleware for distributed and heterogeneous environments," in *Proc. Int. Conf. New Trends Inf. Service Sci.*, 2009, pp. 979–982.
- [13] H. Wirtz, T. Zimmermann, M. Serror, and K. Wehrle, "Collaborative on-demand WI-FI sharing," in *Proc. IEEE 40th Conf. Local Comput. Netw.*, 2015, pp. 19–27.
- [14] A. M. Rahmani et al., "Exploiting smart E-health gateways at the edge of Healthcare internet-of-things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, 2018.
- [15] S. R. Moosavi et al., "End-to-end security scheme for mobility enabled Healthcare internet of things," *Future Gener. Comput. Syst.*, vol. 64, pp. 108–124, 2016.
- [16] L. Gong, "Lower bounds on messages and rounds for network authentication protocols," *Proc. 1st ACM Conf. Comput. Commun. Secur.*, 1993, pp. 26–37.
- [17] C. Lee, S. Chen, and C. Chen, "A computation-efficient Three-Party encrypted key exchange protocol," *Appl. Math. Inf. Sci.*, vol. 6, no. 3, pp. 573–579, 2012.
- [18] X. Li, J. Niu, S. Kumari, M. Khan, L. Liao, and W. Liang, "Design and analysis of a chaotic maps-based three-party authenticated key agreement protocol," *Nonlinear Dyn.*, vol. 80, no. 3, pp. 1209–1220, 2015.
- [19] T. Lee and T. Hwang, "Three-party authenticated key agreements for optimal communication," *J. SomeThing*, vol. 12, no. 3, pp. 1–25, 2017.
- [20] L. Ni, G. Chen, and J. Li, "Escrowable identity-based authenticated key agreement protocol with strong security," *Comput. Math. Appl.*, vol. 65, no. 9, pp. 1339–1349, 2013.
- [21] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions," *Comput. Netw.*, vol. 73, pp. 41–57, 2014.
- [22] Y. Chen, J. Martinez, P. Cattlejo, and L. Lopez, "An anonymous authentication and key establish scheme for smart grid: Fauth," *Energies*, vol. 10, no. 1345, pp. 1–23, 2017.
- [23] X. Jia, D. He, N. Kumar, and K. Choo, "Authenticated key agreement scheme for fog-driven IoT Healthcare system," *Wireless Netw.*, Springer, vol. 25, no. 8, pp. 4737–4750, 2019.
- [24] H. Hamid, S. Rahman, M. Hossain, A. Almogren, and A. Alamri, "A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography," *IEEE Access*, vol. 5, pp. 22313–22328, 2017.
- [25] C. Liu, W. Tsai, T. Chang, and T. Liu, "Ephemeral-secret-leakage secure id-based three party authenticated key agreement protocol for mobile distributed computing environments," *Symmetry*, vol. 10, no. 4, p. 24, 2018.
- [26] S. Patonico, A. Braeken, and K. Steenhaut, "Identity-based and anonymous key agreement protocol for fog computing resistant in the Canetti-Krawczyk security model," *Wireless Netw.*, 2019, pp. 1–13.
- [27] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.
- [28] H. K. Hughes and L. K. Kahl, *A Manual for Pediatric House Officers, the Harriet Lane Handbook*. New York, NY, USA: Elsevier, 2018.
- [29] A. D. Rubin and P. Honeyman, "Nonmonotonic cryptographic protocols," in *Proc. Comput. Secur. Foundations Workshop VII*, 1994, pp. 100–116.
- [30] A. Nicholson, M. Corner, and B. Noble, "Mobile device security using transient authentication," *IEEE Trans. Mobile Comput.*, vol. 5, no. 11, pp. 1489–1502, Nov. 2006.
- [31] M. A. Mughal, X. Luo, A. Ullah, S. Ullah, and Z. Mahmood, "A lightweight digital signature based security scheme for human-centered Internet of Things," *IEEE Access*, vol. 6, pp. 31630–31643, 2018.
- [32] A. Braeken, M. Liyanage, P. Kumar, and J. Murphy, "Novel 5G authentication protocol to improve the resistance against active attacks and malicious serving networks," *IEEE Access*, vol. 7, pp. 64040–64052, 2019.
- [33] P. Shabisha, A. Braeken, P. Kumar, and K. Steenhaut, "Fog-orchestrated and server-controlled anonymous group authentication and key agreement," *IEEE Access*, vol. 7, pp. 150247–150261, 2019.
- [34] "Android developers. 2021. Bluetooth low energy," Accessed: Apr. 8, 2021. [Online]. Available: <https://developer.android.com/guide/topics/connectivity/bluetooth-le>



computing, elliptic curve cryptography, computer and network security.

PLACIDE SHABISHA received the M.Sc degree in communication networks from the University of Sidi Bel Abbes, Algeria, in 2012. Since 2013, he has been an Assistant Lecturer with the Department of Information and Communications Technology, University of Burundi, Bujumbura, Burundi. He is currently working toward the Ph.D. degree in engineering sciences with Vrije Universiteit Brussel, Belgium with a scholarship from the VLIR-UOS project: IUC 2017 Phase 3 UB. His research interests include Internet of Things, cloud



CHAMARA SANDEEPA (Student Member, IEEE) received the B.Sc. Eng. degree (Second Class Honours Upper Division) in electrical and information engineering from the University of Ruhuna, Galle, Sri Lanka. His degree program included subjects from Electrical, Telecommunication, Electronics and Software Engineering while he is specialized under Software Engineering sector. His research interests include IoT, e-Healthcare, computer science, and data science.



current research interests include blockchains, quantum resistant security, 5G, 6G, software defined networking, virtualization, Internet of Things, ambient assisted living, BLE-contact tracing, mobile communications, and artificial intelligence.

CHARUKA MOREMADA (Student Member, IEEE) received the Bachelor of the Science of Engineering degree from the Department of Electrical and Information Engineering, Faculty of Engineering, University of Ruhuna, Sri Lanka, in 2020, specialized in electrical and information engineering B.Sc.Eng.(First Class Hons.). He is currently a Temporary Instructor with the Department of Computer Engineering, Faculty of Engineering, University of Peradeniya, Sri Lanka. He has coauthored some publications in certain research areas and his



NADEEKA DISSANAYAKA (Student Member, IEEE) is a graduate from Faculty of Engineering, University of Ruhuna Galle, Sri Lanka. She received the Bachelor of Science of Engineering degree with Second Class Honours Upper Division, specialized in electrical and information engineering from the Department of Electrical and Information Engineering of the same faculty in 2020. She has coauthored in some publications and her research interests include IoT, e-Healthcare, and electrical engineering aspects.



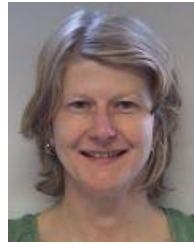
THARINDU GAMAGE received the first degree from the Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka, in 2009. He is currently a Lecturer with the Department of Electrical and Information Engineering, University of Ruhuna, Sri Lanka. He was a Research Assistant for several years with the Department of Electronic and Telecommunication Engineering, University of Moratuwa. During the time he was a Research Assistant, he received his M.Sc. from the same department of University of

Moratuwa, Sri Lanka in 2014. His research interests include IoT, embedded systems, high performance computing, and medical image processing.



AN BRAEKEN received the M.Sc. degree in mathematics from the University of Gent, in 2002 and the Ph.D. in engineering sciences from the KU Leuven at the research group COSIC (Computer Security and Industrial Cryptography), in 2006. In 2007, she became a Professor with Erasmushogeschool Brussel (currently since 2013, Vrije Universiteit Brussel) with Industrial Sciences Department. Prior to joining the Erasmushogeschool Brussel, she worked for almost two years at the management consulting company

Boston Consulting Group (BCG). Her current interests include security and privacy protocols for IoT, cloud and fog, blockchain, and 5G security. She is coauthor of more than 120 publications. She is a Member of the program committee for numerous conferences and workshops (IOP2018, EUC 2018, and ICNS 2018) and a Member of the editorial board for Security and Communications magazine. She has also been member of the organizing committee for the IEEE Cloudtech 2018 conference and the Blockchain in IoT workshop at Globecom 2018. In addition, since 2015, she is a reviewer for several EU proposals and ongoing projects, submitted under the programs of H2020, Marie Curie and ITN. She has cooperated and coordinated more than 12 national and international projects. She has been a STSM Manager in the COST AAPELE project (2014-2017) and is currently in the management committee of the COST RECODIS project (2016-2019).



KRIS STEENHAUT (Member, IEEE) received the master's degree in engineering sciences and the master's degree in applied computer sciences and the Ph.D. degree in engineering sciences from Vrije Universiteit Brussel (VUB) in 1986 and 1995, respectively. She is currently a Professor with the Department of Electronics and Informatics and the Department of Engineering technology, Faculty of Engineering, Vrije Universiteit Brussel, Belgium. Her research interests include the design, implementation and evaluation of wireless sensor

networks for building automation, environmental monitoring, autonomous ground vehicle applications, and smart grids.



MADHUSANKA LIYANAGE (Senior Member, IEEE) received the B.Sc. degree (First Class Honours) in electronics and telecommunication engineering from the University of Moratuwa, Sri Lanka, in 2009, the M.Eng. degree from the Asian Institute of Technology, Thailand, in 2011, the M.Sc. degree from the University of Nice Sophia Antipolis, France, in 2011, and the Doctor of Technology degree in communication engineering from the University of Oulu, Finland, in 2016. From 2011 to 2012, he was a Research Scientist with I3S

Laboratory and Inria, Sophia Antipolis, France. He is currently an Assistant Professor/Ad Astra Fellow with the School of Computer Science, University College Dublin, Ireland. He is also acting as an Adjunct Professor with the Center for Wireless Communications, University of Oulu, Finland. He was also the recipient of the prestigious Marie Skłodowska-Curie Actions Individual Fellowship during 2018-2020. During 2015-2018, he has been a Visiting Research Fellow with CSIRO, Australia, the Infolabs21, Lancaster University, U.K., Computer Science and Engineering, The University of New South Wales, Australia, School of IT, University of Sydney, Sydney, NSW, Australia, LIP6, Sorbonne University, France and Computer Science and Engineering, The University of Oxford, U.K. His research interests include 5G or 6G, SDN, IoT, Blockchain, MEC, mobile and virtual network security. In 2020, he was the recipient of the 2020 IEEE ComSoc Outstanding Young Researcher Award by IEEE ComSoc EMEA.