Universiteit Leiden
The Netherlands

Opleiding Informatica

Relevant e-health application factors for the HHQA-model

Vrinda Badloe

Supervisors:
Dr. Nusa Zidaric & Dr. Rosian Tossaint

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

17/11/2024

Abstract

This thesis provides a cybersecurity perspective to e-health, which is the use of information and communication technology in healthcare. The use of e-health can be combined with more traditional forms of healthcare, such as in-person consults. This is called hybrid healthcare. With the transition to a hybrid system, the quality of care should be preserved. With this in mind, the Hybrid Healthcare Quality Assessment (HHQA) model has been developed as a tool for healthcare organisations. Using this model, they can evaluate and improve their transition to hybrid healthcare. The model lists factors that contribute to the quality of care and contains a self-assessment questionnaire. The factors in the model do not take cybersecurity into account, so the objective of this thesis is to improve the model by recommending factors regarding cybersecurity for e-health apps. The focus here is on network communication security.
The factors were based on the review of the existing European and Dutch regulations about cybersecurity in healthcare and on a case study of an existing e-health app. The security of the communication between the app and the servers of this e-health app was evaluated through network traffic analysis. Based on the information from the traffic analysis, it can be concluded that the communication did not show any obvious weaknesses and abided by the current state-of-the art. The results from the traffic analysis were also used to construct a threat model, listing common threats and whether the app in question was vulnerable to them.
The combination of the regulations and the results of the traffic analysis resulted in a checklist that recommended five factors. The factors indicate that the e-health app:

- ☐ **Uses encryption algorithms that have not been compromised in security**
- ☐ **Performs regular software updates**
- ☐ **Implements proper public key certificates**
- ☐ **Implements authentication mechanisms when pairing devices**
- ☐ **Implements multi-factor authentication**

Contents

1 Introduction

Modern day healthcare knows many challenges, such as a rising amount of patients with a chronic illness, an ageing population and a staff shortage. By using a different approach to healthcare, e-health aims to relieve the healthcare sector from the burden of these challenges and improve the quality of healthcare. E-health is the use of information and communication technology in healthcare systems, such patient portals, measurement devices used by patients and electronic patient records [RIV], [HI]. When a healthcare system combines e-health with existing healthcare services, such as in-person consults, it is called hybrid health [CTHY14].
An example of a hybrid approach is an ongoing project conducted by the Cardio-Thoracic centre Leiden (Hart Long Centrum Leiden) called The Box. This is a package containing (wearable) devices, such as smartwatches and blood pressure monitors. The Box is provided to heart patients, who can take measurements of their vitals by themselves at home, after they have received care at the hospital. The wearable devices are provided along with an e-health app on the patients' smartphone, which handles communication with the device. It displays information to the user and communicates this information to other parties, such as the healthcare provider. The healthcare provider monitors the measurements of the patient remotely and, if necessary, discusses them via a digital consult. According to this project by the LUMC, this approach has multiple benefits. The hybrid approach reduces the amount of patients a healthcare provider needs to see in person, because a hospital visit is not always required. The patient is however still monitored remotely. This is an example of how the pressure on the healthcare system can be relieved by e-health. This hybrid approach also improves the quality of the measurements. The patients are able to take them from the comfort of their own home, where they tend to be more relaxed than at the hospital. The anxiety that a hospital may induce, is avoided by taking such measurements at home. Finally, the hybrid approach has the potential to improve the quality of care by detecting complications earlier [Lei24].
It can however be challenging to implement e-health into existing healthcare systems, since it requires changes to be implemented to different aspects of the healthcare system, such as the organisational structure and the role of the healthcare provider [TSKR+22]. Hybrid healthcare is where business, healthcare and technology intersect [Eys01] and the transition to and development of hybrid health should not interfere with the quality of care.
In order to manage and improve hybrid healthcare, Tossaint-Schoenmakers et al. [TSKR+22] have developed a quality assessment model and a self-assessment questionnaire. They are meant to be used by healthcare organisations as a tool to develop understanding about the implementation and progress of e-health services in their existing systems. The model and questionnaire were developed through concept mapping. This method included brainstorm sessions with participants, sorting the resulting ideas along with literature review data, and a concept map analysis, where a concept map is a visual representation of the relation between the ideas. The participants of the study were patients, healthcare professionals and members of management who have all had experience with e-health. The aim was to find factors that contribute to the effective organisation of e-health. The factors that were the most important according to the participants were included in the model and grouped together into eight clusters. Examples of factors are:

- *"The eHealth application is user-friendly"*
- *"Redesign the current work process and review what contributes to the desired care outcomes"*

- *"Treatment with eHealth has a positive influence on the patient's health"*
- *"Cocreation: eHealth is developed, implemented and redeveloped with different stakeholders"*

The respective clusters of these factors are:

- *"eHealth application"*
- *"Vision, strategy, and organisation"*
- *"End results for the patient"*
- *"Learning system: evaluation and improvement"*

The self-assessment questionnaire contains the same factors as the model and was designed to quantitatively assess the quality of each factor. As a result, organisations that implement this model can track their progress and visualise the points for improvement in their processes. In Appendix A all the factors and clusters are listed.

The transition to a more frequent use of digital services does however have a downside, which is the increased vulnerability to cybercrime. Attackers can use and target computer systems and networks for criminal activity [DN13]. This is especially an important subject matter to the healthcare sector, since it is one of the most vulnerable sectors to cybercrime. The computer systems in healthcare are trusted with sensitive personal information, such as electronic health records, making it a valuable target [TA21]. There are multiple factors that make it less complex to perform an attack on the IT systems in healthcare. For example, it has become easier to perform a cyberattack in general, because of readily-available online access to knowledge and resources, such as malware kits and access to computing resources [Dea17]. In addition to this, IT systems in healthcare often lack adequate security measures in order to withstand cyberattacks [TA21]. Since healthcare organisations are trusted with sensitive personal information, it is of the utmost importance for them to evaluate and strengthen their cybersecurity measures, especially if the amount of e-health services they provide increases.

1.1 Objective

As shown in Appendix A, there are two clusters relating to the quality of the technological services: the cluster *"Quality information technology infrastructure"* and the cluster *"eHealth application"*. As previously explained, it is important for healthcare organisations to consider the cybersecurity of their systems. Currently, this has not been incorporated in the HHQA-model. Therefore, the objective of this thesis is to expand the HHQA-model with requirements for cybersecurity. These requirements will be listed in the form of a checklist. The development of the factors for the checklist are based on the following.
A literature review of existing Dutch and European regulations has been performed, in order to identify high-level cryptographic requirements for healthcare applications. In addition to this, a case study of an existing wearable and accompanying e-health app has been performed. The wearable, also referred to as wearable device or device, takes vital measurements, such as body temperature and heart rate. The accompanying e-health app displays the information that was collected by the

device and communicates this to other parties, such as the healthcare provider or the servers of the app. This thesis focuses on the communication security between the wearable, the app and the server. The goal here was to identify how the authentication, encryption and integrity of the data sent and received by the app were ensured. Based on the information obtained from the case study, a threat model was constructed. This threat model indicates the vulnerabilities that were found in the communication between the app and the server.
The requirements from the regulations and the results from the case study form the basis for the development of factors that contribute to the HHQA-model. The factors in the checklist are meant to improve the cluster *"eHealth application"* and contribute to the quality of healthcare.

1.2 Scope

The e-health application that has been studied is the Withings app. The Withings app is provided along with a smartwatch called ScanWatch 2 1F [Witb], which is offered in the previously mentioned packet called The Box. As illustrated in Figure 1, the wearable is connected to the smartphone, on which the app is installed, through Bluetooth. The app in turn communicates with the servers of Withings through the internet. The original intention was to evaluate the security of both of these communication channels, by capturing the network traffic. As will be explained in Section 4, a network sniffing tool was used for capturing the network traffic and a BLE sniffer was used in combination with the network sniffing tool to capture the Bluetooth traffic. However, the BLE sniffer was not able to capture the Bluetooth traffic, so evaluating Bluetooth communication was not feasible for this project. This could be conducted in future work.
This thesis focuses on the evaluation of the network communication security between the smartphone running the Withings app and the Withings servers. Information such as the internal architecture of the app or the security measures in the rest of the system, such as the use of firewalls, was not available, due to the protection of intellectual property. Only the information that is provided by Withings on their website and the data obtained from the network traffic analysis will contribute to the conclusions of this work.

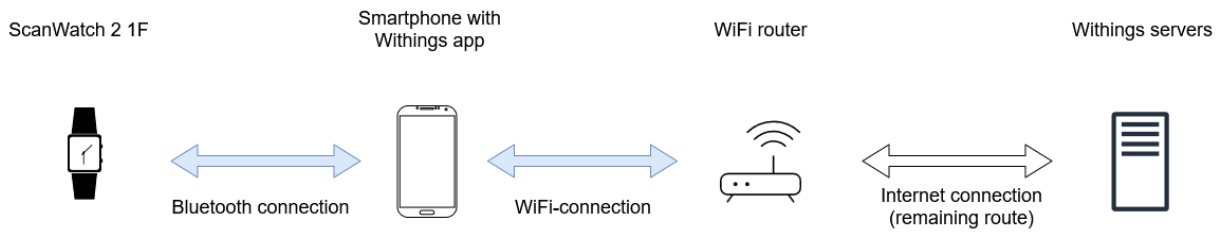

Figure 1: Overview of the system that will be the subject of the case study. The wearable device is a smartwatch with the model name ScanWatch 2 1F. This smartwatch is connected to a smartphone running the e-health app called Withings. The smartphone running the app is connected to the servers of the Withings app through the internet. The focus of this thesis is on the first hop on the route between the smartphone and the servers. This first hop is over WiFi.

1.3 Thesis overview

Section 2 describes the basic concepts of cybersecurity and cryptography and explains the working of communication over WiFi and Bluetooth. Section 3 provides insight to both Dutch and European rules and regulations about the high-level requirements for e-health applications regarding cybersecurity. Section 4 explains the methodology for the network analysis of the application and wearable and shows the results of the traffic captures, as well as a threat model for the communication. Section 5 contains the checklist and the motivation behind its items. Finally, Section 6 concludes the thesis and provides insights for future work.
This bachelor thesis for the Computer Science programme at LIACS, Leiden University was supervised by Dr. Nusa Zidaric (LIACS) and Dr. Rosian Tossaint-Schoenmakers (NeLL).

2 Cybersecurity and communication

As described in Section 1, it is important to take cybersecurity into account when implementing hybrid healthcare. There are several important aspects to cybersecurity. Their understanding is important in order to evaluate how secure a system is. This section explores the methods through which cybersecurity is ensured in a system. It explains several concepts, definitions and principles behind cybersecurity, network communication and Bluetooth communication.
The concepts discussed in this section are important for understanding the rules and regulations that dictate the minimum requirements for cybersecurity in e-health apps. They also provide a basis for understanding how the Withings app communicates and whether this is secure.

2.1 Security goals and STRIDE

Cybersecurity concerns the protection of computer systems from unauthorised access and its consequences through prevention and detection [DJT+22]. The aim is to protect data, hardware, software and communication networks from intentional misuse and from access to these assets by parties other than legitimate, authorised agents.
There are six main properties or goals of cybersecurity, explained in the following.

- **Confidentiality** indicates that private information should only be accessible to those who are authorised, both in transit and rest, meaning when being transmitted and when being stored. Confidentiality and the related concept of privacy, can be provided through data encryption. Only the intended, authorised parties can usefully interpret (decrypt) the information. Anonymity is another related concept and is the assurance that information cannot be linked to someone's identity.

- **Integrity** signifies that data, software and hardware need to remain unaltered by unauthorised parties.

- **Availability** entails that all assets, meaning data, software and hardware, need to remain accessible to those who are authorised.

- **Authentication** is the assurance that an entity, data or software is genuine. The identity of a user, communicating entity or a system process and the source of data or software is who we expect it to be, and not some intruder pretending to be the source.

- **Authorisation** is determined by the resource owner or domain administrator, regulating who has access to certain resources.

- **Accountability** is the property that the entity responsible for taking an action can be identified through logging and transaction evidence. It ensures non-repudiation, meaning someone who takes an action cannot deny having taken the action.

Attackers attempt to find and exploit vulnerabilities in a system and perform actions that violate the aforementioned security goals. In order to prevent attacks, vulnerabilities in the design of the system must be identified and removed. This is done through threat modelling, where threats, threat agents and the ways in which potential attacks can be carried out, are identified [VO21].

The threat modelling approach that will be used in this work, is the STRIDE method. It identifies the following six types of threats to a system.

- **S**poofing is masquerading as an authorised user and using their credentials to gain access to a system.
- **T**ampering is the modification of legitimate information, violating the integrity of the system or data.
- **R**epudiation is the denial of performing an action, evading accountability.
- **I**nformation disclosure is the unauthorised access to non-public, confidential information.
- **D**enial of service violates availability and disrupts the access to the system for legitimate users.
- **E**levation of privilege provides an attacker with higher privileges to the system than they should be allowed to have, providing unauthorised access to the system [KMLS17].

The communication channels that the Withings e-health application uses have been examined by identifying the STRIDE vulnerabilities. A secure e-health app should attempt to mitigate these threats. Hence, it is useful to also take these threats into consideration when constructing the checklist for the enhancment of the HHQA-model.
It is not in the scope of this work to consider the weaknesses in the system of the case study that are caused by attacks in the category of Elevation of privilege. This type of weaknesses require more detailed knowledge about the app, such as the application code and the system architecture, which was not available.

2.2 Cryptography

Cryptography is particularly important in the context of this thesis, because in healthcare, the personal data of the patients is required to be confidential. The personal data that is at rest or in transit between the patient and healthcare provider needs to be illegible to unauthorised parties. This can be achieved through data encryption, part of cryptography, where the original plaintext of a message is converted to ciphertext. Decryption is the reverse of this process and should not be possible for unauthorised parties.
In this section, three different types of cryptographic algorithms will be discussed: symmetric encryption, hashing and asymmetric encryption.

2.2.1 Symmetric encryption

With symmetric encryption, communicating parties use a cryptographic key to encrypt their messages. The key is a pre-shared, secret string of characters. The communicating parties exchange this key with each other before the encrypted communication starts and should keep this secret from anyone else. The key is used by an encryption algorithm that modifies the message content into the ciphertext. Only parties with the key can obtain the plaintext, meaning the original message, from the ciphertext and interpret the message [PP11]. The cryptographic algorithm itself is not a secret, so that the strength of the algorithm can be verified by others. According to Kerckhoffs'

principle, the strength of a cryptosystem does not rely on the secrecy of the algorithm, but on the secrecy of the key [MP08].
A commonly used symmetric encryption algorithm is the Advanced Encryption Standard (AES). The encryption process involves the key being expanded into the amount of rounds the algorithm will perform. In each round, the bytes that the message consists of are added to the round key, shifted and substituted in a pre-defined manner by other bits. The AES algorithm is among the standardised encryption algorithms, which are considered secure and fit for use.
The mode of operation is an important part of symmetric key cryptosystems. It describes how messages with a size greater than the keysize can be encrypted. This determines the level of protection that the algorithm can supply. For example, the Galois Counter Mode is capable of performing authenticity and integrity checks of messages through a Message Authentication Code (MAC), in addition to encryption. [PP11].

2.2.2 Hashing

Hashing is used for verifying the integrity and source of data. Hash functions take a value of arbitrary length as input and produce a short, fixed-length hash value as output by performing mathematical computations, such as bit-shifting, XOR and substitutions. A hash value can be seen as a fingerprint for the original message. Hash functions can be used in digital signatures, message authentication and storing passwords. Hash functions are one-way functions, which means that it is infeasible to compute the original value from the hashed value. This is called **pre-image resistance**. The second property of hash functions is **second pre-image resistance**. This means that for a given input, it is difficult to find a second input that results in the same hash value. The third property is **collision resistance**. It is not likely that two different input values produce the same hashed value, which is useful for checking integrity. If the hash value has changed, it means that the original value, the message, has changed.
Hashing can be divided into two categories. Unkeyed hashing and keyed hashing. **Unkeyed hashing** does not require a key in the hashing process. It meets some basic security requirements, but is not provably secure. Common unkeyed hashing algorithms are MD5 and SHA (Secure Hash Algorithm). MD5 and SHA-1 have proven to be weak with modern computing technologies and can be broken using techniques such as brute-force. The successor of SHA-1, which is SHA-2, is widely used for cryptography. **Keyed hashing** uses a secret key to enhance security. [Buc17], [Sil03], [Pit19], [RS18],[CZ17], [PP11].

2.2.3 Asymmetric encryption

Asymmetric or public key encryption involves a private key and a public key. The owner of the key pair distributes the public key to the party it wants to communicate with. The receiving party uses the public key to encrypt the messages it sends to the owner of this key. Only the owner of the key pair can decrypt these messages using the private key, that is kept secret from everyone.
Public key cryptosystems rely on hard problems. For example, it utilises the mathematical property that it is computationally easy to multiply two large prime numbers, but extremely difficult to factor the product. The entity generating the key pair chooses two large prime numbers and calculates different values with them, eventually resulting in a private and public key. A commonly used public key encryption algorithm is **RSA**, which uses this method for generating the keys.

Another commonly used public key algorithm is the **Diffie-Hellman Ephemeral (DHE)** key exchange algorithm. The communicating parties agree on the domain parameters α and a prime number p, that are not specifically required to be kept secret. Each party then chooses a random value (a and b) to be their own private key and computes their own public key by raising α to the power of the key (α^a and α^b). The public keys are exchanged and in order to obtain the shared secret, both parties raise the other parties public key to the power of their private key ($(\alpha^b)^a \bmod p$ and $(\alpha^a)^b \bmod p$). Both result in the same value, which can be used as a session key. The strength of the algorithm lies in the fact that it is a hard problem to compute ab for $\alpha^{ab} = S$, when α and S are known.
A variation of DHE is **Elliptic Curve Diffie-Hellman Ephemeral(ECDHE)**. The security of these cryptosystems lies in the fact that it is computationally difficult to find k if $Q = kP$, where P and Q are two points on an elliptic curve. For ECDHE, the two communicating parties both agree on a curve and a base point P on the curve. Both parties choose a random number (k and l), which is their private key. Each party computes their public key by multiplying their private key with the base point (kP and lP). To establish the shared secret (S), the parties multiply their private key with the public key of the other party. They compute $S = k \cdot lP$ and $S = l \cdot kP$, which results in the same point on the curve.

All these methods rely on the use of the public key belonging to the other party. In order to ensure that the public key is authentic, the communicating parties can use certificates. The certificates are issued by a trusted third party called a Certificate Authority (CA). The owner of a public key signs the certificate with a key, that should never be used for encryption, because it compromises the secrecy of the key and invalidates the security of the cryptosystem [Buc17], [PP11].

2.3 Network communication

The e-health application exchanges data with the server of the entity that owns the app. This exchange of data takes place over the internet. Computer networks are modelled with of communication layers, each with their own protocols [Los03]. The protocol stack and the their functions in network security are described in this section.

2.3.1 Network layers

The model that describes modern computer networks the best is the TCP/IP-model. This model breaks internet communication down into five layers: the lowest layer is the physical layer, then the data link layer, the network layer, the transport layer and finally the application layer. On each layer, protocols describe the rules of communication. This makes it possible for the communicating parties to understand each other. Figure 2 shows the TCP/IP stack along with the relevant protocols.
The **network layer** is the first of the layers relevant to this work. It organises data in the form of packets. The layer is responsible for delivering the packets across the intermediate systems and networks on the way to reach the intended destination. One of the main network layer protocols is the Internet Protocol (IP).
The **transport layer** is responsible for delivering an entire message to the correct process. Since messages often exceed the maximum size that can be sent over a network, the content needs to be segmented into multiple smaller packets. The protocols responsible for this are the Transmission

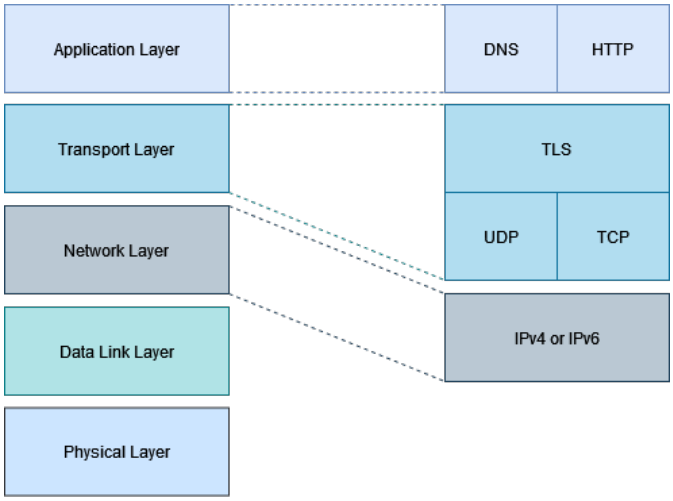

Figure 2: Network layers according to the TCP/IP-model (left) and the protocols on each layer (right). Adapted from Goralski [Gor17] and Oppliger [Opp16].

Control Protocol (TCP) or the User Datagram Protocol (UDP). On top of the TCP protocol, the Transport Layer Security (TLS) protocol can provide encryption mechanisms.
The **application layer** is the final layer of the TCP/IP model. It is responsible for the interface between the network and the application and can translate the way systems represent bits, so other systems can understand them [Gor17].

2.3.2 Transport Layer Security (TLS)

The TLS protocol is a client/server protocol operating between the transport layer and the application layer. The goal of using the protocol is to ensure authentication, confidentiality and integrity. The TLS protocol is a set of protocols, that can be divided into two layers. Figure 3 shows the protocols and sub-protocols in relation to higher and lower layers in the protocol stack.

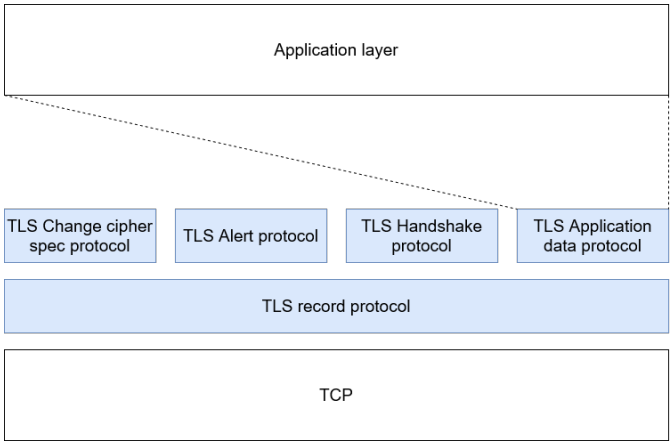

Figure 3: The sub-protocols that make up the TLS protocol. Adapted from Oppliger [Opp16].

The lower TLS record protocol is responsible for fragmenting, compressing and cryptographically protecting the higher-layer protocol data. It operates on top of a reliable transport layer protocol, such as TCP.
The following four protocols make up the higher layer on top of the TLS record protocol. The **TLS handshake protocol** is used for the establishment of a secure connection, where the communicating parties negotiate security parameters. The **TLS change cipher spec protocol** is used to signal a change in the ciphering strategy, where the security parameters negotiated during the handshake are put in place and used. The **TLS alert protocol** is used to signal potential problems. The **TLS application data protocol** is responsible for securely transmitting application data. On the transmitting side of the communication, the TLS alert protocol passes the data on to the lower TLS layer for encryption and compression [Opp16]. The handshake protocol performs the following actions. First, the client initiates the communication session by sending a `client_hello` message. This message contains the TLS-version number, a random number, the session ID and a list of the supported cipher suites. The random value is used to generate the session keys. The cipher suites indicate the public-key encryption algorithms, symmetric-key encryption algorithms and hash functions that the client can support.
The server responds with a `server_hello` message. This message indicates the highest TLS version available on the server, along with the server random, session ID and the cipher suite that the server selected from the list provided by the client [WK15]. Cipher suites indicate what cryptographic algorithms are used for key exchange, certificate verification, bulk encryption and hashing [Nat23]. For example, the cipher suite TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 uses ECDHE for key exchange, RSA for certificate verification, AES256 in Galois Counter Mode for bulk encryption and SHA384 for hashing, as illustrated in Figure 4.

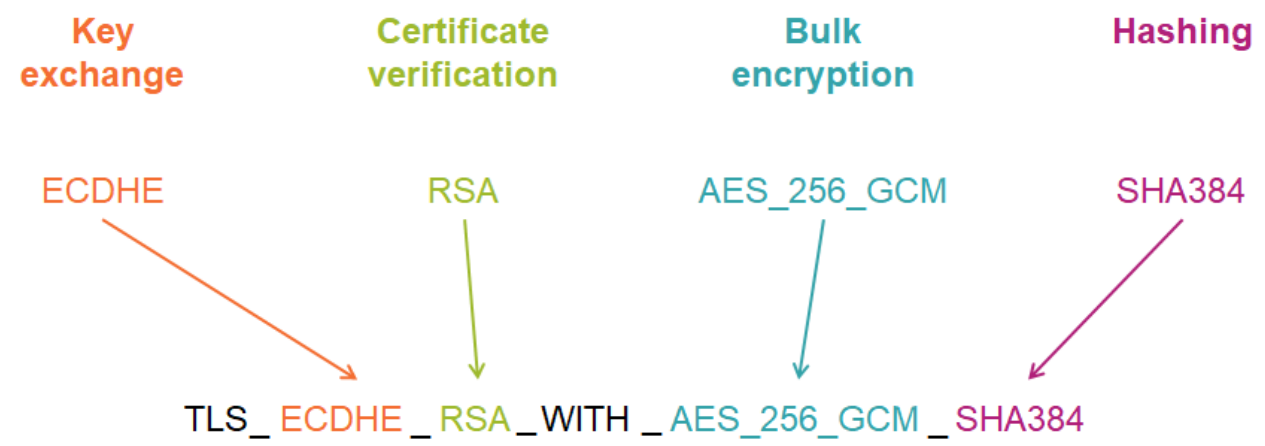

Figure 4: Cipher suites indicate what cryptographic algorithms are used for key exchange, certificate verification, bulk encryption and hashing in a specific order.

After the `server_hello` message, the server can provide its certificate, proving its identity. The server and client also exchange their public keys.
The handshake is completed when the client and server send each other the `change_cipher_spec` message, indicating that further communication will be encrypted with the negotiated parameters, and the `finish` message. This message is used to determine whether the master secret that both parties calculated with the exchanged keys are the same. Figure 5 shows an overview of the handshake messages.

The master secret, along with the client and server randoms are used to calculate secret keys for both the server and client. These are the keys used for generating the MAC secret and the keys used for encryption.
After the handshake protocol has established all parameters, the record protocol takes over the communication. The party that wants to send a message divides it into blocks [WK15]. A Message Authentication Code (MAC) is attached to each block. The MAC is generated by using the MAC key, which is one of the keys generated after the handshake. The server and client both have their own key for this. Other properties of the message, such as the sequence number and the length, are concatenated and hashed along with the key to generate the MAC [Opp16]. The block and MAC are then encrypted, a TLS-header is added and then sent to the other party using TCP. When it is received, the record is extracted and the MAC is verified using the same key that was used for generating the MAC [WK15].
Alternatively, the TLS protocol can employ Authenticated Encryption with Associated Data (AEAD) algorithms for message authentication and integrity assurance. This is used in TLS versions 1.2 and higher. AEAD algorithms, such as AES-256-GCM, produce an authentication tag, that is appended to the ciphertext. They can detect accidental and intentional modifications to messages, providing authentication and integrity alongside confidentiality [Dwo07], [McG08].

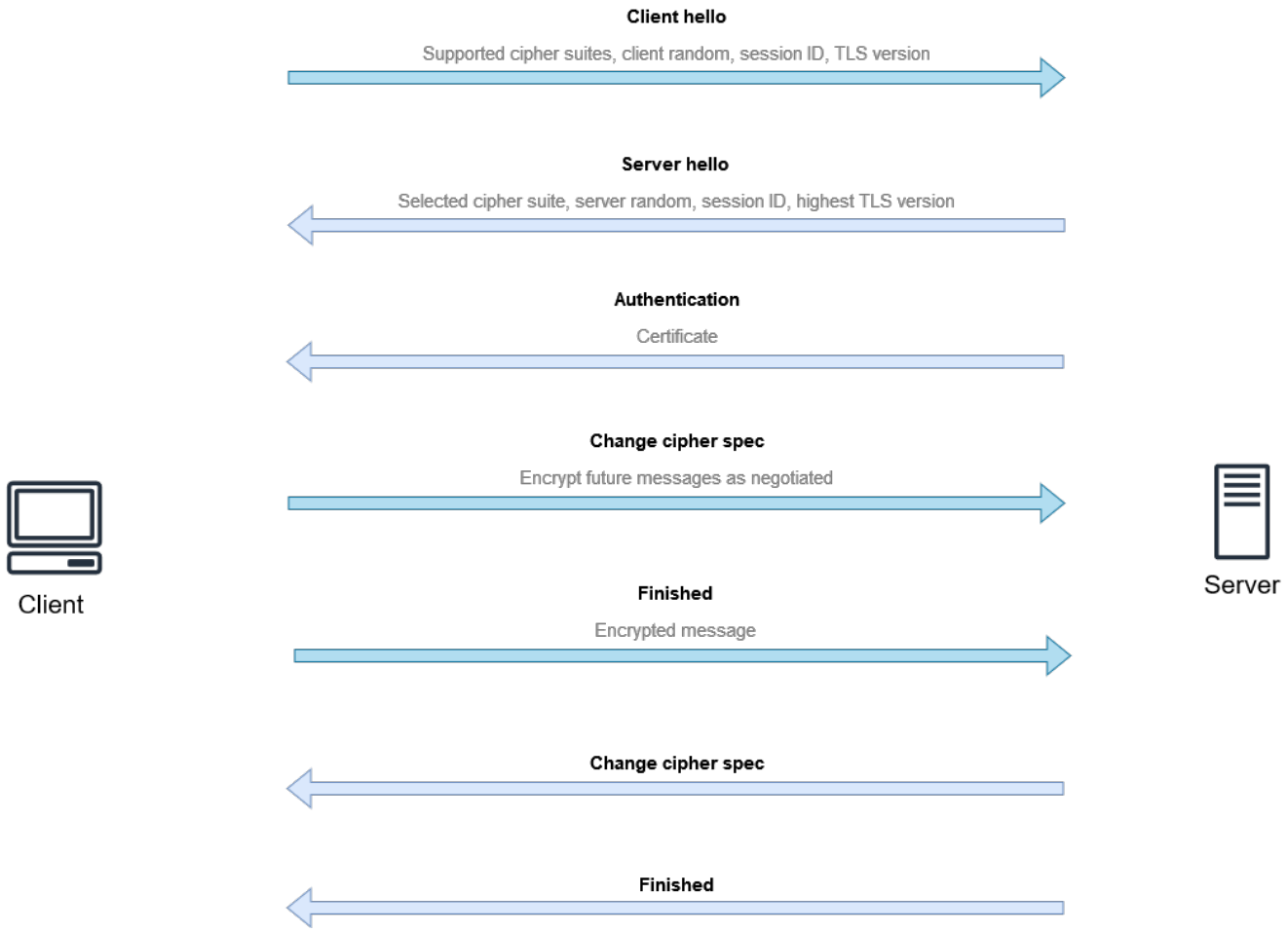

Figure 5: TLS handshake messages between the client and the server.

2.3.3 Transport Control Protocol (TCP)

The Transmission Control Protocol (TCP) is a reliable transport layer protocol. TCP is responsible for delivering segments of messages in the order of the original message and makes sure all the segments arrive at the destination. It uses a three-way handshake to establish a connection between the communicating parties. The connections are established according to the client/server model, where the process initiating the connection is the client and the other process the server.
During the three-way handshake, a number of parameters are negotiated.
The first message in the handshake is sent by the client to the server. The SYN (synchronise) flag is set, indicating a request for connection. The message also contains the arbitrary Initial Sequence Number (ISN). The sequence number will indicate the order of the segments, so that the original message can be reconstructed, once all the segments have been transmitted.
When the server has received the SYN request from the client and has the resources available to establish a connection, it sends a SYN ACK (synchronise, acknowledge) message to the client. In this message, the server's ISN is also exchanged. This way, a two-way connection can be established. Finally, when the client has received the SYN ACK segment from the server, it responds with an ACK message. The handshake has now been concluded and a reliable, two-way connection has

been established between the client and the server [HTC22].
TCP is a transport layer protocol that offers reliability, but this is at the cost of time and computing power. For applications where reliability is considered less important than speed, the User Datagram Protocol (UDP) can be used. UDP can perform error checking using a checksum, however, UDP does not establish a connection and does not guarantee the delivery of data [HTC22].

2.3.4 Internet Protocol (IP)

On the network layer, the Internet Protocol is responsible for transporting data between networks. There are currently two versions of the IP protocol, IPv4 and IPv6. Networks and devices are identified with a unique IP address. IPv6 has a larger address space and the new version offers more implementations for security. An IPv4 address can be represented by the decimal values of each byte separated by a dot, for example 255.255.255.255. An IPv6 address is much longer. This why it uses the hexadecimal notation, where each hexadecimal represents two bytes and is separated by a colon, for example 345f:244f:576a:1456:1345:234b:865c:765c.
When data enters the network layer, an IP header is added to the message containing the source and the destination of the packet. It then passes through one or multiple networks, where packets are forwarded by routers. The routers forward packets to the network closest to the desired destination [HTC22].

2.3.5 Domain Name System (DNS)

Entities on a network are identified with their IP address, but these numerical addresses are not user-friendly and difficult to remember. The Domain Name System is a globally distributed database, mapping domain names to IP addresses. Domain names are user-friendly names indicating the resources on the internet, such as the domain name Withings. The DNS protocol is an application level protocol.
Finding the mapping between a domain name and its corresponding IP address is called resolving and is done by a system component called a resolver. When a connection to a server needs to be made, the resolver first performs a look-up in its own cache. If the cache does not contain the correct record, the resolver will query servers according to the hierarchical structure of the DNS [DK06].

2.3.6 Attacks

There are several common attacks that can be mounted against a communication system. These attacks and their definitions are listed in Table 1.

Attack	Description
Side channel attack	Use power, timing, or electromagnetic radiation from nodes to breach encryption.
Physical attack	Physically damage hardware in the system. This is a type of DoS (Denial of Service) attack
Downgrade attack	Force a connection to use less secure, weaker encryption algorithms.
Brute force attack (exhaustive search)	Trying every possible combination in order to break the protections, such as guessing a cryptographic key by trying every combination of characters.
Eavesdropping/sniffing	Listening in on network communication to detect patterns through traffic analysis, often used to gather information for initiating attacks.
Man-In-The-Middle (MITM) attack	Intercept data by acting as the intended recipient, potentially altering data before passing it to the actual receiver. The communicating parties are unaware of the attacker's presence.
Jamming	Interferes with wireless signals, preventing data transmission. This is a type of DoS attack.
Replay attacks	Capture and resend messages, masquerading as the original sender of that message.
Route attacks	Alter routing information to cause loops, error messages, false routes, or packet dropping.
Viruses	Malware like Trojans and Worms that replicate, self-repair, and cause widespread system or network damage.

Table 1: Types of attacks on network communication[KR25], [Zha21], [Crob], [Croa], [ZP03]

2.4 Bluetooth

The smartwatch and the smartphone with the Withings app in the case study communicate through Bluetooth Low Energy (BLE). Bluetooth is a wireless technology meant for short range and low-cost communication between devices. Bluetooth devices communicate through radio signals on the 2.4 GHz ISM (Industrial, Scientific and Medical) band.
Bluetooth Low Energy (BLE) is suitable for devices that require their communication to have low power consumption. This is especially useful for applications in wearables.
Bluetooth communication also works with communication layers and their protocols. The Bluetooth Low Energy protocol stack is shown in Figure 6. Only the Security Manager is discussed in this section, since it is the only layer relevant to this work. For the explanation of the other layers, refer to Appendix C.

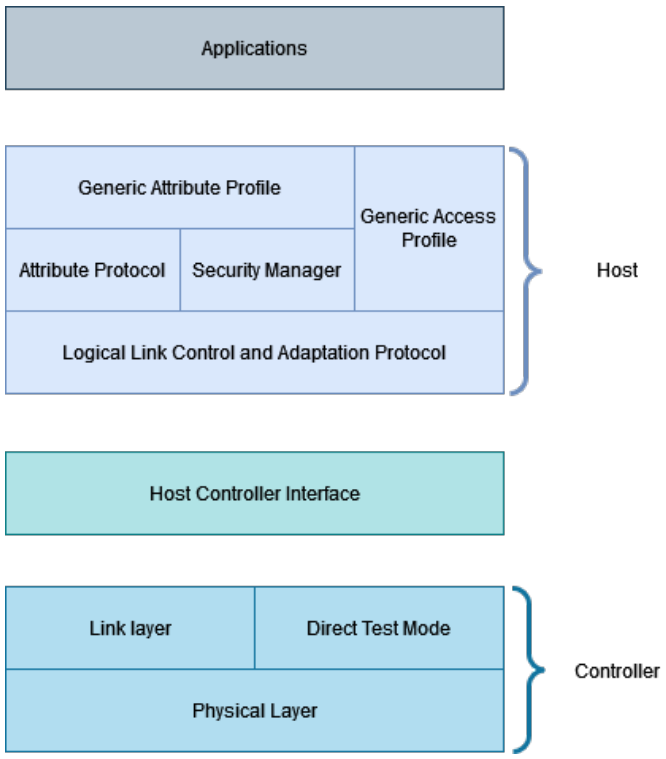

Figure 6: Bluetooth stack. Adapted from Niemen et al. [NSI+15]

2.4.1 Security Manager

The security manager is responsible for pairing mechanisms, authentication and encryption.
There are four methods for pairing. The used method depends on the capabilities of the devices. *Just Works pairing* is the most simple method, where no passkey is exchanged. Unlike with the other methods, Just Works does not provide any protection against Man-In-The-Middle attacks. With *Numeric Comparison*, both devices compute confirmation values and users check whether the values match. If not, the protocol is aborted. With *Passkey Entry*, one device computes and displays a numeric passkey, which needs to be entered on the other device. The last method is the *Out of Band* method, where some other way, such as touch with NFC, is used.
The pairing process is completed in three phases. During the first phase, the devices exchange I/O capabilities, indicating whether the devices can input and display information. The requirements for authentication and information about the keys the devices distribute and expect are also exchanged. In the second phase, a pairing method is selected and initiated. Both devices generate their own public-private key pair and the public keys are exchanged. Using the keys and other cryptographic functions, the devices authenticate each other. In the optional third phase, various different keys are exchanged [NSI+15], [Blu], [Gup16].

3 Rules and regulations

Since cybersecurity concerns an increasing number of industries and services, regulations and directives on the national and EU level have been established. This section describes the essence of these regulations regarding cybersecurity, indicating the minimal requirements for healthcare apps and software in general. These requirements will be used to determine if the Withings app communicates in a secure manner and will also contribute to the factors that will be recommended for the HHQA-model.

3.1 General Data Protection Regulation (GDPR)

The General Data Protection Regulation describes the regulations for the protection of personal data. In compliance with the GDPR, organisations are obligated to encrypt, pseudonymise or anonymise personal data. The "appropriate technical and organisational measures" need to be taken in order to protect personal data from being interpreted by unauthorised parties [GDP19].

3.2 Medical Device Regulations (MDR)

Devices used for medical purposes and accessories to devices, such as software, are defined as a medical device in the context of the MDR. The MDR obligates the manufacturers to implement a Post-market surveillance system, which ensures that manufacturers continue testing the safety and performance of their products, including software, even after placing the product on the market. The manufacturers should implement features for software updates and should track potential risks. This ensures that the software is up to the standard to state-of-the-art principles, to which the manufacturers are required to adhere. It is important that with their implementation, manufacturers minimise the risks associated with cybersecurity. By design, devices must prevent unauthorised access that could harm the safety of the device. The devices are required to have measures in place that ensure integrity, availability and confidentiality of data, which need to be well-documented. This needs to be maintained throughout all components of a system, so it is also important that the interoperability with the other components is secure [mdr].

3.3 Network and Information Security (NIS2) directive

The NIS2 directive is a European regulation concerning the security of network and information systems. The directive specifies that the member states are required to make sure that organisations take the appropriate technical measures to mitigate the risk of incidents and their consequences. Incidents are the disruption of the services provided by the system. When an incident has taken place, organisations are required to notify a Computer Security Incident Response Team (CSIRT), which can help with the mitigation of the consequence of incidents. Every EU member state needs to have at least one CSIRT, which also acts as a point of contact for other member states, in order to boost international cooperation.
The NIS2 directive describes a pro-active approach and describes several technical measures that need to be implemented. It is mandatory for organisations to have policies for the use of cryptography and encryption and use multi-factor authentication when appropriate. Multi-factor authentication is the use of more than one piece of evidence to prove the users identity, such as entering a PIN

along with entering a password [IBM].
Additional measures include risk assessment, cybersecurity training and plans for managing business operations in the event of a security incident and afterwards [Dir23], [Minen].

3.4 Dutch National Cybersecurity Centre (NCSC)

The Dutch National Cybersecurity Centre (Nationaal Cybersecurity Centrum, NCSC) offers guidelines for the implementation of TLS. The guidelines mention four levels of security. "Insufficient" settings should not be chosen, while "Phase out" settings are known to be vulnerable to attacks, but are needed for backwards compatibility. A "Sufficient " setting is fine to use, but the "Good" settings are the most secure. It is stipulated that no TLS configuration will remain secure forever, so it is important that the "Good" settings are updated when newer configurations become available. The guidelines consider version 1.3 of TLS to be "Good"and version 1.2 to be "Sufficient". Other TLS versions should be phased out. The cryptographic algorithms that TLS uses can be divided into four domains: certificate verification, key exchange, bulk encryption and hashing. The security levels for these domains are specified as follows. The RSA algorithm is considered "Good" for certificate verification. The hash functions that are classified as "Good" for the digital signatures on the certificates are all SHA algorithms higher than SHA-1. For key exchange, the ECDHE algorithm is "Good", DHE is "Sufficient "and RSA is "Phase out". For bulk encryption AES-256 and AES-128, both in GCM mode, are "Good". Those same algorithms, but in Cipher Block Chaning (CBC) mode are "Sufficient ". The discussion of modes of operation is beyond the scope of this thesis.
TLS versions older than 1.3 have the option for compression, but this is considered insecure. It is classified as "Good" to not provide that option and "sufficient" to only offer it at the application level.
Another possible weakness is allowing clients to force renegotiation, where a new handshake is performed. This leaves servers vulnerable to DoS-attacks, so it is classified as "Good" if it is not allowed in the implementation.
The guidelines also contain a list of cipher suites, classified as "Good", "Sufficient" and "Phase out". Refer to the document [Nat23] for the full list.

4 Experiments and Threat model

In order to have an understanding of the security of current e-health applications, the Withings app and the ScanWatch 2 1F were studied, as depicted in Figure 1. The main focus here was the communication between the watch, the app and the Withings servers. This section explains how the communication was evaluated and what the findings were.
A network sniffing tool called Wireshark [Wir] was used for the capture of the communication. With this tool, network traffic can be recorded. A total of four sessions were captured, where the watch was connected to the phone and used for measuring temperature and heart rate.

4.1 Network communication

The setup for the capture of the network communication between the Withings app and the servers was as follows. Wireshark can only capture the traffic that is sent from and to the device that it is installed on. In this case, it was installed on a laptop. However, the Withings app was installed on a different device, a smartphone. In order to capture the desired traffic, the smartphone was connected to the laptop using the WiFi hotspot of the laptop. This way, all network traffic sent from and to the phone was directed to the laptop first. There were no additional setup requirements for hardware or software. The next sections describe the findings of the captures.

4.1.1 DNS

As described in Section 2.3.5, networks and devices are uniquely identified with an IP address. When the phone and the server want to communicate with each other, they need to know each others IP address. Through DNS, a domain name such as Withings can be mapped to its IP address. By identifiying the IP-address of the servers, the communication between the app and the servers can be followed in the network capture.

The result of filtering DNS-queries and responses that contain the substring “withings”are shown in Figure 7. Multiple DNS queries have been made and it is shown that three different IPv4 addresses are associated with Withings: 89.30.121.170, 89.30.121.180 and 89.30.121.150.

4.1.2 TCP and TLS

In order to follow and analyse all communication between the servers and the app, all packets with either one of these IP addresses are filtered out.
Section 3 mentions that according to multiple regulations and directives, it is required that data is encrypted, in order to provide protection of personal data. Section 2.3.2, described the best technical ways to do this for network communication: using TLS. The capture showed that this protocol was used in the communication, on top of TCP.
The capture shows the following about TCP. After every new DNS request to one of the Withings servers, the TCP connection is established with the three-way handshake, as described in Section 2.3.3. The client sent a SYN request, to which the server responded with a SYN-ACK. The client responded with an ACK and the connection was established. One of the handshakes that was captured is shown in Figure 8. Figure 9 and Figure 10 show the details of the first two

No.	Info
37	Standard query 0xdc43 A scalews.withings.net
38	Standard query response 0xdc43 A scalews.withings.net A 89.30.121.170
55	Standard query 0x98db A prod.rudderstack.withings.net
56	Standard query response 0x98db A prod.rudderstack.withings.net CNAME front.withings.com A 89.30.121.150
97	Standard query 0x2e04 A static.withings.com
101	Standard query response 0x2e04 A static.withings.com CNAME cdn-eu.withings.net A 89.30.121.180
3866	Standard query 0x4327 A prod.rudderstack.withings.net
3870	Standard query response 0x4327 A prod.rudderstack.withings.net CNAME front.withings.com A 89.30.121.150
6709	Standard query 0x3b56 A prod.rudderstack.withings.net
6710	Standard query 0x3b56 A prod.rudderstack.withings.net
6711	Standard query response 0x3b56 A prod.rudderstack.withings.net CNAME front.withings.com A 89.30.121.150
6713	Standard query response 0x3b56 A prod.rudderstack.withings.net CNAME front.withings.com A 89.30.121.150

Figure 7: The packets that are sent with the DNS protocol and contain the word Withings. This shows that there are three IP-addresses associated with Withings, which are 89.30.121.170, 89.30.121.180 and 89.30.121.150.

No.	Protocol	Info
39	TCP	61977 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=2520752998 TSecr=0 WS=1024
40	TCP	443 → 61977 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
41	TCP	61977 → 443 [ACK] Seq=1 Ack=1 Win=88064 Len=0

Figure 8: Three-way TCP handshake between the app and one of the Withings servers.

packets. In both packets, the Initial Sequence Number, the acknowledgement number and the timestamp are set. After this handshake, both parties are ready to receive packets from each other. All communication was done over TCP. This was confirmed by searching for any packets that did not use this protocol. This search displayed zero packets.

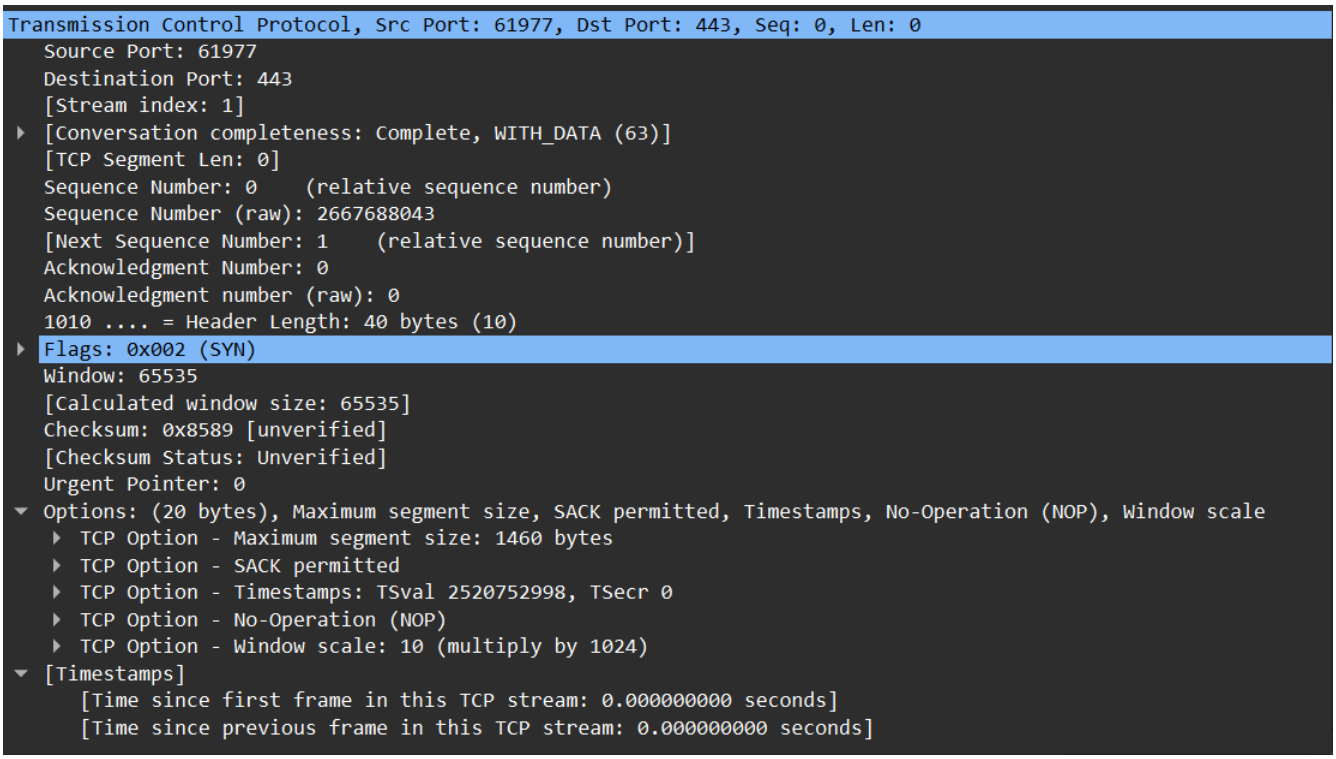

Figure 9: SYN request from client

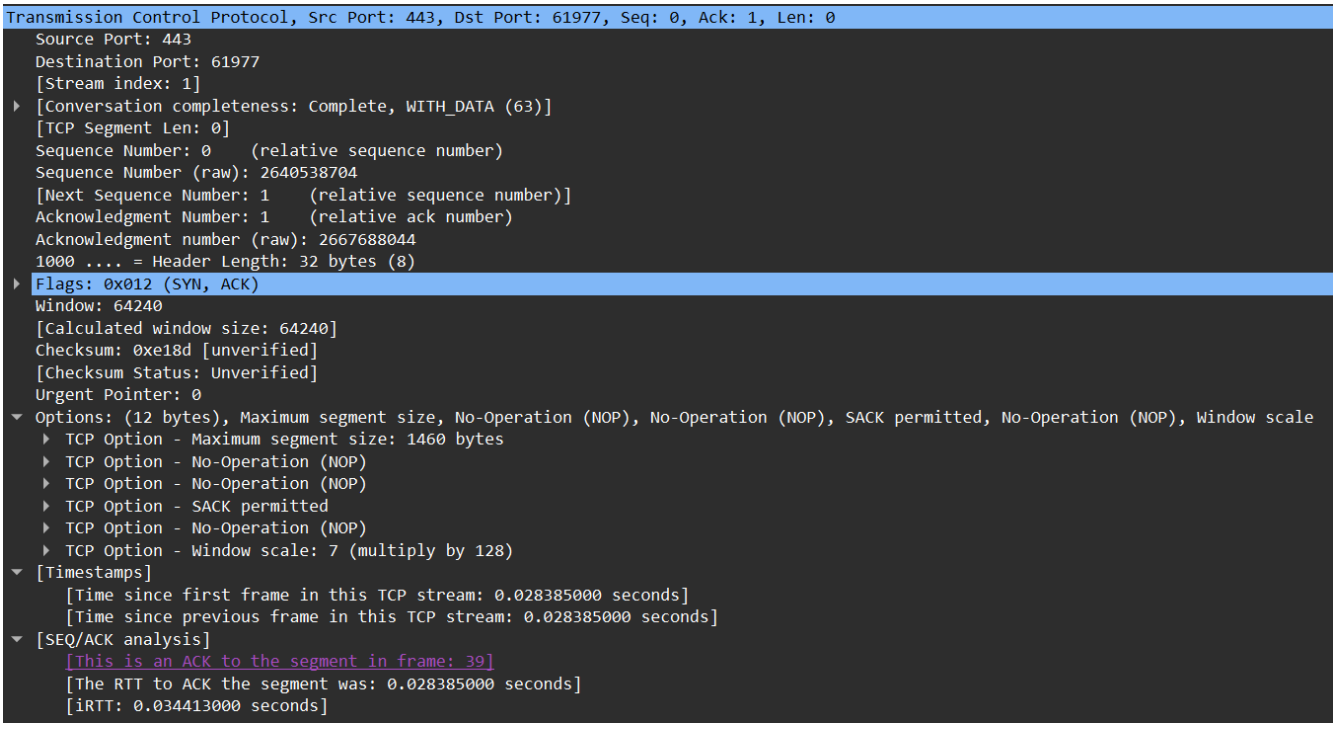

Figure 10: SYN-ACK message from server

On top of the TCP protocol, the TLS protocol provides secure connection and encrypts data. This is shown in Figure 11.

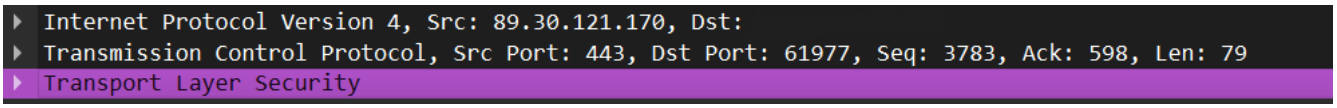

Figure 11: The Internet Protocol delivers packets from one IP address to the other, while TCP transports the segments from process to process. On top of the TCP protocol, the TLS protocol provides security.

Before the server and the app can exchange data, a TLS handshake needs to be performed. Here, the server and app negotiate the security parameters of the connection, as it was explained in Section 2.3.2. Figure 12 shows an overview of the handshake between the server and the client.

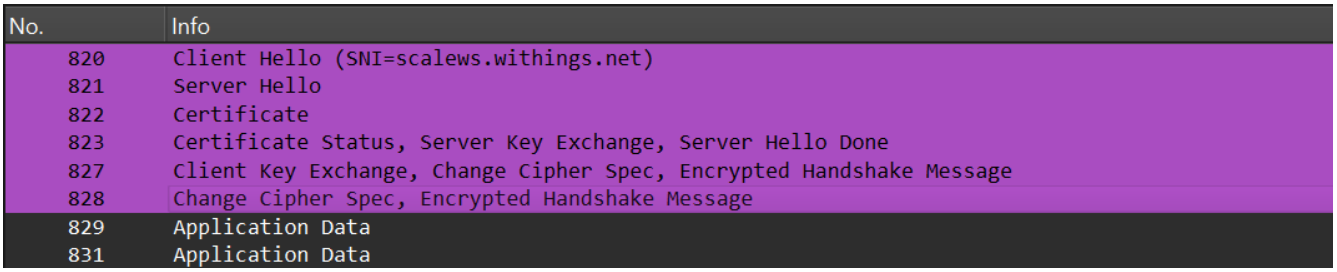

Figure 12: The purple colored packets show the TLS handshake between the server and the client. The last two packets show the Application Data Protocol taking over, where the actual application data is being sent.

One of the parameters that are negotiated, is the TLS version that both parties will use. In the `Client Hello` message, the client proposes the highest TLS-version it is capable of using. In the captures, the client sometimes used TLSv1.2 and sometimes TLSv1.3, as shown in Figure 13. These are the most recent TLS versions. While TLSv1.2 always performs the full handshake, TLSv1.3 allows a shorter handshake to resume a session with an existing pre-shared key. This is the case for the first two ciphersuites that are shown in Figure 14. TLSv1.3 has more security features than the previous versions, but TLSv1.2 is still widely used and considered secure, according to the standards of the NCSC [Nat23].

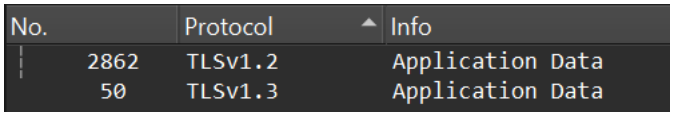

Figure 13: TLSv1.2 and TLSv1.3 were used in the communications. The search for any packets sent with other version numbers showed that these were the only two versions.

The use of two different TLS versions results in two slightly different `Client Hello` messages. Examples of both messages are shown in Figure 14 and Figure 15. In the messages, the client lists all the cipher suites it is capable of using.

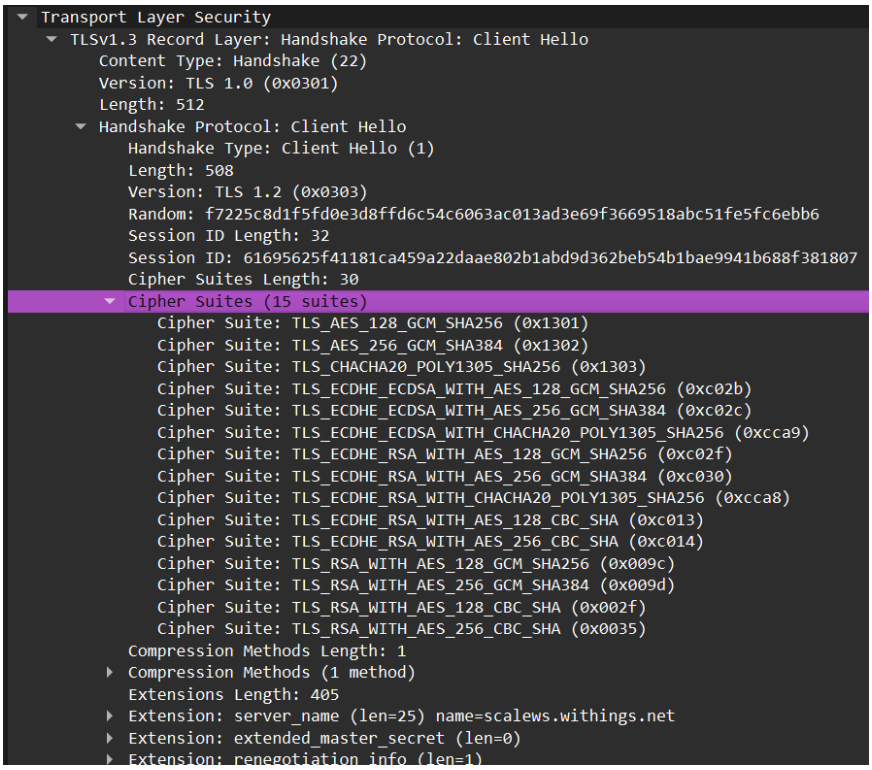

Figure 14: `Client Hello` message for TLSv1.3. The client proposes 15 cipher suites.

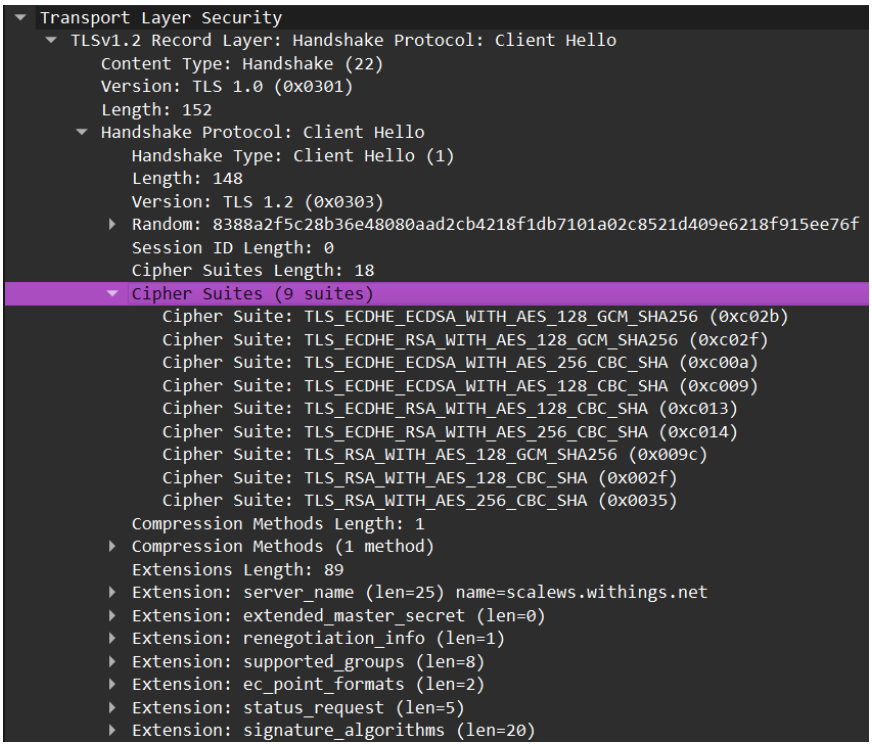

Figure 15: `Client hello` for TLS v1.2. The client proposes 9 cipher suites.

The server responds with a `Server Hello` message with one of the cipher suites it is also capable of using. In all the captures, one of two different cipher suites were used. The cipher suite TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 is used when the communication uses TLSv1.2 and TLS_AES_256_GCM_SHA384 is used when the communication is over TLSv1.3. This is shown in Figure 16 and Figure 17. The NCSC [Nat23] classifies the first one as "sufficient" and the second one as "good ", which means that the algorithms used for encryption, hashing, certificate verification and key exchange are up to standard.

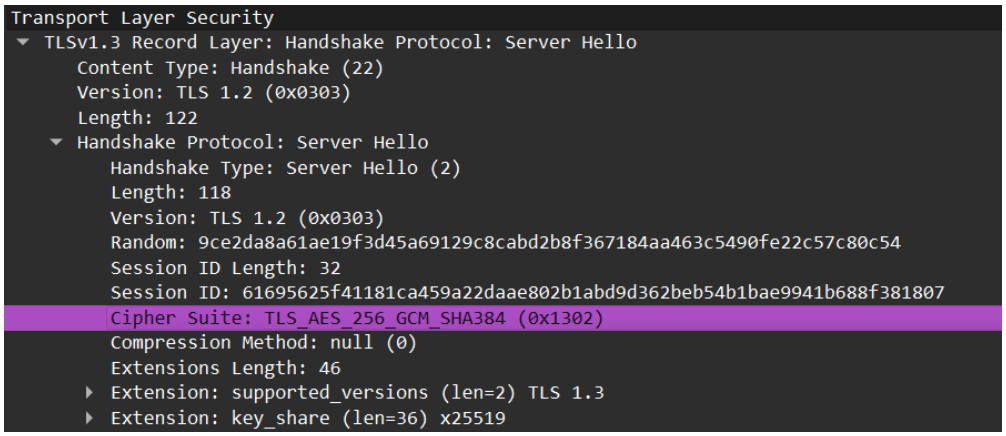

Figure 16: `Server Hello` for TLSv1.3. The server always chooses the cipher suite TLS_AES_256_GCM_SHA384 for this TLS version.

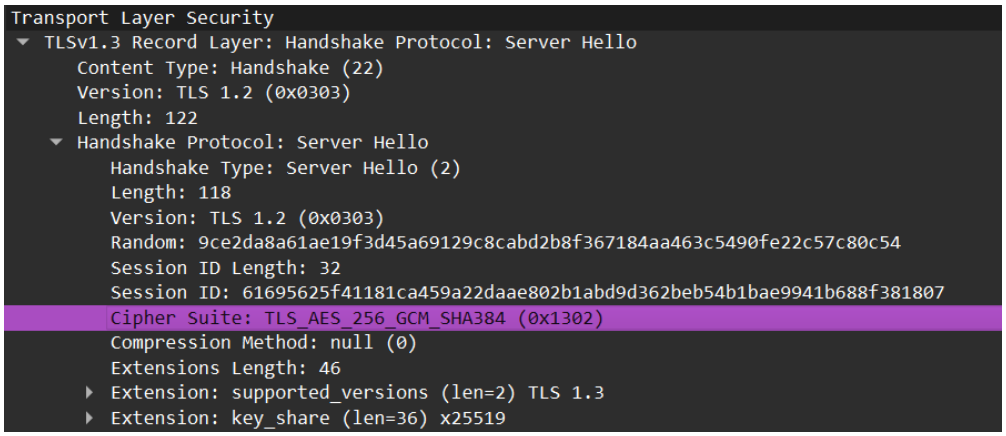

Figure 17: `Server Hello` for TLSv1.2. The server always chooses the cipher suite TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 for this TLS version.

During the handshake, random values are exchanged between the communicating parties. If these values are re-used, attackers might be able to circumvent the security that would otherwise be provided by TLS, for example, by performing a replay attack. In all of the captures, 320 randoms were exchanged. No duplicate random was found, which means that in separate sessions, different randoms are used every time.

The server then exchanges its certificate that needs to have been signed by a trusted Certificate Authority, as mentioned in Section 2.2.3. As shown in Figure 18, the certificate was signed by the non-profit "Let's encrypt", which has provided TLS certificates to 450 million websites [enc].

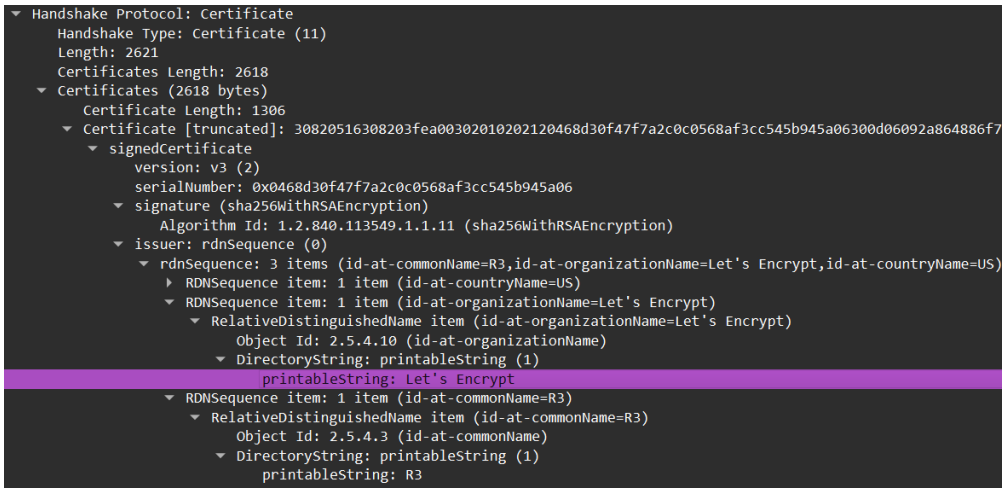

Figure 18: The Certificate Authority Let's encrypt issued the certificate to the servers.

Figure 12 shows that after the certificate exchange, a key exchange takes place. The server and the client exchange their public keys and the server also provides a signature. Figure 19 shows the key exchange message from the server and Figure 20 shows the public key exchange message from the client.

Figure 19: The server exchanges its public key and a signature with the client.

After the client has exchanged its key, it sends a Change Cipher Spec message, indicating that the exchange of messages that will follow, will be encrypted using the negotiated parameters. The client then sends the Encrypted Handshake Message, which is the encrypted `Finished` message. The server responds with a Change Cipher Spec and also sends an `Encrypted Handshake` Message. This concludes the handshake, after which the application data protocol can be seen to take over, also shown in Figure 12.
The security parameters that were used for the TLS connection are adhering to the standards that were required by Section 2 and Section 3, which means that the TLS protocol provides adequate security in the communication between the e-health application and the Withings servers.

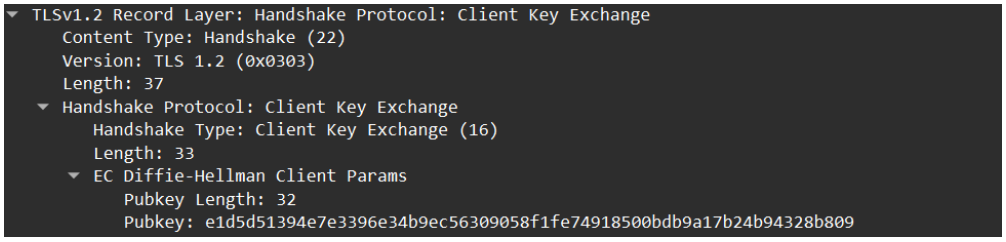

Figure 20: The client exchanges its public key with the server.

4.2 Bluetooth

The Bluetooth connection could not be evaluated in detail. This is because the attempted approach did not reveal useful information. The attempted approach was to use a BLE sniffer, specifically the nRF52840 Dongle by Nordic Semiconductor [Sem]. The only data that was captured was on the Link Layer. The capture showed that an attempt was made by the watch to establish a connection. After that, no other data pertaining to the watch was captured.
The only usable information obtained from the capture about this communication was the fact that the passkey entry pairing mechanism was used. Both devices, the smartphone with the Withings app and the watch, had capabilities for input and for output. This means that the devices are able to display the numerical key and the user can enter the numerical key. This pairing mechanism makes the communication less vulnerable to Man-In-the-Middle attacks, as described in Section 2.4.

4.3 Threat model

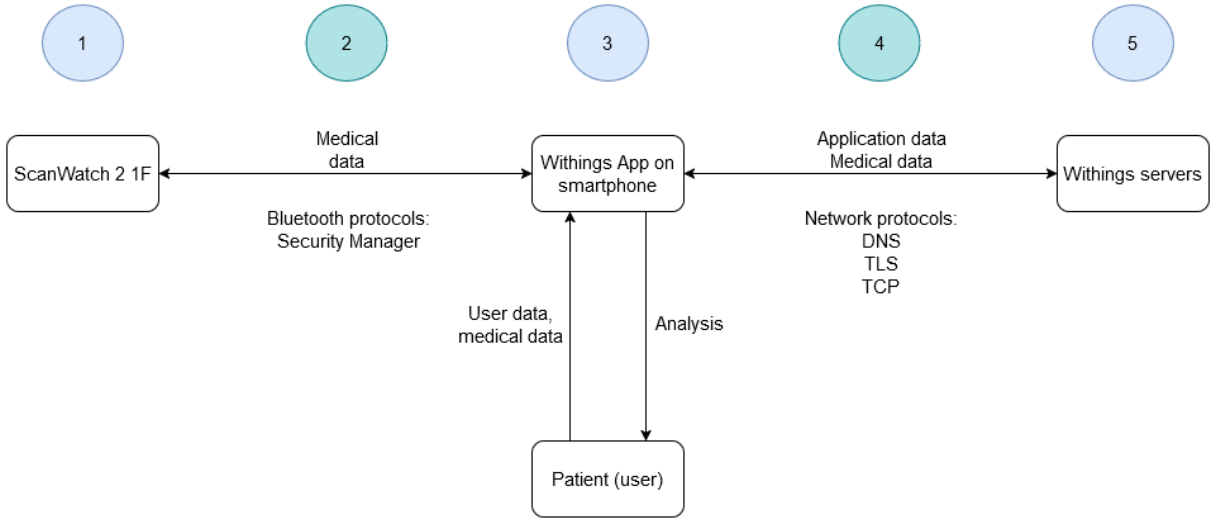

Figure 21: Graphical representation of system, with the relevant communication protocols and the relevant points of attack. The connection between the Scanwatch 2 1F and the Withings app could not be evaluated in this work, as explained in Section 4.2. In future work, a threat model based on the findings of a Bluetooth capture should provide more insights. Also, perhaps the evaluation of IT system structure on the server side could provide more cybersecurity perspectives to the cluster regarding IT.

With the knowledge obtained from the capture in the previous section the threat model of the Withings system can be constructed. Figure 21 shows the graphical representation of the system with the wearable, the Withings app and the Withings servers. For the connection between the Withings App and the servers, the relevant protocols in the communications are shown, which are DNS, TLS and TCP. The figure also indicates the points of exploitation for this connection. Point 1 indicates that the attack directly affects the smartwatch, point 2 indicates that the attack takes place during the Bluetooth communication between the watch and the smartphone and point 3 indicates that the attack directly affects the smartphone. Point 4 indicates that an attack takes place during the communication between the smartphone running the Withings app and the Withings servers and point 5 indicates that the attack affects the server directly. These points of exploitation are used to indicate where in the system the weakness to an attack is present. For example, in a side channel attack, any hardware can be the target, meaning the smartwatch (Point 1), the smartphone (Point 3) and the server (Point 5).

For the threat model regarding the communication between the smartphone and the Withings servers, the threats as described in Table 1 will be classified according to the STRIDE categories as described in Section 2.1. The types of attacks are: Spoofing, Tampering, Repudiation, Information disclosure, Denial of service and Elevation of privilege. Their definitions are:

- **S**poofing is masquerading as an authorised user and using their credentials to gain access to a system.
- **T**ampering is the modification of legitimate information, violating the integrity of the system or data.
- **R**epudiation is the denial of performing an action, evading accountability.
- **I**nformation disclosure is the unauthorised access to non-public, confidential information.
- **D**enial of service violates availability and disrupts the access to the system for legitimate users.
- **E**levation of privilege provides an attacker with higher privileges to the system than they should be allowed to, providing unauthorised access [KMLS17].

In addition to the STRIDE classification, the point of exploitation is indicated, along with whether the vulnerability to the attack is mitigated by the communication protocols in the Withings system.

Attack	**STRIDE**	**Point of Exploitation**	**Protected from by TLS**
Side channel attack	T	1, 3, 5	×
Physical attack	T	1, 3, 5	×
Downgrade attack	T	2, 4	✓
Brute force attack	R, I	2, 3, 4, 5	×
Eavesdropping/sniffing	I	2, 4	✓
Man-In-The-Middle attack	S, T, I	2, 4	✓
Jamming	D	2, 4	×
Replay attacks	T, R	2, 4	✓
Route attacks	T, D	4	×
Viruses	T, D	1, 3, 5	×

Table 2: Types of attacks on network communication that were defined in Table 1. The table lists the STRIDE category, the point in the communication displayed in Figure 21 where the attack is mounted (point of exploitation) and whether the Withings system is protected from the attack. The × indicates that the system is vulnerable to the attack and the ✓indicates that the system has a mechanism to protect itself from the attack.

The side channel attack and the physical attack require physical access. However, in their Security Insurance Plan, Withings [Wita] indicates that they have measures in place to physically protect their data centres. This work does not explore whether the strength of these measures is adequate, but it does provide the indication that it might be difficult to perform such attacks on the Withings servers.
Another caveat is that jamming attacks require proximity to the device utilising the network. TLS does not offer protection against these attacks, but it might still be difficult to mount such attacks, as they require specialised hardware.
Defence against network viruses requires measures such as intrusion detection systems, which is beyond the scope of this thesis. This could however be considered for future work, where

recommendations for IT infrastructure is enhanced with cybersecurity factors. The same is true for route attacks.
Another remark on the threat model is that TLS significantly reduces the weakness to brute-force attacks, when the implementation uses a strong cipher suite.
Finally, it should be noted that downgrade attacks can be mounted against a system if weak cipher suites are used in the TLS implementation. A downgrade attack can be prevented if weak cipher suites are not available to use in the system, so that an attacker cannot choose them. Since RSA for key exchange is considered "Phase out"by the NCSC, we expect the cipher suites using these to be removed in the near future. A few examples of these ciphersuites are listed in Figure 14 and Figure 15.

4.4 Cybersecurity aspects and regulations applied to the Withings app

The objective of performing the case study was to investigate how the authentication, encryption and integrity of the Withings app were ensured. Section 2 indicated several goals and properties of cybersecurity and Section 3 outlined regulations and directive concerning cybersecurity. In this section, the relation between these properties and directives are outlined. Table 3 Indicates for each of the aspects in Section 2 whether it was applicable to the Withings app. Table 4 indicates the same for the regulations.

Aspect of cybersecurity	Applied	Explanation
Confidentiality	✓	The scope of this thesis is to identify how the confidentiality in transit was ensured in the Withings app. When it is being transmitted, the application data is encrypted by the TLS protocol. The communication only uses cipher suites that are considered strong to ensure that unauthorised parties cannot interpret the messages that are being sent by and to the app. In particular, the use of the AES-256-GCM encryption algorithm, ensures confidentiality of each message.
Integrity	✓	The message integrity in the communication is also ensured by the AEAD algorithm AES-256-GCM. The algorithm can detect accidental and intentional modifications to the message through the authentication added to each encrypted message.
Availability	×	
Authentication	✓	The server authenticates itself by providing a certificate that is signed by a trusted Certificate Authority. With the certificate, the client can ensure that the public key that the server will provide is authentic.
Authorisation	×	
Accountability	×	
Symmetric encryption	✓	The symmetric encryption algorithm that is used in the TLS communication is the AES-256-GCM. The AES is state-of-the-art and an industry standard.
Hashing	✓	The hashing algorithm that is used in the TLS communication is SHA-384. It belongs to one of the secure hash algorithms according to NIST, indicating that it the hashing algorithm has pre-image resistance and collision resistance. Changes to the original message result in a different hash value, which indicates that the message has been altered.
Asymmetric encryption	✓	RSA and ECDHE are used in the TLS connection as the key exchange algorithms.

Table 3: Aspects of cybersecurity mentioned in Section 2, with an indication of whether they are applicable to the Withings app. ✓indicates that the Withings app has measures in place that contribute to the aspect and × indicates that this is not the case.

Availability, authorisation and accountability are not directly preserved by the security measures of the network protocols. However, these protocols can indirectly have an impact on these security goals. For example, TLS can contribute to accountability by providing a secure communication channel over which logging data can be transmitted. TLS can contribute to availability by defending against MITM or downgrade attacks that disrupt the communication.
Authorisation is the responsibility of the domain administrator, and the policies for subjects such as access rights cannot be inferred from the traffic analysis.

Directive	**Compliance**
GDPR	The Withings company is in possession of GDPR certification [Wita], indicating that it has provably implemented the appropriate technical and organisational measures to protect personal data from access by unauthorised parties.
MDR	Devices are required to have measures in place to ensure integrity, availability, and confidentiality of data. In table 3, it is outlined how the app ensures this. The software is required to adhere to state-of-the-art principles. This is reflected in the use of TLSv1.2 and higher and the implementation of secure cipher suites.
NIS2	The directive requires policies for cryptography. The network communication between the app and the servers implements the TLS protocol that handles encryption according to the state-of-the-art. In the event of an incident violating cybersecurity, Withings has a disaster recovery plan in place [Wita]. However, Multi Factor Authentication for the user is not implemented in the app.
NCSC	The TLS communication uses two cipher suites that are classified as Good or Sufficient, meaning the cipher suites are not considered weak. This is because the algorithms used for key exchange, bulk encryption, certificate verification, and hashing are classified under the categories Sufficient and Good. Furthermore, TLS versions 1.2 and 1.3 are used, which are Sufficient and Good.

Table 4: The compliance of the Withings app to the regulations outlined in Section 3.

Table 4 Indicates that for the regulations discussed in Section 3, the Withings app complies with the requirements outlined in each chapter.

5 Checklist

The objective of this thesis is to develop cybersecurity factors, that can be added to the HHQA-model, specifically to the cluster "eHealth application". The main focus for these factors was communication security. The factors that are listed in this section, were identified from the case study and threat model of the Withings app and from the rules and regulations concerning cybersecurity. The recommendations that the checklist outlines, are meant to give healthcare organisations an overview of what they should minimally require of an e-health app, in order to consider it secure in its (network) communication. By indicating how to identify a secure e-health app, the recommendations contribute to the quality of healthcare.

The following checklist was constructed.

The e-health application:

- ☐ **Uses encryption algorithms that have not been compromised in security**
- ☐ **Performs regular software updates**
- ☐ **Implements proper public key certificates**
- ☐ **Implements authentication mechanisms when pairing devices**
- ☐ **Implements multi-factor authentication**

Figure 22: Checklist containing cybersecurity factors to be added to the HHQA-model, specifically the cluster "eHealth application".

For the first recommendation, the constantly evolving nature of cybersecurity is considered. It is unlikely that a cryptographic algorithm will remain cryptographically secure forever. For example, as stated in Section 2.2.2, the SHA-1 hashing algorithm were proven to be insufficiently secure for encryption. It is important that e-health applications **do not use encryption algorithms that have been compromised in security**. This is the reason why it was important to confirm that in the network capture of the Withings app that the TLS version was the most recent and that the cipher suites used in the communication consisted of secure algorithms. The Withings servers were capable of the highest currently available version: TLSv1.3, which is the best version to use as of now, as mentioned in Section 3.4.

An indicator that an e-health app adapts to the most current state of cryptography, is that the app **performs regular software updates**. Software updates are important in order to protect against new vulnerabilities of the cryptographic algorithms and other aspects of the system.

When setting up the connections, the smartphone with the e-health app needs to trust both the server of the Withings app and the smartwatch. In order for the server to be trustworthy, the Withings app and server perform the TLS-handshake before setting up a protected TLS connection.

During this process, an important step is the verification of the public key certificate of the server by the client. As mentioned in Section 2.2.3, certificates are used for authentication of the public key. A certificate can be trusted if it is signed by a trusted Certificate Authority (CA). The use of such certificates indicate proper authentication and proper implementation in this area of cryptography. For example, the Withings server used the CA Let's encrypt, as mentioned in Section 4.1, which is a trusted organisation, used by many services worldwide. Therefore, the next recommendation is to require the servers of the app to **implement proper public key certificates**.
For the authentication of the wearable, the app and wearable should **implement authentication mechanisms when pairing devices**. Although the evaluation of the Bluetooth connection between the app and the wearable could not be properly studied, it was noted that the wearable was authenticated, using numerical comparison, which was explained in Section 2.4. This means that it is more difficult to mount attacks such as Man-In-The-Middle.
The final recommendation is mainly derived from the study on the rules and regulations discussed in Section 3. The NIS2-directive required the **implementation of multi-factor authentication** to improve security. This is a measure that ensures that only authorised people can have access to the personal information that is handled by an e-health app. Multi-factor authentication is the use of more than one piece of evidence to prove the users identity, such as entering a PIN along with entering a password.

6 Conclusions

The objective of this thesis is to expand the HHQA-model with factors that account for cybersecurity, with the focus on identifying the most important indicators of secure communication within systems that e-health applications use. These factors have been listed in a checklist, that is meant to improve the cluster "eHealth application" of the HHQA-model. The methods used to do this were a literature review of existing European and Dutch regulations and performing a case study on an e-health app that is currently on the market.
The rules and regulations require data to be protected using encryption and indicate that it is important that the confidentiality of personal data is protected through appropriate technical measures. The regulations also offered the standards for the Transport Layer Security (TLS) network protocol, which provides encryption and authentication when communicating over the internet. The regulations indicated the specific implementations that would result in the best security available.

The case study was performed on a wearable device that collects health data and its accompanying e-health app called Withings. The system consists of the wearable, the smartphone running the Withings app and the servers of Withings. The connection between the Withings app and the servers was evaluated by performing traffic analysis through capturing the communication. From what was captured in this communication, no obvious weaknesses were discovered. The communication abided by the current best practices. TLSv1.2 and TLSv1.3 were used in the communication, where cipher suites of sufficient strength were used. Aditionally, the public key certificate that was provided by the server was signed by a trusted third party. This means that for this specific app and method of evaluation, there were no other recommendations to improve the security in this area.

These findings were summarised in a threat model, that classified common attacks in the STRIDE categories and indicated whether the app was sufficiently protected from these attacks In addition, an overview was constructed of how the cybersecurity aspects discussed in Section 2 and the regulations discussed in Section 3 related to the e-health app from the case study in Section 4.4. The information obtained from the literature and the performed experiments were combined and formed the basis for the checklist in Figure 22.

The recommendations in this checklist are meant to give healthcare organisations an indication of what they should require from a secure healthcare application. The checklist focuses on multiple aspects that were explored in this thesis, such as cryptography. If healthcare organisations do not implement these recommendations, they leave their IT systems open to vulnerabilities. If these vulnerabilities are exploited in a cyberattack, it causes operational disruptions, data breaches and patient safety to be negatively affected, which compromises the quality of care.

Future work should explore the (Bluetooth) communication between the wearable and the e-health app. This was not feasible for this project, as explained in Section 4.2, but could complete the checklist constructed in this work. This research could explore different BLE sniffers and/or different network sniffing software.
Furthermore, the thesis focused on the HHQA-cluster "e-health application", but the IT infras-

tructure and systems also play a significant role in cybersecurity, from implementing firewalls to training employees about safely operating within an IT system. Therefore, the ways in which the "*Information Technology infrastructure and systems*" cluster can be expanded to reflect cybersecurity measures should be explored, perhaps through a similar case study to the one performed in this work.

References

[Blu] Bluetooth. Specification of the bluetooth system. https://www.bluetooth.com/wp-content/uploads/Files/Specification/HTML/Core-54/out/en/consolidated-table-of-contents---compliance-requirements.html.

[Buc17] W. Buchanan. *Cryptography*. River Publishers series in security and digital forensics. River Publishers, 1st edition, 2017.

[Croa] Crowdstrike. What is a Brute Force Attack? https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/brute-force-attack/.

[Crob] Crowdstrike. What is a Downgrade Attack? https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/downgrade-attack/.

[CTHY14] S. Chan, J. Torous, L. Hinton, and P. Yellowlees. Mobile Tele-Mental Health: Increasing Applications and a Move to Hybrid Models of Care. *Healthcare*, 2(2):220–233, May 2014.

[CZ17] L. Chi and X. Zhu. Hashing techniques. *ACM Computing Surveys*, 50(1):1–36, April 2017. https://doi.org/10.1145/3047307.

[Dea17] D. Death. *Information Security Handbook*. Packt Publishing, 2017.

[Dir23] The NIS2 Directive. Nis2 requirements 10 minimum measures to address, 2023. https://nis2directive.eu/nis2-requirements/.

[DJT+22] N. Dutta, N. Jadav, S. Tanwar, H. Sarma, and Pricop E. *Cyber security: issues and current trends*. Studies in Computational Intelligence ; Volume 995. Springer, 2022.

[DK06] L. Dostálek and A. Kabelová. *DNS in action*. Packt Pub Limited, 2006.

[DN13] S. Das and T. Nayak. Impact of cybercrime: Issues and challenges. *International journal of engineering sciences & Emerging technologies*, 6(2):142–153, 2013.

[Don89] A. Donabedian. The quality of care: how can it be assessed? *JAMA*, 261(8):1151–1152, February 1989. https://doi.org/10.1001/jama.261.8.1151.

[Dwo07] M. J. Dworkin. Recommendation for block cipher modes of operation, January 2007. https://csrc.nist.gov/pubs/sp/800/38/d/final.

[enc] Let's encrypt. Let's encrypt. https://letsencrypt.org/.

[Eys01] G. Eysenbach. What is e-health? *Journal of Medical Internet Research*, 3(2):e20, June 2001. https://doi.org/10.2196/jmir.3.2.e20.

[GDP19] GDPR. General Data Protection Regulation (GDPR) Compliance Guidelines, February 2019. https://gdpr.eu/.

[Gor17] W. Goralski. *The illustrated network: how TCP/IP works in a modern network*. Morgan Kaufmann Publishers, Cambridge, Massachusetts, second edition, 2017.

[Gup16] N. Gupta. *Inside Bluetooth low energy*. Artech House mobile communciations series. Artech House, London, second edition, 2016.

[HI] Health and Youth Care Inspectorate. eHealth. https://english.igj.nl/medical-technology/ehealth.

[HTC22] Ltd. Huawei Technologies Co. *Data Communications and Network Technologies*. Springer, 2022.

[IBM] IBM. What is mfa? https://www.ibm.com/topics/multi-factor-authentication.

[KMLS17] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer. STRIDE-based threat modeling for cyber-physical systems. *IEEE*, September 2017. https://doi.org/10.1109/isgteurope.2017.8260283.

[KR25] M. Kokila and Srinivasa Reddy K. Authentication, access control and scalability models in internet of things security–a review. *Cyber Security and Applications*, 3:100057, 2025. https://www.sciencedirect.com/science/article/pii/S2772918424000237.

[Lei24] Leiden University Medical Center (LUMC). The BOX, 2024. https://www.lumc.nl/patientenzorg/specialistische-centra/hart-long-centrum/voor-patienten/the-box/.

[Los03] P. Loshin. *TCP/IP clearly explained*. The Morgan Kaufmann Series in Networking. Morgan Kaufmann Publishers, Amsterdam, 4th ed. edition, 2003.

[McG08] D. McGrew. An Interface and Algorithms for Authenticated Encryption, January 2008. https://www.rfc-editor.org/rfc/rfc5116#section-5.

[mdr] Regulation - 2017/745 - EN - Medical Device Regulation - EUR-LEX. https://eur-lex.europa.eu/eli/reg/2017/745/oj/eng.

[Minen] Ministerie van Economische Zaken. Wet beveiliging netwerk- en informatiesystemen, 2024. https://www.rdi.nl/onderwerpen/wet-beveiliging-netwerk–en-informatiesystemen.

[MP08] S. Mrdovic and B. Perunicic. Kerckhoffs' principle for intrusion detection. *Networks 2008-The 13th International Telecommunications Network Strategy and Planning Symposium*, pages 1–8, 2008.

[Nat23] Nationaal Cybersecurity Centrum (NCSC). Ict-beveiligingsrichtlijnen voor transport layer security v2.1 (tls), 10 2023. https://www.ncsc.nl/documenten/publicaties/2021/januari/19/ict-beveiligingsrichtlijnen-voor-transport-layer-security-2.1.

[NSI+15] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, and C. Gomez. IPv6 over BLUETOOTH(R) Low Energy. RFC 7668, October 2015. https://www.rfc-editor.org/info/rfc7668.

[Opp16] R. Oppliger. *SSL and TLS : theory and practice*. Artech House Information Security and Privacy Series. Artech House, Norwood, Massachusetts, second edition, 2016.

[Pit19] P. P. Pittalia. A comparative study of hash algorithms in cryptography. *International Journal of Computer Science and Mobile Computing*, 8(6):147–152, 2019.

[PP11] C. Paar and J. Pelzl. *Understanding cryptography*. Springer, October 2011.

[RIV] RIVM. E-health (digitale zorg). https://www.rivm.nl/e-health.

[RS18] S. Rubinstein-Salzedo. *Cryptography*. Springer, October 2018.

[Sem] Nordic Semiconductor. NRF52840 Dongle. https://www.nordicsemi.com/Products/Development-hardware/nRF52840-Dongle.

[Sil03] J. E. Silva. An overview of cryptographic hash functions and their uses. *GIAC*, 6, 2003.

[TA21] N. Thamer and R. Alubady. A survey of ransomware attacks for healthcare systems: Risks, challenges, solutions and opportunity of research. In *2021 1st Babylon International Conference on Information Technology and Science (BICITS)*, pages 210–216, 2021.

[TSKR+22] R. Tossaint-Schoenmakers, M. J. Kasteleyn, A. Rauwerdink, N. Chavannes, S. Willems, and E. P. W. A. Talboom-Kamp. Development of a quality management model and self-assessment questionnaire for hybrid health care: Concept mapping study. *JMIR Form Res*, 6(7):e38683, July 2022.

[VO21] P. C. Van Oorschot. *Computer Security and the Internet*. Springer, 11 2021.

[Wir] Wireshark. Wireshark · Go Deep. https://www.wireshark.org/.

[Wita] Withings. Medical cloud protection and security insurance plan — Withings. https://www.withings.com/us/en/data-security#sub-pdcs.

[Witb] Withings. Scanwatch 2 withings. https://www.withings.com/nl/en/scanwatch-2.

[WK15] J. Wang and Z. A. Kissel. *Introduction to network security: theory and practice*. Wiley, Singapore, second edition, 2015.

[Zha21] N. Zhang. Research on the application of data encryption technology based on network security maintenance in computer network security. *Journal of Physics: Conference Series*, 1744(2):022060, February 2021. https://dx.doi.org/10.1088/1742-6596/1744/2/022060.

[ZP03] Y. Zuo and B. Panda. Network viruses: their working principles and marriages with hacking programs. *IEEE Systems, Man and Cybernetics SocietyInformation Assurance Workshop, 2003.*, pages 306–307, 2003.

A HHQA-model

Appendix A on the next page shows all the factors that were included in the HHQA-model and the clusters they were grouped in. Figure 24 shows an overview of the clusters in relation to the Donabedian Structure-Process-Outcome model. "Structure" refers to the available resources and organisational structure. A good structure increases the likelihood of a good process. The "Process" refers to what is done when giving and receiving care by practitioners and patients respectively. A good process increases the likelihood of a good outcome. The "Outcome" refers to the effects of the healthcare on the patient [Don89].
As can be seen below, the cluster e-health application has only one factor, which does not account for the requirements for cybersecurity. Section 1 described the need for adequate cybersecurity and Section 3 described that it was required for organisations to implement adequate technology, in order to protect personal data. This is not reflected in the model, so this work has aimed to provide the cybersecurity perspective and add to the model in the form of a checklist.

Vision, strategy, and organization

- Support the implementation and development of eHealth in the organization with good project management.
- Mobilizing funding for working with eHealth.
- Clear internal policies regarding the use of eHealth.
- Vision supported by the line, "Why are we doing this?".
- Care delivery with eHealth complies with laws and regulations.
- Financial reimbursements for eHealth deployment.
- Redesign the current work process and review what contributes to the desired care outcomes.

eHealth application

- The eHealth application is user-friendly.

Information Technology infrastructure and systems

- Information technology architecture available within the health care organization.
- Back-up scenario during technical problems

Providing support toward health care professionals

- Health care professionals have easy access to information technology resources; for example, device, internet, screen, or headset.
- Embedding eHealth in the daily practice of health care professionals.
- Training and supervision for health care professionals.
- Help desk for health care professionals.
- Information on the treatment with eHealth is clear and accessible to the health care professional.

Learning system: evaluation and improvement

- Cocreation: eHealth is developed, implemented and redeveloped with different stakeholders.
- Monitoring and evaluation of service and treatment results.

Attentiveness to the patient

- Clear communication to the patient about how care is offered.
- Personalized care, considering patient needs with regard to (deployment of) eHealth.
- The patient has easy access to the necessary information technology resources; for example, device, Internet, and so on.
- Patients receive practical support in using the eHealth application; for example, a help desk.
- The patient has confidence in the eHealth application.
- The patient has the flexibility to use eHealth wherever and whenever it is convenient.

Skills, knowledge, and attitude of health care professionals

- Good balance between face to face and eHealth for the health care professional.
- The health care professional has confidence in the eHealth application.
- The health care professional is satisfied with working with eHealth.

End results for the patient

- The patient can integrate the use of eHealth in their daily life.
- Treatment with eHealth has a positive influence on the patient's health.
- Treatment with eHealth contributes to the patient's self-reliance.
- The patient is satisfied.
- The patient has easy access to care.
- eHealth provides logistical convenience for the patient.
- eHealth has added value for the patient.

Figure 23: Included factors of the HHQA-model. Adapted from Tossaint-Schoenmakers et al. [TSKR+22].

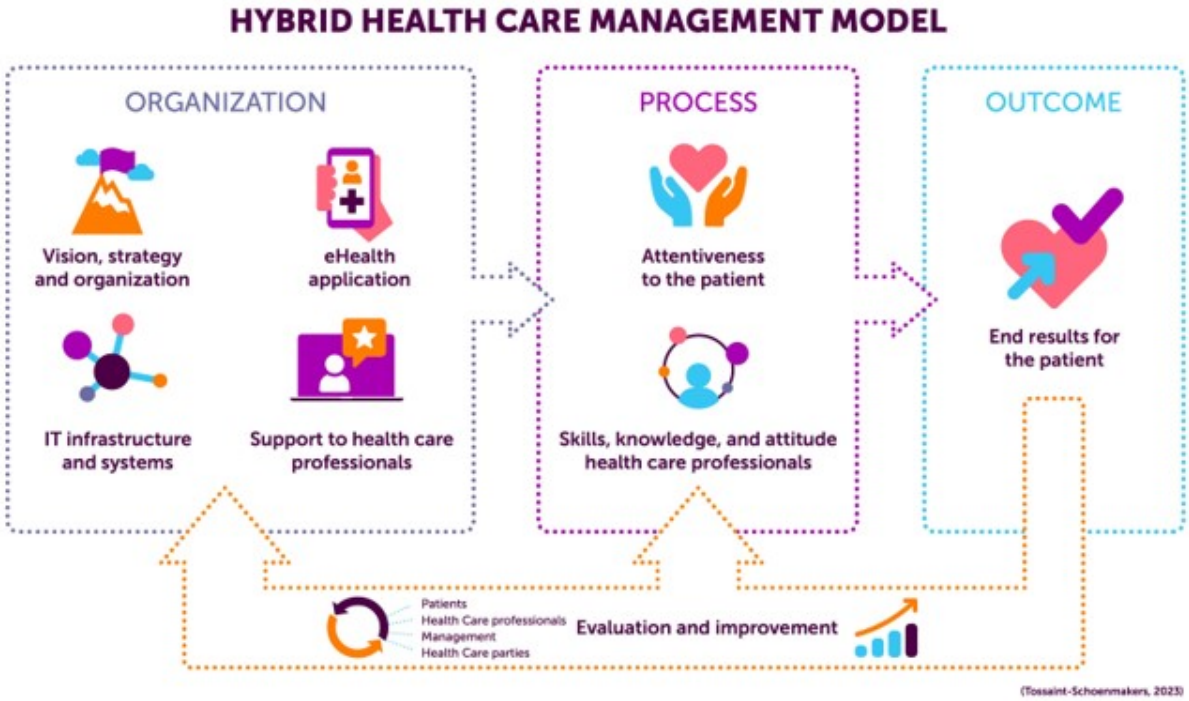

Figure 24: The clusters of the HHQA-model. By Tossaint-Schoenmakers et al. [TSKR+22]. The clusters pertain to the structure of the organisation, the requirements for the process of delivering care to the patients through e-health and what benefit e-health has on the patients. All these clusters are meant to be continuously evaluated and improved, with the help of the self-assessment questionnaire.

B Wireshark

Wireshark was used to capture and analyse packets that were sent over the network. The program was downloaded from the Wireshark website [Wir] and required no additional settings to be selected. For the experiments, measurements such as steps were taken by the watch and the app was opened on the smartphone. Since it was connected with a WiFi hotspot to the device on which Wireshark was installed, Wireshark captured the communication. The analysis in the experiments was done by filtering out the necessary information. These filters are listed below. The filters are case sensitive.

Filter	**Function**
dns & frame contains "withings"	Filters out the DNS queries to any Withings server
(ip.addr == x || ip.addr == y || ip.addr == z)	Filters out packets sent to any of the three IPv4 addresses x, y, and z. The filters below are used in combination with this filter by adding two ampersands (&) in between.
frame.number ¿= a	Shows the frames captured from a certain point. This was used to determine if with every connection to a server a TCP handshake was performed.
!tcp	Shows all packets not sent over TCP
_ws.col.protocol != "TLSv1.3"	Shows all packets sent over a TLS version lower than 1.3
tls.handshake	Shows the TLS handshake protocol messages
x509sat.printableString == "x"	Shows the CA that signed the server's public key certificate
tls.handshake.ciphersuite == 0x000	Filters the chosen cipher suite

Table 5: Network packet filters and their functions

C Bluetooth

Physical Layer

The **Physical Layer** is the lowest layer. It is responsible for sending and receiving data packets over the air through radio signals. In order to prevent interference from other Bluetooth devices, frequency hopping is used. The connected devices change the radio frequency of the connection according to a predefined pattern.

Link Layer

The second layer, the **Link layer**, provides the access to the wireless connection. It establishes connections between devices, checks for and handles errors and manages data flow. Error checking is done using the Cyclic Redundancy Check (CRC), which is a checksum. CRC checks for bit-errors and is especially useful in noisy environments with a higher chance of multiple bit errors.
With BLE, a device needs to have either a Public Address, a Random Address or both. A Public Device Address is a globally unique device address, comparable to an Ethernet MAC address. The address is assigned to the controller. A Random Address provides a privacy feature. The real address of the device can be hidden by the use of a random address that changes over time, in order to prevent tracking. Only devices that have been authenticated can map the random address to the real address
The **Direct Test Mode** is on the same layer as the link layer, but only used for testing and does not play a role in the actual communication.

Host Controller Interface

The Host Controller Interface enables communication between the lower layers in the controller and the higher layers in the host. The HCI is necessary when the controller is present on a separate Bluetooth controller chip.

L2CAP

The next layer is the **Logical Link Control and Adaptation Protocol (L2CAP)**. It handles channel multiplexing, which means that this layer routes the connection to the appropriate channel in the layer above it. It also handles segmentation and reassembly of large data packets.

Attribute Protocol and General Attribute Protocol

The Attribute Protocol (ATT) enables devices to discover, read and write attributes, which is a representation of any data, such as speed or size. The attributes have types, handles and access permissions, which indicate whether a device requesting the data has permission to read or write to the attribute.
The General Attribute Protocol (GATT) defines the client and server roles, since the ATT protocol is a client-server model. It can define permissions for groups of attributes, services and characteristics, which make up a service. This makes it easier to manage attributes. A GATT profile manages one or multiple services, for example a service for keeping track of body measurements, such as heart rate and temperature.

Generic Access Profile

The Generic Access Profile (GAP) defines the procedures related to discovery, connecting and

security that all Bluetooth devices have in common, for the sake of interoperability. It also defines the Bluetooth Device Name that is visible to the end user. [NSI+15], [Blu], [Gup16].