Managing Mobile Relays for Secure E2E Connectivity of Low-Power IoT Devices

Pawani Porambage*, Ahsan Manzoor†, Madhsanka Liyanage‡, Andrei Gurtov§ and Mika Ylianttila¶

*†‡¶Centre for Wireless Communications, University of Oulu, Finland
{firstname.lastname}@oulu.fi
§Department of Computer and Information Science, Linköping University, Sweden
gurtov@acm.org

Abstract—The widespread Internet of Things (IoT) ecosystems empower the deployment of various Bluetooth Low Energy (BLE) sensor nodes in many ambient assisted living (AAL) type applications. Regardless of their limitations, these low-power IoT sensor nodes need pervasive and secure connections to transfer the aggregated data to the central servers located in remote clouds which will perform further processing and storing functions. The common practice is to use one or multiple dedicated gateways to assist the communication between the sensor and the cloud. This paper presents a mobile-based relay assistance solution for establishing secure end-to-end (E2E) connectivity between low-power IoT sensors and cloud servers without using a dedicated gateway. The prototype implementation and the described security features verify the technical readiness of the proposed solution.

Index Terms—Bluetooth Low Energy, Internet of Things, Relay, Sensors, Ambient Assisted Living

I. INTRODUCTION

With the proliferation of the Internet of Things (IoT) technologies, in addition to the key purpose of using a particular smart object, people are fascinated to access or obtain multiple other services from the same device [1], [2]. Moreover, the seamless connectivity and increased quality of experience are highly regarded for better user experiences. Many IoT devices in health-care and Ambient Assisted Living (AAL) applications are equipped with unlicensed band short-range radio access technologies, including Bluetooth Low Energy (BLE), HaLow, ZigBee, and Smart Utility Networks (SUNs) [3]. Among them, BLE is the best-known and most used low-power communication technology that supports connectivity for Body Area Networks (BANs) and a large number of medical IoT devices which operate with coin cell batteries [4]. Although the Low Power Wide Area Network (LPWAN) radio technologies such as Narrow Band IoT (NB-IoT) and LoRaWAN are recently becoming popular, they are not widely deployed in medical and wearable IoT applications.

BLE is a stateless protocol where the requests are independently transmitted and it allows a very flexible topology which can be adjusted to fit into a large number of use cases. The resource-constrained IoT devices that use standalone BLE for communication purposes require dedicated nodes that serve as local gateways (GWs) to provide back-end connectivity with the remote IoT cloud/data centers. In the typical AAL applications, the BLE wearable pair with the user’s mobile phone which is acting as the GW to obtain end-to-end (E2E) connectivity with the cloud servers. Instead of using a committed single GW, if the wearable can get the data forwarding services from unknown relaying nodes in the closest proximity, it will provide high user mobility even to foreign environments.

The exploitation of mobile-based relays for the back-end connectivity of BLE devices is described for an AAL use case in Figure 1. The elderly or the people with chronic conditions may require continuous monitoring of their health records or localize with the help of different wearable BLE sensors. The interested parties (e.g., family or caretakers) can track their behavior or examine the health conditions based on the data retrieved from the remote central cloud. By using a dedicated mobile phone, the back-end connectivity between the BLE wearable sensor and the cloud data center may drop when the user is going beyond the comfort zone or the phone-battery is dead. The uninterrupted E2E connectivity can be maintained between the wearable and the cloud data centers with the help of some random mobile users who are performing as relays.

In order to keep the in-line with this mechanism, the unknown mobile user needs to be rewarded by the remote cloud for his relaying service.

![Fig. 1: Usecase scenario](image)

Our key contributions of this paper include the design and development of on-demand sharing network instantiating for universal accessibility of BLE sensors with the help of mobile relays. The simulation results and the implementation with the commercial off-the-shelf devices prove the technical readiness of the proposal. The security considerations of the protocol are discussed along with the possible attack scenarios. To the best of our knowledge, this will be the first attempt of...
exploiting third-party unknown mobile relays for the forwarding of medical data generated by BLE sensors.

The remainder of the paper is organized as follows: Section II provides the related work. Section III and IV describe the proposed network architecture and protocol. Section V and VI present the simulation and implementation results, and the security considerations of the protocol. Finally, Section VII concludes the paper by drawing the future research directions.

II. RELATED WORK

BLE version 5.0 is the latest addition from the Bluetooth Special Interest Group [5]. It is a wireless personal area networking technology that can be exploited in a very broad range of applications including health-care, fitness and elderly care, smart homes, smart cities, etc. BLE plays a vital role in many other short-range wireless communication protocols such as Zigbee, Near Field Communication (NFC), 6LoWPAN [6]. Moreover, the coexistence of BLE with the different varieties of devices ranging from mobile phones to very resource-constrained medical sensors has proven its usability as an IoT technology. As presented in [4], it is feasible to implement a complete IP-based protocol stack on constrained BLE devices and enable gateway operations covering global IPv6 connectivity and proxy-cache functionality.

The proposed relay-assisted E2E connection establishment for low-power IoT devices was initiated by the idea behind collaborative on-demand Wi-Fi sharing (COWS) appeared in [7]. The ubiquitous access to the foreign private Wi-Fi network is mostly prohibited since the private Access Points (APs) have no means of authenticating foreign users before granting access to their network. COWS comprehensively discover roaming networks by opportunistically broadcasting connection requests and offers 802.11 authentications at private APs by embedding user credentials. At the end of successful authentication at private APs, the Internet traffic is tunneled through the user’s Home Provider (HP). Although COWS has a rather small footprint in terms of communication, it is not really designed for resource-constrained devices. Simply, we commenced this work as an extension of COWS for IoT sensors. In our design, we intend to adopt the message flow of COWS and tailor it according to protocol specifications of BLE. Instead of APs in COWS, the intermediary needs to be the smart-phone which supports BLE and has the connectivity to the Internet.

In [8], Raza et. al. present a design of a novel open hardware platform for BLE and discuss the possibility of implementing 6LoWPAN-connected Bluetooth Smart. They use nRF24Cheep, the custom designed BLE beacon platform, to perform experiments on adjusting BLE message formats. Moreover, they provide Contiki operating system port to the same open BLE broadcast platform. Although this work is in a very abstract form, it provides an initial guidance on how to establish a BLE connected IoT setup and its prototyping for a full open-source BLE stack.

The newly introduced edge computing paradigms (E.g. Fog computing and Multi-Access Mobile Edge Computing) leverage the low-latency requirements of IoT applications. The work presented in [9] describe the exploitation of smart e-health gateways, called UT-GATE, at the edge of health-care IoT in clinical environments. According to their proposed fog-assisted system architecture, the smart e-health gateway provides an intermediary layer of intelligence between sensor nodes and cloud. The resource consuming tasks of the sensors can be outsourced at the gateway and support ubiquitous characteristics such as mobility, energy efficiency, scalability, and reliability issues. They have demonstrated the functionality of UT-GATE with different network topology such as mesh and star and using wireless sensor technologies like 6LoWPAN, Wi-Fi, and BLE. However, this solution still lacks the universal connectivity in the roaming situations.

Another research followed by Haus et. al., in [10], investigate the usability of an iConfig edge-driven platform for IoT device management including BLE beacons in smart cities. In this experiment, the user with a smart-phone running the iConfig edge module will assist the BLE beacons, emitted by IoT devices, in three phases namely registration, configuration, and maintenance. The automatic configuration and further processing are performed by the edge module with the minimal user interaction. This particular work does not address how the IoT sensors can maintain E2E connectivity with back-end cloud data centers.

With the advent of technological upgrades, BLE will no longer support only the monitoring and data collecting applications. The researches are underway to investigate its applicability for different areas such as multimedia streaming [11]. However, in any case, security is a mandatory property to guarantee in E2E IoT communication scenarios [12].

III. USE CASE, NETWORK ARCHITECTURE AND ASSUMPTIONS

This section describes the network architecture, key assumptions, and prerequisites taken into account for designing the protocol.

A. Network Architecture

The network architecture is illustrated in Figure 2 with reference to the AAL use case. BLE sensor advertises its availability of data. There can be one or number of anonymous mobile phones who receive the advertisement and accept to cooperate with further communication as a relay node. The best relay node is selected based on the received signal strength indicator (RSSI). The link between the mobile and the central server (CS) in IoT cloud will be securely established over the Internet in a conventional manner (e.g., Hypertext Transfer Protocol Secure (HTTPS)). When the data is received from the BLE device, CS will update the database which is dedicated to that particular user (or device).

B. Pre-requisites and Assumptions

The BLE device should undergo an initial registration with the CS, which can be performed offline or via a trusted third party (e.g., trusted GW). The sensor should have the identity
of the CS and the CS should maintain the records of all the legitimate sensors and their functioning databases. In order to maintain E2E secure communication, the BLE device and CS should share the cryptographic keys for data encryption and decryption, and the authentication credentials (e.g., User ID (UID) and hash chain).

Throughout the protocol, we consider the guaranteed link quality for all the communication channels. Furthermore, it is assumed that the connections remain uninterrupted over the period of communication and the mobile device will operate in the same proximity. No mutual or transitive trusts are required between the relay device and the other entities (i.e., sensor and CS). However, the mobile phone can always proceed the relaying functions (forwarding sensed data OR traffic) as long as it is rewarded. For the sake of rewarding mechanism, the mobile needs to be registered with CS in advance and the secure links (i.e. Transport Layer Security (TLS) protocol) should be established between the two entities. The functionality of CS is utterly trusted which will grant the incentives to the relay device at the end of successful service.

IV. PROTOCOL

This section describes the message flow of the protocol which is divided into multiple phases as illustrated in Figure 3.

1) Phase 1: When the data is available with the BLE sensor it creates an advertising HELLO packet and broadcasts to find a mobile as a relay node. This HELLO packet includes device MAC address, application ID (i.e., Universal Unique Identifier (UUID) 128bit) and time-stamp. This step is the initiation of non-connectable status to connectable status between the sensor and the relay mobile.

2) Phase 2: Upon receiving the HELLO message from the sensor, one or multiple mobile phones (i.e., which have the health-care application) in the closest proximity may respond. If the acceptances are received by multiple mobile phones during a predefined time-out, the best relay among them is elected based on the RSSI value. If the sensor does not receive any connection acceptance messages during that time-out, it will repeat Phase 1 again.

3) Phase 3: Once the connection between the sensor and the mobile is established, the sensor sends another connection request (CR) message to the central CS via the mobile relay. This message includes the CS address as plain text and the connection request to CS in an encrypted format. Advanced Encryption Standard-Counter with Cipher Block Chaining-Message Authentication Code (AES-CCM) is used for symmetric encryption whereas the keys are shared off-line between the sensor and CS. A time-stamp is also contained, in order to keep the freshness of the connection request. Upon receiving the connection request and sensor MAC address, the CS will authenticate the sensor and validate its legitimacy by decrypting the message with corresponding keys.

4) Phase 4: If the connection request has been received by a legitimate user, and the message freshness is encountered, the CS will continue the communication and send back the connection approving the message.

5) Phase 5: Once the connection is established, the BLE sensor sends the encrypted data packets to CS via the mobile relay. When the buffered data is over, the sensor will send a request to cease the connection. After receiving this, the CS will reply with an acknowledgment for ceasing the connection which ensures the successful reception of data. Finally, the BLE sensor will send the last Ack message and terminate the connection. This final two-way handshake is performed in order to guarantee the successful data transmission from the sensor to the cloud and to validate the genuine support from the mobile relay.

6) Phase 6: At the end of successful data forwarding process, the mobile user has to be rewarded for his service. The reward is automatically calculated by the application.
depending on the amount of data uploaded by the sensor. Based on the sensor acknowledgment about the completion of data forwarding, the CS will grant the computed reward to the mobile user. Due to the rewarding mechanism, the initial user authentication is important. Furthermore, we consider that the remote CS is an entirely trusted entity where the final reward is guaranteed to the mobile user for his service.

V. SIMULATION AND IMPLEMENTATION

A. Simulation

In order to justify the applicability and the reliability of this protocol in a real environment, we performed simulations in MATLAB. For that, we estimate the probability for a successful meeting with at least one mobile user who is willing to act as a relay. The BLE sensor needs to meet a mobile user before its data buffer overflows. The arrival of mobile users can be modeled as a Poisson process with the expected value \( \lambda \), the number of mobile users arrive during \( t \) time interval, where \( t = B/\lambda_0; B \) is the data buffer size (KB) of the BLE sensor and \( \lambda_0 \) is the data generating rate of the sensor (Kbps). The probability \( (P) \) of meeting at least one mobile user (i.e., the probability of successful data transmission) during \( t \) time period is expressed as eq. 1.

\[
P = 1 - e^{-\frac{\lambda B}{\lambda_0}}
\]  

(1)

In the first simulation, as shown in Figure 4, we illustrate how the probability distribution affects at different mobile arrival rates with a constant data generation rate of 0.2 Kbps (where \( \lambda_0 = 0.2 \) Kbps) at varying buffer size. We use CC2650 for the experimental setup as it is a very common platform for BLE application development. Since CC2650 \(^1\) has 128 KB Flash memory, the maximum possible buffer size will be limited to that value. When the buffer size is increased, the probability of meeting a mobile user will reach one with a less mobile arrival rate. With the maximum buffer size of 128 KB, the probability of success \( P \) will be one if three mobile users arrive per hour. When buffer size is changed to 100 KB, \( P \) will reach one for the arrival of five users per hour.

Figure 5 illustrates the results of the second experiment with the probability distribution for varying sensor data generation rates. The buffer size is kept constant as 100 KB. Accordingly, if the data generation rate of the sensor is low, the highest success rate can be achieved with a low arrival rate of mobile user. For instance when \( \lambda_0 = 0.2 \) Kbps, the probability of success \( P \) will reach one with arrival of five mobile users per hour.

In the third experiment, we try to find the proximity requirement to hold the assumption of holding an uninterrupted communication link between the BLE sensor and mobile relay. In order to maintain an uninterrupted communication link between the BLE sensor and mobile relay, the RSSI value should be above the minimum detectable level throughout the data transmission time. Consequently, the distance between

\[RSSI_D = RSSI_{D_0} + 10n\log\frac{D}{D_0} + X(0, \sigma)\]  

(2)

The two values \( RSSI_D \) and \( RSSI_{D_0} \) respectively represent the RSSI (dBm) at the distance \( D \) (m) and the reference distance \( D_0 \) (m). The path loss exponent \( (n) \) is dependent on many environmental factors and obstacles such as walls, objects, and people. The zero-mean Gaussian noise with variance \( \sigma^2 \) is represented by \( X(0, \sigma) \). According to the empirical analysis given in [14], for BLE communication, the path loss exponent varies from 1.35 to 3 for indoor and outdoor environments. As stated in [14], the reason for the lower path

\(^1\)http://www.ti.com/lit/ds/swr158b/swr158b.pdf
loss exponent could be that the indoor environments could be acting like a wave-guide, resulting in constructive interference.

**TABLE I: Path loss exponent \( n \) values for BLE communication in indoor and outdoor environments [14]**

<table>
<thead>
<tr>
<th>Environment</th>
<th>Category</th>
<th>( n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indoor</td>
<td>Line of sight</td>
<td>1.98</td>
</tr>
<tr>
<td></td>
<td>Non line of sight</td>
<td>1.35</td>
</tr>
<tr>
<td>Outdoor</td>
<td>0(^\circ)</td>
<td>2.89</td>
</tr>
<tr>
<td></td>
<td>45(^\circ)</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td>90(^\circ)</td>
<td>2.47</td>
</tr>
</tbody>
</table>

Considering the reference values observed from CC2650, \( RSSI_{D_0} \) value was measured as -72 dBm at 1m distance \( D_0 \) when the transmission power at the sensor is adjusted to the minimum of -21 dBm. In line with the manufacturer specifications of the CC2650 platform, the lowest identifiable RSSI is -100 dBm. We use these values along with the path loss model to estimate the distance between the BLE sensor and the mobile phone in order to maintain an uninterrupted service. As shown in Figure 6, the plots are drawn by varying the path loss exponent which increases with the growth of available obstacles between two communicating entities. Accordingly, for an uninterrupted data transfer in the indoor environments, the distance between the sensor and the mobile needs to be maintained less than 120 m during the entire time which the connection is working. In contrast, at the outdoor environments, the sensor and the mobile have to keep at most 9m proximity to maintain the connection. Thereby, we can conclude that our protocol will support higher user mobility in the indoor environments than in outdoors.

**B. Implementation**

In order to show the feasibility of this protocol, we have accomplished the prototype implementation on a test-bed with a BLE sensor, mobile, and cloud platform (Figure 7). The Internet access was achieved by the general university WiFi network (i.e., PanOULU network). The implementation is also demonstrated in [15]. Texas Instrument sensor tag CC2650 and Samsung Galaxy Note 8 were respectively used as the sensor and mobile hardware platforms. In accordance with the protocol, we slightly modified the BLE stack 2.2.1 on CC2650 using SmartRF Flash programmer 2. The mobile application was developed on Android 7.1.1 operating system using Android Studio 3.0 libraries. This mobile application scans in the background to discover devices and connects to the BLE sensors. This BLE sensor is then paired with the mobile automatically, using the passcode 0000. After paring, sensor initiates data uploading directly to the Cloud platform. The last part of the implementation was to deploy the cloud server on Google Firebase\(^2\) where the user of the mobile application can authenticate himself and the sensor can upload the sensed data to a JSON database. The user needs to login the application for authentication by CS and the collection of rewards. The application monitors the amount of data transferred from the sensor to the cloud and after the confirmation from the CS, the application automatically credits the reward to the user account. In order to keep the reward mechanism simple and profitable, for transferring every 1 KB of data, the user gets one point which can later be used for different purposes (e.g., convert into digital currency). Moreover, Firebase uses HTTPS connection over TLS for secure communications between the mobile and cloud server along with real-time database security.

**TABLE II: BLE configuration settings for CC2650**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Configured values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmission power</td>
<td>0 dB</td>
</tr>
<tr>
<td>Number of running services</td>
<td>6 services</td>
</tr>
<tr>
<td>Periodic event</td>
<td>1000 ms</td>
</tr>
<tr>
<td>Advertising interval</td>
<td>100 ms</td>
</tr>
<tr>
<td>Connection timeout</td>
<td>1000 ms</td>
</tr>
<tr>
<td>Broadcast delay</td>
<td>500 ms</td>
</tr>
<tr>
<td>Packet size</td>
<td>18 byte</td>
</tr>
</tbody>
</table>

Under normal network conditions, we measured the maximum response time of the mobile to discover a sensor, authenticate the sensor from the cloud, establish connection and discover one service, upload data (i.e., one packet includes 18 byte application data from sensor to cloud), and terminate

\(^2\)https://firebase.google.com/
the connection. Each experiment was performed ten times and the average was taken. In order to synchronize the clock time between sensor and mobile, computer clock time was sent to sensor when flashing the modified BLE stack. The application data inside one packet, which is 18 byte, is equivalent to one data reading. Figure 8 shows the distribution of timing values for those five operations. Accordingly, the total response time for the mobile was computed as 5.4 s.

![Figure 8: Time distribution for the protocol operations](image)

The duration for connection initializing (2.8 s) (i.e., including sensor discovery, authentication, connection establishment, and service discovery) and termination (0.2 s) will remain constant, irrespective of the amount of data transferred.

The power consumption for Android application running on the mobile was reported as 33 mAh. Therefore, with 3.85 v nominal voltage, the power consumption of the application on the mobile is 127.05 mW per hour. It is only 1% of battery. Thus energy consumption is quite low for our implementation. The distribution of power consumption of CC2650 sensor was observed from Monsoon power monitor tool, as depicted in Figure 9. The average power consumption values of the sensor during the connection establishment and data upload to the cloud were recorded as 239 mW and 262 mW. On an average, the sensor uploads the data to cloud server once every hour. We assume that every transaction takes approximately 1 minute and the sensor deactivates its BLE until the next transaction. With the Cell capacity of 300mAh, the sensor uses approximately 0.5% of battery for each transaction and it can upload data 200 times without recharging the sensor.

**VI. SECURITY CONSIDERATION**

In this section, we describe the possible attacks that can occur on the proposed protocol, the attack mitigation techniques, and the related security properties. Typically, BLE faces three types of security vulnerabilities naming passive eavesdropping, man-in-the-middle (MITM) attacks and identity tracking.

Passive eavesdropping can be a critical issue in these kinds of relay-based systems. Passive eavesdropper can learn about users’ privacy information (e.g. location and current health condition) from the sensed data. Moreover, the eavesdropper can obtain the network parameters of the CS to perform network-level attacks such as TCP reset attacks. In order to eliminate the risk of passive eavesdropping, we use AES-CCM for E2E data encryption. AES-CCM is an authenticated encryption algorithm designed to provide both authentication and confidentiality. In brief, since the cryptography of the protocol takes "authenticate-then-encrypt" approach, the risks of eavesdropping attacks are prevented. Furthermore, the establishment of secure E2E connections between the source (BLE sensor) and destination (CS) will eliminate the prerequisite for the trustworthiness of the mobile relay.

In the caliber of MITM attacks, two types of attack scenarios are possible with the proposed architecture. MITM attacks can be imposed either on the link between the BLE sensor and the mobile relay or on the link between the mobile relay and CS. In the first scenario, a malicious relay mobile can collect the sensor data, but it will not forward the collected traffic to the CS. As explained in figure 3, the sensor node will not delete the transmitted data until it receives an acknowledgment from CS. If the sensor does not receive an acknowledgment within a predefined timeout value, the sensor will select another relay to transmit the data and blacklist the malicious relay which is unable to complete the transmission. Since we use E2E encryption between the sensor node and the cloud server, it is not possible for the mobile relay to generate an acknowledgement. In the second case, a malicious relay can try to alter the data or augment bogus data to receive extra rewards from CS. Since we use E2E encryption between the sensor node and the cloud server, it is not possible for the relay node to generate new data or alter the data without the consent of the cloud server. If the CS notices such attempts by a relay node, it will blacklist and remove the corresponding mobile relay from the system. Furthermore, by including the time-stamp values to the messages generated by the sensor, the message freshness is guaranteed and the CS will identify if there is an attempt to impose replay attacks.

Identity tracking occurs when a malicious entity is able to associate the address of a BLE device with a specific user and then physically track that user based on the presence...
of the BLE device. Typically, such attacks are overcome by periodically changing the BLE device address.

As the CS is responsible for storing sensor data of different individuals, it becomes the most potential attack point. The attackers will try to act as legitimate sensors or relays. These kinds of impersonate attacks can be also eliminated by using a strong authentication process for the mobile-relays at the CS. The application running on the mobile relay should contain the login credentials for authentication which are acquired during the pre-registration phase with the CS. Furthermore, the pre-shared keys between the sensor and the CS will provide an implicit authentication. Moreover, we establish a secure tunnel (i.e TLS) between the mobile-relay and the CS, which is mandatory before the sensor data transmission.

VII. CONCLUSIONS

Under the proliferation of low-power IoT technologies and their applications, universal accessibility is a noteworthy attribute that needs to be possessed by the sensors. Throughout this paper, we have addressed how to provide such an ubiquity for the low-power BLE sensors with the help of mobile relay nodes. We have explained the protocol in detail taking into account a particular AAL application. We have also provided the theoretical evaluation of the protocol and the implementation details along with the security considerations. According to our experimental evaluations, the protocol will successfully work even with the arrival of three mobile users per an hour for the proximity of 120 m for indoor and 9 m for outdoor environments. Finally, the prototype implemented verified the technical readiness of the proposed solution. Experiments revealed that Texas Instrument sensor tag CC2650 can upload data for 200 times without recharging the sensor. Moreover, mobile phone uses only 1% battery power for one hour of operation. In the future, we plan to address the identified challenges and limitations, and implement and evaluate our proposed solution as a commercial product.

With the advent of mobile networking technologies, the users will speculate that every device is not only a simple user device but also a networking device that can act as a relay or an access point. This idea is materialized with the recently introduced edge computing paradigms. The research efforts beyond 5G need to enable user mobility and ultra-densification of networking devices. We intend to extend the very same idea of exploiting mobile relay nodes with drone technologies.

ACKNOWLEDGEMENT

This work is supported by three projects including 6Genesis Flagship (grant 318927), “SECUREConnect” and “Industrial Edge”, which are funded by Academy of Finland and TEKES, Finland. A. Gurtov was supported by the Center for Industrial Information Technology (CENIIT).

REFERENCES