Communication and security trade-offs for battery-powered devices: a case study on wearable medical sensor systems

Winderickx, J.; Bellier, P.; Duflot, P.; Mentens, N.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 13, 2021, accepted April 6, 2021, date of publication April 27, 2021, date of current version May 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3075980

Communication and Security Trade-Offs for Battery-Powered Devices: A Case Study on Wearable Medical Sensor Systems

JORI WINDERICKX[1], PIERRE BELLIER[2], PATRICK DUFLOT[3], AND NELE MENTENS[1,4], (Senior Member, IEEE)

[1]IMEC-COSIC and Embedded Systems and Security - KU Leuven, 3000 Leuven, Belgium
[2]Montéfiore, Department of Electrical Engineering and Computer Science, ULiège, 4000 Liège, Belgium
[3]Centre Hospitalier Universitaire de Liège, 4000 Liège, Belgium
[4]LIACS - Leiden University, 2333 CA Leiden, The Netherlands

Corresponding author: Jori Winderickx (jori.winderickx@kuleuven.be)

This work was supported in part by the WearIT4Health Project through the Interreg V-A Euregio Meuse-Rhine, in part by the European Union and the European Regional Development Fund, in part by the Province of Limburg - Belgium, and in part by the CyberSecurity Research Flanders under Grant VR20192203.

ABSTRACT In order to reduce the workload of hospital staff and to provide better services to hospitalized patients, attempts are made to integrate patient monitoring systems directly into hospital networks. Monitoring systems must respond to more and more technological challenges. They are ideally portable and wireless, to free the patient from the hospital bed. At the same time, to enable better patient follow-up, a large amount of information needs to be transmitted and processed in real time. Challenges in the design of such systems include energy-efficient processing and communication, and guaranteeing the security of the measured data. This paper describes a wearable sensor system, integrated into a hospital network, that supports high data rates generated by multiple sensors. With a strongly motivated focus on end-to-end security, we explore trade-offs with respect to security schemes and implementations, and wireless network protocols. The results show that the energy efficiency of the resulting system is comparable to existing systems that support far less sensor data and that compromise on end-to-end security by offloading security operations to a delegation server. To our knowledge, this is the first work that explores the impact of the security scheme and the wireless network protocol on the energy consumption of a wearable device, while providing true end-to-end security.

INDEX TERMS Data security, energy consumption, Internet of Things, wireless communication.

I. INTRODUCTION

The hospital of the future consists of a multitude of interconnected devices that monitor the health of the patients and send medical sensor data to a server that interfaces with the Electronic Medical Record (EMR). The EMR refers to the comprehensive medical records of an individual that is accessible in electronic form. In order to measure vital sign parameters, wearable devices are used. Since these devices are battery-powered, it is important to minimize their energy consumption. Both the processing and the communication of data contribute to the energy consumption. That means that the communication data rate of the measured vital sign parameters as well as the wireless communication protocol have an influence on the lifetime of the wearable device's battery.

Besides energy efficiency, security is an important issue in wireless networks. The General Data Protection Regulation (GPDR) [1], strengthened by national law, enforces data protection and privacy for all individuals at the European level. As a consequence, sensor data measured on hospitalized patients should be protected against potential adversaries that retrieve or manipulate the data. It is clear that security mechanisms should be installed during the communication between the wearable device and the hospital server. But security is at least as important during the process in which hospital

The associate editor coordinating the review of this manuscript and approving it for publication was Wen Chen.

staff associate a wearable device to a patient. To achieve this, the device needs to be anonymously and wirelessly linked to the patient. Both during communication and association, end-to-end security needs to be provided, i.e. third parties should be prevented from accessing data while these data are being transferred from one end system to another. In our case, these end systems are the wearable device and the hospital server.

The system under consideration in this paper, consists of a wireless waterproof wearable device, communicating with a hospital server. The features of the monitoring system were determined based on (1) co-creation sessions with future users - healthcare professionals and potential patients, (2) market analysis and (3) the collaboration with three hospitals. The wearable device continuously monitors heart rate, blood pressure variation, breathing rate, oxygen saturation, skin temperature and human activity (intensity and posture). Raw data used for the estimation of these parameters are the electrocardiogram (ECG), the photoplethysmogram in three wavelengths (PPG), bio impedance (BioZ), the temperature (T) and the 3-axes accelerometers (ACC). These data are transmitted wirelessly from the wearable device on the patient, hereafter named patch, to the local hospital server. The patch is intended for use on nursing ward patients. It is designed to allow mobility for adult patients ($>=$ 18 years old) to provide physiological information. However, it is neither intended for use on critical care patients nor for diagnosis. The local server will process the measured data and use it for two applications: monitoring and reporting to the EMR. Moreover, the local server can also notify healthcare professionals when physiological data fall outside specified ranges of selected parameters. The data measured by the wearable device are intended for use by healthcare professionals as an aid to monitor patients.

Our contributions can be summarized as follows:

- We implement end-to-end security in a wearable health monitoring system, intended for real-life use in a hospital, with a multitude of measured sensor data.
- We explore the impact on the energy consumption of different security schemes and implementations, as well as different wireless network protocols. As such, our work serves as a guideline for researchers and practitioners setting up a wearable medical sensor system or any battery-powered sensor system that needs to communicate a relatively large amount of data while providing end-to-end security.
- We present a proof-of-concept implementation of the resulting system and the corresponding measurement results.

First, related work is analyzed in Section II. Then, the system architecture is described in Section III. For the selection of the most suitable RF network for our system, low-power RF networks are compared in Section IV. Next, to ensure the protection of the measured data, the security requirements are analyzed and used to create a security architecture, described in Section V. Then, after the theoretical analysis in Sections IV and V, a practical exploration is presented in Section VI. It implements and compares the different security solutions using the most suitable RF protocol. Finally, a conclusion is made in Section VII.

II. RELATED WORK

Two popular healthcare projects in research are CodeBlue [2] and MEDiSN [3], but many more projects and platforms have been devised, as analyzed by Javdani and Kashanian [4]. CodeBlue [2], proposed by Malan *et al.* [2] in 2004, is an ad hoc sensor network infrastructure for emergency medical care. The authors used MICA2 motes to monitor the pulse oximetry of patients. MEDiSN [3], created by Ko *et al.* [3] in 2010, provides medical emergency detection in sensor networks. The authors used the Sentilla Tmote Mini platform for two use cases: ECG monitoring, and pulse oximetry with LCD screen. However, in more recent works, researchers also began looking into optimizations. For example, Samie *et al.* [5] used data compression in a wearable ECG monitoring platform to reduce the overall energy consumption and it resulted in a device that could theoretically monitor the ECG of a patient for 10 days using a 400 mAh battery.

As stated, the number of healthcare platforms in research is increasing, as interest in integrating portable devices continues to grow. Nevertheless, the security issue is still a concern. In this work, we analyze the implementation and integration of a wearable healthcare platform in a hospital while providing true end-to-end security to protect the patient's personal data.

End-to-end security is important to take into account when dealing with sensitive data. Without end-to-end security, intermediate devices like gateways can compromise the confidentiality of the transmitted data. This was already specified by the earliest works about patient monitoring. For example, both the CodeBlue [2] and MEDiSN [3] projects expressed the importance of security by referring to privacy issues and national regulations like the 1996 Health Insurance Portability and Accountability Act (HIPAA), but neither provided end-to-end security. In the first published paper of the CodeBlue [2] project, security was put as future work and it was revisited by Kambourakis *et al.* [6] in 2007. It was, however, to our knowledge, never fully implemented with end-to-end security. In the MEDiSN [3] project, the authors did implement secured communication between the entities in the network but they did not provide end-to-end security.

Security in constrained environments has been a hot topic for many years. For example, the concept of a delegation server or trusted third party to offload the authentication and authorization computations from constrained devices has been researched extensively by, e.g., Hummen *et al.* [7], Raza *et al.* [8], and Kerberos [9]. The main motivation for using a delegation server is that the impact of the authentication and authorization protocols on the energy consumption of constrained devices is too large. Furthermore, it is often assumed that the constrained device has a preconfigured secure communication channel with the

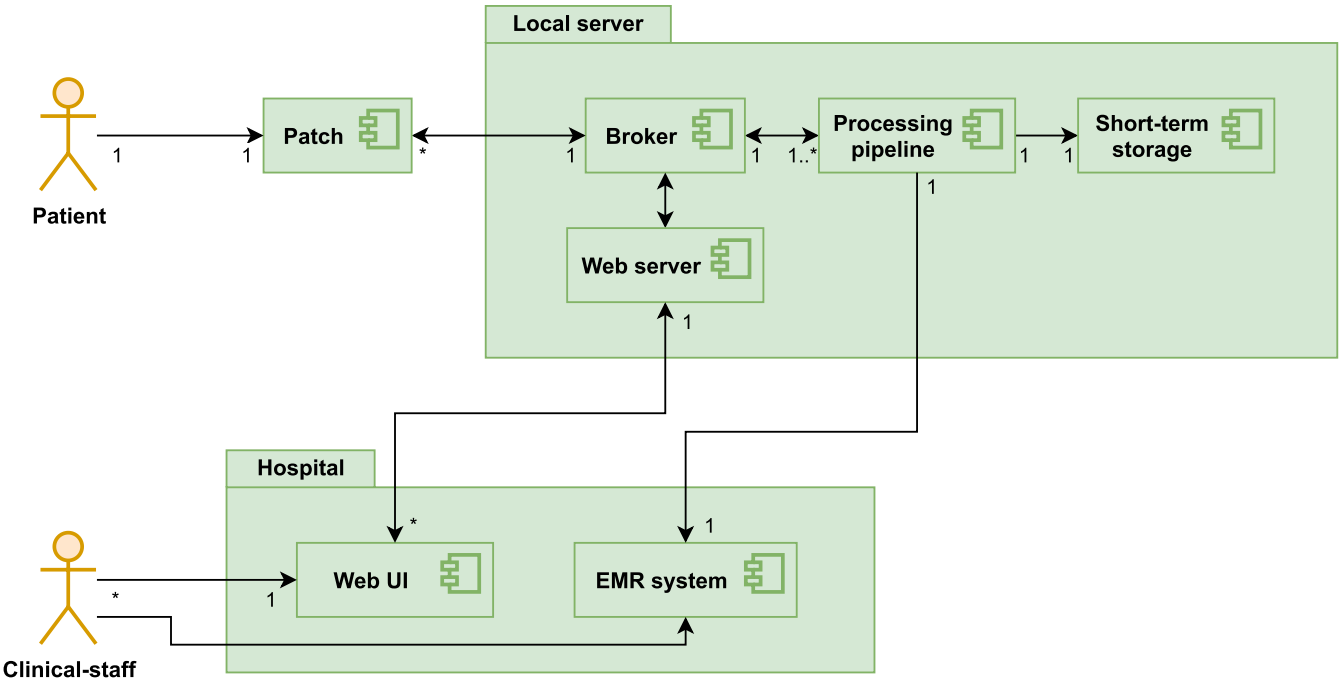

FIGURE 1. Overall system architecture (1: one entity; *: multiple entities).

delegation server. The delegation server can setup connections with remote parties and it can then pass along the required session information to the constrained device. After this procedure, the constrained device and remote server can securely communicate. By offloading the authentication and authorization, the constrained device does no longer need to utilize the computationally expensive algorithms that are typically used for these purposes like the asymmetric-key cryptography of RSA or Elliptic Curve Cryptography (ECC). For example, the paper by Moosavi *et al.* [10] showed that it is also possible to use the fog to provide a distributed set of delegation servers to offload the authentication/authorization. The advantage is that a Denial-of-Serice (DOS) attack is less likely because the delegation server is no longer centralized.

The issue with the delegation server is that it is a target for attackers to compromise one or more constrained devices. In this work, we avoid the use of a delegation server to narrow the range of targets available to the attacker. We analyze and limit the impact on the constrained device by leveraging the newest technologies and techniques with regards to communication security. This work extends the work-in-progress results presented in [11].

III. SYSTEM MODEL

The system architecture, presented in Figure 1, consist of three entities: the patch, the local server and the hospital infrastructure. In this section, first, the patch is described. Then, the functionalities of the local server are given. Finally, the workflow of the system is detailed to explain how the patch is used.

The local server also interconnects with the hospital infrastructure for monitoring, managing and reporting purposes. Clinical staff can access the patient's data through a web-based user interface (Web UI) or through the EMR system. We do not elaborate further on the hospital infrastructure because it depends on specific choices made by the IT department of the hospital. Nevertheless, the system described in this paper was validated on three different hospital networks and can therefore be considered to be broadly usable. The usage and application of the system is the same for each hospital.

A. PATCH

The patch is composed of three different parts: (1) two textile electrodes, attached to the housing via snap buttons, (2) an electronic board and (3) a battery. Both the electronic board and the battery are contained in the housing. The electrodes have to be changed for each patient as they could lead to cross contamination if used on multiple patients. The electronic board contains a microcontroller, a network interface and all the required on-board sensors, namely ECG, PPG, BioZ, T and ACC, as introduced in Section I. The battery is easily expendable when the patient is wearing the patch in order to facilitate its continuous use.

The on-board sensors produce about 3.3 kB of raw data every second. In order to reduce the data overhead and to lower the number of times the device should awake from sleep (wake-up overhead), 33 kB of data is sent to the server every 10 seconds.

B. LOCAL SERVER

The local server runs three applications: a message broker, a processing pipeline and a web server. The message broker

is used by the patch to publish its raw data, by the processing pipeline to convert the raw data into vital parameters, and by the web server to display these vital parameters. Furthermore, commands to and from the patch and the web server are also communicated via the message broker. Regarding the conversion of raw data to vital parameters, a 30-second time window is used. At the start of each window, a copy of the raw data is saved into short-term storage. It can later be used for analysis. Then, within each 30 seconds, raw data is first preprocessed to eliminate noise and then processed to extract the vital parameters of the patient. Furthermore, the processing pipeline is responsible for assessing the patient's state. It estimates the early warning score (EWS) for each patient and triggers notifications to the clinical staff if abnormal conditions are detected in the patient's vital parameters. The EWS is estimated every 10 s. Depending on the instructions of the clinical staff, the pipeline can also update the patient's electronic medical record (EMR) with specific data points.

C. WORKFLOW

The workflow describes how the patch is used within the involved hospitals, as depicted in Figure 2. At first, every patch needs to be configured. This includes setting up the network *configuration* and security parameters for the connection to the hospital network, and registering the device into the system database. The patch configuration as well as future *updates* are done by the IT department of the hospital using a programmer with a wired connection to a PC.

After configuration, the patch is delivered to the clinical staff. Note that any user that wants access to the system first needs to go through an *authentication* step. This influences which actions he/she can perform with the system and how the data are filtered on the display.

During the *association* step, a given patch is associated with a patient. Each time a patch is assigned to a patient, the nurse scans the patient's EMR identifier on the patient's wristband and the patch serial number on the patch's label using an off-the-shell barcode reader. Based on these two identifiers, a unique association identifier (UID) is generated by the local server. This UID is wirelessly sent from the local server to the patch each time it boots up for the duration of the association. The patch will append the UID to each packet it will subsequently emit. This design has two main advantages:

- It does not expose the patient's EMR identifier inside the message broker.
- It avoids any session mismatch in the data processing pipe-lines and allows the pipelines to cache UIDs and related metadata for efficiency.

The nurse will now *install* the patch on the patient's chest. First, he/she checks the battery level of the patch, second, he/she attaches the patch with an adhesive on the left chest at the level of the heart, and third, he/she attaches both electrodes to the patch via snap buttons and to the chest using the adhesive part of the electrodes. All vital parameters of the patient except the blood pressure are now displayed continuously on the user interface.

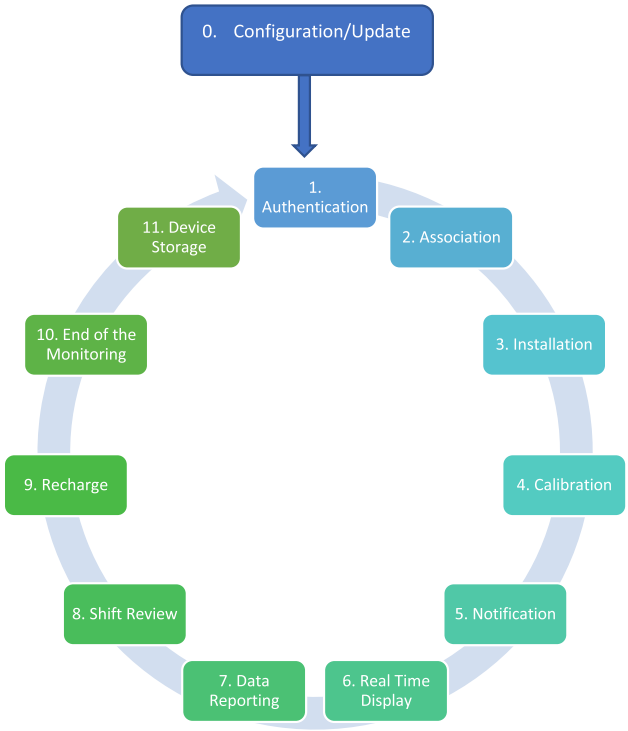

FIGURE 2. Usage workflow.

It is then required to *calibrate* the patch for the blood pressure measurement. A reference blood pressure measurement is taken by the nurse and sent to the blood pressure measurement algorithm via the user interface.

All vital parameters of the patient are now monitored in (near) real time and the patient's early warning score (EWS) is continuously evaluated. The *real-time display* can be accessed on any web browser or mobile device on the hospital's network.

If a deviation is detected in the vital parameters, a *notification* is sent to the clinical staff. The clinical staff can comment and/or acknowledge this notification. Furthermore, all actions are tracked by the local server. This offers an audit log and a communication channel between clinical staff. With the right permissions, a user can adjust the thresholds of the notification detection for a specific patient.

The clinical staff can report specific data points to the EMR. The *data reporting* mimics the work they were already doing with discrete measuring devices. However, compared to the discrete measurement approach, the process is faster via our interface, and, it is more detailed because the clinical staff is able to select more data points.

At the end of a staff member's shift, he/she can *review* the notifications and the evolution of vital parameters. He/she can also review and export a PDF report to the EMR of any ECG sample.

For the entire duration of the session, the battery charge level will be visible on the user interface. If the battery is almost empty, the clinical staff has to replace the battery. The system will notify the clinical staff whenever the battery level

drops below 20 and 5 percent, such that a timely *recharge* can be done.

When the patient no longer needs to be monitored, the adhesive is thrown away and the patch is put back in the stock after cleaning. This refers to the *end of monitoring* and the *device storage* steps in Figure 2. The clinical staff instructs the system that the patch is now available to be used for another patient. Note that this will happen automatically if no data are received for a long period of time.

IV. WIRELESS COMMUNICATION TRADE-OFFS

The requirements of the patch can be summarized as follows. It should be wirelessly connected to the local server and comfortable for the patient to wear. That means that it is necessary to reduce the size and the weight of the patch, which is directly related to the battery size. Therefore, the way of processing and transmitting data should be power efficient. It should be possible to install a patch on each patient in the hospital, thus, the network should be designed to handle a large number of patches. As patients can move inside the hospital, the network should cover the entire hospital. This sections makes a theoretical comparison of different network topologies and wireless communication protocols in order to select the most suitable solution for our application scenario.

A. NETWORK TOPOLOGY

There are three main types of network topologies:

- *Ad-hoc*: one device can communicate directly with every other device within its range. This topology is adapted to a relatively small number of devices. As the link is direct between two devices, the range is limited by the protocol and the power of the signal transmission. The advantage is that the implementation of this type of network is relatively simple as it does not require protocols for e.g. routing.
- *Mesh*: the mesh is an extension of the ad-hoc topology. Some devices can act as router and relay data between two nodes. These routers extend the coverage at the expense of an increased power consumption for transferring data. Furthermore, all devices share the same network, thus, they can easily be integrated into the network.
- *Star*: a central access point communicates directly with all peripherals in its range. There can be more than one access point linked together to extend the coverage to the entire area. Generally, theses access points are not power constrained because they are plugged into the electrical grid. This kind of network can be seen as a mesh with only access points as routing nodes. Thus, the peripherals in the network use less power than in the mesh network, but, the network requires an extensive network infrastructure to cover the entire area.

In the scenario that we consider in this paper, there is only one local server collecting all the data. Moreover, taking into account the typical size of a hospital, an ad-hoc network topology cannot ensure connectivity for all patch devices. The mesh topology is also less suitable because of its increased power consumption and the fact that the coverage relies on the number of devices and their distribution. Therefore, the star topology is the most convenient for our application at the expense of the cost of fixed access points in the hospital.

B. PROTOCOLS

There are a lot of existing wireless communication protocols, e.g. Bluetooth Low Energy (BLE), Wi-Fi, ZigBee, LoRaWAN and Sigfox. These protocols can be divided into different types of networks: Wireless Personal Area Networks (WPAN), Wireless Local Area Networks (WLAN) and Low-Power Wide-Area Networks (LPWAN). The classification is typically based on the range of the protocol. Furthermore, each network protocol has its own advantages and disadvantages. An overview of the network protocol specifications is given in Table 1.

TABLE 1. Comparison of RF protocols suitable for indoor application.

Features	BLE	ZigBee	Wi-Fi
IEEE specification	802.15.1	802.15.4	802.11b/g/n
Frequency band	2.4 GHz	868/915 MHz 2.4 GHz	2.4 GHz
Data rate ($Mbps$)	1-3	0.250	54-150
Nominal range (m)	55-100	10-60	100-180
Max active devices	8	65000	2007
Security	AES-CCM	AES-CCM	WPA2
Power consumption			
TX ($\mu W/kb$)	3.7-88.8	108.2-435.6	3.8-465
RX ($\mu W/kb$)	3.6-72.6	129.3-369.6	3.1-120.8
Idle current (μA)	2-200	0.7-2.5	690
Max payload (B)	339	102	1500
Max overhead (B)	158/8	31	58
Efficiency	94.46%	76.67%	96.28%

The WPAN protocols like BLE and ZigBee have low throughput (250 kbps - 1 Mbps) and operate over short ranges indoor (10 - 100 m), but, they have the advantage to be low-power. In contrast, a WLAN protocol like Wi-Fi is well adapted to send large amounts of data (54 - 150 Mbps) over slightly larger distances (100 - 200 m). LoRaWAN and Sigfox can transmit data over very long ranges of a few kilometers. However, they are not optimized for indoor use and feature typically very low throughput (300 bps - 50 kbps) and can handle only a limited amount of data. Furthermore, some LPWAN networks like Sigfox are hosted by an operator which results in a paid subscription.

LoRaWAN and SigFox are clearly not applicable for the application studied in the paper at hand. We selected the other three protocols, Bluetooth, ZigBee and Wi-Fi, and made a theoretical comparison through a set of the most recent chips available on the market: BL652-SA, RN4020, BT800 and LM931 as Bluetooth chips, JN5168, MGM11, Xbee ZB SMT and EM351 as ZigBee chips, and LM821-0463, WGM160, CX53111 and CC3120 as Wi-Fi chips.

We made an estimate of the efficiency of the 3 protocols to send the amount of data required by our system architecture (33 kbytes). The efficiency is defined by the payload

size (*Ndata*), the number of packets (*Npackets*) and the data overhead per packet (*Noverhead*) [12], see Equation 1. The number of packets is defined by the fragmentation of our data that is required in the respective RF protocol. Then, for each packet sent, the header overhead needs to be added.

$$\text{Efficiency} = \frac{N\ data}{N\ data + (Npackets * Noverhead)} \quad (1)$$

As can be seen in the last line of Table 1, it turns out that ZigBee is less efficient in our scenario, because, we need to send a large amount of data. It takes more time to transmit our data via the ZigBee protocol, which results in less time in low-power sleep mode and consequently in more energy consumption. Regarding the network size, ZigBee is the one that can deal with the largest number of nodes. However, bandwidth needs to be shared between all patches which greatly reduces the number of nodes ZigBee can handle, since each patch requires a high data rate.

On the other hand, BLE and Wi-Fi are equivalent with respect to the energy consumption per byte, but Wi-Fi has a better coverage and can deal with a larger number of devices per cell. Additionally, Wi-Fi has three advantages over BLE for our setup: (1) less gateways need to be deployed because the range is larger, (2) a Wi-Fi infrastructure is often already present in hospitals, and (3) the patch could also easily be used for monitoring patients at home, as Wi-Fi is already widespread in almost every house.

It should be mentioned that these protocols could be extended to more complex structures that can improve the number of devices that can connect to the network at the expense of more access points. Nevertheless, we conclude this section by selecting Wi-Fi as the most suitable protocol for our application, given that 33 kbytes of data need to be sent every 10 seconds, targeting a low energy consumption and the ability to handle many patches.

V. SECURITY TRADE-OFFS

The objective is to secure the entities and data flows as presented in Figure 1. This section concentrates on two entities, the patch and the local server, and the data flow between them. First, the security requirements are compiled using the STRIDE threat modelling technique. Next, three techniques are evaluated to provide an end-to-end secured communication channel. Finally, the online and offline security requirements in terms of access control are discussed. The theoretical analysis made in this section, will be brought to practice in Section VI, which implements and compares the different security solutions.

A. SECURITY REQUIREMENTS

At the start of developing the security architecture, the relevant threats to the system must be identified. We use the STRIDE threat model [13] and the STRIDE-per-interaction technique to find the threats. An abbreviated compilation of the threats is given below, according to the six categories: Spoofing, Tampering, Repudiation, Information disclosure, Denial-of-service and Elevation of privilege:

- *Spoofing:* refers to a scenario in which data are disguised as coming from a known, trusted source, while they are coming from an unknown, untrusted source. In our setting, the system can contain multiple patches but only one local server. The patch measures sensitive data, which is communicated to and analyzed by the local server. In order to prevent spoofing, mutual authentication is required between the patch and the local server. This can be achieved through *entity authentication* and *data origin authentication* in order to validate the entities and the dataflow, respectively.
- *Tampering:* is the act of altering or damaging data while being communicated from one entity to the other. In our system, the data collected by the local server is used for operations like monitoring and assessing the health of a patient. It is, therefore, important to prevent tampering by ensuring the *data integrity* of the communication flow.
- *Repudiation:* is the rejection or the refusal of acknowledgement of a previously made agreement. Preventing repudiation typically involves the use of logging to prevent interactions to be denied. In our scenario, repudiation protection can be enabled on the server by using logs of interactions. It is, however, difficult to implement on the patch, since the patch is a constrained device in which the storage is already filled with application code and measured data. Nevertheless, the local server is the central device that is involved in all interactions, which means that logging on the local server prevents the repudiation of all interactions.
- *Information Disclosure:* refers to an attacker gaining unauthorized access to valuable information. Since the measured vital sign parameters are personal in the considered patient monitoring system, *data confidentiality* is imposed by law [1]. Furthermore, the scope of the attack should be limited if a device is compromised. A compromised device should not affect previous or future patients. This can be achieved by providing Perfect Forward Secrecy (PFS). PFS ensures that the previous sessions are not compromised when the current session is compromised. Confidentiality protection is, therefore, required for the data transmitted and stored by the patch as well as the local server.
- *Denial-of-Service:* attacks intend to make a network device unavailable to its intended users by, e.g., flooding the network or the local server with traffic. The considered monitoring system operates in the internal secured network of the hospitals where Denial-of-service attacks are highly unlikely. However, disconnections or weak reception can render the patch unable to send its data. It is, therefore, advised to reserve storage on the patch to store data temporarily. As an additional security measure, the local server could be duplicated to provide redundancy as it could be targeted as a single point of

failure. We did not opt for this security measure because of the locality of the implementation, as discussed previously.
- *Elevation of Privilege:* concerns the situation in which a lower-privilege user accesses functionality or content reserved for higher-privilege users. In order to protect the system from this threat, *authorization* should be implemented. This is important because only authorized devices should be able to upload data for specific patients, and only authorized personnel should have access to the data of specific patients.

In summary, based on the STRIDE approach, the five desired cryptographic properties that are identified in our system are *entity authentication*, *data origin authentication*, *confidentiality*, *data integrity* and *authorization*.

B. END-TO-END SECURED COMMUNICATION

In most end-to-end secured tunnels, two different types of protocols are used: key establishment and secure communication. Key establishment is typically a handshake which is based on either symmetric-key cryptography, public-key cryptography or both. It can provide the required entity authentication. During the handshake, the authenticity of both communicating identities is confirmed while a session key is derived. The session key can then be used to secure the actual communication. This is typically done using a symmetric-key encryption algorithm like AES [14]. AES is a block cipher which can be used in different modes of operation to achieve a specific set of security requirements. For example, CTR (Counter) mode and CBC (Cipher Block Chaining) mode provide confidentiality, and GCM (Galois/Counter Mode) and CCM (Counter with CBC-MAC Mode) are authenticated modes of encryption that ensure both confidentiality and authentication. In the following paragraphs, we discuss three methods for key establishment that enable end-to-end security. In our proof-of-concept implementation in Section VI, we compare different cipher suites based on these three approaches.

1) SYMMETRIC-KEY APPROACH

In constrained environments, the Pre-Shared Key (PSK) technique for authentication and key establishment is often used for its efficiency. It is based on symmetric-key cryptography, where both communicating entities share a key, and this shared key is used to generate a master key. The key is shared through an out-of-band communication channel, e.g. through configuration in advance. The session keys that are used to encrypt the data are derived from the master key. The authentication is based on the implicit use of this master key. If an entity cannot derive a session key, it is assumed that it does not know the master key, and thus, the entity is unable to authenticate itself. In order to prevent attackers from performing a brute-force attack, it is important to regularly update the key and to use key lengths that are considered to be long enough for the envisioned protection level [15].

2) PUBLIC-KEY APPROACH

A technique based on public-key cryptography that is often used during the handshake, is the combination of the Diffie-Hellman (DH) protocol [16] for key exchange and the RSA [17] algorithm or the Elliptic Curve Digital Signature Algorithm (ECDSA) for entity authentication. Public-key cryptography uses two separate keys for each entity: a private and a public key. For entity authentication, the private key is used to sign data and the public key is used to verify the signature. If the verification succeeds, the data are authenticated to be originating from the corresponding entity. Moreover, the public key is often packed into a certificate which contains the public key, the corresponding entity's information, and a signature. This signature is generated by a trusted third party such that it can be validated by all entities.

3) COMBINED APPROACH

In a more recently proposed technique, a combination of PSK and Diffie-Hellman is devised [18]. It uses DH key exchange authenticated with a PSK. This way, the authentication is more efficient than the approach based on public-key cryptography only. By using DH, perfect forward secrecy can be achieved.

C. ACCESS CONTROL

Access control is important for the offline and online security of entities. We denote offline security as the secure commissioning of the devices, and online security as access authorization.

1) LOCAL SERVER

The IT department of the hospitals will be responsible for hosting and managing the local server. It can be continuously monitored and no unauthorized physical and network access should be possible. At minimum, authorization measures like login credentials should be used to limit access to the local server.

2) PATCH

Access to the patch cannot be controlled by the IT department since it is used on patients. Attackers, thus, may have physical access. Steps should be taken to secure all available interfaces of the patch, e.g. using authenticated firmware and password protection.

VI. PROOF-OF-CONCEPT IMPLEMENTATION + MEASUREMENT RESULTS

The protocols that are used in the proof-of-concept implementation are presented and mapped to the OSI model in Figure 3. First, the Wi-Fi network was chosen for its efficiency and to facilitate interoperability and integration in hospitals. Secondly, we opted for using TCP to enable reliable communication over the IP network. Next, MQTT was used to provide a lightweight and proven messaging transport. It is also a standardized protocol of the International Organization

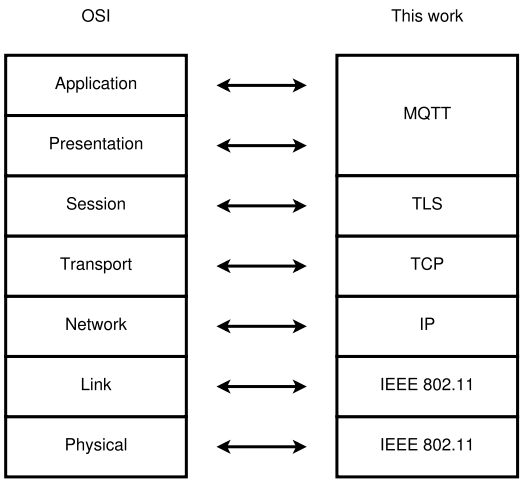

FIGURE 3. Selected protocols for the communication channel.

for Standardization (ISO) [19]. Finally, the TLS protocol is used to secure the MQTT communication. It is a protocol designed by the Internet Engineering Task Force (IETF) [20].

The patch is implemented using a custom platform. This platform features the MSP432P4011 [21] as application MCU, CC3120 [22] as network interface, and the sensor ICs required to measure the vital parameters described earlier. The MSP432P4011 is a SimpleLink Ultra-Low-Power 32-Bit Arm Cortex-M4F MCU with Precision ADC, 2MB Flash and 256KB RAM. The CC3120 is a SimpleLink Wi-Fi Network Processor. It features an ARM Cortex-M3 MCU that can completely offload the Wi-Fi and Internet Protocols from the application MCU.

The local server is Linux based. The software on the server uses the container technology of Docker [23]. The server runs RabbitMQ [24] as message broker and TimescaleDB as database. This combination has been validated to cope with the amount of data generated by 500 monitored patients using a single machine with 8GB of RAM and 4 CPU cores. The WebServer is implemented in Java using the Spring framework. The WebServer serves an Angular application to manage, monitor and inspect near real-time and historical data. Finally, the streaming engine is a combination of Java code for the aggregation and routing logic, and MATLAB/C++ code for the actual signal processing algorithms.

A. SECURITY OVERHEAD

To achieve the five required cryptographic properties, identified in Section V-A, we selected four cipher suites for our analysis:

1) TLS_PSK_WITH_AES_128_GCM_SHA256
2) TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256
3) TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
4) TLS_DHE_RSA_WITH_AES_128_GCM_SHA256

For secure communication, all cipher suites use AES, either in GCM or in CBC mode, in combination with the SHA256 hash function. For key establishment, the first cipher suite uses only symmetric-key cryptography based on the PSK approach (as explained in Sect. V-B1). It does not provide Perfect Forward Secrecy (PFS); it is primarily added for reference. The second cipher suite uses a combination of symmetric-key and public-key cryptography for the key handshake (as explained in Sect.V-B3). It is based on Elliptic Curve Diffie Hellman Exchange (ECDHE) in combination with PSK. The third and fourth cipher suite correspond to the public-key approach explained in Sect. V-B2. The third cipher suite uses ECDHE in combination with RSA, while the fourth cipher suite uses Diffie Hellman Exchange (DHE) in combination with RSA. The second, third and fourth cipher suite provide all the required properties of an end-to-end secured channel including Perfect Forward Secrecy (PFS). The authorization aspect is implemented by using user/password login credentials in the MQTT protocol. After successfully establishing the secured channel, the patch needs to provide these credentials to gain access to the MQTT broker.

The selected key sizes are a 2048-bit RSA key pair, a 256-bit ECC key pair, a 128-bit session key and a 256-bit hash. These key sizes match the basic recommendations for new systems, reported by for example the European Union Agency for Network and Information Security (ENISA) and ECRYPT-CSA in the Algorithms, key size and parameters report [15], [25]. An overview of multiple standardization agencies can be found on the website of BlueKrypt [26]. Our key sizes are estimated to be safe to use until 2028.

The TLS protocol is implemented in two ways. For the first and second cipher suite, we have used the mbed TLS library (v2.11.0) [27]. Moreover, the SECP256K1 curve was used for the ECDHE implementation of this library. The third and fourth cipher suite, using public-key cryptography for both the authentication and key establishment, are hard to implement on the MCU because of storage and memory constraints. For this reason, we opted for the secure socket feature of our CC3120 network processor. It provides a public-key based TLS protocol implementation. In this scenario, the TLS protocol is offloaded from the MCU to the network processor. Moreover, the network processor uses a hardware accelerator for the RSA and the AES algorithm. The ECC curve used by the CC3120 chip is SECP256R1.

The performance of the key handshake depends on the used algorithms and hardware, presented in Figure 4. The first cipher suite based on the PSK approach takes the least amount of time, about 60 ms, to establish a session. The other cipher suites involve public-key algorithms that are more computationally expensive. The MCU is a low-power processor that is limited in performance. This is notable in the execution of the ECDHE_PSK handshake. It takes about 2.6 seconds to establish a session. The secure socket feature of the network processor is much faster than the MCU for these types of calculations since it can perform the handshake under 0.5 seconds for ECDHE_RSA and DHE_RSA. The elliptic curve discrete logarithm based version (ECDHE_RSA) is about twice as fast as the discrete logarithm based version (DHE_RSA). This may be because of the

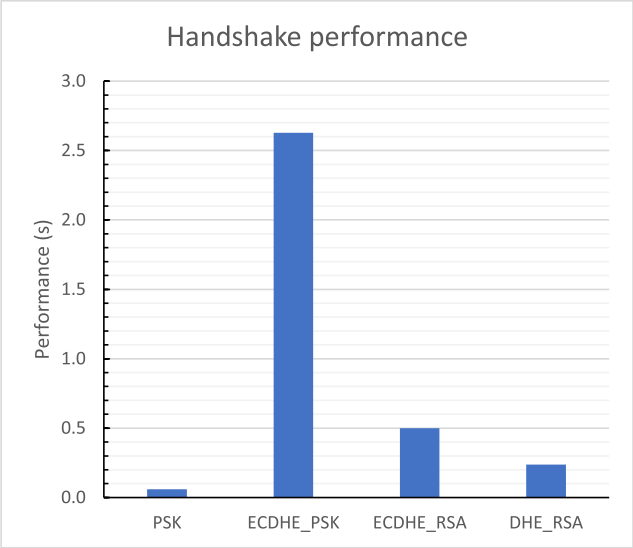

FIGURE 4. Performance evaluation for each of the handshake configurations, where PSK, ECDHE_PSK, ECDHE_RSA and DHE_RSA stand for the handshake protocols in the first, the second, the third and the fourth cipher suite introduced in this section. The first and the second approach are implemented on the MCU, while the third and the fourth approach are implemented using the secure socket feature of the network processor.

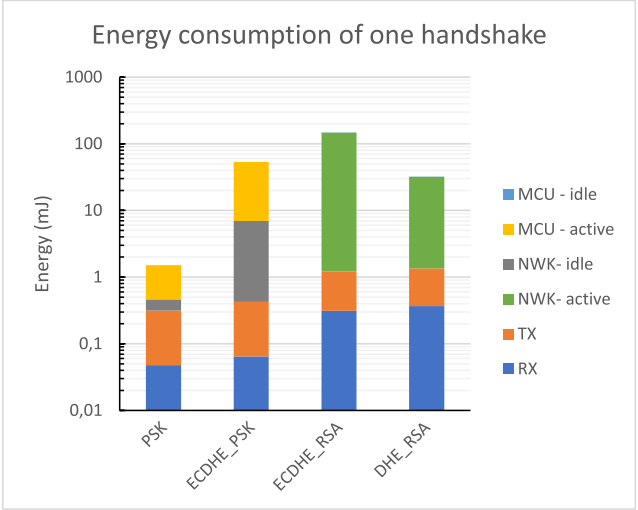

FIGURE 5. Estimated energy requirements for the each of the handshake configurations, where MCU, NWK, TX and RX stand for the energy consumption of the MCU, the energy consumption of the network processor and the energy consumption for transmitting and receiving data, respectively. A distinction is made between the energy consumption when the component is idle and when the component is active.

TABLE 2. Implementation overhead required for security protocols on the MCU.

Configuration	Storage (kB)	Stack (kB)	Heap (kB)
PSK	39.6	1.0	4.1
ECDHE_PSK	56.6	1.1	5.0
(EC)DHE_RSA	1.1	0.0	0.0

available accelerators. However, the exact details of the implementation of the network processor is not known.

The estimated energy results of the handshake process are shown in Figure 5. The energy results were generated using the performance results and the specifications of the MCU and the network processor. The first cipher suite (PSK) is the most lightweight solution, using about 1.5 mJ. The second cipher suite (ECDHE_PSK), which is the slowest of the four cipher suites, does not consume the most energy. It uses the low-power MCU for its calculations, resulting in only 53 mJ. The third cipher suite (ECDHE_RSA), executed on the network processor, is much faster, but consumes much more energy to do the calculations. It takes around 147 mJ to perform a handshake. The fourth cipher suite (DHE_RSA) is both fast and energy efficient in comparison to the other public-key based handshakes. It uses about 32 mJ to establish a connection.

The implementation overhead of the TLS protocol is given in Table 2. Results were derived from the Memory Allocation report provided by Code Composer Studio. The first and the second cipher suite require additional code on the MCU because of the mbed TLS library. The first cipher suite (PSK) requires about 40 kB of storage and 5 kB of memory (stack and heap). The second cipher suite (ECDHE_PSK) uses an additional amount of around 17 kB of storage and 1 kB of memory in comparison to the PSK cipher suite. If the TLS protocol is offloaded to the network processor, the MCU only requires about 1 kB of additional storage. This is the case for the third and the fourth cipher suite (ECDHE_RSA and DHE_RSA).

B. SECURE COMMUNICATION

In terms of implementation size, the firmware, which uses the network processor for the TLS protocol, uses about 41 kB of storage and 142 kB of memory. In total of 2% of Flash and 55% of RAM memory of the MCU is used. This leaves enough room for the other TLS implementation configurations.

On powering up the patch, it first establishes a secured connection with the server. This connection is maintained for as long as the patch is used. The handshake process of the TLS protocol is, therefore, only executed once. Next, the patch requests its session parameters (e.g. the UID). Finally, the patch starts measuring and periodically sends data to the local server. Every 10 seconds, the sensor data is compiled and pushed to the server using MQTT. The QoS level 1 configuration is used for the MQTT connection to ensure arrival of the data. For the remaining time, the platform is measuring the vitals and listening for commands from the local server.

The average power consumption and lifetime estimation of the patch is measured and compared in Table 3 to related work introduced in Sect. II. For our work, we use the results of the fourth cipher suite (DHE_RSA). The lifetime estimates are based on a 400 mAh battery. The results of related work were compiled and estimated using the values available in those papers. Our result was measured using

TABLE 3. Comparison of average power and lifetime estimation of the platforms referred to in Section II, with the following vital parameters: the pulse oximetry (PO), the electrocardiogram (ECG), the photoplethysmogram in three wavelengths (PPG), the bio impedance (BioZ), the 3-axes accelerometers (ACC) and the temperature (T).

Project	Vitals	Power (mW)	Lifetime (days)	End-to-end security
CodeBlue [2]				
	PO	87.78	0.63	no
MEDiSN [3]				
	ECG	11.29	4.87	no
	PO	105.30	0.52	no
Samie et al. [5]				
	ECG	5.87	9.36	no
This work				
	ECG, PPG, BioZ, ACC, T	69.82	0.79	yes

the Keithley 2000 Ammeter. Our platform consumes about 20% to 33% less than the other platforms that measure only the pulse oximetry (PO) parameter (CodeBlue and MEDiSN PO). However, it consumes about 6 to 12 times more power than the platforms which measure only the ECG parameter (MEDiSN ECG and the platform of Samie *et al.*). Nevertheless, our platform measures a larger range of parameters, and consequently, it must also send more data. Furthermore, the energy required for the key handshake using the fourth cipher suite (DHE_RSA) is only around 5% of the energy required for measuring and reporting the vital parameters of one period. In one period where the patch measures and reports the data, the patch uses about 698.2 mJ. The DHE_RSA based handshake process requires only 32 mJ. Given that the handshake only needs to be performed once in the lifetime of the patch, we can conclude that the energy consumption for providing end-to-end security is negligible to the energy spent during the entire lifetime of the patch.

VII. CONCLUSION

A wearable health monitoring system intended for real-life use in a hospital with a multitude of measured sensor data is designed and implemented. Using this system, six vital parameters are continuously being monitored: hearth rate, blood pressure variation, breathing rate, oxygen saturation, skin temperature and human activity (intensity and posture). Furthermore, an early warning score is added to notify clinical staff if abnormal conditions are detected. Additionally, clinical staff are able to use the system to report specific vitals to the Electronic Medical Record. The system is validated using a proof-of-concept implementation tested in three different hospitals. In comparison to related work, our patch provides a more energy-efficient monitoring system, taking into account that it supports a significantly higher amount of sensor data.

The system adds two custom entities to the hospital: the patch and the local server. The patch is a wireless wearable battery-powered device, and the local server is a Linux based device hosted by the hospital. The patch contains a MSP432P4011 MCU, a CC3120 network processor and the required on-board sensors. The impact of the wireless network protocol and the security architecture on the energy consumption is explored, resulting in the choice for Wi-Fi as the most efficient RF protocol for our use case. Furthermore, to provide the end-to-end security between the patch and local server, adhering to the security requirements determined through the STRIDE threat modelling approach, the TLS protocol is chosen. Four cipher suites of the TLS protocol are analyzed via two implementations: the mbed TLS library on the MCU and the secure socket feature on the network processor. Offloading the security to the network processor shows to be the most optimal solution. Furthermore, by leveraging long session lifetimes, the energy consumption overhead of establishing a true end-to-end secured channel is deemed negligible.

REFERENCES

[1] The European Parliament and the Council of the European Union, "Regulation (eu) 2016/679 of the European parliament and of the council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)," *Off. J. Eur. Union*, vol. 2016, no. 679, p. 88, 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj

[2] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton, "CodeBlue: An ad hoc sensor network infrastructure for emergency medical care," in *Proc. Int. Workshop Wearable Implant. Body Sensor Netw.*, 2004, p. 3. [Online]. Available: http://nrs.harvard.edu/urn-3:HUL.InstRepos:3191012

[3] J. Ko, R. P. Dutton, J. H. Lim, Y. Chen, R. Musvaloiu-E, A. Terzis, G. M. Masson, T. Gao, W. Destler, and L. Selavo, "MEDiSN: Medical emergency detection in sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 10, no. 1, pp. 1–29, Aug. 2010. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1814539.1814550

[4] H. Javdani and H. Kashanian, "Internet of Things in medical applications with a service-oriented and security approach: A survey," *Health Technol.*, vol. 8, nos. 1–2, pp. 39–50, May 2018.

[5] F. Samie, L. Bauer, and J. Henkel, "An approximate compressor for wearable biomedical healthcare monitoring systems," in *Proc. Int. Conf. Hardw./Softw. Codesign Syst. Synth. (CODES+ISSS)*, Oct. 2015, pp. 133–142. [Online]. Available: http://ieeexplore.ieee.org/document/7331376/

[6] G. Kambourakis, E. Klaoudatou, and S. Gritzalis, "Securing medical sensor environments: The CodeBlue framework case," in *Proc. 2nd Int. Conf. Availability, Rel. Secur. (ARES)*, Apr. 2007, pp. 637–643. [Online]. Available: https://ieeexplore.ieee.org/document/4159858/

[7] R. Hummen, H. Shafagh, S. Raza, T. Voig, and K. Wehrle, "Delegation-based authentication and authorization for the IP-based Internet of Things," in *Proc. 11th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2014, pp. 284–292. [Online]. Available: http://ieeexplore.ieee.org/document/6990364/

[8] S. Raza, L. Seitz, D. Sitenkov, and G. Selander, "S3K: Scalable security with symmetric keys—DTLS key establishment for the Internet of Things," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1270–1280, Jul. 2016.

[9] D. C. Neuman, S. Hartman, K. Raeburn, and T. Yu, *The Kerberos Network Authentication Service (V5)*, document RFC 4120, Jul. 2005. [Online]. Available: https://rfc-editor.org/rfc/rfc4120.txt

[10] S. R. Moosavi, T. N. Gia, E. Nigussie, A. M. Rahmani, S. Virtanen, H. Tenhunen, and J. Isoaho, "End-to-end security scheme for mobility enabled healthcare Internet of Things," *Future Gener. Comput. Syst.*, vol. 64, pp. 108–124, Nov. 2016, doi: 10.1016/j.future.2016.02.020.

[11] J. Winderickx, P. Bellier, P. Duflot, D. Coppieters, and N. Mentens, "Work-in-progress: Communication and security trade-offs for wearable medical sensor systems in hospitals," in *Proc. Int. Conf. Embedded Softw. (EMSOFT)*. New York, NY, USA: IEEE Press, 2019, p. 2.

[12] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *Proc. IECON-33rd Annu. Conf. IEEE Ind. Electron. Soc.*, Nov. 2007, pp. 46–51. [Online]. Available: http://ieeexplore.ieee.org/document/4460126/

[13] A. Shostack, *Threat Modeling: Designing for Security*. Hoboken, NJ, USA: Wiley, 2014. [Online]. Available: https://books.google.be/books?id=YiHcAgAAQBAJ
[14] J. Daemen and V. Rijmen, *The Design of Rijndael*. New York, NY, USA: Springer-Verlag, 2002.
[15] N. P. Smart, "Algorithms, key size and protocols report (2018), h2020-ict-2014-project 645421, d5.4," ECRYPT-CSA, Bristol, U.K., Tech. Rep. D5.4, Feb. 2018.
[16] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.
[17] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.
[18] H. Tschofenig and P. Eronen, *Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)*, document RFC 4279, Dec. 2005. [Online]. Available: https://rfc-editor.org/rfc/rfc4279.txt
[19] *Information Technology—-Message Queuing Telemetry Transport (MQTT) V3.1.1*, Standard ISO/IEC 20922:2016, International Organization for Standardization (ISO), Jun. 2016. [Online]. Available: https://www.iso.org/standard/69466.html
[20] E. Rescorla and T. Dierks, *The Transport Layer Security (TLS) Protocol Version 1.2*, document RFC 5246, Aug. 2008. [Online]. Available: https://rfc-editor.org/rfc/rfc5246.txt
[21] T. I. Incorporated. (2019). *Msp432p4011: Simplelink Ultra-Low-Power 32-Bit Arm Cortex-M4f MCU With Precision ADC, 2Mb Flash and 256Kb RAM*. Accessed: Mar. 2019. [Online]. Available: http://www.ti.com/product/MSP432P4011
[22] T. I. Incorporated. (2019). *Cc3120: Simplelink Wi-Fi Network Processor, Internet-of-Things Solution for MCU Applications*. Accessed: Mar. 2019. [Online]. Available: http://www.ti.com/product/CC3120
[23] D. Incorporated. (2019). *Docker Enterprise is the Industry-Leading Container Platform*. Accessed: Mar. 2019. [Online]. Available: https://www.docker.com/products/docker-enterprise
[24] P. S. Incorporated. (2019) *Rabbitmq is the Most Widely Deployed Open Source Message Broker*. Accessed: Mar. 2019. [Online]. Available: https://www.rabbitmq.com/
[25] N. P. Smart, V. Rijmen, B. Gierlichs, K. Paterson, M. Stam, B. Warinschi, G. Watson, and R. Tirtea, "Algorithms key sizes and parameters report-2014," *Eur. Union Agency Netw. Inf. Secur.*, vol. TP-05-14-084-EN-N, p. 113, Nov. 2014. [Online]. Available: https://www.enisa.europa.eu/publications/algorithms-key-size-and-parameters-report-2014
[26] D. Giry. (2018). *Cryptographic Key Length Recommendation*. Accessed: Apr. 2019. [Online]. Available: https://www.keylength.com/en/
[27] A. Holdings. (2019). *Arm MBED TLS*. Accessed: Mar. 2019. [Online]. Available: https://tls.mbed.org/

JORI WINDERICKX received the master's degree in industrial sciences electronics and ICT from the collaboration between KU Leuven and the University of Hasselt, in 2015, and the Ph.D. degree from the Research Group Embedded Systems and Security (ES&S) and in close collaboration with the Research Group COSIC, KU Leuven, in 2020. He is currently a Voluntary Researcher with KU Leuven and he has joined Flanders Make to research wirelessly connected industrial and vehicular applications. His research interests include energy-efficient implementations of security algorithms and protocols for the IoT systems.

PIERRE BELLIER started to work with the Microsys Laboratory, Department of Electrical Engineering and Computer Science, University of Liège. He is currently a Civil Engineer in electronics and computer science with the University of Liège. As a Project Engineer, he worked on various projects including research, industrial, smart buildings, medical, and the IoT applications. His research interests include energy efficient electronic design and microcontroller programming as well as power optimized wireless communication for the IoT systems.

PATRICK DUFLOT received the master's degree in computer engineering from the University of Liege. He occupied different functions in the software development lifecycle (developer, architect, team leader, data engineer) in several industries (banking, broadcast and healthcare). He joined Centre Hospitalier Universitaire de Liège, in 2018, to develop a wearable vital signs monitor. He is currently supporting project partners in IT, analysis, legal, and dissemination tasks for publicly funded research projects the hospital is involved in. https://www.linkedin.com/in/patrickduflot-a767601/.

NELE MENTENS (Senior Member, IEEE) received the master's and Ph.D. degrees from KU Leuven, in 2003 and 2007, respectively. She was a Visiting Researcher with Ruhr University Bochum, in 2013, and EPFL, in 2017. She is currently an Associate Professor with the COSIC Group, Electrical Engineering Department (ESAT), KU Leuven. She was/is the PI in around 15 finished and ongoing research projects with national and international funding. She is the (co)author in approximately 100 publications in international journals, conferences, and books. Her research interests include configurable computing for security, design automation for cryptographic hardware, and security in constrained environments. She serves as a Program Committee Member of renowned international conferences on security and hardware design, such as NDSS, CHES, SAC, DATE, FPL, and ESWEEK. She was the General Co-Chair of FPL, in 2017, and the Program Chair of EWME and PROOFS, in 2018. She serves as an Expert for the European Commission.