University of Zurich[UZH]

MASTER THESIS – Communication Systems Group, Prof. Dr. Burkhard Stiller

Design and Evaluation of Ultra-Wideband (UWB) Architectures with a Focus on Privacy-Preserving Characteristics

Charlotte Eder
Zürich, Switzerland
Student ID: 17-826-090

Supervisor: Katharina Müller
Date of Submission: July 24, 2023

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

ifi

Master Thesis
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
URL: http://www.csg.uzh.ch/

Universität Zürich[UZH]

Institut für Informatik

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel (inklusive generativer KI wie z.B. ChatGPT) angefertigt habe. Mir ist bekannt, dass ich die volle Verantwortung für die Wissenschaftlichkeit des vorgelegten Textes selbst übernehme, auch wenn (nach schriftlicher Absprache mit der betreuenden Professorin resp. dem betreuenden Professor) KI-Hilfsmittel eingesetzt und deklariert wurden. Alle Stellen, die wörtlich oder sinngemäss aus veröffentlichten oder nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Zürich, 24.07.2023

Unterschrift Student:in

Abstract

Ultra-Wideband (UWB) technology has gained significant popularity in indoor localization applications. These applications often generate vast amounts of personal information, increasing the need to ensure compliance with privacy-preserving principles to safeguard user data. In this thesis, the privacy-preserving characteristics of several UWB localization architectures were analyzed. Firstly, UWB localization architectures were examined based on their privacy-preserving characteristics. Subsequently, two versions of a time difference of arrival (TDOA) localization system were implemented, including privacy best practices provided by the IEEE 802.15.4 standard during the implementation process. Additionally, the privacy-preserving characteristics of the implemented UWB localization systems were evaluated with the help of a privacy criteria catalog based on COPri V.2 ontology [1].

This thesis found that the localization system employing a passively listening tag fulfills seven out of eight privacy criteria. In contrast, the system where the tag actively sends out UWB signals only fulfilled three out of eight criteria in its minimal version. However, the privacy-preserving characteristics of the active system could be greatly improved by using tools such as dynamic addressing, encrypting packages containing personal information, using a message integrity code (MIC), and using a scrambled time sequence (STS). Finally, the limitations of the current systems' implementations are addressed which provides directions for future research.

Zusammenfassung

Die Ultrabreitband (UWB)-Technologie erfreut sich bei Indoor Lokalisationsanwendungen immer grösserer Beliebtheit. Diese Anwendungen erzeugen grosse Mengen an personenbezogenen Informationen, wodurch die Notwendigkeit wächst, den Datenschutz zu gewährleisten. In dieser Arbeit wurden die datenschutzfreundlichen Eigenschaften verschiedener UWB-Lokalisierungsarchitekturen analysiert. Zunächst wurden UWB-Lokalisierungsarchitekturen auf ihre datenschutzfreundlichen Eigenschaften hin untersucht. Anschließend wurden zwei Versionen eines Lokalisierungssystems mit time difference of arrival (TDOA) Architektur implementiert, wobei die vom IEEE 802.15.4 Standard zur Verfügung gestellten Werkzeuge zur Privatsphäreerhaltung in den Implementierungsprozess miteinbezogen wurden. Danach wurden die datenschutzfreundlichen Eigenschaften der UWB-Lokalisierungssysteme mit Hilfe eines auf der COPri V.2 Ontologie [1] basierenden Kriterienkatalogs bewertet.

In dieser Arbeit wurde festgestellt, dass das Lokalisierungssystem mit einem passiv zuhörenden Tag sieben von acht Datenschutzkriterien erfüllt und damit das System, bei dem der Tag aktiv UWB-Signale aussendet, übertrifft, welches in seiner Minimalversion nur drei von acht Kriterien erfüllt. Die datenschutzfreundlichen Eigenschaften des aktiven Systems können jedoch durch die Verwendung von Tools wie dynamische Addressen, die Verschlüsselung von Paketen mit persönlichen Informationen, die Verwendung eines Message Integrity Codes (MIC) und die Verwendung einer Scrambeled Time Sequence (STS) deutlich verbessert werden. Abschließend werden die Limitationen der aktuellen Systeme erwähnt, die Richtung für zukünftige Forschung vorgeben.

Contents

1

Introduction

UWB technology is a type of radio communication technology with excellent timestamping capabilities [2]. For this reason, it has established itself as one of the go-to technologies for all sorts of indoor localization applications. What further propelled the advance of UWB technology was the decision of phone companies such as Apple and Samsung to integrate UWB antennas into their flagship smartphones [3; 4]. Furthermore, car manufacturers have recently begun exploring the integration of UWB technology into devices such as car keys [5; 6].

As UWB applications have begun to spread into all domains of life [7], the amount of applications that generate and process personal information has also increased. This causes potential privacy concerns. It is therefore important to investigate the level of privacy protection of current UWB solutions to find potential weaknesses and to help improve these.

One tool to identify privacy-related weaknesses is to perform a privacy analysis. However, when screening the current body of research, it became evident that the usage of UWB technology itself is often regarded as a sufficient privacy-preserving mechanism [8; 9; 10; 11]. While this assertion holds true on a basic level, several other works focusing on other communication technologies showcased that also the architecture of such systems and the mechanisms implemented in them can have a significant impact on privacy [12; 13; 14; 15; 16]. Hence, there is a gap in research focusing on UWB privacy, and more specifically its architectural level.

A second important aspect that arises from the current body of research is that few works actually provide a framework for performing a privacy analysis on an architectural level. There is a multitude of works that give a general overview of privacy requirements or high-level analysis [17; 12; 18; 19]. There are also works that implement privacy-enhancing features on a network level [13; 14; 20; 21]. However, these works focus on one concrete feature such as encryption or dynamic addressing, and do not perform a privacy analysis or implement privacy improvements. Therefore, there is a need for works that further adapt privacy analysis frameworks so that they can be used on a network level.

Based on the gaps mentioned above, three research questions were deduced. The first research question asks, whether or not architectural choices concerning the UWB system have an effect on privacy. If this is the case, the second question focuses on what kind of effects such architectural choices have. Lastly, it should also be checked if and what

tools are provided in the institute of electrical and electronics engineers (IEEE) 802.15.4 standard that can help improve the privacy of a UWB localization system.

To answer these questions, two UWB localization systems with different architectures are designed and constructed over the course of this work. During the design process, the privacy tools provided by the IEEE 802.15.4 standard are evaluated and implemented where possible. Additionally, a network-level privacy analysis framework is derived based on the work of [1].

1.1 Structure

This work is structured into eight parts. Chapter 1 gives an introduction to the topic of UWB technology and its applications. Chapter 1 also highlights the current research gap when it comes to the privacy analysis of UWB localization applications on an architectural level. Furthermore, Chapter 1 introduces the research questions that are answered within this work.

Chapter 2 gives an overview of UWB technology and the standards that lay the foundation for the usage of the technology. In particular, this section focuses on techniques that make confidential and authenticated UWB communication possible. Additionally, different UWB localization architectures are introduced. On the other hand, an overview of the topic of privacy is given and a privacy ontology is presented, based on which the built UWB localization systems will be analyzed [1]. There is also Chapter 3 which gives an overview of the related work in the domain of UWB localization architectures, papers focusing on UWB and privacy as well as privacy-related papers that are useful for this thesis.

In Chapter 4, the structure used for the privacy analysis is described. This includes the use case based on which the localization systems will be designed, and the selection of the privacy evaluation criteria. Next, in Chapter 5, the design decision based on which the UWB localization systems were built is presented. Additionally, the design of the constructed localization systems is showcased in detail.

After that, in Chapter 6, the designed systems are analyzed on a technical level. After that, the core part of this work, the privacy analysis is performed. In Chapter 7, the results are summarized and further discussed in the context of the background. Chapter 7 also highlights the limitations of this work. Chapter 8 concludes the work by summarizing the key findings and giving starting points for future research directions.

2

Background

This Section gives an overview of UWB technology and its advantages. It also introduces the standards and regulations that guide the usage of UWB technology. In particular, there is a focus on the IEEE 802.15.4 standard which defines the frame structure on the Physical (PHY) and medium access control (MAC) layer for UWB packages. Additionally, the standard also provides tools for security and privacy.

After that, the next Section gives an overview of the different UWB localization approaches. This Chapter also gives an introduction to the topic of information privacy and presents COPri V2, a core ontology for privacy requirements upon which the privacy analysis of this work builds.

2.1 Introduction to UWB

As defined by the federal communication commission (FCC), UWB is a technology used for short-range radio communication. It either uses a fractional bandwidth of larger than 20% or an absolute bandwidth of at least 500 MHz [22]. As shown in Figure 2.1, UWB occupies a much wider frequency band compared to other communication technologies such as Bluetooth, the global positioning system (GPS), or wireless fidelity systems (WI-FI). Moreover, the energy level used for communication by UWB is lower than the average energy level used by other technologies.

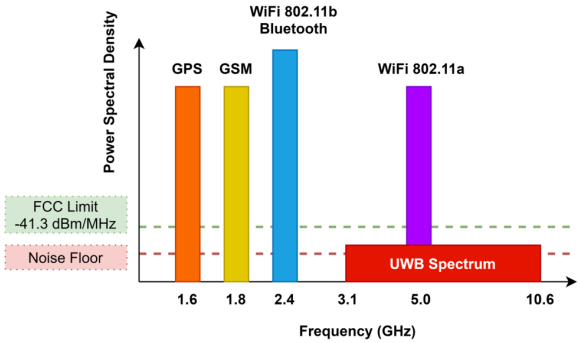

Figure 2.1: Frequency Spectrum of Several Wireless Communication Technologies [23].

A type of UWB technology that has gained a lot of traction during the past years is UWB impulse radio (IR). As described by [24], each information symbol is transmitted by multiple pulses in an IR system. The information itself is encoded by the position or polarity of the pulses. In addition, each pulse resides inside a time interval called a frame. A time-hopping code determines the position of the pulses inside the frames to reduce the probability of collisions with other pulses. As an example, a possible pulse encoding of the signal (1, -1, 1) is visualized in Figure 2.2. Each information bit is represented by two pulses. For the first bit, the time-hopping codes are (2, 1). Therefore, the first pulse in the first frame is shifted by two pulse lengths and the second pulse is shifted by one pulse length. The shift length is also called chip interval T_c. In this example, 1 and -1 are conveyed by the polarities of the pulses.

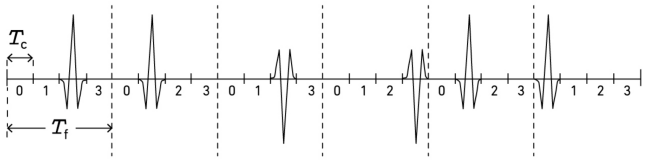

Figure 2.2: (1, -1, 1) Encoded as an Impulse Radio Signal. The position of the pulses is defined by the time-hopping codes 2, 1, 2, 3, 1, 0. T_c represents the chip interval and T_f the frame length [24].

The usage of UWB technology has multiple advantages. First of all, the large bandwidth allows UWB to support a high channel capacity, which in turn allows for low communication transmission power. This avoids narrowband interference with other wireless technologies [25]. Next, especially IR UWB signals have a short life-time which gives signals a high time-resolution [26]. This means that a receiver can very accurately determine the time of arrival of a signal, allowing for the construction of time of arrival (TOA), time difference of arrival (TDOA) and two-way ranging positioning systems with centimeter accuracy [25]. Moreover, the high time-resolution of UWB also strengthens the technology against signal fading and multipath interference [26]. Lastly, UWB systems are suitable for high-speed communication due to their high bandwidth [24]. Another characteristic of UWB technology is that it can use low carrier frequencies. This makes UWB signals pass through obstacles more easily [26]. Lastly, UWB's signal shape is similar to noise which reduces the chance of eavesdropping [26].

2.2 UWB Regulations and Standards

The first worldwide regulation that allowed a limited usage of UWB devices was published by the FCC in 2002 for the United States of America (USA) [24]. The focus of the regulation was to prevent interference with other communication technologies. This entailed setting the lower bound of allowed frequencies for UWB to 3.1 gigahertz (GHz) and limiting the maximum mean power spectral density of the signal to -41.3

decibel-milliwatts per megahertz (dBm/MHz), the part 15 limit for unintentional radiators. After the authorization of UWB in the USA, other countries introduced their own regulations for UWB. However, these regulations are not congruent on a global level with each other which makes it difficult to introduce UWB devices that conform with all regulations worldwide [24].

Based on these regulations, several UWB standards were introduced in the following years. These standards can be organized based on the layer of the open system interconnection (OSI) model they target. The most important UWB standards were organized this way by [25] in Figure 2.3. The foundation of UWB communication on the PHY and MAC layer is given two standards of the institute of electrical and electronics engineers (IEEE), the IEEE 802.15.4a [27] and the IEEE 802.15.4z [28] standard. On top of these standards, application-specific standards have been designed by several other organizations. This includes standards such Apple's nearby interaction protocol for third-party devices [29] or the digital key 3.0 of the car connectivity consortium [30].

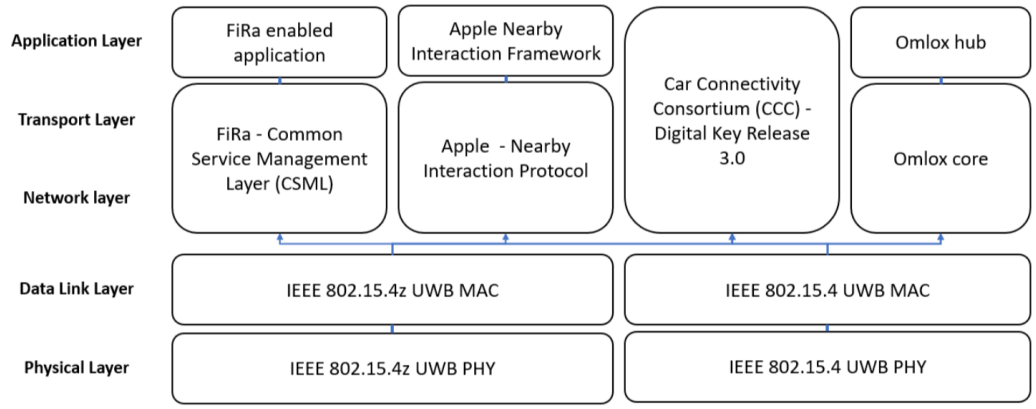

Figure 2.3: UWB Standards Inside the OSI Reference Stack [25].

2.2.1 IEEE

UWB standardization efforts started in the early 2000s by two IEEE task forces [24]. The IEEE 802.15.3a task force worked on an amendment for the PHY layer based on the IEEE 802.15.3 high-rate wireless personal area network (WPAN) standard. However, the standard was never finalized and the task force was dissolved in 2006.

The other task force worked on an amendment of the IEEE 802.15.4 standard for low-rate WPANs. The first official version of the UWB amendment was introduced in 2007 under the name IEEE 802.15.4a. The 802.15.4a standard specifies two optional signaling formats for IR-UWB and for the chirp spread spectrum UWB [24]. In addition, optional ranging procedures are described for IR-UWB. After its introduction in 2007, the 802.15.4a amendment was incorporated into the main body of the standard in 2011. A second addition to the standard was published in 2012. The IEEE 802.15.4f-2012

amendment introduced additional specifications for a high-rate (HRP) and low-rate pulse (LRP) mode. This amendment was incorporated into the main body of the standard in 2015. After that, fewer smaller, non-UWB-related amendments were added to the standard [31]. For the remainder of this work, IEEE 802.15.4a will refer to the UWB-specific sections of the 2015 version of the IEEE 802.15.4 standard [27]. When mentioning the IEEE 802.15.4 standard itself, the 2020 version of the standard is referred to [32].

The newest UWB-related amendment to the IEEE 802.15.4 standard was added in 2020. The IEEE 802.15.4z amendment focuses on functionalities for increased integrity and accuracy. The evolution of the IEEE 802.15.4 standard and its most important additions are summarized in Figure 2.4.

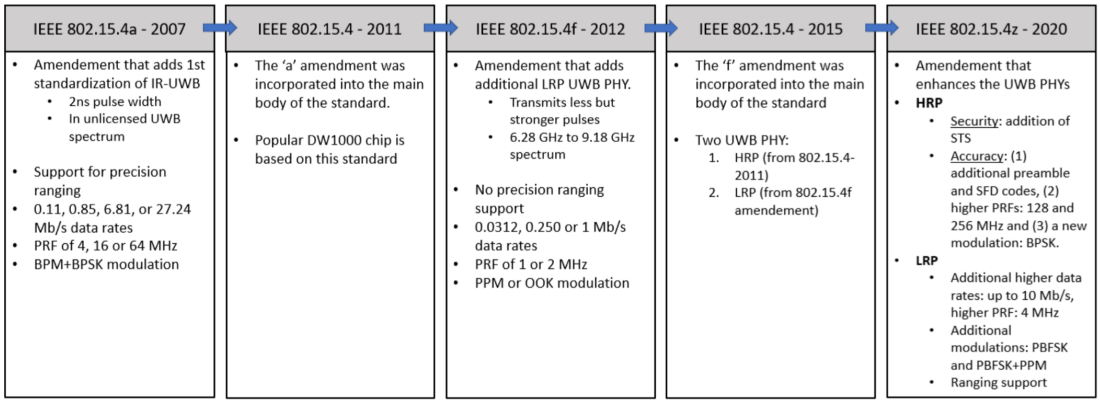

Figure 2.4: Overview of the Evolution of the UWB Amendments for the IEEE 802.15.4 Standard [25].

The IEEE 802.15.4 standard is related to UWB ranging and localization applications. However, it is important to note that there are also several IEEE standards for other application domains of UWB. For example, the IEEE 802.15.6 standard regulates the usage of UWB in applications close to, and inside the human body. The IEEE 802.15.8 standard defines mechanisms that enable peer-aware communications that can be used for applications such as social networking, advertisement or gaming [25].

2.2.2 FiRa and Further UWB Standards

Due to the increased interest in UWB fine-ranging applications, the fine-ranging consortium (FiRa) was brought to life in 2019 [33]. This industry consortium aims to provide solutions to the current interoperability challenges of the UWB ecosystem. On one hand, FiRa provides specifications on top of the IEEE 802.15.4 standard such as the common service management layer specification [34]. This specification enables interoperability among FiRa devices. It also serves as a framework for deploying UWB service applications. The exact FiRa specifications are only available to FiRa members. On the other hand, FiRa provides a certification process including a FiRa label to help the general public identify FiRa compatible devices [35].

Apart from FiRa, other organizations released UWB standards and frameworks on top of the IEEE 802.15.4 standard. This includes Apple with the nearby interaction protocol for third-party devices [29], the car connectivity consortium with its digital key 3.0 standard for location-aware smartphone-to-car connectivity [30], or the open-source omlox standard for real-time location services (RTLS) [36]. A summary of other UWB standards and their evolution can be found in [25].

2.3 UWB Frame Structure

The IEEE 802.15.a standard defines the following frame structure [27]: synchronization (SYNC) field, start-of-frame delimiter (SFD) field, physical header (PHR) field, and PHY payload as can seen in Figure 2.5. The SYNC header marks the beginning of a package and helps synchronize devices communicating with each other. The SYNC header is constructed by using one of the predefined preamble codes from the IEEE 802.15.4a standard. For packages following the 802.15.4a standard, the preamble codes are either of length 31 or 127. For the 802.15.4z standard, there are predefined preamble codes of lengths 91 and 127. The preamble codes are drawn from the ternary alphabet {–1,0,1}.

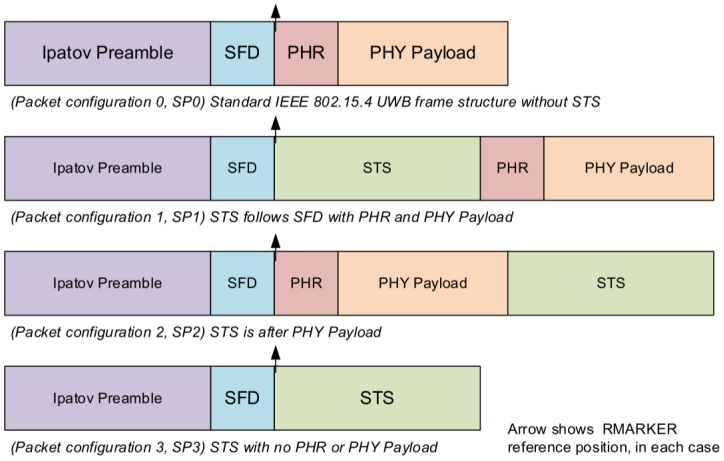

Figure 2.5: UWB Frame Structures. Mode 0 was defined in the IEEE 802.15.4a standard, Mode 1 to 3 were added in the IEEE 802.15.4z amendment [37].

The SFD field indicates the end of the SYNC field and the exact start of the PHR [27]. For the 802.15.4a standard, there is a short SFD code of length 8 and a long SFD code of length 64. These codes also use the ternary alphabet {–1,0,1}. The IEEE 802.15.4z standard drops support for the ternary code with a length of 64 and defines four new codes of length 4, 8, 16, and 32. The new codes use a binary system which shortens the code and makes ranging more accurate [28].

The PHR contains information about the payload that is transmitted [27]. For the 802.15.4a standard, the PHR contains information such as the data rate, the frame length, whether the package is used for ranging or not, the preamble duration and the single error correct double error detect SECDED field (see Figure 2.6) [27].

Bits: 0–1	**2–8**	**9**	**10**	**11–12**	**13–18**
Data Rate	Frame Length	Ranging	Reserved	Preamble Duration	SECDED

Figure 2.6: PHR Field Format [27].

Lastly, the scrambled timestamp sequence (STS) field was introduced in the IEEE 802.15.4z amendment. It can be placed at three different positions as shown in Figure 2.5. As the amount of possible preamble codes is limited, they are repeated multiple times in the SYNC field which can be a starting point for attacks [38]. In order to prevent these attacks, the STS field is added. It consists of a sequence of pseudo-randomized pulses generated by a deterministic random bit generator that uses a 28-bit key and a 128-bit nonce [25; 28]. Each zero bit produces a pulse with positive polarity and each one bit produces a pulse with negative polarity. These pulses are spread. The receiver can then only decode the STS if it knows the keys and cryptographic scheme used for STS generation. Moreover, STS correlation only works if it is started at the same time [25].

Apart from the general frame structure, the IEEE 802.15.4 standard also defines specific types of MAC frame formats [32]. One of these is the data frame visualized in Figure 2.7, which is used for ranging exchanges [39].

PHY Payload								
MAC Header (MHR)							MAC Payload	MAC Footer (MFR)
Frame Control	Sequence Number	Destination PAN Identifier	Destination Address	Source PAN Identifier	Source Address	Aux Security Header	Frame Payload	FCS
2 octets	1 octet	0 or 2 octets	0, 2 or 8 octets	0 or 2 octets	0, 2 or 8 octets	0, 5, 6 10 or 14 octets	Variable number of octets	2 octets

Figure 2.7: Mac Frame Format for Data Frames [37].

The frame control field indicates the frame type and which components are part of the MAC header (MHR) [25]. More specific information about the frame control field can be found in the next section. After the frame control field, there is a sequence number followed by the addressing fields which consist of the destination personal area network PAN identifier (ID), destination address, source PAN ID, and destination address. These fields are followed by an optional auxiliary security header. Then, the information el-

ements (IEs) with header and payloads are defined. These are optional fields that can be used to extend the IEEE 802.15.4 standard [32]. After the MHR, the MAC payload containing the data follows. In the end, a frame checking sequence (FCS) is attached to make sure that all bytes have been transmitted [32].

Bits: 0–2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10–11**	**12–13**	**14–15**
Frame Type	Security Enabled	Frame Pending	AR	PAN ID Compression	Reserved	Sequence Number Suppression	IE Present	Destination Addressing Mode	Frame Version	Source Addressing Mode

Figure 2.8: General Structure of the Frame Control Field for Data Frames [32].

The general structure of the frame control field for data frames is given in Figure 2.8. The first three bits define the frame type. A data frame is defined by the sequence 001. Other frame types can be found in Section 7.2.2.2 of the IEEE 802.15.4 standard [32]. Next, the security-enabled bit should be set to one if the payload of the frame is protected on the MAC layer. In addition, the field also indicates whether the frame has an auxiliary security header or not. The frame pending field is set to one if the device sending has more data for the recipient after this frame. The acknowledge required field is set to one if the sender expects an acknowledgment from the recipient of the message. The PAN ID compression field is used to indicate the presence of the PAN ID field. To see when the PAN ID compression field should be set to one based on source and destination ID length and presence as well as source and destination PAN ID presence, consult Section 7.2.2.6 of the IEEE 802.15.4 standard [32].

The sequence number suppression bit shall be set to one if the sequence number is omitted. The IE present field is set to one if any IEs are present. Destination and source addressing mode consist of two bits. They are set to 00 if no addresses are present, 10 for short 16-bit addresses, and 11 for long 64-bit addresses. Lastly, there is the frame version field. For data frames, it is set to 00 if the frame conforms to the IEEE 802.15.4-2003 standard, 01 for the IEEE 802.15.4-2006 standard, and 10 if the type of IEEE 802.15.4 is not further specified. For other frame types, consult Section 7.2.2.10 of the IEEE 802.15.4 standard [32] to find valid values of the frame version field. In case the security-enabled field in the frame control field is set to one, an auxiliary security header shall be added after the source address (see also Figure 2.7).

Octets: 1	0/4	0/1/5/9
Security Control	Frame Counter	Key Identifier

Figure 2.9: Auxiliary Security Header [32].

The auxiliary security header field has a variable length and contains information such as the security control field, a frame counter and a key identifier that are used for the secure processing of messages (see Figure 2.9). The frame counter field is a four-byte counter that is used for nonce generation and replay protection. The key identifier field implicitly or explicitly defines the key used for cryptography protection.

Bit: 0–2	3–4	5	6	7
Security Level	Key Identifier Mode	Frame Counter Suppression	ASN in Nonce	Reserved

Figure 2.10: Security Control Field [32].

Similarly to the frame control field, the security control field provides information about the type of secure processing used (see Figure 2.10). The first three bits indicate the security level of the frame. Zero indicates that no encryption and no message integrity code (MIC) is used. The numbers one to three indicate that a MIC is added at the end of the message but the content of the payload is not encrypted. Five to seven indicate that the payload is encrypted and a MIC is added. Note that the standard does not allow payload encryption without a MIC. The numbers One to seven mentioned are provided in binary representation with three bits.

The key identifier mode field whether the key is determined implicitly or explicitly and how long the key source field is. If the frame counter suppression is set to one, the four byte frame counter inside the auxiliary security header is omitted. The autonomous system number (ASN) in Nonce field is set to zero if the frame counter is used to generate the nonce. The last bit is reserved. More information about the auxiliary security header can be found in Section 9.4 of the IEEE 802.15.4 standard [32].

2.4 Security and Advanced Encryption Standard (AES)

The IEEE 802.15.4 standard supports three security services: data confidentiality, data authenticity, and replay protection (except for time-slotted channel hopping (TSCH) mode) [32]. The services can be provided with the help of a block cipher that is defined in the AES-128 [40]. The AES-128 block cipher is a specific variant of the Rijndael block cipher family. The AES-128 block cipher takes a block and key of size 128 bits as input parameters for encryption and decryption [41]. Compared to its predecessor, the data encryption standard (DES), the AES-128 offers higher levels of protection mainly due

to its longer key size at roughly the same level of computational complexity [42]. In addition, the AES can be easily implemented on devices with low computational power [42].

This requires the block cipher to be operated in a mode called cipher block chaining - message authentication code (CCM*) [32]. This mode generates a MIC during the encryption process. The MIC is added at the end of the payload before the FCS and can be used to verify that the payload and source address were not tampered with. As inputs, the AES-128 in CCM* mode takes the message's payload that shall be encrypted, a 128-bit key, and a 13-byte nonce [32]. For decryption, AES-128 in CCM* mode takes the encrypted payload with the MIC, and the same 128-bit key and a 13-byte nonce the message was encrypted with. Note that in other words, the MIC is sometimes also called the message authentication code or the message integrity check [43].

Octets: 8	**4**	**1**
Source Address	Frame Counter	Nonce Security Level

Figure 2.11: AEAD Nonce for Non-TSCH Mode [32].

The generation of the nonce is also defined in the IEEE 802.15.4 standard [32]. A nonce for authenticated encryption with associated data (AEAD) in non-TSCH mode is generated by taking the eight-byte source address, the four-byte frame counter from the auxiliary security header, and the nonce security level which has the value as the security level field of the security control field inside the auxiliary security header (see Figure 2.11).

With the help of the AES-128, data confidentiality is ensured by encrypting the message's payload during transmission. Data authenticity is guaranteed as the generation of the MIC uses a nonce that includes the source address of the message sender and the generation requires knowing the shared key. Replay protection is created as the nonce used for the creation of the MIC contains the frame sequence counter.

2.5 Ranging and Localization Approaches

UWB technology is known for its precise time-stamping capabilities [2]. Therefore, it is widely used for ranging and indoor localization applications [44]. Two-way ranging processes are explicitly described in the IEEE 802.15.4z standard [28]. For localization, several methods of indoor localization are used in combination with UWB technology such as TOA, TDOA, angle of arrival (AOA), received signal strength (RSS) and hybrid approaches [45; 44; 46; 47].

2.5.1 UWB ranging

When it comes to ranging, the processes of single-sided two-way ranging (SS-TWR) and double-sided two-way ranging (DS-TWR) are described in the IEEE 802.15.4z standard [28].

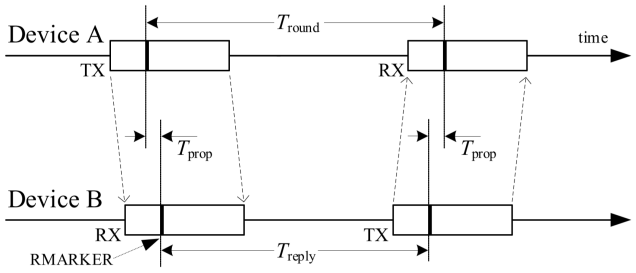

Figure 2.12: SS-TWR [28].

For SS-TWR, two messages are exchanged between a device A and a device B (see Figure 2.12) [28]. Based on the round-trip time measured at device A and the reply time measured at device B, the propagation time can be calculated with Equation (2.1). The propagation time is also called the time of flight (TOF) [24; 37].

$$T_{prop} = \frac{1}{2}(T_{round} - T_{reply}) \tag{2.1}$$

As T_{round} and T_{reply} are measured at two different devices and with two different clocks, an error C_{offset} is introduced by the clock offset between the two clocks [28]. This clock offset grows with the size of T_{reply}. To compensate for the clock offset, some remote transmitters can measure the clock offset. With the Equation (2.2), the propagation time can be corrected by the clock offset [37].

$$T_{prop} = \frac{1}{2}(T_{round} - T_{reply}(1 - C_{offset})) \tag{2.2}$$

In case the remote transmitters cannot measure the clock offset or T_{reply} is very long, an extended version of SS-TWR, called DS-TWR, can be used [28]. For DS-TWR, devices A and B exchange four messages as shown in Figure 2.13.

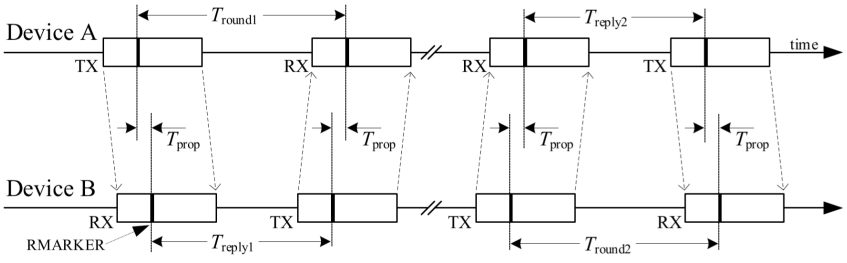

Figure 2.13: DS-TWR [28].

The round-trip and reply times of the messages are used to calculate the average propagation time of the messages with Equation (2.3). Note that the first package sent by device B can also be used as the initiating package of the second SS-TWR exchange. This reduces the total amount of packages sent between the devices to three [28].

$$T_{prop} = \frac{(T_{round1} \times T_{round2} - T_{reply1} \times T_{reply2})}{(T_{round1} + T_{round2} + T_{reply1} + T_{reply2})} \tag{2.3}$$

With the average propagation time calculated with either SS-TWR or DS-TWR, the distance between the two devices can be calculated by using the speed of light in the respective propagation medium and the propagation time [48]:

$$Distance = c \times T_{prop} \tag{2.4}$$

2.5.2 TOA Localization

TOA localization is one of the main UWB localization methods [47]. TOA localization estimates the position of a tag based on TOF measurements between the tag and multiple anchors [24]. The TOF measurements can be determined via SS-TWR or DS-TWR [28]. In two dimensions, the TOFs between tag and anchors multiplied by the speed of light represent radii of circles around the anchors (see Figure 2.14) Geometrically, the tag's position is then determined by the intersection of these circles [44]. In two dimensions, at least three anchors are needed to get a unique solution for the position of the tag [49]. In three dimensions, four anchors are needed [49].

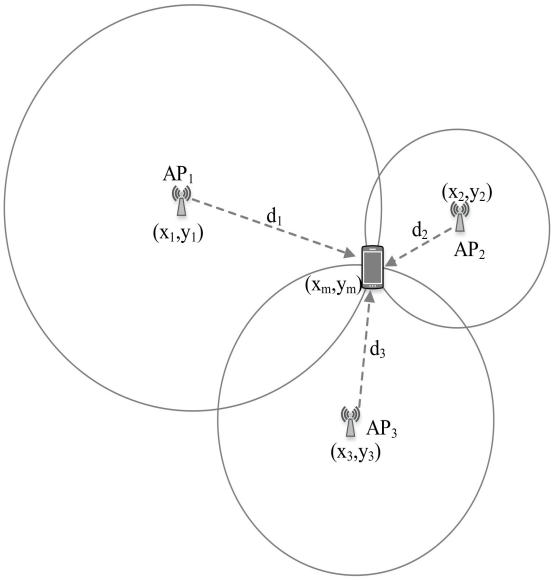

Figure 2.14: Geometric Visualization of TOA Localization in Two Dimensions [50].

As there is noise in real-world measurements, statistical approaches are used in practical applications to find the tag's position [24]. These methods express the tag's position

as an optimization problem that involves the measured distances, the anchor's positions, and some unknown noise components. The optimization problem is then solved by common estimators such as the minimum mean-square error estimator [24], the maximum a posteriori estimator [24] or the maximum likelihood estimator [51].

2.5.3 TDOA Localization

TDOA localization uses the difference in arrival time of one or multiple signals to determine the position of a tag. The signal can either be sent by the tag and then received by multiple anchors or the anchors send a synchronized signal which is then received by the tag [44]. For the remainder of this work, a TDOA localization where the tag sends a signal to the anchors will be called an active TDOA localization system. When the anchors send a synchronized signal to the tag, the system will be called passive TDOA localization system. In order to send the signal at the same time or to compare the arrival time of the tag's signal, the clocks of the anchors have to be synchronized [24].

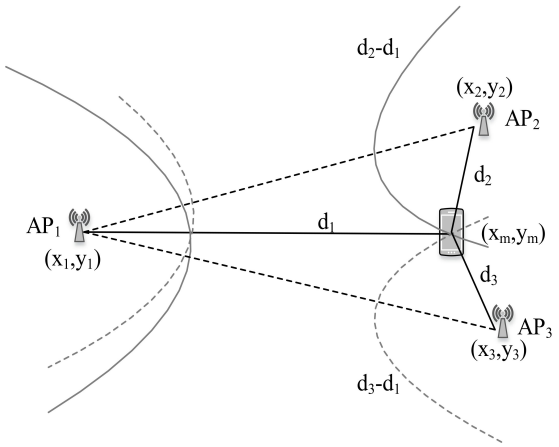

Figure 2.15: Geometric Visualization of TDOA Localization in Two Dimensions [50].

In two dimensions, the difference in arrival time of a signal between two anchors describes a parabola [24]. The parabolas between three anchors are visualized in Figure 2.15. The position of the tag can be determined by intersecting the parabolas. In two dimensions, four anchors are needed to find a unique solution for the position of the tag [49]. In three dimensions, the minimum required number of anchors is five [49]. Again, a statistical approach is normally chosen to find the position of the tag in a real-world scenario. A possible model for this can be designed as follows [24]:

$$z = f(x, y) + \eta \tag{2.5}$$

The measurement model described in Equation (2.5) consists of z, a vector of the measured distance differences between the anchors and a reference anchor, $f(x, y)$, a vector of the true values of the signal parameters and η, the probability density function

of the noise. The distance differences in vector z can be determined by calculating the TDOA values between the anchors and a reference anchor and multiplying it them with the speed of light. The true values of the signal parameters can be expressed in two dimensions with the formula $f(x, y)$. For the ith element, $f_i(x, y)$ can be described by subtracting two distances, expressed as circles around the ith anchor and the reference anchor, from each other. This is described in Equation (2.6).

$$f_i(x, y) = \sqrt{(x - x_i)^2 + (y - y_i)^2} - \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{2.6}$$

The variables x and y are unknown and describe the tag's position. The variables x_i and y_i are the coordinates of the ith anchor and are assumed to be known. The coordinates x_0 and y_0 describe the position of the reference anchor and are also assumed to be known.

With the measurement model described in Equation (2.5), a parametric approach such as maximum likelihood estimation can be employed to find the position of the tag [24]. In maximum likelihood estimation, the goal is to identify a set of unknown parameters represented as the vector $\theta = [x, y, \eta]$ that will maximize a given likelihood function. [24]. As $f(x, y)$ is a deterministic function, the maximum likelihood function can be expressed as conditional probability density function [24]. In the case of TDOA, a correlated gaussian noise component with the mean μ and the covariance matrix Σ is assumed. Based on this, and assuming mean of zero and a known covariance matrix. the maximum likelihood position estimate can be expressed as the following Equation (2.7):

$$\theta = \underset{[x,y]}{\arg\min} \, (z - f(x, y))^T \Sigma^{-1} (z - f(x, y)) \tag{2.7}$$

This is the same as the weighted least squares solution for the model in Equation (2.5) [52]. If one further assumes that the position of the anchors is perfectly known, the covariance matrix can be reduced to an identity matrix and θ can then be expressed as Equation (2.8):

$$\theta = \underset{[x,y]}{\arg\min} \sum_{i=1}^{N} (z_i - f_i(x, y))^2 \tag{2.8}$$

There is a multitude of approaches to solve this least squares problem. These include methods where the gradient or the jacobian needs to be known such as stochastic gradient descent [53] or methods that work without the gradient such as the Nelder-Mead algorithm [54].

2.5.4 AOA Localization

AOA localization uses the angle of arrival of a signal at multiple anchors or at the tag to find the position of the tag [24]. Determining the angle can be done by using the phase difference of arrival (PDOA). For this, either two receiver antennas or an antenna array are needed.

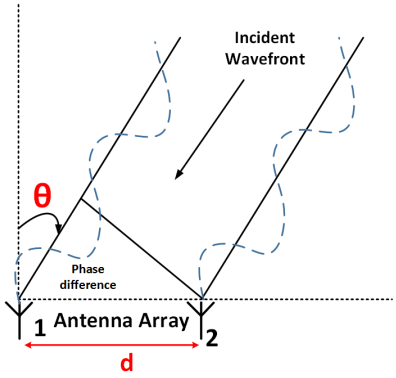

Figure 2.16: Geometric Visualization of PDOA to Get the Angle at an Antenna [55].

Considering a signal is sent from a distant transmitter, it can be assumed that the incoming signal arrives as a planar wave-front [24]. This is also visualized in Figure 2.16. Based on trigonometric laws, the PDOA value calculated between the two antennas and the wavelength λ of the incoming signal, the angle of arrival can be expressed as Equation (2.9) [56]. It is important to note that this equation only holds up as long as the difference between the two antennas is smaller than $\frac{\lambda}{2}$ [56].

$$\theta = arcsin(\frac{\alpha\lambda}{2\pi d}) \tag{2.9}$$

When having determined the AOA at multiple anchors or for multiple signals at the tag, the position of the tag can be calculated by drawing angle lines from each source and intersecting them (see Figure 2.17) [44]. To find a unique solution in two dimensions, at least two anchors are needed and three anchors for three dimensions [24]. It is also important to note that the AOA is determined in relation to the orientation of the receiver antenna [56]. Therefore, the orientation of the receiver's antenna needs to be known and taken into account to calculate the position of the tag [56].

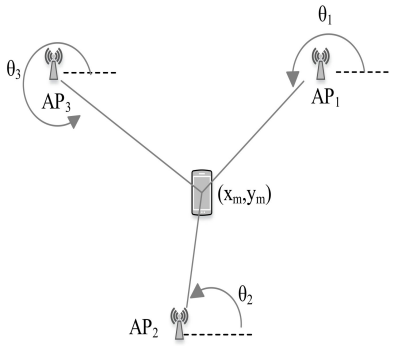

Figure 2.17: Geometric Visualization of AOA Localization in Two Dimensions [50]

2.6 Privacy

With the proliferation of ubiquitous computing in all domains of life, an increasing amount of personal information is being generated, processed, and stored. Consequently, privacy has become a crucial concern when designing and evaluating digital applications. What makes it challenging to address these concerns however is the conceptual ambiguity surrounding the term privacy. During a span of more than 100 years, privacy has been studied and defined in various fields, including social sciences, economics, law, and natural sciences [57]. The definitions often vary widely and range from defining privacy as the right to be left alone [58] to the concept of information privacy, which entails one's ability to access and control one's personal information [59]. To address the broadness of the term privacy, taxonomies, and ontologies have been developed to categorize terms relating to privacy [60; 61; 62].

One of these ontologies is COPri V2, which is a comprehensive ontology for developing privacy requirements [1]. The conceptual model of the ontology can be found in the Appendix at Appendix A.1. COPri V2 groups the concepts relating to privacy into four dimensions: Organisational, risk, treatment, and privacy. The organizational dimension contains concepts relating to social and organizational aspects. In particular, it introduces the concept of actors. Actors are autonomous entities that carry out strategic actions toward a goal. The organizational dimension also contains the concept of information. Two types of information are distinguished: personal and non-personal information. Personal information is information that can be related to a legal entity. As these entities have the right to control personal information according to the definition of information privacy [59; 1], personal information is the key concept of privacy.

In the risk dimension, concepts that can potentially endanger privacy needs are introduced. For example, the concept of a threat is defined, which is a potential incident that for example exploits vulnerabilities and thus threatens personal information. A vulnerability is a weakness in the system that a threat can exploit.

Next, in the treatment dimension, concepts that mitigate risks such as privacy goals, policies, and mechanisms are introduced. Lastly, the privacy dimension introduces eight fine-grained concepts that can be used as privacy requirements. These concepts are confidentiality, anonymity, unlinkability, unobservability, notice, transparency, minimization, and accountability. They are defined as follows:

.

Confidentiality. Confidentiality advises that personal information should be inaccessible by threats. Confidentiality is further split up into the concepts of non-disclosure, need to know, and purpose of use. **Non-disclosure** means that information shall only be disclosed if the data subject gives permission to do so. **Need-to-know** describes that an actor should only have access to and use as much personal information as is necessary for completing a specific task. Lastly, **purpose-of-use** dictates that personal information should only be used for the use case to which the data subject gave permission.

Anonymity. Demands that personal information shall be used in a manner that does not reveal the identity of the data subject. This can be achieved by using anonymization techniques.

Unlinkability. Dictates that it should not be possible to link personal information back to its data subject.

Unobservability. The goal of unobservability is to hide activities performed by a data subject.

Notice. Notice means that a data subject should be informed when its personal information is being collected.

Minimization. Minimization advises that the collection of personal information should be kept to a minimum.

Transparency. Requires that a data subject knows who is using its information and for what purposes. This can be achieved by using authentication and authorization. **Authentication** is a mechanism that verifies if are who they claim they are. **Authorization** on the other hand checks if actors have the right to use information in accordance with their credentials.

Accountability. Dictates that data subjects can hold information users accountable concerning their actions with the personal information of the data subject.

Based on these privacy requirements, privacy-preserving applications can be designed. In addition, the privacy requirements can be taken to evaluate existing applications based on their privacy-preservation characteristics.

Within the scope of this work, it is important to also mention a particular type of information privacy which is location privacy. [63] defines location privacy as a special case of information privacy. Location privacy describes a data subject's ability to prevent other actors from learning about its current or past location. For this, location information needs to be safeguarded which consists according to [17] not only of a data subject's location but may also include a data subject's identity (such as name or identifier (ID), spatial information (coordinates and trajectories), or temporal information (when the location was collected).

All in all, ensuring privacy is a crucial concern in the context of UWB localization systems. To ensure the protection of privacy, tools like COPri V2 provide privacy requirements that can guide privacy-preserving development and that can be used to evaluate the level of privacy of existing systems. When it comes to UWB localization systems, safeguarding location information is particularly important, as this is the main type of personal information such a system produces.

3

Related Work

This Chapter provides an overview of the relevant research concerning UWB technology and privacy. It begins with highlighting works that focus on UWB localization systems. After that, it explores works related to the topic of privacy such as location privacy, privacy-preserving applications, the intersection of UWB technology and privacy, privacy considerations within the IEEE 802.15.4 standard, and privacy analysis methods.

3.1 UWB Localization Systems

First of all, there is an extensive body of research that gives an overview of the general mechanisms involved in indoor localization and what kind of architectures can be used to perform indoor localization with the help of wireless technologies [45; 44; 55; 64]. Some of these surveys also specifically focus on UWB technology such as [65; 66; 46; 47].

Next, there are works that specifically focus on a single ranging or localization approach. When it comes to ranging, there are works that focus on ranging-related attacks [67] or making the ranging results more accurate [68; 69]. Examples of UWB localization systems that use a TOA approach can be found in [70; 71; 69]. Other works explore UWB systems with a TDOA localization architecture [72; 73; 74]. There are also works that implement UWB AOA localization systems that specifically use a PDOA approach [56; 75; 76].

A multitude of works also combined the aforementioned approaches [77; 78; 79]. A combination of TOA and TDOA is applied by [77]. [78] measures the accuracy of hybrid AOA and TDOA system and [79] design an algorithm that synthesizes the data of an inertial measurement system with the data of a hybrid UWB AOA and TOA system. There are also hybrid approaches that combine multiple localization technologies such as WI-FI and UWB [80], Bluetooth and UWB [81], or GPS and UWB [82].

UWB ranging is utilized in numerous applications. In research, UWB ranging is used among other things for asset location [83], pedestrian tracking [84] and to determine the distance between vehicles [85]. There are also products provided by companies that rely on UWB ranging. For example, Kinexon provides UWB ranging systems for collision protection and for save zoning [86; 87]. Save zoning for example can be used to mitigate the spread of COVID-19 by giving people so-called SaveTags that visually and auditorily

warn the wearer when standing too close to another wearer of a SaveTag [87]. Another company that uses UWB ranging in its products is Estimote. Estimote produces UWB proximity beacons that can be programmed to send contextual notifications, use zone-based positioning, or to perform presence detection [88]. For example, users can build applications that help employees find their co-workers within seconds, that support users with way-finding, or that provide extra context about artworks in a museum [88].

UWB indoor localization can be used for similar use cases as UWB ranging. UWB is often evaluated in industrial settings, for example for employee tracking [2] or drone-based inventory management [89]. Another use case of UWB is elderly care [10; 75; 90]. Such systems can be used to monitor accidents when elderly patients fall in bathrooms [10], or help provide location-based services inside the homes of elderly patients [75], which can, for example, include continuously monitoring their health parameters inside their homes [90].

When it comes to UWB indoor location systems designed by companies, Kinexon provides a real-time player and ball tracking system for several sports [91]. Another example is the use case by Qorvo in which a TDOA system was used in a museum to provide navigation for visitors [92].

3.2 Privacy in Internet of Things (IoT) Networks

Numerous studies have examined the domain of location privacy. One of the pioneering works on location privacy is [63]. The authors develop a threat model for location privacy, based on which countermeasures were developed and summarized in a privacy framework. Further research focuses on systematically identifying threats to localization privacy and methods that can be used against it [17; 12; 18]. The privacy-preserving methods can be split up between the devices they are happening on (on device, during transmission, on server) [12] and include processes such as encryption, anonymization and pseudonymization, obfuscation and a reduction in personal information shared [17; 12; 18]. Other papers implement privacy-preserving applications. For example, [13] developed PILOT, a privacy-preserving indoor localization tool based on WI-FI. Another example of a privacy-preserving system is the work of [14]. Here, a localization system for an underwater environment is designed. Compared to other works, [14] focuses on information-hiding techniques instead of using a cryptographic approach (e.g. [20; 21]) due to the limited bandwidth and computation power of underwater sensors. On a more general level, [93; 19; 94] conducted surveys of general IoT privacy topics, blurring the line between security and privacy concerns.

Few works also consider privacy in combination with UWB technology. When it comes to applications that track and observe people, many authors argue that the usage of UWB itself increases the privacy-preserving characteristics of a system. For example, [8; 9; 10; 11] all point out that the usage of UWB sensors instead of a camera-based system is in itself privacy-enhancing. However, none of the authors concretely consider

the impact of the UWB systems architecture on privacy. Other papers that analyze UWB and privacy focus on cryptographic methods [95; 96; 97] or modulation techniques [98].

Apart from work that focuses specifically on UWB, there are also works that examine the IEEE 802.15.4 standard. Again, several works focus on cryptographic systems that can be used together with the 802.15.4 standard [99; 100; 101; 102]. One paper focusing on the IEEE 802.15.4 standard that stands out is the work of [15]. Apart from proposing a system with whom nodes in a wireless sensor network can periodically change their addresses, the authors also construct a threat model, develop privacy countermeasures based on the model, conduct a privacy analysis, and test the impact of the privacy measures on the system's performance. Another paper that focuses on identifying security and privacy-related issues in UWB positioning systems is the work by [103]. [103] highlights a concrete approach for building a scalable and security-preserving UWB system.

Lastly, two works that need to be mentioned in the context of performing a privacy analysis are the works of [1] and [16]. [1] introduces a core ontology for privacy requirements, which the authors recommend should be used during the design phase of application development to make applications more private by design. However, even though [1] introduces a use case the introduced privacy requirements of the ontology could be applied to, there is no step-by-step guide on how to use the ontology as a whole to design a privacy-preserving system.

another work that should be mentioned in the context of privacy analysis is the work of [16]. [16] conducts an investigation of IoT data privacy in regards to healthcare IoT. In particular, the authors create an overview of all privacy concerns and the appropriate privacy mechanisms to tackle the concerns along each step of the data flow of a healthcare IoT application. However, no concrete examples are given of a system that implements the privacy mechanisms mentioned in the work.

4

Methodology

The methodology Chapter first gives an overview of the hard- and software components involved in building the UWB localization systems. After that, the framework of the privacy analysis and the steps involved in performing the analysis are presented.

4.1 Hardware and Software Components

In this Section, an overview of the used technologies is given. This includes the microcontrollers utilized as well as the software needed to control these microcontrollers.

4.1.1 Nordic Semiconductors

Nordic Semiconductors is a semiconductor company specializing in wireless technology [104]. Their product palette includes WI-FI, Bluetooth, and ZigBee solutions. Among other products, Nordic Semiconductors also produces a series of nRF52 development kits (DKs). These DKs are equipped with Bluetooth 5.3 system on chips (SoCs) that integrate a 64 MHz Arm Cortex-M4 CPU. The DKs are also Arduino Uno Revision 3 compatible, making it possible to mount 3rd-party shields [105]. Two of these boards are the nRF52840 and the nRF52833 (see also the right Figure in Figure 4.1). [105]. They can be programmed by using nRF5 software development kit (SDK) [106]. To flash programs onto the nRF52 boards, Nordic Semiconductor recommends using the integrated development environment Segger Embedded Studio [107].

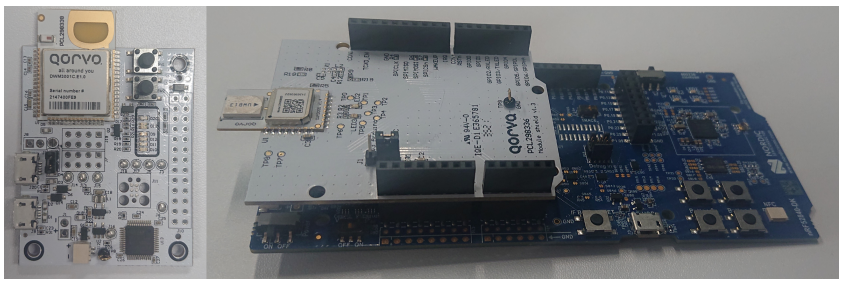

Figure 4.1: Left: DWM3001CDK. Right: nRF52840 and DWM3000EVB.

4.1.2 Qorvo

Qorvo is a manufacturer of UWB transceiver modules [108]. Qorvo produces two lines of UWB transceiver modules, the DWM1000 and DWM3000 line [109; 110]. The DWM1000 is one of the most popular UWB modules [25]. It supports channels 1,2,3,4,5, and 7 and is compliant with the IEEE 802.15.4a standard [109]. The DWM3000 module is the successor to the DWM1000 module. It is based on the DW3110 integrated circuit (IC). It can operate on channels 5 and 9 and supports both the IEEE 802.15.4a and IEEE 802.15.4z standards. Moreover, it is compatible with Apple's U1 chip and FiRa compliant devices [110].

For this work, DWM3000EVB Arduino shields and DWM3001CDK DKs were used for UWB communication (see Figure 4.1) [111; 112]. The DWM3001CDK is a fully-fledged DK and can be used without any other devices. The DWM3000EVB Arduino shield was mounted on top of an nRF52840 DK.

The devices were used in combination with the DW3xxx SDK. This SDK can be downloaded under the section documents on the website of the DWM3000EVB shield [111]. For the combination of nRF52840 DK and DWM3000EVB Arduino shield, the SDK includes a build platform that works out of the box together with the nRF SDK. For the DWM3001CDK, the Qorvo SDK had to be ported to the nRF52833 board as at the time of building the experiments, no build platform or separate SDK was available for the DWM3001CDK. This process included changing the nRF system files, adjusting the RAM and FLASH ranges, and updating the pin map.

4.1.3 Estimote

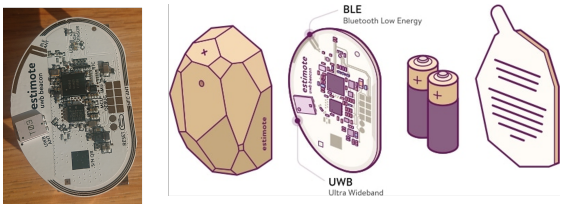

Figure 4.2: Left: Real-World UWB Beacon. Right: Schematics of a UWB Beacon [113].

Estimote is a location technology company providing an operating system for the physical world [114]. In order to do this, Estimote provides real-world beacons equipped with UWB and Bluetooth modules (see Figure 4.2) [115; 113]. One type of beacon produced by Estimote is called UWB beacon [113]. These beacons are shipped as part of the UWB DK and can be evaluated with a demo app provided by Estimote [116]. The beacons are battery-powered and include a Qorvo DW3120 module [117] as its UWB transceiver and an nRF52840 SoC [105] for Bluetooth communication. Apart from the demo app, Estimote also provides an SDK for developers to write their own apps [116]. The SDK lets users build swift applications for the iPhone on top of UWB two-way ranging [116].

4.1.4 UWB Sniffer

During a previous project, a network sniffer was built based on the DWM3000EVB and the nRF52840 DK to listen to UWB network traffic (see Figure 4.3). This sniffer was taken as a basis for the sniffer used in this project and expanded by the capabilities of sniffing encrypted UWB traffic and UWB traffic that uses an STS.

The sniffer pipeline has several components. The DWM3000EVB mounted on top of the nRF52840 DK acts as the core part of the sniffer. The UWB module is configured to listen on a specific UWB channel. When the device receives a well-formatted UWB package, the DK forwards the package via a universal asynchronous receiver transmitter (UART) connection to the host computer. In case the UWB module is not able to correctly receive the package, an error message is forwarded via UART to the host computer.

On the host computer, a Python script is run which takes the packages that arrive at a communication port (COM) via the UART connection, transforms them into the package capture next-generation dump file format (PcapNg), and writes them into a pipe. Wireshark, a program for visualizing packages and network traffic, can then read the packages from the pipe in real time and visualize the UWB network traffic.

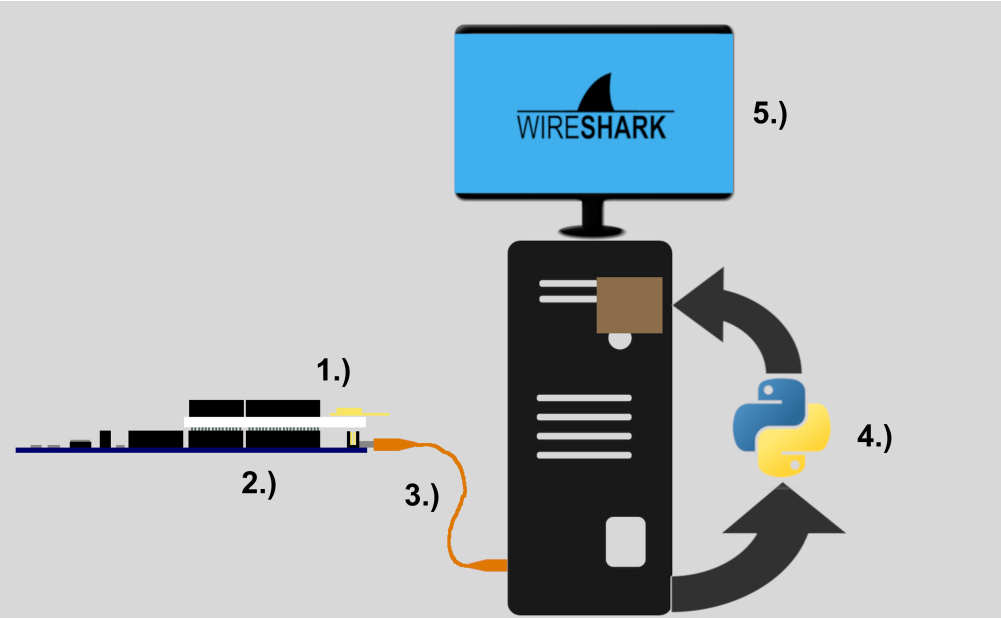

Figure 4.3: UWB Sniffer to Capture UWB Traffic. 1.) DWM3000EVB 2.) nRF52840 DK 3.) USB Cable 4.) Computer with Python Script 5.) Wireshark.

4.2 Privacy Analysis

One part of this work consists of deriving a process for privacy analysis on a network level. This process will be used to perform a privacy analysis of the UWB localization systems that will be built during the course of this work. The approach chosen for the privacy analysis is predominantly based on the works of [1; 15; 17; 118].

First, similarly to [1] and [15], a use case is designed in which the localization system will operate. Next, the components of the use case are fleshed out which includes identifying the actors that engage with the system [1; 118]. In particular, the roles of the data subject, the data controller, and the data processor are assigned to the different actors [1; 118]. In addition, the motivation of the different actors for sharing and processing personal information is highlighted [118]. After that, the personal information involved in the process is identified [1]. Particularly, there is a focus on location information. This location information consists of data related to position, identity, and time according to [17]. Lastly, potential threats to privacy are identified [1].

After fleshing out the components of the use case, the evaluation criteria for the privacy analysis are chosen. For this, the privacy requirements in [1] are taken as a basis and those applicable to the analysis on a network level are selected.

4.2.1 Derived Use Case

Longevity among the elderly has resulted in an increased demand for ambient assisted living (AAL) and e-health solutions [1; 119]. The goal behind these solutions is to relieve the burden on caregivers, to enable the elderly to live more autonomously and comfortably, and to provide medical personnel with health-related information continuously, remotely, and in real-time [1]. At the same time, privacy concerns are some of the most prominent criticisms of such systems [1; 75].

This work focuses on a UWB localization system that is used to track the movements of an elderly patient in a retirement home. The system is mounted inside the room of the patient. The room is assumed to have a square layout of four times four meters.

The use case is visualized in Figure 4.4. 1.) represents the elderly patient wearing a UWB tag. 2.) shows a UWB anchor. The anchor can send and receive UWB signals and it can do simple calculations. 3.) is a master anchor. It can also send and receive UWB signals and perform calculations. Moreover, it is connected to the retirement home's network. 4.) represents a server. Here, the more complex processing of movement data happens. The server is connected to the database of the retirement home 5.). Here, all or some parts of the movement data are stored for analysis. 6.) is a caretaker. He or she uses the data to monitor the patient. 7.) Is a medical professional. He or she uses the data for predictive care. 8.) Is an employee at the localization system company. He or she wants to use the data to debug the system and to improve other products such as machine learning algorithms.

The scope of this work includes all steps between 1.) and 3.). In particular, the focus lies on the transmission between the devices and the data that is processed on them. Out of scope is all processing that happens on the server, for example, long-term analysis of the patient's movement or a combination of the movement data with further datasets such as the patient's medical history.

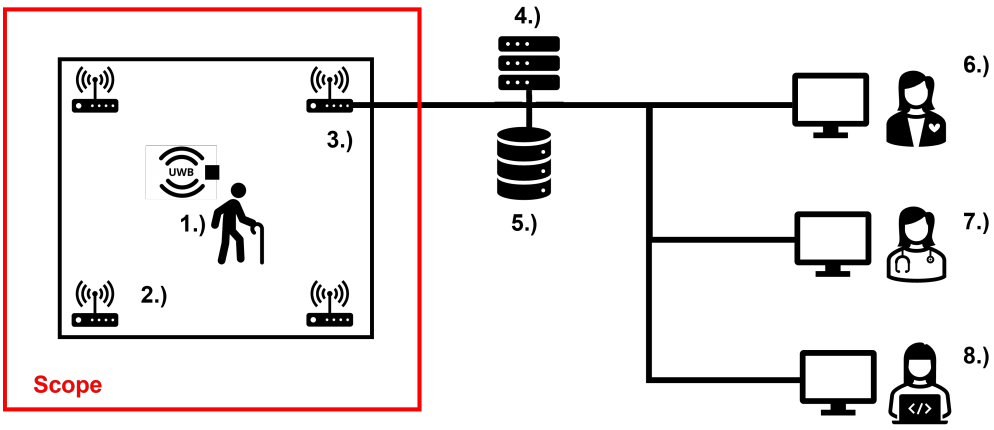

Figure 4.4: Representation of a Possible UWB Localization System Architecture Used for Elderly Care.

4.2.2 Components of the Derived Use Case

This Section describes in detail the components involved in the derived use case. This includes actors, the personal information involved in the process, and potential threats to privacy.

The data subject in this scenario is the elderly patient. He or she provides personal information to profit from improved healthcare services and preventative care. The data controller is the caretaker of the elderly patient. He or she wants to use the data to remotely monitor the behavior of the patient to detect abnormalities and intervene accordingly. The caretaker also shares the gained information with medical personnel such as doctors. These are also data controllers. The medical personnel uses the data to find patterns in the behavior of the patient that can be used for preventative care. The system itself is assumed to be provided by an external company. This company acts as a data processor. The company works in close cooperation with the retirement home and shares most of their goals. However, the company also wants to use the movement data to improve their localization system. In addition, they want to store the data to use it for further products such as machine learning algorithms for preventative care.

The personal information shared in this scenario is the location information of the patient. This includes position, identity, and time information [17] about the patient. This information can be combined with the medical history of the patient.

Looking at threats, as for any wireless system, it is possible to eavesdrop on the lo-

calization system [15]. It is assumed that an attacker has access to a UWB sniffer with multiple antennas (e.g. an iPhone), making it possible to potentially sniff on all UWB channels and to determine the angle of arrival of a signal. In addition, the attacker could also potentially eavesdrop on the communication inside the hospital network. Another potential threat is posed by inappropriate behavior of the caretaker, the medical personnel, or the company's employees. Examples of such behavior are accidental publishing of confidential data, deanonymization by combining localization data with other data sources, or storing personal information at insecure an insecure location. The personal information mentioned here will be refined in Subsection 6.4.1 based on the actual implementation of the localization systems.

4.2.3 Criteria Selection

For the privacy analysis, the privacy requirements from [1] were used as a basis. However, some of the requirements cannot be applied to the network level as they focus on topics like giving consent to share personal data. Thus, these requirements were identified and omitted. Namely, the following privacy requirements were excluded based on the fact that they for example target the behavior of the actors described in Subsection 4.2.2:

Non-disclosure (part of confidentiality): Non-disclosure dictates that personal information can only be disclosed upon the data subject's permission. This requirement targets the implementation of processes on top of the localization itself and is therefore excluded.

Notice: emphasizes that data subjects should be notified when their personal information is collected. This privacy requirement might be partially triggered on the data level inside the OSI model, however its main implementation targets again a higher processing level than the TDOA localization operates on.

Accountability: data subjects should be held accountable for their actions concerning personal information. This criteria also targets the behavior of the actors and not directly the implementation of the localization process.

With these requirements excluded, the privacy criteria upon which the system will evaluated are: Confidentiality including need-to-know and purpose of use, anonymity, unobservability, unlinkability, minimization, and transparency which includes authentication and authorization (see also Section 2.6).

4.3 Experiments for the Technical Evaluation

Apart from the privacy evaluation, the UWB localization systems are also evaluated based on their technical capabilities. The primary objective of the technical evaluation is to ensure the proper functioning of the systems. For this, an experimental setup was designed, with the help of which multiple experiment suites were conducted.

For the experimental setup, the anchors of the localization system are placed on the floor of a room in the form of a square with edges of a length of three meters. This setup is supposed to emulate a realistic placing of the anchors inside a patient's room

as described in the derived use case (see Subsection 4.2.1). The antennas of the anchors placed at the north of the room are oriented towards the south and the antennas placed at the southern border of the room are oriented toward the north. Where not indicated differently, the antenna of the tag faces west. This is also marked in the right Figure in Figure 4.5.

To describe the location of objects inside the room, a coordinate system with its origin (0,0) located in the middle of the room is used. The x-direction points towards the east of the room and the y-direction towards the north of the room. One unit within the coordinate system corresponds to one 0.75 m in the real world. Therefore, the anchor in the northeast corner is for example placed at (2, 2).

After placing all devices at their supposed location, the localization system is activated. The first type of experiment consists of placing the tag at the location (0, 1.5) and placing a sniffer at the western border of the room with the same y-coordinate. The sniffer is then turned on in order to simulate a potential attacker eavesdropping on the localization system. The sniffed packages are stored in the form of a PcapNg file and visualized with the help of Wireshark. The collection duration of the sniffer is ten minutes.

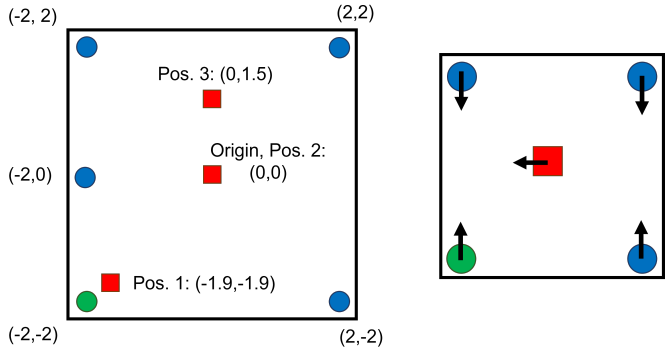

Figure 4.5: Left: Setup for the Localization Experiments. Right: Antenna Orientations during the Experiments. The green circle represents the master anchor, the blue circles show the other anchors and the red square indicates the position of the tag for positions one, two, and three. The origin of the coordinate system lies in the middle of the room at (0,0).

The rest of the experiments do not include the sniffer and instead focus on collecting the TDOA values and calculated locations of the localization system. For the first experiment suite, four anchors in the corners of the room are used. The tag is placed in one of three locations: for position one, the tag is placed northeast of the anchor in the southwest corner at the coordinates (-1.9, -1.9). For position two, the tag is placed in the middle of the room at the origin of the coordinate system at (0, 0). For position three, the tag is placed north of the middle of the room at (0, 1.5). These three positions have been chosen to cover a variety of positions inside the room with different distances between anchors and the tag. The setup of the anchors and the positions are visualized

in Figure 4.5. For the second experiment suite, the tag is placed at the same three positions, but a fifth anchor is added at the coordinates (-2, 0).

For the experiment suites with four anchors, the collection duration for positions one and three is 15 minutes. For position two, the data was collected for 30 minutes. The same collection durations were used for the experiments with five anchors.

After the experiment suites with four and five anchors, further experiment suites were performed that focused on a specific characteristic of the localization systems. This includes an experiment suite where the tag is placed at (0,0) and turned around in the horizontal plane to see the impact of changing the antenna orientation, an experiment suite where the UWB signals are partly blocked by having a person sit inside the room and a suite where the tag is placed at (0, 1.5) and its signals are collected for multiple hours. All these experiments used four anchors in the corners of the room.

4.4 Technical Localization Evaluation Metrics

Following the completion of the experiments, the localization systems are evaluated based on three factors for the technical evaluation: whether or not the systems are capable of performing basic localization, how accurate they are, and what level of precision they have. Note that in this case, accuracy is defined as the degree to which the observed values align with the ground truth whereas precision is defined as the repeatability or refinement of the results [120].

To get a first overview of the localization capabilities of the localization systems, the collected locations are visualized in a two-dimensional plot. To provide visual orientation within the plot, the positions of the anchors and the actual tag position are also plotted. Based on the position of the point cloud of the collected locations relative to the actual tag position, a rough estimate of the overall working of the localization system and its accuracy can be made. Additionally, the diameter of the point cloud gives a first impression of the precision of the localization system.

To further quantify these observations, two metrics are used. To assess the accuracy of the systems, the distance between the actual tag location and the median of the calculated tag locations is used. The median was specifically as the measurement of the point cloud as it is not as heavily skewed by outliers as the arithmetic mean.

To describe the precision, the mean absolute deviation (MAD) of the collected values is utilized. The MAD is calculated by subtracting a measure of position from each calculated observation, taking the absolute value of each difference, summing these values up, and dividing them by the number of observations as shown by Equation (4.1) [121]. As the measure of position, the median is used again because it is not as heavily skewed by outliers as the mean.

Both metrics are calculated with the help of the coordinate system and then transformed into meters. One unit inside the coordinate system is equivalent to 0.75 meters, the transformation factor is thus 0.75 from coordinate units to meters.

$$MAD = \sum_{i=1}^{n} \frac{|x_i - \bar{x}|}{n} \tag{4.1}$$

Lastly, to get a deeper insight into the composition of the calculated locations, the histograms of the TDOA values at each anchor are constructed. Here, the focus specifically lies on the distance between the expected TDOA value at that anchor and the median of the collected TDOA values. For the sake of conciseness, the histograms are also summarized into tables for some experiments showing the median of the TDOA values of each anchor minus the expected TDOA value at each anchor (see Equation (4.2)).

$$Distance = median\_tdoa\_values_{anchor\_i} - expected\_tdoa\_value_{anchor\_i} \tag{4.2}$$

This helps identify if systematic errors are already introduced on a physical level or if they occur later on during the localization process.

5

Results

This Chapter is structured into three parts. The first two Sections explain the reasoning behind the choice of the localization approach and the localization architecture design decisions. The subsequent two Sections describe in detail the architecture of the two localization systems constructed. Lastly, the remaining Sections present the results of the experiment suites described in Section 4.3.

5.1 Choice of the Localization Approach

One of the most important steps in designing a localization system is the choice of the localization approach. According to its product page [122], the DW3110 transceiver IC supports implementations of TOA, TDOA, and PDOA localization systems. These approaches have all different advantages and disadvantages that are summarized in Table 5.1.

TOA approaches provide high localization accuracy [122]. TOA systems employ two-way ranging to achieve a localization accuracy of 10 centimeters (cm) according to Qorvo [122]. In order to perform TOA localization in two dimensions, at least three anchors are needed [49]. TOA localization can only support a limited number of tags as both

Criterium	TOA	TDOA	PDOA	Source
Accuracy	High	High	Medium	[37]
Min Nr of Anchors 2D	4	3	2	[24; 49]
Max Nr of Tags	Few	Many	Many	[86]
Clock Synchronization	No	Between Anchors	No	[24]
Antenna Complexity	Low	Low	High	[56]
Privacy	Low	High (Passive Tag)	High (Passive Tag)	Observation

Table 5.1: Summary of Localization Characteristics of Different Localization Approaches.

anchors and tags need to send and receive messages. The clocks do not have to be synchronized and no complex antenna structure is needed. When it comes to privacy, both the tags and the anchors send messages that contain personal information. This potentially poses a risk to privacy.

TDOA approaches can also achieve high localization accuracy in the range of 10 cm [37]. At least four anchors are needed to perform TDOA localization in two dimensions [49]. In the case of passive tags (see Subsection 2.5.3 for a definition of active and passive system), the TDOA system can, in theory, support an infinite number of tags. If the tags are active, the number of tags is limited by the signal processing speed of the anchors. In order to perform TDOA localization, the clocks between the anchors need to be synchronized [24]. No complex antenna architecture is required for TDOA localization. TDOA systems that use passive tags can be considered privacy-preserving as no personal information is transmitted via UWB by the system. TDOA localization systems that use active tags can potentially pose a privacy risk as the tags transmit signals based on which a person's location can be determined. In addition, the signals might also contain pieces of personal information such as the tag ID.

PDOA localization systems are considered to be less accurate than TOA and TDOA systems [37; 75]. In return, PDOA localization only needs two anchors to determine the position of a tag in two dimensions [24]. Additionally, no clock synchronization is required for PDOA localization [37]. However, the antenna structure needed for PDOA localization is more complex than the antenna structure used for TOA or TDOA localization. For PDOA localization, at least two antennas or an antenna array is needed to calculate the PDOA values. When it comes to the number of tags supported, PDOA localization has similar characteristics to TDOA localization as a PDOA system with passive tags can, in theory, support an unlimited amount of tags whereas PDOA localization with active tags only supports a limited amount of tags. Building upon a comparable line of reasoning to that of TDOA localization, active PDOA localization carries a higher privacy risk than passive PDOA localization. The characteristics of the different localization approaches are summarized in Table 5.1.

Since the focus of this work lies on constructing a privacy-preserving localization system, TOA localization was excluded as a potential approach due to its low privacy-preserving characteristics. Subsequently, the criteria of localization accuracy and the ease of constructing the system were considered in choosing the localization approach.

Typically, TDOA systems can achieve higher localization accuracy than PDOA systems. However, unlike TDOA systems, PDOA systems do not require the synchronization of anchor clocks, which simplifies the design and construction of a PDOA system. Given that the use case does not require a centimeter level of accuracy, the decision was made to adopt the PDOA approach.

When starting to implement the PDOA system, several points were noticed: First of all, even though the product page of the DW3110 IC highlights that the DW3110 IC is capable of performing +/- five-degree angle measurements [122], the DW3000 family

user manual [37] mentions that the DW3110 does not support PDOA localization. The claim of the family user manual was further backed up by the Qorvo UWB product page [123] and a post on the Qorvo forum [124].

In order to validate that the DW3110 is not capable of performing PDOA measurements, the PDOA examples provided by Qorvo SDK were tested. The examples consist of a UWB sender that periodically sends out a UWB signal and a receiver that calculates the PDOA value of the signal sent by the sender. First, the files in the folders *ex_01h_simple_tx_pdoa* and *ex_02h_simple_rx_pdoa*, *simple_tx_pdoa.c* and *simple_rx_pdoa.c*, were copied to a separate folder and slightly modified. In both files, all references to test_run_info() were omitted. This function prints information onto a screen that the standard devices are not shipped with.

Next, the transmission delay between packages TX_DELAY_MS of the sender in *simple_tx_pdoa.c"* was set to 100 milliseconds instead of 500 milliseconds. For the PDOA receiver in *simple_tx_pdoa.c*, an UART connection was set up and the PDOA values were printed to the UART connection with the help of the function dimi_print from the *dimi_print.c* file (see also Subsection 5.3.5). Lastly, a Python script was written that collected the PDOA values from the UART connection. The script saved the PDOA values to a file and converted the PDOA values from radian to degrees. After that, it calculated distribution values such as minimum, maximum, and mean of the samples, and visualized the PDOA values inside a histogram. The formula for converting the radian units to degrees was taken from a comment inside the *deca_device.c"*file inside the Qorvo SDK.

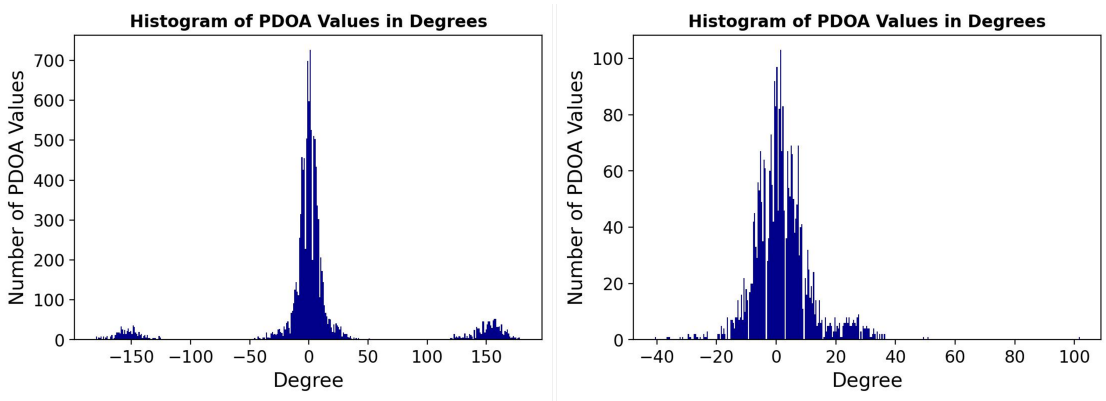

Figure 5.1: Histogram of PDOA Values. Left: First run of the experiment with the devices rotated 180 degrees relative to one another. Right: Second run with the devices rotated 90 degrees counter-clockwise relative to one another.

After the modification of the files, the programs for the sender and receiver were flashed onto two nRF52840 boards equipped with DWM3000EVB shields. The results of the measurements are visualized in Figure 5.1. The devices were placed in a straight line two meters from each other. Both devices were rotated 180 degrees relative to one another and placed on the floor. The receiver was connected to the host computer via

a USB cable and both devices were turned on. The Python script was executed for five minutes during which it collected 2958 PDOA values. The highest recorded PDOA value was 179.3 degrees, while the lowest was -180.0 degrees. The mean of the sample was 11.33 degrees and the median was 0.22 degrees. The histogram of the PDOA values is shown in Figure 5.1 on the left.

The histogram on the left shows a trimodal distribution, characterized by a big peak near zero degrees and two smaller peaks around -160 and 160 degrees. When repeating the experiment while turning the receiver by a variable amount of degrees, the resulting distributions looked very similar to the distribution in Figure 5.1 on the left. However, all of them were missing the two smaller peaks. Instead, the distributions showed a single peak around zero degrees such as the histogram of the second run of the experiment visualized in Figure 5.1 on the right. For this experiment, the devices were rotated 90 degrees relative to each other. The expected outcome of the experiments would have been histograms displaying unimodal distributions, where the mean value varies depending on the relative orientation of the sender and receiver to each other. Since this was not the case for the performed experiments, it was concluded that the DW3110 IC is not capable of calculating the PDOA value of an incoming signal.

Based on the experiments, the product pages, and the Qorvo DW3000 family user manual it was decided to not build a PDOA localization system. Instead, the TDOA approach was chosen over the TOA approach since TDOA systems with a passive tag also exert high privacy-preserving characteristics similar to PDOA systems.

5.2 Localization at the Anchors or at the Tag?

When it comes to designing a localization system, the choice between active or passive tags has significant implications for privacy. A definition of active and passive localization systems can be found in Subsection 2.5.3. Whenever a tag sends a package, the package can potentially be intercepted, even when the package is encrypted. The attacker can then gain personal information from the package's header, its arrival time at the attacker's position, and further metadata. While using an STS makes it very hard for the attacker to properly receive the package and thus augments privacy, the mere presence of the package still reveals information about the tag. Therefore, a localization system relying on a completely passive tag is more privacy-preserving by design than a localization system using an active tag.

For some use cases, however, it is essential to know the position of the tag not only at the tag itself but also at an external position. Looking at the use case described in Subsection 4.2.1, the caretakers want to monitor the position of the elderly patient in real-time. This implies that the patient's location or the necessary information to calculate the position of the patient has to be transmitted to the caretaker in some way. This can be achieved by using a passive system and forwarding the information from the tag to the caretaker. Alternatively, an active system can be used where the information

is forwarded from one of the anchors to the caretaker. In the passive scenario, the information is forwarded from the tag to the caretaker or an intermediary device in a wireless manner, turning the passive tag into an active tag which undermines the privacy-preserving benefits of using a passive system.

To fulfill the requirement of real-time monitoring, it was therefore decided to implement a localization system based on an active tag. An active system transmits by default only pieces of information to calculate the tag's position over the air. In addition, compared to a passive system, real-time monitoring with an active system does not involve extra transmission steps as the active system by default collects all the data at a master anchor from which the data can easily be transferred to a network and subsequently to the caretaker.

When relaxing the condition of real-time monitoring, a passive localization system becomes viable too. For example, it would be possible to build a system where the tag itself detects abnormal or dangerous patterns inside the movement of the elderly patient and then warns the caretakers about it. Such a system maintains a higher degree of privacy than a system with an active tag while still keeping the patient safe. In order to also provide long-term movement data for the medical personnel to analyze, the data can be periodically transferred to the retirement home's network via a USB cable. Like this, real-time tracking of the elderly patient's position by an eavesdropping attacker is not possible anymore. It was therefore decided to implement and analyze both a passive and an active localization system.

5.3 TDOA Localization at the Tag

Initially, this Section describes the general setup of the passive TDOA localization system. This includes an overview of the steps performed during the localization process, the messages used, and the general communication settings utilized. Subsequently, specific components of the DW3xxx SDK are described in more detail as they are necessary to understand the design choices for the code parts of the master anchor, anchors, and tag. Lastly, code snippets of the C programs for master anchors, anchors, and tags are highlighted.

5.3.1 Structure of the Passive TDOA Localization System

The passive TDOA localization system works by sending out slightly time-shifted localization messages from the anchors that are caught by the tag, corrected by the time-shift, and forwarded to the host computer where the location of the tag is determined (see Section 5.5). The process is visualized in a sequence diagram in Figure 5.2.

First, all devices initialize themselves. This includes setting the serial peripheral interface (SPI) rate, resetting the IC, and configuring the basic IC settings as well as the communication settings (see Subsection 5.3.2). Next, the master anchor sends a synchronization message to all the anchors. After the synchronization message, each

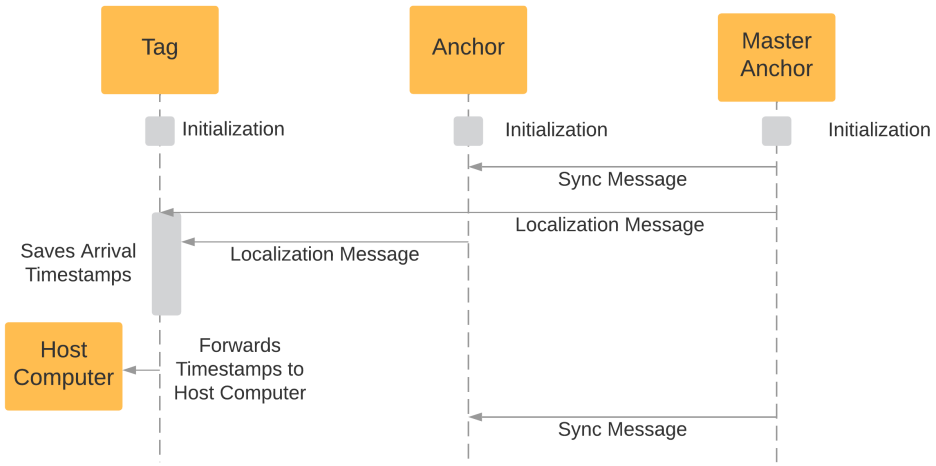

Figure 5.2: Sequence Diagram of the Passive TDOA Localization System.

anchor, including the master anchor, sends a localization message to the tag. These localization messages are slightly time-shifted compared to each other since the tag needs time to process each incoming package from the different anchors. Next, the tag collects the incoming messages and saves their arrival time. It corrects the arrival time by the anchor-specific delay and forwards the timestamps to the host computer. Except for the initialization, the described process is repeated for an infinite amount of localization exchanges.

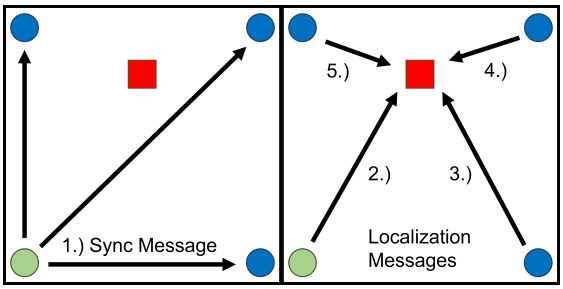

Figure 5.3: Setup of the Passive TDOA System. The master anchor is represented by a green circle, standard anchors are denoted by blue circles, and the red square represents the tag within the system. Left: Sync message sent by the master anchor. Right: Localization message sent with a short delay by each anchor.

To monitor an elderly patient inside their room, the anchors are placed in a square of three times three meters as shown in Figure 5.3. The master anchor is placed in the bottom left corner. The other three anchors are placed in the other corners, with the anchor ID increasing in the counterclockwise direction. The tag can be moved freely inside the square that is described by the anchors. To perform the localization of the tag,

the master anchor sends a sync message to all other anchors. With a short delay that is specific for each anchor, the anchors then send a localization message to the tag. The order with whom the localization messages are sent out is represented by the numbers in Figure 5.3. The tag saves the arrival time of the localization messages and calculates its position based on the arrival times. This ranging exchange can be infinitely repeated.

5.3.2 Communication Configurations

```
static dwt_config_t config = {
    5,               // Channel number.
    DWT_PLEN_128,    // Preamble length
    DWT_PAC8,        // Preamble acquisition chunk size
    9,               // TX preamble code
    9,               // RX preamble code
    1,               // sfd type 1, 8-bit decawave standard: ----+-00
    DWT_BR_6M8,      // Data rate.
    DWT_PHRMODE_STD,// PHY header mode.
    DWT_PHRRATE_STD,// PHY header rate.
    (129 + 8 - 8),   // SFD timeout
    DWT_STS_MODE_OFF,// STS disabled
    DWT_STS_LEN_64, // STS length
    DWT_PDOA_M0      // PDOA mode off
};
```

Code 5.1: UWB Communication Configurations for the Passive TDOA Localization System.

The passive communication system uses the standard communication configurations from the simple_tx example in the ex_01a_simple_tx folder from the DW3xxx SDK. These settings were selected to keep a high level of compatibility with the other examples. Notably, compatibility was valued stronger than privacy considerations since the passive TDOA localization system does not transmit personal information via UWB.

The communication settings are shown in Code 5.1. The communication channel used is channel five with a center frequency of 6489.6 MHz. Channel five is supported by both legacy devices and newer chips and therefore ensures high compatibility. A preamble length of 128 symbols was used and a preamble acquisition chunk (PAC) size of eight symbols was chosen. The TX and RX code number nine that can be found in the IEEE 802.15.4 standard was u. As SFD-type, type 1 was chosen which corresponds to the decawave specific eight symbols SFD $- - - - + - 00$. The data rate was set to 6.8 megabits per second (Mb/s) and the PHR rate to 850 kilobits per second (kb/s). PHY header mode and PHY header rate are set to standard mode. The SFD timeout is calculated with the formula $(preamblelength + 1 + SFDlength - PACsize)$. No STS is used and an arbitrary STS length of 64 is provided as this value is required. Lastly, PDOA mode is turned off.

5.3.3 Message Structure

Several different messages are used by the passive TDOA localization system, all of which have the same basic message structure. The passive system uses package configuration zero for all its packages (see Figure 2.5). In package configuration zero, no STS is added and the packages consist of the preamble, SFD, PHR, and PHY payload. Both the overall package structure and the MHR comply with the IEEE 802.15.4 standard (see Figure 2.7).

To keep the code simple, the MHR is specified inside an array as shown in Code 5.2. Note however that the DW3000 SDK also provides the files 802.15.4.h and 802.15.4.c to handle IEEE 802.15.4 compliant frames. These files include structures such as enums and structs to construct and interact with frames on the MAC level. The files also include functions to access and modify specific parts of the message's header and payload.

```
1 uint8_t localization_msg[] = {0x41, 0x88, 0, 0xCA, 0xDE, 'T', 'X', 'A', '
      2', 0x21};
2
3 #define FRAME_CTRL 0
4 #define SEQUENCE_NR 2
5 #define DEST_PAN_ID 3
6 #define DEST_ID 5
7 #define SRC_ID 7
8 #define FUNCTION_CODE 9
```

Code 5.2: Example of a Localization Message Used by an Anchor.

As an example for the MHR header and MAC payload, a localization message sent by an anchor is shown in Code 5.2. The first two bytes define the frame control sequence. Compared to Figure 2.6 in Section 2.3, the ordering of the bits inside a byte is partially backwards. This has to do with the fact that the DK saves integer numbers little-endian formatted. The way the frame control field is constructed inside the array is highlighted in Figure 5.4.

Reserved	PAN ID Compression	Acknowledge Required	Frame Pending	Security Enabled	Frame Type	Source Addressing Mode	Frame Version	Destination Addressing Mode	IE Present	Sequence Number Suppression
Bits: 7	6	5	4	3	0–2	14–15	12–13	10–11	9	8
0	1	0	0	0	001	10	00	10	0	0
Hex: 0x41						0x88				

Figure 5.4: Frame Control Field in Array.

The first hexadecimal number $0x41$ specifies the first eight bits of the frame control field. The frame type is data frame, which is denoted by the bits 001. The fields security enabled, frame pending and acknowledge required are all set to zero. The PAN glsID compression is set to one as source and destination address are 16-bit short and only the destination PAN glsID is present. The second byte $0x88$ specifies the sort addressing mode for source and destination address with the bits 10. The frame version is compliant with the IEEE 802.15.4-2003 standard which is denoted by 00. No IEs are present and the sequence number is not suppressed. For more information about the exact choice of the individual bits, consult section 7.2.2 in the IEEE 802.15.4-2020 manual [32].

Note that the sequence of bits inside the two bytes that define the frame control field inside the array in Code 5.2 is similar to how Wireshark shows the sequence of bits Figure 5.5 inside the frame control field. The only difference is that Wireshark displays the second byte of the frame control field before the first byte.

```
▾ Frame Control Field: 0x8841, Frame Type: Data, PAN ID Compression, Desti
    .... .... .... .001 = Frame Type: Data (0x1)
    .... .... .... 0... = Security Enabled: False
    .... .... ...0 .... = Frame Pending: False
    .... .... ..0. .... = Acknowledge Request: False
    .... .... .1.. .... = PAN ID Compression: True
    .... .... 0... .... = Reserved: False
    .... ...0 .... .... = Sequence Number Suppression: False
    .... ..0. .... .... = Information Elements Present: False
    .... 10.. .... .... = Destination Addressing Mode: Short/16-bit (0x2)
    ..00 .... .... .... = Frame Version: IEEE Std 802.15.4-2003 (0)
    10.. .... .... .... = Source Addressing Mode: Short/16-bit (0x2)
```

Figure 5.5: Frame Control Field in Wireshark.

After specifying the frame control, the sequence number is added to the array in Code 5.2. After that, two bytes are used to specify the PAN ID of the destination PAN network. The PAN ID is followed by a two byte destination and then two byte source address. To keep the addressing simple, the first byte of an address is used to specify the type of device. 'M' denotes the master anchor, 'A' is an anchor, and 'T' stands for tag. The second byte of the address is the ID of the specific device. At the moment, single digits denoted by universal coded character set 8 (UTF-8) encoded strings are used. Note however that in a real-world application with more than ten anchors or tags, a sequence number should be used instead of a string for the second byte.

After the source address, the payload of the package follows. For the passive TDOA system, all packages have a payload containing one byte. This byte is a custom function code that specifies the function of the frame. $0x21$ specifies a localization message and $0x11$ a synchronization message. The message ends with the FCS that is automatically added by the device before transmitting the message. Specific parts of the message's MHR and payload can be accessed via the defined indices between lines 3 and 8 in Code 5.2.

5.3.4 Managing Time on the DK

The DW3000 measures time in several different units. First of all, the IC implements the mandatory chipping rate of 499.2 MHz that is specified in the IEEE 802.15.4 standard. Accurate timestamping of message transmission and reception times is crucial for achieving a theoretical ranging accuracy in the millimeter range. To accomplish this, the 499.2 MHz clock is oversampled by a factor of 128, resulting in a sampling clock frequency of 63.8976 GHz. The sampling clock has a period of 15.65 picoseconds. During this period, the light travels 4.69 millimeters resulting in a theoretical distance measuring accuracy of roughly five millimeters. Inside the code of the DW3xxx SDK, the clock period of 15.65 picoseconds is equivalent to one device time unit (DTU). The conversion factor from DTU to seconds is defined in the file deca_device_api.h as $1.0/499.2e6/128.0$.

In other parts of the SDK such as the examples for SS-TWR and DS-TWR, time durations are specified in UWB microseconds. One UWB microsecond is defined as $512/499.2e6$ seconds or approximately 1.0256 microseconds [125]. The conversion factor from UWB microseconds to DTU is 63898. UWB microseconds are mainly used to set transmission and reception delays, for example in the SS-TWR and DS-TWR examples of the DW3xxx SDK.

Finally, the systems clock frequency is sometimes quoted as 125 MHz, which is an approximation of the actual 124.8 MHz system clock frequency. The approximate clock period is eight nanoseconds or $1/(124.8\times106)$ seconds. In addition, sometimes a one GHz phase locked loop (PLL) is referenced, which is an approximation of the actual PLL frequency of 998.4 MHz.

Assuming the DW3000 works similarly to the DW1000, all clocks are generated with the help of a 38.4 MHz oscillator crystal and PLLs [126; 127]. The 38.4 MHz clock signal can also be provided from an external source via the SYNC input pin [37]. This allows for external synchronization of multiple ICs, for example in order to send a synchronized signal for TDOA localization.

For low-power operations, the DW3000 is also equipped with a 20 kilohertz (kHz) oscillator. It is used to time the sleep state of the IC [127]. Note that there is also a deep sleep state of the IC in which all clocks are turned off.

The DW3000 SDK provides multiple functionalities for handling time. First of all, it is important to note that some of the functions are only available in certain operating states of the IC. For example, querying the current system time can only be done in IDLE_PLL, TX and RX mode as the function requires DW3000 to operate at nominal 125 MHz rate. More about the different states of the DW3000 can be found in the DW3000 family user manual [37].

As already mentioned, the system time of the DW3000 can be checked with the function dwt_readsystimestamphi32() inside the deca_device_api.h. The function returns the high 32 bits of the 40 bits system counter. The 40 bits system counter is given in DTU. Therefore, the system counter is provided by the function dwt_readsystimestamphi32() in units of roughly eight nanoseconds. The biggest time that can be held by the system

counter is close to 17.2 seconds.

There are no functions in the deca_device_api.h that return the low order byte of the system counter as this byte is always zero. This is due to the fact that the DW3000 operates at a frequency of approximately 125 MHz in IDLE_PLL mode. If the system counter is queried in other modes than IDLE_PLL, TX or RX, dwt_readsystimestamphi32() returns zero.

Next, the functions get_tx_timestamp_u64() and get_rx_timestamp_u64() return the full 40-bit timestamp as an unsigned 64-bit integer variable of a message transmission or reception time. These functions are defined in shared_functions.c. Note there are multiple variations of these functions in deca_device_api.h, returning either the high 32-bit, low 32-bit, or the complete timestamp.

The timestamp that is returned via the get_timestamp_u64() functions is saved to the timestamp buffer during frame transmission or reception precisely timed to when the RMARKER of the message passes through the antenna (see Figure 2.5). The timestamp is created with the help of the sampling clock, therefore filling out the whole 40 bits and having accuracy in the range of a DTU. When configuring the antenna delay with the functions dwt_settxantennadelay() and dwt_setrxantennadelay() in DTUs, the antenna delay is automatically added to the tx and rx timestamps. For the SS-TWR and DS-TWR examples of the DW3xxx SDK, an average antenna delay of 16385 DTUs is set. Note however that for accurate two-way ranging, the antenna delay of each device should be configured separately.

Another functionality provided by the DW3000 SDK is the configuration of delayed message transmission and reception. This is done by setting a reference time at which the IC starts the transmission or reception. This can be done with the help of the function dwt_setdelayedtrxtime(), which is used both for delaying TX and RX operations. The function takes the high 32 bits of the system time as input. This timestamp is called the reference time. In addition, the lower order bit of the reference time is ignored, essentially setting the TX or RX time in units of approximately eight nanoseconds.

For transmission, the reference timestamp at which the message is sent does not include the antenna delay. Instead, the antenna delay will be added later in the transmission process. For reception, the reference time specifies the time at which the receiver will be turned on.

The delayed transmission mode can be activated by giving the dwt_starttx() the parameter DWT_START _TX_DELAYED. Similarly, a delayed reception is invoked by providing the function dwt_rxenable() with the parameter DWT_START_RX_DELAYED. Empirical testing showed that setting a transmission delay too close or too far away from the current system time of the delay setting causes the transmission to fail. The minimum acceptable delay, relative to the reference time at which it was set, is around 500 UWB microseconds or around 31949000 DTU. The lower bound is likely related to the processing speed of the IC. This is further backed up by the two-way ranging examples that mention platform-specific processing delays. Regarding the upper bound, it was empirically tested that the transmission fails when the delay is larger than 0x80010000 DTUs compared to the current system time. The reason for this upper bound remains unclear and requires further investigation.

Unsuccessful delayed transmissions need to be handled. This can be done by checking the return value of the dwt_starttx() function and implementing the appropriate behavior according to the successful or unsuccessful transmission.

Lastly, the delay can also be specified by first setting a reference time with dwt_setreferencerxtime() to which the delay set with dwt_setreferencerxtime is added to. When no reference time is set, dwt_setdelayedtrxtime() sets an absolute timestamp.

Apart from delaying the transmission or reception of a message, there are also functions provided in the DW3xxx SDK that can delay the execution of any code parts. Since these functions are implemented with the help of the 20 kHz oscillator, their accuracy is much lower than the eight nanosecond accuracy of the dwt_setdelayedtrxtime() function. The function that is used most often in the DW3xxx SDK is called Sleep() and is defined in the port.c file. Sleep() takes a delay in milliseconds as input. Sleep() itself is implemented with the help of nrf_delay_ms() which in turn uses the function nrf_delay_us(). Both of these functions are implemented in the nrf_delay.h file. The function nrf_delay_ms() takes a delay in milliseconds as input and nrf_delay_us() takes a delay in microseconds as input.

5.3.5 UART Setup

To transfer data from the DK to the host computer, an UART connection is used. Two helper files, uart_setup.c, and dimi_print.c are utilized to manage the UART connection.

The uart_setup.c file includes two functions, uart_setup() and uart_handle_error(). The uart_setup() function initializes the UART connection by first defining standard parameters of the connection and then initializing the UART connection from the DK side. This process includes setting TX, RX, request to send (RTS), and clear to send (CTS) pins as well as setting the baudrate to 115200 bauds per second. The uart_handle_error() performs error handling by using standard functions from the nRF SDK.

The file dimi_print.c has been reused from a previous project and implements multiple functions to write data to the UART connection. With the help of the function app_uart_put() from the nRF SDK, sequences of characters as well as integers of different lengths can be written to the UART connection. One function in the dimi_print.c file is dimi_print_buffer(). The function takes a pointer to an array of uint8_t symbols as well as the array's length as input, prints the content of the array to the UART connection, and prints the delimiter "zueri" at the end of the buffer to the UART connection.

The functions of both files have only been tested and validated for the nRF52840 and DWM3000EVB. To use the UART connection with the DWM3001CDK, the pins would have to be changed and project configurations of the .emProject file would need to be updated.

5.3.6 Master Anchor

```
1
2 static uint8_t localization_msg[] = {0x41, 0x88, 0, 0xCA, 0xDE, 'T', 'X',
      'M', '0', 0x21};
3 static uint8_t sync_msg[] = {0x41, 0x88, 0, 0xCA, 0xDE, 'A', 'X', 'M', '0
    ', 0x11};
4
5 #define SYNC_INTERVAL 500 // milliseconds
6 #define ANCHOR_DELAY 1000 // UWB microseconds
7
8 uint64_t tx_ts;
9 uint32_t tx_time;
10
11 // before: application entry point, initialization
12 while (1)
13     {
14     //send the sync message
15     dwt_writetxdata(sizeof(sync_msg), sync_msg, 0);
16     dwt_writetxfctrl(sizeof(sync_msg)+FCS_LEN, 0, 0);
17     dwt_starttx(DWT_START_TX_IMMEDIATE);
18     while (!(dwt_read8bitoffsetreg(SYS_STATUS_ID, 0) &
     SYS_STATUS_TXFRS_BIT_MASK)){ };
19     dwt_write8bitoffsetreg(SYS_STATUS_ID, 0, SYS_STATUS_TXFRS_BIT_MASK);
20     //configure the delay
21     tx_ts = get_tx_timestamp_u64();
22     tx_time = (tx_ts+(ANCHOR_DELAY*UUS_TO_DWT_TIME))>> 8;
23     dwt_setdelayedtrxtime(tx_time);
24     // send delayed localization message
25     int ret;
26     dwt_writetxdata(sizeof(localization_msg), localization_msg, 0);
27     dwt_writetxfctrl(sizeof(localization_msg)+FCS_LEN, 0, 0);
28     ret = dwt_starttx(DWT_START_TX_DELAYED);
29     /* If error, abandon frame and goto next one. */
30     if (ret == DWT_SUCCESS){/*...*/}
31     Sleep(SYNC_INTERVAL);
32     }
```

Code 5.3: Main Loop of the Master Anchor that Sends Synchronization and Localization Messages.

The code snippet in Code 5.3 shows the core parts of the master anchor's code. First, two messages are initialized that are periodically broadcasted by the master anchor. The first message, localization_msg, is sent to the tag so that the tag can localize itself based on the arrival time of the message. The same message is also sent by the other anchors with a slight time-shift. The function code of the localization_msg is specified inside the first byte of the payload and has the value $0x21$. The value $0x21$ determines that this message should be used for localization. The second message, sync_msg, is used to transmit a synchronization signal to the other anchors based on which they time the transmission of their individual localization message. The function code of the sync_msg

is $0x11$, which specifies this message should be used for synchronization purposes.

Next, the synchronization interval and the anchor delay are specified at line five and six. The synchronization interval is given in milliseconds and it times how long the master anchor waits in-between sending synchronization messages to the anchors. The actual synchronization interval is longer than the value specified in SYNC_INTERVAL as the execution of the code inside the main loop takes time as well. The sync_interval is executed with the help of the Sleep() function on line 31. More about timing functions such as Sleep() can be found in Subsection 5.3.4. Next, the anchor delay is specified in UWB microseconds. It is used to delay the transmission of the localization message after the transmission of the sync message. For the master anchor, this is necessary as the operations at lines 18 and 19 take a variable amount of time of plus or minus 1000 nanoseconds. If no delay was specified, an error with a magnitude of 1000 nanoseconds would be introduced making accurate localization impossible.

The delay of 1000 UWB microseconds at line 6 was chosen based on empirical testing and the SS-TWR and DS-TWR examples in the DW3xxx SDK. This delay is large enough to ensure the proper transmission of the localization message by the IC while still being considerably smaller than the delay of anchor 'A0', preventing clashes with its localization message.

Next, in the omitted code parts after defining the synchronization interval and anchor delay, the device is initialized and the communication configurations are set. At line 12, the main loop of the master anchor starts. The loop infinitely sends a synchronization message tightly followed by a localization message. To send the synchronization message, the content of the sync_msg array is written to the transmission buffer with the help of the function dwt_writetxdata(). The function takes the length of the sync_msg, a pointer to the beginning of the message, and the buffer offset as arguments. Next, the function dwt_writetxfctrl() configures the TX frame control register before the transmission. It takes the frame length including the FCS length, the buffer offset, and a variable specifying whether or not the frame is a ranging frame as arguments. After that, the transmission is immediately started by giving the dwt_starttx() function the keyword DWT_START_TX_IMMEDIATE. Next, the loop at line 18 waits for the end of the transmission. This is detected by observing the status buffer waiting for a TX frame sent event. At the next line, the TX frame sent event in the status buffer is cleared.

In the following steps beginning at line 21, the delayed transmission of the tx_message is prepared. First, the timestamp of the transmission of the synchronization message is determined with the help of the function get_tx_timestamp_u64(). The function returns a 40-bit timestamp in DTUs that is saved to a uint64_t variable. Next, the time of transmission for the delayed message is calculated. The time of transmission needs to be specified in DTUs. This is done by converting the anchor delay from UWB microseconds to DTU by multiplying it with the conversion factor 63898 and adding it to the time of transmission of the synchronization message. Additionally, the calculated time of transmission for the localization message is shifted by eight bits to the right as the dwt_setdelayedtrxtime takes only the high 32-bits of the transmission time as an input parameter. After setting the delay, the message is written to the TX buffer and the TX

frame control register is configured. Then, a delayed transmission is started by handing the keyword DWT_START_TX_DELAYED to the dwt_starttx() function. The device will wait until the system time that was specified with the dwt_setdelayedtrxtime is reached and then start to initialize the transmission.

At line 30, the if statement checks that the dwt_starttx() function has not returned an error. If no error was returned, the IC is polled for the TX sent event. After that, the event is cleared and the frame sequence numbers inside the sync and localization message are augmented. The process inside the while loop is repeated forever.

5.3.7 Anchor

The central parts of the code of an anchor can be found in Code 5.4. After the initialization of the device, the UWB receiver is activated at line 8. At line 9, the IC is polled until a frame is properly received or an error occurred. If no error occurred, the good RX frame event is cleared at line 9 and the frame length is determined at line 13. If the frame length is shorter than FRAM_LEN_MAX which is defined as 127 bytes in shared_defines.h, the frame is read into the rx_buffer with the function dwt_readrxdata() at line 18. This function takes a pointer to the buffer the data will be read into, the frame length, and the offset inside the rx_buffer to read from as input parameters.

At line 22, it is checked if the received frame was a synchronization message. If this is the case, the delayed transmission of the localization message for the tag is triggered. The sending of the localization message works the same way as sending the localization message at the master anchor (see Subsection 5.3.6). Note that each anchor has a specific ANCHOR_DELAY, which is 5000 for anchor 'A0', 10000 for anchor 'A1', 15000 for anchor 'A2', and 20000 for anchor 'A3'. Again, these delays were chosen to be small enough to minimize the clock drift between the arrival of the synchronization message and the transmission of the localization message but still large enough so that the tag has sufficient time to process all the localization messages. What is also different compared to the master anchor is that the sequence number of the anchor's localization message is taken from the synchronization message sent by the master anchor. At line 30, the sequence number of the synchronization message sent by the master anchor is taken as the sequence number for the localization message.

```
uint8_t rx_buffer[50];
uint64_t rx_ts;
uint32_t tx_time;

while (1)
    {
      // Activate RX and receive frames
      dwt_rxenable(DWT_START_RX_IMMEDIATE);

      while (!((status_reg = dwt_read32bitreg(SYS_STATUS_ID)) &
      (SYS_STATUS_RXFCG_BIT_MASK | SYS_STATUS_ALL_RX_ERR))){};
      if (status_reg & SYS_STATUS_RXFCG_BIT_MASK){
        dwt_write32bitreg(SYS_STATUS_ID, SYS_STATUS_RXFCG_BIT_MASK);
        frame_len = dwt_read32bitreg(RX_FINFO_ID) &
        RX_FINFO_RXFLEN_BIT_MASK;

        if (frame_len <= FRAME_LEN_MAX){
            dwt_readrxdata(rx_buffer, frame_len, 0);
        }
        // if a sync frame was received
        // start delayed transmission of localization message
        if (rx_buffer[FUNCTION_CODE] == 0x11)
        {
          int ret;
          rx_ts = get_rx_timestamp_u64();
          tx_time = (rx_ts+ANCHOR_DELAY*UUS_TO_DWT_TIME) >> 8;
          dwt_setdelayedtrxtime(tx_time);

          // take sync seq nr as localization seq nr
          localization_msg[SEQUENCE_NR] = rx_buffer[SEQUENCE_NR];

          dwt_writetxdata(sizeof(localization_msg), localization_msg, 0);
          dwt_writetxfctrl(sizeof(localization_msg)+FCS_LEN, 0, 0);
          ret = dwt_starttx(DWT_START_TX_DELAYED);

          if (ret == DWT_SUCCESS)
          {
              while (!(dwt_read32bitreg(SYS_STATUS_ID) &
              SYS_STATUS_TXFRS_BIT_MASK)){ };
              dwt_write32bitreg(SYS_STATUS_ID,SYS_STATUS_TXFRS_BIT_MASK);
          }

        }
      }
```

Code 5.4: Example of a Localization Message Used by an Anchor.

5.3.8 Tag

```
/* Loop forever initiating ranging exchanges. */
while (1)
{
  /*A frame has been received and written into rx buffer*/
  if (rx_buffer[FUNCTION_CODE] == 0x21)
  {
    // Add data to ts buffer
    ts_buffer[anchors_received*package_length + 0] = rx_buffer[SRC_ID];
    ts_buffer[anchors_received*package_length + 1] = rx_buffer[SRC_ID+1];
    ts_buffer[anchors_received*package_length + 2] = 'T';
    ts_buffer[anchors_received*package_length + 2+1] = tag_id;
    ts_buffer[anchors_received*package_length + 4] = rx_buffer[
   SEQUENCE_NR];
    rx_ts = get_rx_timestamp_u64();

    // correct for anchor delays
    if (rx_buffer[SRC_ID+1] == '0' && rx_buffer[SRC_ID] == 'A'){
    rx_ts = rx_ts - ANCHOR_DELAY_0*UUS_TO_DWT_TIME;
    }
    else if (rx_buffer[SRC_ID+1] == '1' && rx_buffer[SRC_ID] == 'A'){
    rx_ts = rx_ts - ANCHOR_DELAY_1*UUS_TO_DWT_TIME;
    }
    else if (rx_buffer[SRC_ID+1] == '2' && rx_buffer[SRC_ID] == 'A'){
    rx_ts = rx_ts - ANCHOR_DELAY_2*UUS_TO_DWT_TIME;
    }
    /*write rx ts into ts_buffer*/
    uint8_t i;
    for (i = 0; i < 5; i++)
    {
        ts_buffer[anchors_received*package_length + 5+i] = (uint8_t)rx_ts
   ;
        rx_ts >>= 8;
    }
    anchors_received++;
    // if packages are received from all anchors, pass to host computer
    if (anchors_received == (number_of_anchors)) {
    for (int i = 0; i < number_of_anchors; i++){
       dimi_print_buffer(&ts_buffer[i*package_length], package_length);
    }
    ts_buffer[0] = '\0';
    anchors_received = 0;
    //Sleep a bit in case we are not in sync anymore as to not receive
   the other packages
    Sleep(15);
       }
    }
  }
}
```

Code 5.5: Example of the Code for a Passive Tag.

The main loop of the passive tag determines the arrival time of the localization messages from the anchors, saves them into a buffer called tx_buffer, and forwards the content of the tx_buffer to the host computer after all localization messages from the anchors have been received. An overview of this main loop is shown in Code 5.5.

After receiving a frame and writing it into the rx_buffer, the tag checks at line 5 if the function code of the received package specifies a localization package. If this is the case, the tag saves the information such as the anchor ID at line 8 and 9, the tag ID of the tag itself at line 10 and 11, the sequence number of the localization message at line 12 and the time of arrival at the tag from the rx_buffer into the tx_buffer. The time of arrival of the localization messages are found with the help of the function get_rx_timestamp_u64(); the function returns a 40-bit timestamp in DTUs. The time of arrival is then corrected by an anchor-specific delay (see Subsection 5.3.7) with the if statement at lines 16 to 24. For this, the ID of the anchor is determined and its hard-coded anchor delay is subtracted from the time of arrival.

After the correction, the forty-bit timestamp is written to the ts_buffer with the help of a for-loop and a right-shift operator. At line 32, counter anchors_received is augmented by one. Next, at line 34, it is checked if the same number of localization messages has been received as anchors exist. The number_of_anchors is hard-coded at the beginning of the tag.c file. If the same number of messages as number_of_anchors has been received, the content of the ts_buffer is transferred to the host-computer with the help of the dimi_print_buffer function and a for-loop. After that, the ts_buffer is emptied at line 38 and the number of anchors received is set to zero. In addition, the anchor sleeps for 15 milliseconds to make sure the buffer does not fill up with other packages in case of a missed package.

5.4 TDOA Localization at the Anchors

This Section presents the general setup of the passive TDOA localization systems. First, an overview of the steps involved in the localization process is given. Second, the messages and communications settings used are further described. Thirdly, more details are given about specific aspects of the active localization system, in particular the AES engine used for encryption. Finally, code snippets that highlight the working of the master anchor, anchor, and tag as described.

5.4.1 Structure of TDOA Localization at Anchors System

The basic structure of the TDOA localization system is identical to the TDOA approach described in Subsection 2.5.3 (see Figure 5.6). The tag sends a localization package to all anchors. The anchors receive the localization package, save the time of arrival of the tag's package and forward the arrival time to the main anchor which will be called the master anchor. The master anchor collects the packages from the other anchors and saves the arrival time to an array. When the localization package's arrival time is received from all anchors, the master anchor forwards the timestamps to the host computer where the localization calculations take place (see Section Subsection 5.5.1 for more details on the localization calculations).

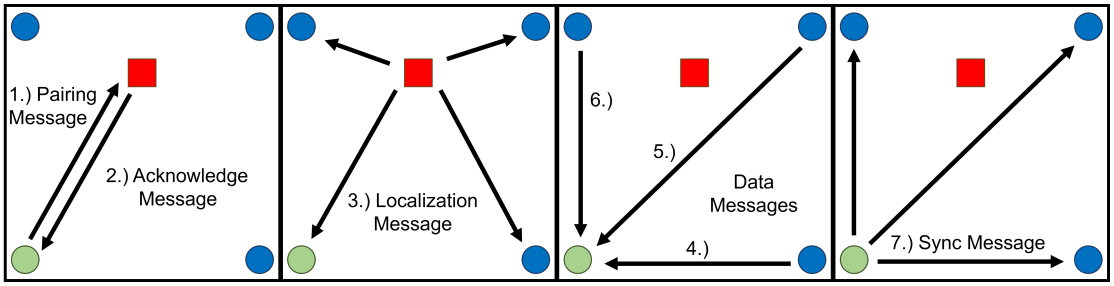

Figure 5.6: Steps involved in the Active TDOA Localization System.

After the reception of all the anchor's packages, the master anchor sends a synchronization package to the other anchors. The anchors save the arrival time of the synchronization package to correct their local clocks with it. This is necessary to keep the anchor's clocks synchronized. The synchronization signal is sent as closely as possible to the next expected localization package from tag without colliding with it. This helps keeping the clock drift between the anchors as small as possible and thus augments localization accuracy.

As the packages send from the anchors to the master anchor contain personal information as payload, the timestamp, the packages can be potentially be eavesdropped on by an attacker. It is therefore important to encrypt the payload of the packages. This is done via the AES engine that is part of the DW3000's SDK.

Finally, before the localization exchanges start, the master anchor and the tag perform a form of pairing. During this pairing, the tag chooses a temporary ID which is used for the duration of the localization session. The goal behind the temporary ID is to make it difficult for a potential attacker to collect personal information about the tag over multiple sessions. At the same time, it should still be possible for the medical personnel to analyze location data over multiple sessions. To achieve this, the temporary tag ID is generated with the help of the AES engine. The master anchor sends a nonce to tag with which the tag encrypts its original ID and a shared key. The master anchor can then based on the new temporary address embedded in the acknowledge message find the association with the original tag ID.

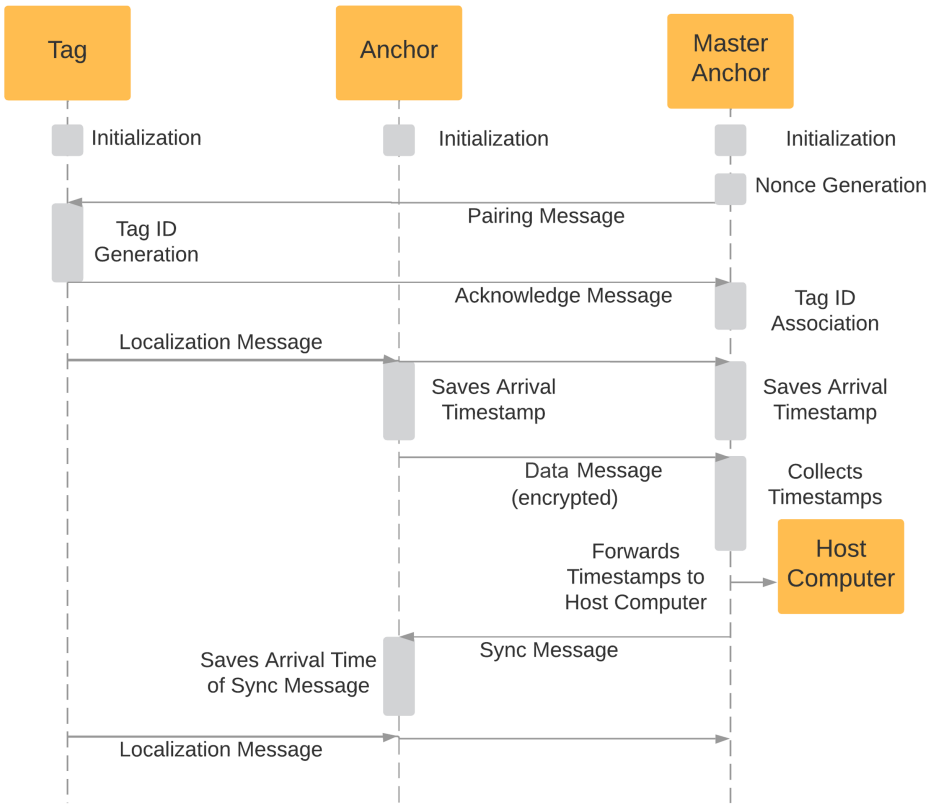

Figure 5.7: Sequence Diagram for the Active TDOA Localization System.

The chronological order of the packages is visualized in Figure 5.7. First, all devices are initialized. Then, the master anchor generates a nonce based on its system time and a counter which it forwards to the tag via a pairing message. The tag reads the nonce out of the pairing message and generates its temporary tag ID with it. Then, the tag sends an acknowledge message to the master anchor. The master anchor takes the temporary ID out of the acknowledge message and associates it with the original tag ID with the help of the nonce and the encryption engine.

Then, the localization exchange starts which can be infinitely repeated. First, the tag sends a localization messages to all anchors. The anchors save the arrival time of the tag's localization message and forward it via an encrypted timestamp package to the master anchor. The master anchor collects the timestamp packages, decrypts them and forwards anchor ID, tag ID and the timestamps via an UART connection to the host computer. Lastly, the master anchor sends a synchronization package to the other anchors shortly before the next localization exchange is expected to start.

5.4.2 Active Communication Configurations

```
static dwt_config_t config = {
    9,                /* Channel number */
    DWT_PLEN_256,     /* Preamble length */
    DWT_PAC16,         /* Preamble acquisition chunk size*/
    11,                /* TX preamble code */
    11,                /* RX preamble code */
    3,                /* 3 for 4z 8 symbol SDF type */
    DWT_BR_6M8,       /* Data rate */
    DWT_PHRMODE_STD, /* PHY header mode */
    DWT_PHRRATE_STD, /* PHY header rate */
    (257 + 16 - 16),   /* SFD timeout */
    DWT_STS_MODE_1 | DWT_STS_MODE_SDC, /* deterministic STS mode */
    DWT_STS_LEN_64,/* STS length */
    DWT_PDOA_M0       /* PDOA mode off */
};
```

Code 5.6: UWB Communication Configurations for the Active TDOA Localization System.

In Code 5.6, the UWB communication configurations are shown for the master anchor, the anchors, and the tag. As the communication channel, channel nine was chosen. This channel is newly supported by the DW3000, as opposed to the DW1000 [37]. As older devices do not support channel nine, it requires a potential attacker to use up-to-date hardware in order to eavesdrop on the communication. For all further settings, non-standard configuration settings were chosen to further augment the effort a potential attacker has to exert in order to observe the UWB communication. Standard configurations can be found in the examples provided the DW3xxx SDK and the DW3000 family user manual [39; 37].

For the preamble length and PAC size, the values 256 and 16 were chosen. Transmitter (TX) and receiver (RX) preamble codes are set to 11. As the start of frame delimiter (SFD), the standard 8 symbol type for the IEEE 802.15.4z standard is chosen. Standard PHY header mode and rate are chosen. The SFD is calculated with the formula $preamble_length + 1 + SFDlength - PACsize$.

As STS mode, STS mode one is chosen which places the STS directly behind the SFD. This mode ensures that if the STS is not correctly processed for example because of a missing key, the whole MHR and payload containing personal data cannot be read. As implementing full STS mode would have been out of scope for this work, deterministic STS mode was chosen instead. STS length was set 64 and PDOA mode was turned off.

5.4.3 Message Structure

The active TDOA localization system uses two message structures, longer encrypted messages that have an auxiliary security header and MIC and shorter, unencrypted messages. The longer, encrypted messages are used for the data packages sent by the

anchors to the master anchor, and the shorter, unencrypted messages are used for all other exchanges such as the pairing request of the master anchor or the localization message sent by the tag. Similarly to the passive TDOA localization system, the MHR and payload of the messages specified inside arrays (see also Subsection 5.3.3). Specific parts of the MHR and payload can be accessed with the help of variables defined below the arrays. The first message structure is used for the encrypted data packages sent by the anchors. An overview of the general structure of the encrypted data packages is shown in Figure 5.8. Further, the content of the frame control field of data packages is visualized in Figure 5.9.

Field	Frame Control	Sequence Number	Destination PAN ID	Destination Address	Source Address	Auxiliary Security Header			Payload	MIC	FCS
						Security Control	Frame Counter	Key Identifier			
Bytes	2	1	2	8	8	1	4	1	8	16	2
Index	0	2	3	5	13	21	22	26	27	35	37

Figure 5.8: Frame Structure, Number of Bytes and Indices of Data Messages.

The first difference between the passive and active TDOA message structure is that the active system uses eight-byte addresses compared to the two-byte addresses of the passive system. This is because the creation of the nonce in AEAD mode requires eight-byte source addresses (see Figure 2.11). In order to keep all addresses the same length, both source and destination addresses were set to a length of eight bytes for both encrypted and unencrypted packages. However, to make it possible that the localizer program on the host computer can handle both data from the active and the passive localization system, only the two first bytes of the eight-byte addresses are used for this use case. In a real-world application, the whole address space should be used as the address. The next difference is that the encrypted data messages have an auxiliary security header. This header contains information relevant to the used encryption in the security control field, a frame counter for the nonce generation, and a key identifier.

Reserved	PAN ID Compression	Acknowledge Required	Frame Pending	Security Enabled	Frame Type	Source Addressing Mode	Frame Version	Destination Addressing Mode	IE Present	Sequence Number Suppression
Bits: 7	6	5	4	3	0–2	14–15	12–13	10–11	9	8
0	0	0	0	1	001	11	10	11	0	0
Hex: 0x09						0xEC				

Figure 5.9: Frame Control Field of the Data Messages.

As content of the security field, The data messages contain the value $0x0F$. As shown in Figure 5.10, this value denotes that the frame has a security level of seven which corresponds to 111 in binary, meaning that the frame has both an encrypted payload and uses a MIC of length sixteen bytes. More information about the values the fields of the auxiliary security header can take can be found in the IEEE 802.15.4 standard [32]. The key identifier mode field is set to 01, meaning that the key is determined from the one-byte key index field. The frame counter is not suppressed and no ASN is used for the nonce generation, therefore the fields Frame counter suppression and ASN in the nonce are both set to zero.

Reserved	ASN in Nonce	Frame Counter Suppression	Key Identifier Mode	Security Level
Bits: 7	6	5	3–4	0–2
0	0	0	1	111
Hex: 0x0F				

Figure 5.10: The Security Field of the Data Messages.

All other messages used by the active TDOA localization system have the same basic structure as the encrypted messages except that they do not have the auxiliary security header and that they are missing the MIC. The payload of the unencrypted messages starts at index 21. An overview of all messages used for the active localization system is shown in Figure 5.11.

Message Type	Frame Control Field	Destination Address	Source Address	Security Control Field	Payload Length	Total Message Length
Pairing	0xEC21	TX	M0	-	13	36
Acknowledge	0xEC02	0	Tvar	-	0	23
Localization	0xEC01	AX	Tvar	-	8	31
Timestamp	0xEC09	MX	AX	0x0F	8	53
Synchronization	0xEC01	AX	M0	-	4	27

Figure 5.11: Messages Used by the Active TDOA Localization System.

5.4.4 Cryptography, STS and AES Engine

A standard approach to increase the privacy-preserving characteristics of a communication system is by using cryptographic methods. For this, the IEEE 802.15.4 standard provides three security services: data confidentiality, data authenticity and replay protection [32; 37]. The DW3xxx SDK provides these services with the help of an AES engine.

The AES engine can perform the encryption and decryption of packages with the help of an IEEE compliant CCM* encryption core. The engine can use keys of length 128, 192 or 256 bits. Additionally, the AES engine also provides the functionality to automatically add a MIC to the end of an encrypted message for data authenticity. Moreover, the AES engine can directly encrypt and decrypt messages inside the TX and RX buffers, enabling rapid message processing speeds.

To effectively profit from the services provided by the AES engine, the first step is to identify packages that contain personal information. For the active TDOA localization system, these are the localization packages sent by the tag and the data packages sent by the anchors. The localization package of the tag contains the source address of the tag giving indications about the originator of the package. However, as the source address is part of the package's header, it cannot be encrypted as this would make the package unprocessable for other devices. Instead, it was decided to use dynamic addresses for the localization packages of the tag as further described in Subsection 5.4.5.

The data packages also contain personal information as their body contains the arrival timestamp of the localization message at that specific anchor as well as the tag ID of that localization message. As these pieces of information are part of the body of the data packages instead of the header, they can be encrypted during message transmission.

It was decided to encrypt the data packages with the help of an IEEE 802.15.4 compliant use of the AES engine. Firstly, this meant that a CCM* core was chosen to handle the encryption. As encryption key, a 128-bit shared secret key was chosen and the nonce was generated based on the nonce pattern in Figure 2.11 for the AEAD nonce for non-TSCH mode. Additionally, a 16-byte MIC was added at the end of the encrypted messages to ensure the highest authenticity level possible within the IEEE 802.15.4 standard.

Apart from the primary function of the AES engine of encrypting and decrypting messages, the AES engine can also be used to generate the STS. As already mentioned in Section 2.3, the STS consists of a sequence of randomised pulses that are added either after the SFD or the PHY payload mainly to prevent distance reducing attacks but also to add better timestamping accuracy and to contribute to the overall message authenticity.

For the active TDOA localization system, it was decided to use a deterministic STS with package configuration one (see Figure 2.5) for the following rationale: First of all, the STS was introduced in the IEEE 802.15.4z amendment in 2020, making it a relatively new addition. Therefore, using the STS places higher demands on the hardware

requirements for a potential attacker, as the sniffing devices used need to be up to date.

Secondly, using the STS with package configuration one means that UWB module expects a specific STS while processing incoming packages. When there is a mismatch between expected and actual STS, there is a drop in reception performance of the package's data due to poor correlation performance [37]. This makes it harder for a potential attacker to properly eavesdrop on traffic that uses an unknown STS.

Additionally, when testing the implementation of the STS processing within the DW3xxx SDK, it was noted that when receiving an unexpected STS in package mode one, the IC would not properly process the package and throw an error during the dwt_readrxdata() command. This is why in the DS-TWR examples with deterministic STS (example 5c and 5d), it is first checked that the STS quality is high with the command dwt_readstsquality() before processing a package. However, the exact behavior of the IC when handling bad STS quality was not thoroughly evaluated and there may be an option provided by the SDK to read the content of a package with bad STS quality. However, the evaluation of the SDK showed that adding the STS will increase the effort a potential attacker has to exert to read the package containing an STS.

Lastly, to remain within the scope of this work, a deterministic STS was used instead of comprehensive, non-deterministic STS, where a new STS is generated for each package with the help of a shared secret key and a nonce that includes a counter. The deterministic STS does not effectively protect against replay attacks and does not guarantee authenticity. Once an attacker has figured out that a deterministic STS is used, he or she can read incoming packages and use the deterministic STS for replay attacks.

However, implementing the deterministic STS can be done with low effort to demonstrate some of the advantages of using an STS. In addition, implementing the deterministic STS makes it possible to test out how the UWB module processes the STS. For future projects, it is important to also build a system that uses a non-deterministic STS. The build process will help understand all the challenges involved in implementing the non-deterministic STS, such as for example handling the local counters for the nonce generation when a package with a STS is not received by one of the anchors.

5.4.5 Dynamic Addresses

One privacy concern involves the long-term tracking of individuals based on the tag's ID. For example, a patient can be tracked based on the tag ID that is embedded in the source address of the localization messages sent by the tag. To address the concern of long-term tracking, [15] suggests implementing dynamic addresses for the tag.

There are multiple challenges involved in using dynamic tag addresses. First of all, the generation of the dynamic addresses should be achievable with reasonable effort. Secondly, the address generation should be non-deterministic to prevent attackers from deducing the pattern behind the dynamic address generation. Lastly, it is important to ensure that the localization system itself is still able to perform long-term tracking for medical monitoring purposes.

Based on these requirements, it was decided to generate a new tag address at the beginning of each activation of the TDOA localization system. Further, to enable long-

term tracking for the master anchor, a system was needed to associate the dynamic tag ID with the original, static tag ID. This was achieved by introducing a pairing between master anchor and tag, during which the dynamic tag ID is generated with the help of the AES engine. The basic idea behind the pairing is that the master anchor periodically broadcasts a nonce based on which the tag can generate its dynamic address by passing its original address, a shared key, as well as the nonce to the encryption engine. The master anchor, knowing the nonce and the shared key, can then re-associate the temporary tag ID with the original tag ID.

5.4.6 AES helper

```
1  static dwt_aes_key_t     keys_options[3]=
2  {
3      {0x53504552, 0x4150524f, 0x4c4f4e47, 0x4c495645, 0x00000000, 0
       x00000000, 0x00000000, 0x00000000}, ...
4  };
5  static dwt_aes_config_t aes_config=
6  {
7      .key_load           = AES_KEY_Load,
8      .key_size           = AES_KEY_128bit,
9      .key_src            = AES_KEY_Src_Register,
10     .aes_core_type      = AES_core_type_CCM,
11     .mic                = MIC_16,
12     .aes_key_otp_type   = AES_key_RAM,
13     .key_addr           = 0
14 };
15
16 int initialise_aes_tx(dwt_aes_job_t *aes_job_tx, uint8_t *header,uint8_t
       header_length, uint8_t *payload, uint16_t payload_length, uint8_t *
       nonce) {
17     aes_job_tx->mode        = AES_Encrypt;
18     aes_job_tx->src_port    = AES_Src_Tx_buf;
19     aes_job_tx->dst_port    = AES_Dst_Tx_buf;
20     aes_job_tx->nonce       = nonce;
21     aes_job_tx->header      = header;
22     aes_job_tx->header_len  = header_length;
23     aes_job_tx->payload     = payload;
24     aes_job_tx->payload_len = payload_length;
25     aes_job_tx->mic_size = 16;
26 }
```

Code 5.7: Code Snippet that Performs the Pairing of the Tag with the Master Anchor.

The aes_helper.c file provides multiple variables and functions that help handle the encryption engine of the DW3xxx SDK. First of all, the aes_helper file provides an array called key_options with keys for encryption and decryption. The only key used for the implementation of the active TDOA localization is the key with the index zero in the first key struct of the key_options array. The key is shown in Code 5.7 at line 3. It is

a 128-bit custom key that is not used in any of the examples provided by the DW3xxx SDK. The last four bytes are set to zero as the dwt_aes_key_t can also be used to store 256-bit keys.

Next, the aes_helper file provides an aes_config struct where all configuration settings needed for the encryption engine are defined. First of all, AES_KEY_Load keyword specifies that the key is loaded from the key register. Next, the key size is set to 128 bits. After that, the key register is given as the source of the key, rather than a position in the random-access memory (RAM) or the one-time pad (OTP). As block cipher mode, CCM* is chosen as it is the required mode for the IEEE 802.15.4 standard and the mode used in the DW3xxx SDK examples. Next, the MIC size is set to 16. After that, the aes_key_otp_type is set to AES_key_RAM. Note that for this variable, only AES_key_RAM and AES_key_OTP are the only two options offered. As the AES example from the DW3xxx SDK uses RAM here, it was decided to also use RAM. It was noted that the comment about the usage of this enum in deca_device_api.h is confusing, as it specifies that aes_key_otp_type and key_load need to match, but then goes on to give the example that for OTP, the otp_type should be set to RAM. Lastly, the key address is set to zero which indicates that the key is placed at the beginning of the key register without any offset.

Apart from the variables and definitions, the aes_helper file also offers some useful functions for handling the encryption and decryption of packages. One of these functions is the initialise_aes_tx() function at line 16 in Code 5.7. It takes an aes_job as input argument as well as several other parameters and initializes the aes_job with these values.

First of all, the mode is set to encryption at line 17. Next, the TX buffer is set as both the source and destination buffer for the encryption job. After that, several pointers to arrays containing the nonce, the header of a package and its payload are assigned. In addition, the length of the header and payload are also provided. These values need to be provided by the caller of the function as function parameters. Note that the content of the header will not be encrypted but the content of the payload will. Lastly, the mic_size is set to 16.

Note that the aes_helper file also provides an initialise_aes_rx() function. Here, the mode is set to decryption and the source and destination port are set to the RX buffer instead of the TX buffer. The remaining fields stay the same.

The aes_helper file also provides several functions for encoding and decoding packages. To explain the working of these functions, the function aes_tx() will serve as a representative example.

At line 10, the function aes_tx() takes a pointer to an initialized aes_job as its input parameter. As a first step, it selects the appropriate key from the key_options array based on the content of the key identifier field of the package header inside the aes_job (see also Figure 2.9). Note that the key needs to be set before every encryption job when using CCM* encryption, even if the key does not change between jobs.

Next, the nonce is generated with the help of the create_nonce() function which is defined in line 1. This function assembles the nonce based on parts of the header. To create the nonce, it fills an array with the eight-byte source address of a given package header, its frame counter from the auxiliary security header, and the first three bits

```
int create_nonce(uint8_t *nonce, uint8_t *header){
      uint8_t i;
      for (i = 0; i < 8; i++)
      {nonce[7-i] = header[SRC_ID+i];}
      for (i = 0; i < 4; i++)
      {nonce[11-i] = header[FRAME_CNT+i];}
      nonce[12] = header[SECURITY_CTRL] & 0x7;
}

int aes_tx(dwt_aes_job_t *aes_job_tx){
      dwt_set_keyreg_128(&keys_options[aes_job_tx->header[KEY_ID]]);
      create_nonce(aes_job_tx->nonce, aes_job_tx->header);
      aes_config.mode = AES_Encrypt;
      dwt_configure_aes(&aes_config);
      status=dwt_do_aes(aes_job_tx,aes_config.aes_core_type);
}
```

Code 5.8: AES Functions.

interpreted as a one-byte integer of the security control field. It therefore creates an AEAD nonce for non-TSCH mode which is compliant with the IEEE 802.15.4 standard [32]. Note that as with previous processes such as defining the frame control field, the order of the bytes is partly reversed compared to the descriptions inside the IEEE 802.15.4 standard [32].

After the generation of the nonce, the aes_tx() function sets the aes_config mode to encryption. Then, the configuration settings are passed to the dwt_configure_aes() function, which configures the encryption engine. As a last step, the encryption of the message's body is performed. The dwt_do_aes() function returns whether or not the encryption was successful and writes the results to the TX buffer in the case of the TX job. It also automatically adds the MIC at the end of the message's body. The function aes_tx() and create _nonce are shown in Code 5.8.

A further function provided by the aes_helper file is the aes_rx() function which goes through the same steps as the aes_tx() function, the difference being that it takes a decryption job as input, sets the aes_config and mode to decrypt and writes the decryption result to the RX buffer. The dwt_do_aes() function inside the aes_rx() will return an error in case the MIC does not match the nonce from the provided header of the aes_rx_job and the key used.

Lastly, the aes_helper file also provides the functions aes_tx_custom_nonce_no_mic() and aes_rx_custom_nonce_no_mic() that can be used for generating the temporary tag ID and associating the original tag ID with the temporary tag ID. Between aes_tx() and aes_tx_custom_nonce_no_mic(), there are three key differences. First of all, the aes_tx_custom_nonce_no_mic() does not choose the key based on the key identifier field of the security header but always uses the key with the index zero. Secondly, the nonce is not generated based on the security header but assumed to be already present in the nonce array of the aes_job. Lastly, the aes_tx_custom_nonce_no_mic() function sets the

aes_config.mic to zero as no MIC is used for the temporary ID. The aes_rx_custom_nonce_no_mic() contains the same changes but performs a decryption job instead of an encryption job.

5.4.7 Tag.c

```
1  static uint8_t ack_msg[] = {0x2, 0xEC, ...};
2  tag_id[0] = '0';
3
4  while(!paired)
5      {
6        dwt_rxenable(DWT_START_RX_IMMEDIATE); //receive pairing request
7        dwt_readrxdata(rx_buffer, frame_len, 0); // ommitted lines above
8        //Check that message came from master anchor and is pairing request
9        if(rx_buffer[SRC_ID] == 'M' && rx_buffer[FRAME_CTRL] == 0x21){
10         //configure the encryption job
11         initialise_aes_tx(&aes_job_tx,  &ack_msg[0], 0, &tag_id[0], 1, &
       nonce[0]);
12         //create nonce
13         uint8_t i;
14         for(i = 0; i<13; i++){nonce[i] = rx_buffer[PAYLOAD + i];}
15         //use scratch buffer
16         aes_job_tx.src_port = AES_Src_Scratch;
17         aes_job_tx.dst_port = AES_Dst_Scratch;
18         //Encrypt the address, ignore MIC
19         aes_tx_custom_nonce_no_mic(&aes_job_tx);
20         dwt_read_rx_scratch_data(tag_id, aes_job_tx.payload_len, 0);
21         ack_msg[SRC_ID+1] = tag_id[0];
22         localization_msg[SRC_ID+1] = tag_id[0];
23         ack_msg[SEQUENCE_NR] = rx_buffer[SEQUENCE_NR];
24         dwt_writetxdata(sizeof(ack_msg), ack_msg, 0);
25         dwt_writetxfctrl(sizeof(ack_msg)+FCS_LEN, 0, 0);
26         dwt_starttx(DWT_START_TX_IMMEDIATE);
27         //set paired to true
28         paired = 1;
29      }
30     }
```

Code 5.9: Code Snippet that Performs the Pairing of the Tag with the Master Anchor.

The tag performs two processes: pairing and localization exchanges. First, the pairing is shown in Code 5.9. After that, the important code parts of the localization exchange are highlighted in Code 5.10.

The main code parts of the pairing are shown in Code 5.9. First, at lines 6 and 7, the tag listens to any incoming messages and writes them to the rx_buffer. Next, at line 9, the tag checks whether the received package was sent by the master anchor and if it contains the number $0x21$ in the first byte of the frame control field. If these conditions are met, the tag assumes that the received message is a pairing message and proceeds to execute the code within the if statement.

Next, based on the nonce embedded in the pairing message, the tag calculates its temporary tag ID. For this, the tag first extracts the nonce from the pairing message at line 14. After that, the addresses of the scratch buffer are set to the source and address port field of the aes_job object. Here, the scratch_buffer is used instead of the TX buffer to be able to read out the new temporary ID after the encryption and to save it as the temporary tag ID for the tag.

At line 19, the tag ID is generated with the help of the encryption engine of the DW3xxx SDK. The function used stems from the aes_helper file and allows for providing a custom nonce instead of extracting the nonce from the message header. In addition, it does not generate a MIC.

After activating the encryption engine, the temporary tag ID is read to the tag ID buffer at line 20. Next, the temporary tag ID is set as the source address for the acknowledge message and localization message at lines 21 and 22. Then, the acknowledge message with the temporary tag ID is transmitted via UWB at lines 24-26. Finally, On line 28, the pairing variable is set to one so that the pairing loop shown in Code 5.9 is exited.

```
1  static uint8_t localization_msg[] = {0x1, 0xEC, ...};
2  #define RNG_DELAY_MS 500
3
4  while (1)
5  {
6      localization_msg[ALL_MSG_SN_IDX] = frame_seq_nb;
7      dwt_writetxdata(sizeof(localization_msg), localization_msg, 0);
8      dwt_writetxfctrl(sizeof(localization_msg)+FCS_LEN, 0, 1);
9      dwt_starttx(DWT_START_TX_IMMEDIATE);
10     while (!(dwt_read8bitoffsetreg(SYS_STATUS_ID, 0) &
       SYS_STATUS_TXFRS_BIT_MASK)){ };
11     dwt_write8bitoffsetreg(SYS_STATUS_ID, 0, SYS_STATUS_TXFRS_BIT_MASK);
12     frame_seq_nb++;
13     Sleep(RNG_DELAY_MS);
14 }
```

Code 5.10: Code that Periodically Transmits the Tag's Localization Message.

The second code snipped that performs the periodic transmission of localization messages is visualized in Code 5.10. The while loop periodically transmits a localization message which can be captured by the anchors to localize the tag. The content of the while-loop is infinitely executed after the pairing of the tag and master anchor.

First, the sequence number of the localization message is updated at line 6. Then, the localization message is written to the TX buffer, the TX frame control register is configured, and the transmission of the localization message is started. After the transmission, the status_id is cleared and the frame_sequence_number is augmented. Lastly, the tag waits for a specified delay, in this case 500 milliseconds, before sending the next localization message.

5.4.8 Anchor.c

```
1  while (1){
2    dwt_rxenable(DWT_START_RX_IMMEDIATE);
3    dwt_readrxdata(rx_buffer, frame_len, 0); // omitted lines above
4    if (rx_buffer[SRC_ID] == 'T' && rx_buffer[FRAME_CTRL] == 0x1){
5        // Add data to tx message
6        data_msg[TX_IDENTIFIER] = rx_buffer[RX_IDENTIFIER];
7        data_msg[TX_IDENTIFIER+1] = rx_buffer[RX_IDENTIFIER+1];
8        data_msg[TX_COUNTER] = rx_buffer[RX_COUNTER];
9        rx_ts = get_rx_timestamp_u64();
10       rx_ts = rx_ts - rx_ts_sync;
11       uint8_t i;
12       for (i = 0; i < 5; i++){
13         data_msg[TX_TS+i] = (uint8_t)rx_ts;
14         rx_ts >>= 8;
15       }
16       data_msg[SEQUENCE_NR] = frame_seq_nb;
17       Sleep(ANCHOR_DELAY);
18     ...}
```

Code 5.11: In this Code Snippet, The Anchor Extracts the Important Pieces of Information from the Tag's Localization Message and Saves the Localization Message's Arrival Time to the Data_msg Buffer.

The basic loop of an anchor consists of listening to the localization message of the tag, storing its arrival time as well as information about the localization message, and forwarding the data to the master anchor with the help of an encrypted data package. In addition, the anchor also listens to synchronization messages of the master anchor and updates its clock accordingly. The most important parts of this process are visualized in Code 5.11 and Code 5.12.

In Code 5.11, the process of identifying a localization message and extracting the important pieces of information from it is highlighted. As a first step at line 2, the UWB receiver is turned on and the received packages are written to the rx_buffer. Note that some supplemental code lines that guarantee the proper working of dwt_readrxdata have been omitted for the sake of brevity.

Next, at line 4, it is checked if the received package is a localization package from the tag. For this, it is checked if the received package's first byte of the source_id is a 'T'. Additionally, it is also checked if the frame_control field's first byte is '0x1', as only the location message has this frame control byte out of all the tag's messages.

If both conditions are fulfilled, the anchor extracts the tag's ID from the tag's source address field, and the sequence number of the localization message from the localization package at lines 6 to 8. Additionally, it retrieves the exact arrival time of the package with the help of the get_rx_timestamp_u64() function (see Subsection 5.3.4 for more information about time management functions).

Next, at line 10, the arrival time is corrected by the synchronization timestamp. This step keeps the clocks of the anchors synchronized. After that, between lines 11 to 15,

the 40-bit timestamp of the localization message's arrival time is saved to the data_msg array. This is done by using a for loop which writes one byte at a time to the array.

Lastly, the frame sequence number of the localization message is assigned as the frame sequence number of the data package. Next, the anchor sleeps for its anchor-specific delay before sending its data package. This delay allows the master anchor to process each package properly. The anchor-specific delay for anchor 'A0' is five milliseconds, 10 milliseconds for anchor 'A1', 15 milliseconds for anchor 'A2', and 20 milliseconds for anchor 'A3'.

```
1  static uint8_t data_msg[] = {0x9, 0xEC, ...}
2  #define ANCHOR_DELAY 15 // 5 for 'A0', 10 for 'A1' etc.
3
4  while (1){...
5    /*Prepare and Send Data Package*/
6    aes_tx(&aes_job_tx);
7    dwt_writetxfctrl(HEADER_LEN+PAYLOAD_LEN+MIC_LEN+FCS_LEN, 0, 1);
8    dwt_starttx(DWT_START_TX_IMMEDIATE);
9    while (!(dwt_read8bitoffsetreg(SYS_STATUS_ID, 0) &
       SYS_STATUS_TXFRS_BIT_MASK))
10   { };
11   dwt_write8bitoffsetreg(SYS_STATUS_ID, 0, SYS_STATUS_TXFRS_BIT_MASK);
12   frame_seq_nb++; frame_cnt++; frame_cnt_copy = frame_cnt;
13   for (i = 0; i < 5; i++)
14   {
15       data_msg[FRAME_CNT+i] = (uint8_t)frame_cnt_copy;
16       frame_cnt_copy >>= 8;
17   }
18   dwt_setleds(DWT_LEDS_ENABLE | DWT_LEDS_INIT_BLINK);
19   rx_buffer[0] = '\0';
20   }
21   /*if sync signal, do clock reset*/
22   if (rx_buffer[SRC_ID] == 'M' && rx_buffer[FRAME_CTRL] == 0x1)
23   {rx_ts_sync = get_rx_timestamp_u64();}
24   else{} // error handling
25 }
```

Code 5.12: Second Code Snippet of the Anchor. It shows the transmission of the data package and the handling of the master anchor's synchronization package.

After writing the important pieces of information to the data_msg array, the content of the array is encrypted and sent to the master anchor. The code part which executes these steps is highlighted in Code 5.12.

As a first step at line 6, the body of the data_msg is encrypted with the help of the aes_tx() function of the aes_helper.c file. Additionally, the function also adds a MIC at the end of the message and transfers the message to the TX-buffer. Next, the data package is sent to the master anchor at lines 7 to 11. Note that for the input parameter of the function dwt_writetxfctrl(), the total frame length consists of header length, payload length, MIC length and FCS length.

After the transmission of the data package, all counters are augmented. This includes

the one byte long frame sequence number and the four-byte long frame counter of the security header between lines 13 to 17. As a last step, the rx_buffer is cleared at line 19. After that, all code for handling the tag's localization message has been executed and the if-statement is closed.

Next, at line 22, it is checked whether the received package was a synchronization package from the anchor, in case it was not a localization package from the tag. If the source address of the package contains an 'M' for master anchor and the frame control field holds the value $0x1$, it is assumed that a package inside the rx_buffer is a synchronization message from the master anchor. This means that at line 23, the arrival time of the package is saved to the rx_ts_sync and will be used as the new zero value of the clock. Note that this value is not corrected by the time of flight of the synchronization message. This will be done later on by the configurer script as this value changes depending on the position of the anchors. The last part of the code inside the else-statement is omitted. Here, errors are handled.

5.4.9 Master Anchor.c

```
1  run_implementation()
2  {
3   while(tags_paired < number_of_tags){
4     Sleep(50);
5     sys_time = dwt_readsystimestamphi32();
6     for(int i = 0; i < 4; i++) {
7        pairing_msg[NONCE_START+5+i] = (uint8_t)sys_time;
8        sys_time >>= 8;
9     }
10    pairing_msg[NONCE_START+9] = pairing_msg[NONCE_START+9] + 1;
11    pairing_msg[NONCE_START+10] = pairing_msg[NONCE_START+10] - 1;
12    pairing_msg[NONCE_START+11] = pairing_msg[NONCE_START+11] + 2;
13    pairing_msg[NONCE_START+12] = pairing_msg[NONCE_START+12] - 2;
14
15    dwt_setrxaftertxdelay(RX_AFTER_TX);
16    dwt_setrxtimeout(LISTEN_TIMEOUT);
17    dwt_write32bitreg(SYS_STATUS_ID, SYS_STATUS_TXFRS_BIT_MASK);
18    dwt_writetxdata(sizeof(pairing_msg), pairing_msg, 0);
19    dwt_writetxfctrl(sizeof(pairing_msg)+FCS_LEN, 0, 1);
20    dwt_starttx(DWT_START_TX_IMMEDIATE | DWT_RESPONSE_EXPECTED);
21    Sleep(1);
22 ...}
```

Code 5.13: First Part of the Master Anchor Pairing.

This Section highlights the code parts of the master anchor involved in the pairing process between the tag and master anchor as well as the code parts used for the localization process. The first two code snippets in Code 5.13 and Code 5.14 give an overview of the steps performed during the pairing process. Following these snippets, Code 5.15 and Code 5.16 show the steps the master anchor executes during the localization process.

The first code snippet in Code 5.13 shows the beginning of the pairing process between master anchor and tag. First of all at line 4, the master anchor sleeps for 50 milliseconds in-between pairing attempts to conserve energy. Next, the nonce for creating the temporary tag ID is generated. Note that the current way of creating the nonce is for experimental purposes only. In a real-world application, the nonce should be generated in a non-deterministic way to avoid the repetition of the temporary tag IDs.

In this example, the nonce consists of the characters 'NONCE' for the first five bytes of the nonce, followed by the high 32-bit of the current system time and four one byte counters for the last four bytes of the nonce. The current system time and the counters are updated for each run of the pairing loop at the lines 6 to 13.

After the nonce generation, the pairing message is sent and the anchor waits for a potential response from the tag. This is achieved by using the DWT_RESPONSE-_EXPECTED keyword inside the dwt_starttx() function at line 19. In addition, a RX after TX delay at line 14 and a listening timeout at line 15 are set. The first delay times when the reception after the transmission is activated and the second delay determines how long the IC listens to incoming UWB packages.

During the testing phase of the code, it was noticed that when setting the RX after TX to a duration in the millisecond range, the code would still continue to be executed after several microseconds. Therefore, another delay of one millisecond was added at line 21 with the help of the Sleep() function. For a future system, further work is necessary to ensure the correct working of the RX after TX delay and the RX timeout.

```
1  while(tags_paired < number_of_tags){
2     ...
3     while (!((status_reg = dwt_read32bitreg(SYS_STATUS_ID)) & (
     SYS_STATUS_RXFCG_BIT_MASK | SYS_STATUS_ALL_RX_TO |
     SYS_STATUS_ALL_RX_ERR))){ };
4     dwt_readrxdata(rx_buffer, frame_len, 0); \\omitted lines above
5     if(rx_buffer[SRC_ID] == 'T' && rx_buffer[FRAME_CTRL] == 0x02)
6     {
7       initialise_aes_rx(&aes_job_rx,  &rx_buffer[0], SRC_ID+1, &rx_buffer
     [SRC_ID+1], 1, &nonce[0]);
8       for(i = 0; i<13; i++){
9         nonce[i] = pairing_msg[NONCE_START + i];
10      }
11      current_ids[tags_paired] = rx_buffer[SRC_ID+1];
12      aes_rx_custom_nonce_no_mic(&aes_job_rx);
13      original_ids[tags_paired] = rx_buffer[SRC_ID+1];
14      tags_paired++;
15    }
16 }
```

Code 5.14: Second Part of the Master Anchor Pairing.

After sending out the pairing message, the master anchor waits for a potential response of the tag. At lines 3 and 4 in Code 5.14, the master anchor listens to UWB traffic and writes a received package to the rx_buffer. Note that for the sake of conciseness, several

lines for appropriately handling incoming packages were omitted before line 4.

After receiving a package, the master anchors checks whether the source address field is equal to a 'T' and if the first byte of the frame control field contains the value $0x02$. If these conditions are fulfilled, the master anchor assumes that the received package is an acknowledge message from the tag and proceeds with the execution of the code inside the if-statement.

At line 7, the master anchor initializes a decryption job for finding the original tag ID based on the source address of the acknowledge message. Next at lines 8 to 10, the nonce is extracted and saved to the nonce array from the pairing message the master anchor broadcasted in Code 5.13. Following the extraction, the temporary tag ID is added to the current_ids array. Then, the decryption engine is used with the help of the aes_rx_custom_nonce_no_mic() function to find the original tag ID.

At line 11, the original tag ID is saved to at the same index inside the original_ids array as the temporary ID inside the current_ids array. Later on, The current_ids array and original_ids array can be used in combination with the help of the find_original_id() function to find the association between the temporary tag ID and the current tag ID. Lastly, the tags_paired counter is augmented by one. This counter is used to exit the pairing loop once all tags have been paired. For the current setup, the number of tags is always one.

After the pairing of the master anchor with the tag, an infinite loop is started which is responsible for handling the behavior of the master anchor during the localization exchanges with the tag and the other anchors. The core parts of this loop are highlighted in Code 5.15 and Code 5.16.

As a first step of the main loop in see Code 5.15, the master anchor turns on the receiver and saves any incoming UWB package to the rx_buffer. Again, several reception handling steps before line 4 have been omitted for the sake of conciseness.

After receiving a package, similarly to the steps performed by the anchor shown in Subsection 5.4.8, the master anchor checks whether the received package is a localization message from the tag. If this is the case, the master anchor extracts the same pieces of information as the anchor. This includes the tag ID, the sequence number and the package's time of arrival. However, in contrast to the anchor, the master anchor stores these pieces of information to the ts_buffer instead of appending them to the body of a data package. Lastly, the anchors_received counter is augmented at line 8. Note that the ts_buffer is a 100-byte long buffer in which the localization information from each anchor is stored for a single localization exchange. This includes the anchor ID, tag ID, sequence number and time of arrival of a localization message for each anchor, including the master anchor. Per anchor, these pieces of information have a package_length of 10 bytes.

At line 10, the master anchor checks if the received package is a data package of one of the anchors. It identifies a data package by checking that the source address starts with the character 'A' and that the frame control field has the value $0x9$ as the first byte.

If these conditions are fulfilled, the master anchor first decrypts the body of the data package with the help of the aes_rx() function at line 12. The decryption job for the

input parameter was defined during the omitted initialization of the master anchor. After a successful decryption, some pieces of information are taken out of the data package. This includes the anchor ID at lines 15 and 16, the tag ID at lines 17 and 18, and the arrival timestamp at that specific anchor from line 19 to 21 in Code 5.15. Note that for the extraction of the tag ID, the function find_original_id() is used to find the original tag ID based on the temporary tag ID. To do this, the function uses the current_ids and original_ids arrays that have been initialized during the pairing. Lastly, the anchors_received counter is augmented.

```
1  while(1)
2  {
3    dwt_rxenable(DWT_START_RX_IMMEDIATE);
4    dwt_readrxdata(rx_buffer, TX_IDENTIFIER, 0); // omitted lines
5    if (rx_buffer[SRC_ID] == 'T' && rx_buffer[FRAME_CTRL] == 0x1)
6      {
7      /* same as anchor but storing to ts_buffer */
8      anchors_received++;
9      }
10   if (rx_buffer[SRC_ID] == 'A' && rx_buffer[FRAME_CTRL] == 0x9)
11     {
12     aes_rx(&aes_job_rx);
13     /* write to Anchor buffer */
14     arp = anchors_received*package_length
15     ts_buffer[arp + 0] = rx_buffer[RX_IDENTIFIER];
16     ts_buffer[arp + 1] = rx_buffer[RX_IDENTIFIER+1];
17     ts_buffer[arp + 2] = rx_buffer[TX_IDENTIFIER];
18     ts_buffer[arp + 3] = find_original_id(rx_buffer[TX_IDENTIFIER+1]);
19     for(int i = 2; i < 7; i++) {
20       ts_buffer[arp + 2 + i] = rx_buffer[TX_IDENTIFIER + i];
21       }
22     anchors_received++;
23     } ...
24 }
```

Code 5.15: First Part of the Master Anchor Localization Loop.

For a standard localization exchange of an active TDOA system with four anchors, the master anchor goes through four iterations of the main loop before entering the if statement at line 3 in Code 5.16. The following lines of code are responsible for transmitting the content of the ts_buffer via an UART-connection to the host-computer and for sending out the synchronization message to the anchors.

The code between lines 4 and 6 transfers each 10-byte timestamp collection from the ts_buffer to the host_computer with the help of the dimi_print_buffer() function. For this, it is necessary that a UART connection is initialised before the main loop.

After transferring the content of the ts_buffer via the UART connection to the host computer, the master anchor sends a synchronization package to the anchors shortly before the next localization message of the tag is expected. This is achieved at line 8 by sleeping for the localization message tag interval minus the data package delay of

the anchor with the highest ID (in this case 'A3' with 20 milliseconds) minus 7 more milliseconds from empirical testing to make sure all devices have enough time to process the synchronization message before the next localization message of the tag.

The localization message itself is transmitted with the help of the code between lines 9 to 14 in Code 5.16. After the transmission of the localization message, the master anchor stores the timestamp of the transmission of the message as its new clock correction value. It uses this value the same way the other anchors do to correct the arrival timestamp when receiving a localization message.

Lastly, the master anchor cleans up the buffers by writing zeros into all fields of the ts_buffer and adding the array finisher \0 at the beginning of the ts_buffer and rx_buffer. The step of writing the zeros into the ts_buffer was introduced after observing that the timestamps of the last anchor would suddenly stop being correct after more than 10 minutes of localization exchanges. Although the cause of this behavior could not be determined, it was observed that writing zeros to all fields of the ts_buffer resolved the problem. However, for future projects, it is important to identify the root cause of the problem.

```
1  while(1)
2  { ...
3    if (anchors_received == (number_of_anchors)) {
4      for (int i = 0; i < number_of_anchors; i++){
5         dimi_print_buffer(&ts_buffer[i*package_length], package_length);
6      }
7      anchors_received = 0;
8      Sleep(TAG_INTERVAL - 27);
9      dwt_writetxdata(sizeof(sync_msg), sync_msg, 0);
10     dwt_writetxfctrl(sizeof(sync_msg)+FCS_LEN, 0, 1);
11     dwt_starttx(DWT_START_TX_IMMEDIATE);
12     while (!(dwt_read8bitoffsetreg(SYS_STATUS_ID, 0) &
       SYS_STATUS_TXFRS_BIT_MASK))
13     { };
14     dwt_write8bitoffsetreg(SYS_STATUS_ID, 0, SYS_STATUS_TXFRS_BIT_MASK);
15     rx_ts_sync = get_tx_timestamp_u64();
16     for(int i = 0; i < sizeof(ts_buffer); i++){
17       ts_buffer[i] = 0;
18     }
19     ts_buffer[0] = '\0';
20     }
21   rx_buffer[0] = '\0';
22   }
23 }
```

Code 5.16: Second Part of the Master Anchor Localization Loop.

5.5 Localizer Program on Host-Computer

On the host computer, the tag's position is calculated with the help of a localizer program written in Python. The same localizer program can be used both for the passive and the active TDOA localization system.

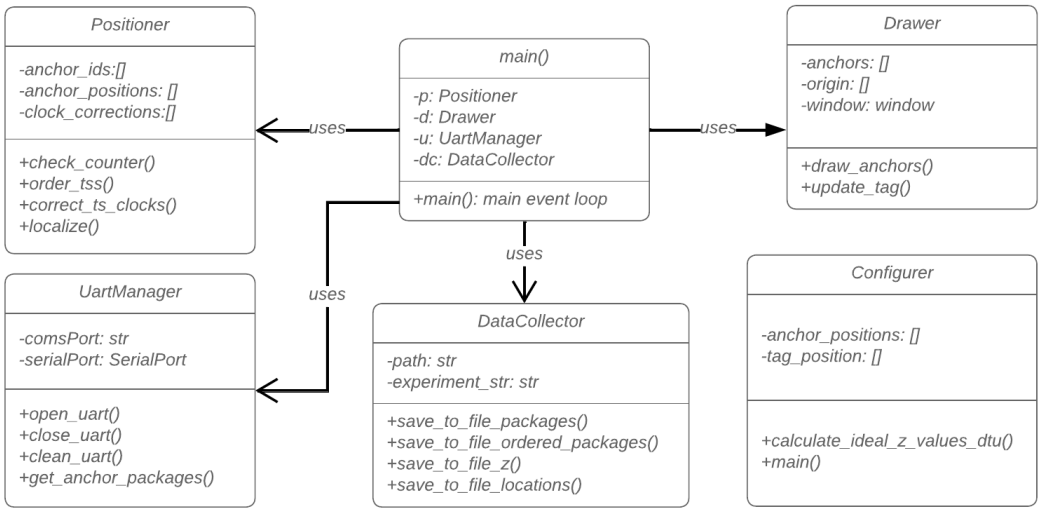

Figure 5.12: UML-Diagram of the Localizer Program. The diagram shows the most important variables and functions of the involved classes. Note that configurer is not a class but a separate script.

The localizer program consists of multiple components as shown in Figure 5.12: The uartmanager is responsible for opening and closing an UART connection to the DK. Additionally, it can receive and format packages transmitted by the DK through the UART connection.

The positioner preprocesses packages, calculates the TDOA values of a localization exchange, corrects the TDOA values based on anchor-specific clock correction values, and calculates the tag's position. The clock correction values are determined with the help of the configurer script and are manually entered into the clock_corrections field of the positioner class before the execution of the localizer script.

To receive real-time feedback, the localizer provides a simple graphical user interface (GUI). The GUI is created with the help of the PySimpleGUI library and managed by the drawer class. The drawer provides functions for creating a window, drawing the position of the anchors in it, and plotting the position of the tag. An example of the GUI is shown in Figure 5.13.

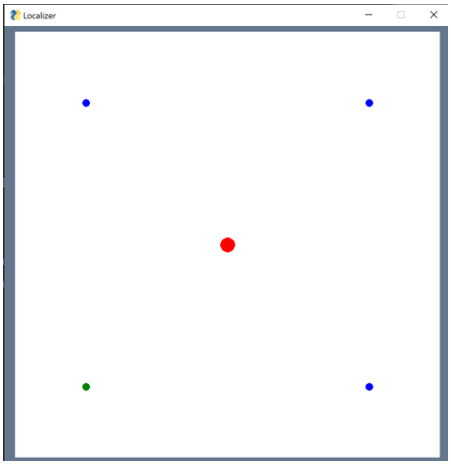

Figure 5.13: The GUI of the Localizer. The master anchor is represented by a green circle, anchors are represented by blue circles, and the tag is depicted by a red circle.

Lastly, the datacollector class can be used to write the data generated during the localization process to a file. It can collect data such as the tag's position, the TDOA values, or the packages from the localization exchange.

The code of the main.py file is represented in Code 5.17. Between lines 1 and 13, multiple variables and an instance of each helper class are initialized. The variables include the number of anchors, the anchor_ids, the anchor_positions, and the paths to save the collected data to.

In a first step at line 15, the UART connection to the DK is emptied in case there is still data stored on the DK side. Following this, an infinite loop is entered which continuously polls for packages from the DK based on which the tag's location is calculated. As a first step inside the loop, the GUI is updated to reflect the new position of the tag. Next, the packages from the UART connection are collected with the help of the function get_anchor_packages() at line 19. The packages have the same format as the packages that are sent by the passive tag (see Subsection 5.3.8). Next, the received packages are saved to a file with the help of the datacollecor.

Following this, the positioner checks that the counters of the received packages contain the same value. This step is needed to make sure that the packages are part of the same localization exchange. If the criterion is not met, the loop continues at line 31. In case the packages do have the same counter, the timestamps are taken out of the packages and ordered in the same way as the anchor_ids list is ordered. Subsequently,

the ordered timestamps are saved to a file. Next, the timestamps from the different anchors are correct by the anchor_specific correction values previously determined with the help of the configurer file. At line 25, the TDOA-values are calculated and saved to a file. Finally, the tag location is calculated, visualized with the help of the function update_tag, and saved to a file. Lastly, the code checks if the window has been closed and if so breaks the loop and closes the UART-connection.

```
1  if __name__ == '__main__':
2      nr_of_anchors = 4
3      anchor_ids = ["M0", "A0", "A1", "A2"]
4      anchor_positions = [[-2, -2], [2,-2], [2,2], [-2, 2]]
5      x_initial = np.array([0,0])
6      tag_position = 0
7      path = r"..\\Data\\"
8      experiment_str = r'ActiveTDOA_position1_try1'
9
10     p = Positioner(nr_of_anchors, anchor_ids, anchor_positions)
11     d = Drawer(anchor_positions)
12     u = UartManager()
13     dc = DataCollector(path, experiment_str)
14
15     u.clean_uart()
16
17     while True:
18         event, values = d.window.read(timeout=100)
19         packages = u.get_anchor_packages(nr_of_anchors)
20         dc.save_to_file_packages(packages)
21         if p.check_counter(packages['counters']):
22             ordered_tss = p.order_tss(anchor_ids, packages['anchors'],
       packages['tss'])
23             dc.save_to_file_ordered_packages(ordered_tss)
24             corrected_tss = p.correct_ts_clocks(ordered_tss)
25             z = p.calculate_z(corrected_tss)
26             dc.save_to_file_z(z)
27             tag_position = p.localize(x_initial, z)
28             save_to_file(filename_position, tag_position)
29             d.update_tag(tag_position)
30             event, values = d.window.read(timeout=100)
31         if event == psg.WIN_CLOSED:
32           break
33     d.window.close()
34 u.close_uart()
```

Code 5.17: Main Function of the Localizer.

5.5.1 Positioner

The positioner is the most important class of the localizer script as it contains the logic to solve the optimization problem for finding the tag's location. The positioner class implements a solver for the maximum likelihood estimation described in Subsection 2.5.3 [24]. The solver can be found in the method localize() shown in Code 5.18 at line 19. The solver is implemented with the help of the function minimize() from the scipy optimize package. The minimize() function takes multiple input parameters.

```
1  def f(self, x, xi, x0):
2      return math.sqrt(math.pow(x[0] - xi[0], 2) + math.pow(x[1] - xi[1],
       2)) - math.sqrt(math.pow(x[0] - x0[0], 2) + math.pow(x[1] - x0[1], 2))
3
4  def g(self, x, *args):
5      # Pass references for better readability
6      xis = args[0]
7      x0 = args[1]
8      z = args[2]
9      vec = []
10     # calculate z - f(x,y)
11     for xi, zi in zip(xis, z):
12         vec.append(zi - self.f(x, xi, x0))
13     # calculate MMSE
14     vec = np.array(vec)
15     vecT = np.transpose(vec)
16     theta = vecT.dot(vec)
17     return theta
18
19 def localize(self, x_initial, z):
20     bounds = ((-20, 20), (-20, 20))
21     res = minimize(self.g, x_initial, method='Nelder-Mead', options={'
       xatol': 0.1, 'disp': False}, bounds=bounds, args=(self.
       anchor_positions[1:], self.anchor_positions[0], z))
22     return res.x
```

Code 5.18: Core Parts of the Positioner.

The first parameter specifies the function to be optimized. The function must adhere to a specific format. The function's first argument should be a one-dimensional array of shape (n,). This argument will contain the values that will be optimized. Supplemental arguments that are needed to completely specify the function can be passed by using the tuple args. For the TDOA optimization problem, the function that shall be minimized is the least squares estimator from Equation (2.8). In order to also keep compatibility with more general solutions of the estimator, Equation (2.7) with sigma set to the identity matrix was implemented instead. Mathematically, Equation (2.7) with sigma as the identity matrix and Equation (2.7) describe the same equation. The Equation (2.7) is implemented in the method g(). The method g() takes two parameters, x containing possible x and y coordinates of the tag, and args, containing the coordinates of the anchors, the reference anchor, and the TDOA-values of one localization exchange. The

method g() itself uses the method f() as a helper function. The method f() implements the function with the same name from Equation (2.6) that describes the true value of the ith signal parameter. Both methods f() and g() can be found in Code 5.18.

The second parameter of the scipy minimize() function is an initial guess for the values that should be optimized. In the case of g(), the initial guess is a possible position of the tag. In a test version of the positioner, the last known position of the tag was taken as input for the next localization run. However, this approach had the problem that when the algorithm diverged to one of the bounds, the following initial guess would be extremely poor, often causing the function to become stuck at one of the bounds. Therefore, it was decided based on empirical testing, to always take the coordinates (0,0) as the initial guess for the tag's location as this approach yielded satisfactory results while demonstrating good performance.

The next argument of the minimize function() is the method argument. This parameter determines the type of solver that will be used for the minimization algorithm. For this use case, the Nelder-Mead solver was selected. Nelder-Mead is a pattern search optimization algorithm that does not rely on gradient or Jacobian information for the minimization process. It was decided to use Nelder-Mead as empirical testing showed that it converged for most cases in less than 100 milliseconds and because it was easy to implement as no gradient of the function to be optimized had to be calculated.

Further, some options are specified for Nelder-Mead solver as a parameter in minimize(). The variable after the keyword 'xatol' specifies the absolute error in x between iterations that is acceptable for convergence. Here, the value 0.1 was chosen based on empirical experimentation as it gives both reasonable performance and convergence. The other keyword, 'disp': False specifies that the solver should not output any information during its run.

Next, the boundaries ((-20, 20), (-20, 20)) for the output values of the function g() are defined after the bounds keyword. Lastly, the supplemental arguments of the function g() are passed to it via the args parameter. Args consist of the anchors coordinates, the reference anchors coordinates, and the TDOA-values denoted by z.

5.5.2 Configurer

The configurer script can be used to find the anchor correction values that are used by the localizer.

The underlying concept of anchor correction values stems from the observation that even when an anchor is positioned at a known localization, there will still be a difference between the expected TDOA values at that location and the in reality measured TDOA values. This difference exists because of constant factors such as the difference in time of flight of the localization message to the different anchors or antenna-specific transmission delays for each of the devices. Since these factors are constant, they can be measured and corrected for, which is what the configurer accomplishes.

The first step for using the configurer is to supply it with the number of anchors of the localization system, the anchors' positions as well as defining a position at which the tag will be placed. Next, the theoretical TDOA values for the tag's location are calculated

for each anchor. After that, the device is placed at the previously defined tag location. Subsequently, the device is connected to the host computer via a universal serial bus (USB) cable.

When all devices are set in place, the configurer script is run. For a predefined amount of localization exchanges (referred to as av_size), the configurer calculates and collects the TDOA-values for each anchor. It also subtracts the expected TDOA values from the TDOA-values, resulting in the TDOA errors of that specific exchange. When the av_size amount of collected TDOA differences are reached, the configurer calculates the median of collected TDOA differences for each anchor. Here, the median is used as it is more robust to outliers than the arithmetic mean. The median difference then can be used as the correction value for that specific anchor. The correction values are printed to the console and a histogram of the collected TDOA values is plotted.

The transfer of the anchor correction values to the positioner script happens manually. Typically, the correction values are determined before a series of experiments, and the same correction values are employed throughout all experiments within that specific series.

5.6 Experiment Results of the Passive TDOA Localization

This Section shows the results of the experiments performed with the passive TDOA localization system. More information about the experimental setup can be found in Section 4.3. The experiments focus on the technical evaluation of the system.

First, an overview of the packages sniffed by a potential attacker is shown in Subsection 5.6.1. After that, the results of the experiment suite with a four anchor localization system are presented in Subsection 5.6.2. After that, the results of the experiment suite where the tag is turned around in the horizontal plane are highlighted in Subsection 5.6.3. Lastly, the results of the experiment suite with five anchors are shown in Subsection 5.6.4.

5.6.1 Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
228	09:43:27.667485	0x304d	0x5841	IEEE 802.15.4	12	Data, Dst: 0x5841, Src: 0x304d
229	09:43:27.667485	0x3041	0x5854	IEEE 802.15.4	12	Data, Dst: 0x5854, Src: 0x3041
230	09:43:27.668480	0x3141	0x5854	IEEE 802.15.4	12	Data, Dst: 0x5854, Src: 0x3141
231	09:43:27.670806	0x3241	0x5854	IEEE 802.15.4	12	Data, Dst: 0x5854, Src: 0x3241
232	09:43:28.157427	0x304d	0x5841	IEEE 802.15.4	12	Data, Dst: 0x5841, Src: 0x304d
233	09:43:28.172992	0x3041	0x5854	IEEE 802.15.4	12	Data, Dst: 0x5854, Src: 0x3041
234	09:43:28.172992	0x3141	0x5854	IEEE 802.15.4	12	Data, Dst: 0x5854, Src: 0x3141
235	09:43:28.173985	0x3241	0x5854	IEEE 802.15.4	12	Data, Dst: 0x5854, Src: 0x3241

Figure 5.14: Traffic of the Passive TDOA Localization System Shown in Wireshark. Yellow packages are sent by anchor 'A0', orange packages by anchor 'A1', red packages by anchor 'A2', and turquoise packages are sent by the master anchor 'M0'.

Figure 5.14 shows an excerpt of the traffic a potential attacker could capture from the passive TDOA localization system. The excerpt was generated by sniffing a passive TDOA localization system with four anchors and one tag with the help of the sniffer pipeline mentioned in Subsection 4.1.4. The sniffer was activated for ten minutes and fifteen seconds. During this time, it captured a total of 4693 packages.

In Figure 5.14, packages sent by the anchor with the ID 'A0' are marked in yellow. Packages from anchor 'A1' are highlighted in orange, and packages sent by anchor 'A2' are red. Packages from the master anchor are highlighted in turquoise. What is notable is that only the synchronization message of the master anchor was captured but not the localization message. This discrepancy arises from the architecture of the sniffer. In its current iteration, the sniffer passes each package to the host computer after receiving it. During the transmission of the package from the sniffer to the host-computer, the sniffer cannot capture any other packages. Consequently, when two packages arrive at the sniffer within a very short time interval, the sniffer will not capture the second package. This highly likely happened to the localization package of the master anchor as it is sent very shortly after the synchronization package.

When observing the received packages, a regular pattern arises. First, the master anchor sends a synchronization message. After that, the localization message of anchors 'A0', 'A1', and 'A2' are captured in a sequential manner. This pattern can be observed throughout the whole capture file. When examining individual packages, the package structure defined in Subsection 5.3.3 can be observed again. As the packages comply with the IEEE 802.15.4 standard, Wireshark correctly dissects them and provides an overview of particular package parts. An excerpt of a single package is shown in Figure 5.15. Notably, details such as the short addresses of the source and destination address or the frame control field are highlighted by Wireshark.

```
> Frame 192: 12 bytes on wire (96 bits), 12 bytes captured (96 bits) on interface \\.\pipe\wireshark, id 0
v IEEE 802.15.4 Data, Dst: 0x5841, Src: 0x304d
    > Frame Control Field: 0x8841, Frame Type: Data, PAN ID Compression, Destination Addressing Mode: Short/1
      Sequence Number: 176
      Destination PAN: 0xdeca
      Destination: 0x5841
      Source: 0x304d
      FCS: 0x800d (Correct)
v Data (1 byte)
<

0000  41 88 b0 ca de 41 58 4d  30 11 0d 80              A····AXM 0···
```

Figure 5.15: General Package Information in Wireshark about a Synchronization Package Sent by the Master Anchor.

5.6.2 Standard Settings

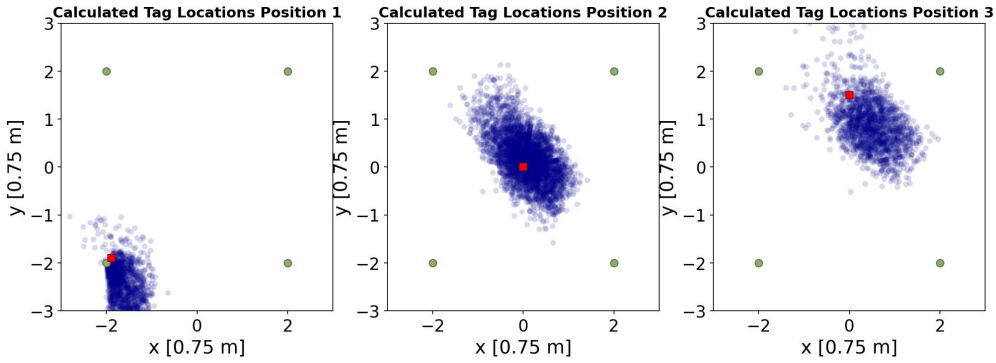

Figure 5.16: Measured Locations for the Three Positions Described in Section 4.3.

The first experiment suite consisted of collecting the location of a tag with the help of a four-anchor passive localization system at three different positions. Figure 5.16 displays the estimated position of the tag based on the TDOA values gained from the passive localization system. The green circles show the locations of the anchors, the red square represents the actual position of the tag, and the blue circles show the estimated tag positions.

For the first experiment, the tag was placed at position (-1.9, -1.9). When visually examining the point cloud of the calculated locations in Figure 5.16, the calculated position of the tag lies roughly at the correct location. The key metrics of the collected point cloud can be found in Table 5.2. With a median of (-1.70, -2.44), the center of the point cloud lies southeast of the actual tag position. This amounts to a distance between the actual tag's location and the median of the calculated location of 0.44 m.

For the second experiment, the calculated median lies at (0.07,0.19) compared to the actual tag location of (0,0). This results in a distance of 0.14 meters (m) between the

Experiment	**Actual Position**	**Median**	**Distance Median/AP**	**MAD CP**
Position 1	x -1.9 y -1.9	x -1.70 y -2.44	0.44	0.38
Position 2	x 0.0 y 0.0	x 0.09 y 0.17	0.14	0.41
Position 3	x 0.0 y 1.5	x 0.47 y 0.84	0.60	0.42

Table 5.2: Metrics for the Four Anchor Passive TDOA System. Distance Median/AP and MAD CP in [m]. AP = Actual Position, CP = Calculated Position.

calculated and actual tag position. The third experiment's calculated tag position sits at (0.47, 0.84) compared to the actual tag position of (0, 1.5). The distance between the two points is 0.60 m.

When it comes to the mean absolute deviation (MAD) between the median of the point and the calculated locations, the average MAD of the three positions sits around 0.40 m. This metric shows the variability of an univariate sample.

Finally, the TDOA-values for each anchor are shown in Figure 5.17. Each histogram shows a histogram of the calculated TDOA-values between the master anchor and a second anchor. The shape of the TDOA value distributions resemble a uniform distribution for all anchors. The majority of the distributions span approximately ten nanoseconds. One thing that is notable is that the distributions of the TDOA values between anchor 'A2' and the master anchor exhibit a slight left skew for all three experiments.

Other important observations can be made by comparing the theoretical TDOA value of the location for each experiment with the median of the actual measured TDOA values. For the first location, the TDOA median and the theoretical TDOA value lie within a nanosecond from each other. For anchor 'A1', the expected TDOA value is much larger than the measured median. Lastly, for anchor 'A2', the expected TDOA value lies a bit more than six nanoseconds lower than the calculated mean.

For position two, all calculated medians lie a bit higher but still close to the expected TDOA values. For position three, the TDOA medians of anchor 'A0' and 'A1' are close to the expected TDOA values. However, the median of anchor 'A2' lies around four nanoseconds lower than the expected TDOA value.

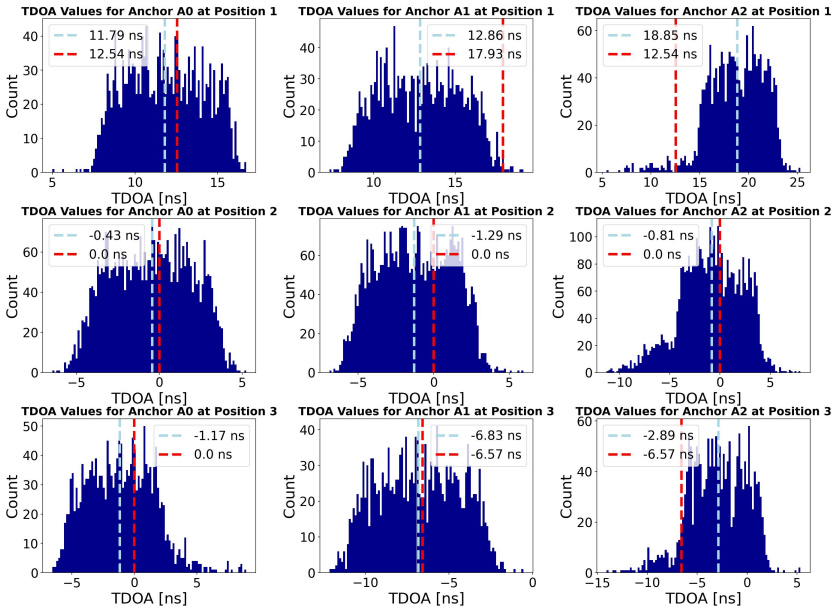

Figure 5.17: Histograms for TDOA Values of the Passive Localization System with Four Anchors. The red line shows the expected TDOA value and the blue line the median of the collected TDOA values.

5.6.3 Turning Tag

When performing the experiments with the standard settings, it was noticed that turning the tag resulted in different measured locations, even when the tag was placed at the same coordinates. It was therefore decided to perform a series of experiments during which the tag was placed at the same location but turned in different directions. The correction values for the anchors were determined before the experiment suite while the tag was facing west at the coordinates (0,0).

The left Figure in Figure 5.18 shows the antenna directions of the anchors and the tag for one of these experiments. For all experiments, the two anchors in the southern corners face north and the two anchors in the northern corners face south. For each experiment, the antenna of the tag is turned in one of four directions: northeast, southeast, southwest, and northwest. At each direction, the TDOA localization system was run for approximately five minutes. The results of these measurements can be found in the right Figure in Figure 5.18.

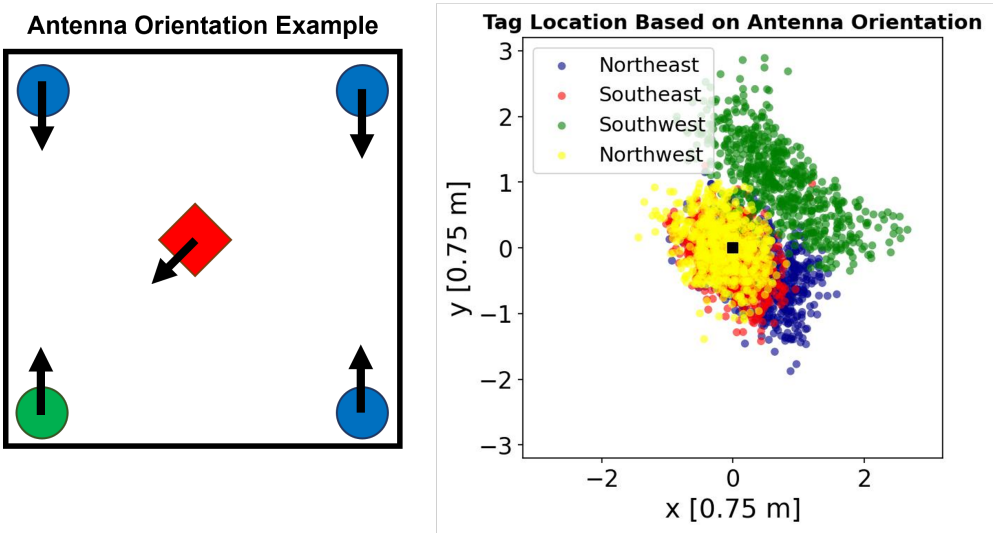

Figure 5.18: Left: Antenna Orientation of the Anchors and The Tag. Right: Results of Orienting the Antenna of the Tag in Four Different Directions. The tag is placed at coordinates (0,0) for all orientation experiments.

When visually examining the point clouds of the different experiments, it is notable that they all lie at different positions, even though the tag is placed at the same location. Especially the point cloud of the tag facing southwest lies considerably more northeast than the other point clouds.

These observations are further backed up by the metrics in Table 5.3. The medians of the point clouds differ from each other by a distance of up to more than a meter. The medians of the experiments with the tag facing northwest and southeast lie closer to the true position of the tag than the medians of the point clouds from the experiments with the tag turned southwest and northeast.

Position	AP	Median	Distance Median/AP
Northeast	x 0.0 y 0.0	x 0.51 y -0.36	0.47
Southeast	x 0.0 y 0.0	x 0.03 y -0.12	0.09
Southwest	x 0.0 y 0.0	x 0.67 y 0.99	0.9
Northwest	x 0.0 y 0.0	x -0.15 y 0.02	0.15

Table 5.3: Metrics of the Experiment Suite With the Turning Tag. Distance in [m].

Lastly, the differences between the median of the TDOA values of a specific anchor and the expected TDOA value of that anchor are calculated and shown in Table 5.4. Visually, this metric corresponds to the distance between the blue and red lines in Figure 5.17 but for the values of the turning tag experiment suite.

The first thing noticeable is that the difference between the medians and ideal TDOA values for the southwest measurement is very big for all anchors. This indicates that the signal's arrival time from the anchors was slightly shorter than the arrival time of the master anchor's signal compared to the TDOA values of the orientation during which the anchor correction values were configured. For the other orientations, no particular pattern is apparent. The differences vary between -2.85 and 1.83 nanoseconds (ns).

Orientation	A0	A1	A2
Northeast	-2.85	-0.36	1.45
Southeast	0.49	-0.2	1.14
Southwest	-6.4 -	5.63	-7.21
Northwest	1.83	-0.42	1.04

Table 5.4: Differences Between the Median of the Collected TDOA Values of Each Anchor and the Expected TDOA Value for that Respective Anchor in [ns].

5.6.4 Five Anchors

For the last series of experiments, a fifth anchor was added to the passive TDOA localization system in the middle between the master anchor and anchor "A2" at the coordinates (0, -2). The anchor's ID was 'A3'. Subsequently, the three experiments from Subsection 5.6.2 were repeated.

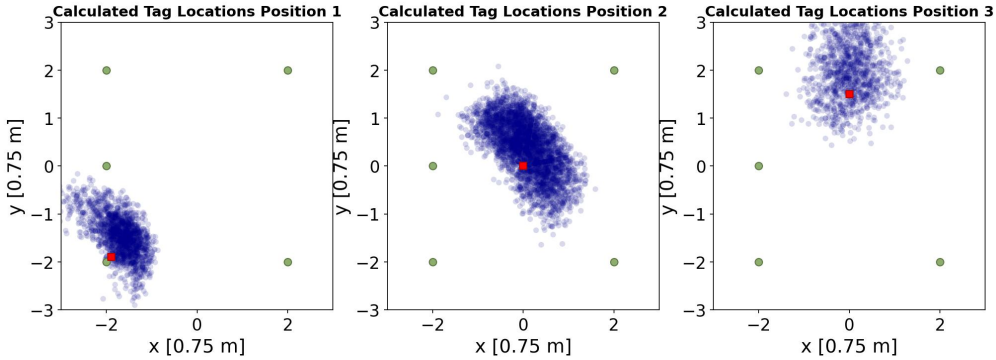

Figure 5.19: Measured Locations for the Experiments with a Passive Five Anchor TDOA Localization System.

A visualization of the experiment's results can be found in Figure 5.19. For all three experiments, the point clouds of the calculated tag positions lie roughly at the correct spot based on a visual examination. However, all three point cloud centers seem to be slightly above the actual tag position. This suggests that there might have been a slight error in choosing the anchor correction values.

When looking at the metrics in Table 5.5, the observation is confirmed that the medians of the point clouds lie higher than the actual position of the tag. In addition, the calculated medians also lie more east than the actual tag positions. The distance between the actual tag position and calculated median is 0.38 m and 0.34 m for the first two experiments. For the third experiment, the distance is bigger with 0.77 m. This can be explained by the fact that the dataset of the third experiment contains a lot of outliers.

In the Figure for the third experiment in Figure 5.20, there is a gathering of calculated tag locations in the form of a horizontal line at the northern side of the Figure. The line-like distribution of points can be explained by the fact that the scipy.optimize() solver used for optimizing the localization problem was given the borders (-20,20) for

Position	**AP**	**Median**	**Distance Median/AP**	**MAD CP**
Position 1	x -1.9 y -1.9	x -1.65 y -1.46	0.38	0.35
Position 2	x 0.0 y 0.0	x 0.09 y 0.44	0.34	0.46
Position 3	x 0.0 y 1.5	x 0.20 y 2.51	0.77	4.13

Table 5.5: Metrics of the Experiments with Five Anchors. Distance Median/AP and MAD CP in [m].

the x and y coordinates. Consequently, when the algorithm diverges, it halts at these boundaries.

The high amount of outliers is also reflected in the MAD of the third experiment in Table 5.5. With a value of 4.13 m, it is nearly ten times bigger than the MADs of 0.35 m and 0.46 m for experiments one and two.

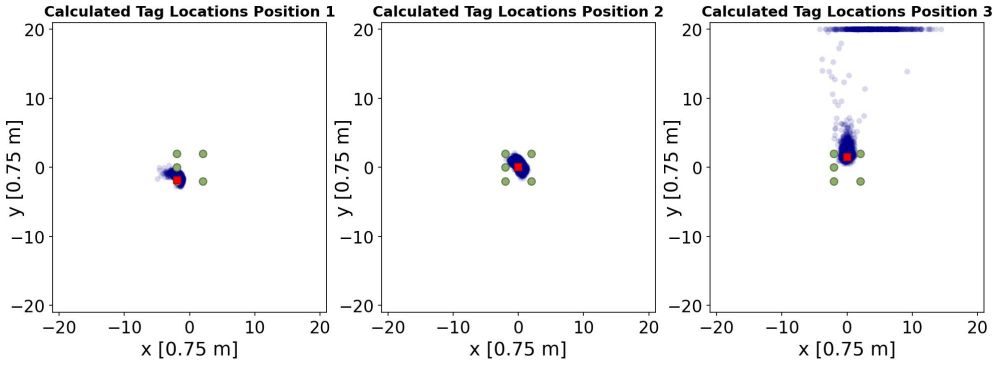

Figure 5.20: Measured Locations for the Experiments with a Five Anchor TDOA Localization System Large Window.

Lastly, the difference between the median of the TDOA values and the expected TDOA value are represented in Table 5.6. For position one, all medians lie below the expected TDOA value. For anchors 'A1' and 'A3', the difference is especially big. For experiment two, the TDOA difference is negative for all anchors and lies roughly around minus two nanoseconds. Lastly, for experiment three, the TDOA differences are again lying around two nanoseconds except for anchor 'A3', where the difference is nearly minus eight nanoseconds.

Orientation	**A0**	**A1**	**A2**	**A3**
Position 1	0.41	-4.85	-0.69	-4.01
Position 2	-1.19	-1.72	-2.79	-1.13
Position 3	-1.58	-3.82	-1.59	-8.45

Table 5.6: Difference Between Median TDOA an Expected TDOA for Experiments with Passive Five Anchor TDOA Localization System in [ns].

5.7 Experiment Results of the Active TDOA Localization

This Section presents the results of the experiments performed with the active TDOA localization system. The experiments performed are similar to the ones performed with the passive TDOA localization system. The description of the experimental setup for the active system can be found in Section 4.3.

The first Subsection gives an overview of the data packages sniffed by a potential attacker in Subsection 5.7.1. Following this, Subsection 5.7.2 presents the results of the experiment suite with four anchors and Subsection 5.7.3 the experiment suite with five anchors. Lastly, Section 5.8 gives an overview of further experiments performed, where data was collected as part of this work but where the evaluation lies out of scope for this work.

5.7.1 Wireshark

When a potential attacker tries to sniff the active TDOA localization system, the attacker is met with multiple challenges: First of all, the active TDOA localization system uses non-standard communication settings (see Subsection 5.4.2), meaning that the attacker needs to know these settings beforehand in order to sniff the UWB traffic. In addition, the active TDOA system uses a deterministic STS meaning that the attacker also needs to have a device that is able to handle the STS. Moreover, the attacker needs to know that a deterministic STS is used in order to handle it correctly and process the incoming packages.

Figure 5.21 shows an excerpt of the traffic a potential attacker could capture when knowing about the aforementioned communication settings. Similar to Subsection 5.6.1, some packages are missing as the sniffer was not fast enough to capture packages closely following each other.

No.	Time	Source	Destination	Protocol	Length	Info
10	16:50:17.493458	00:00:00:00:00:00:30:4d	00:00:00:00:00:00:58:54	IEEE 802.15.4	36	Data,
11	16:50:18.611605	00:00:00:00:00:00:30:4d	00:00:00:00:00:00:58:54	IEEE 802.15.4	36	Data,
12	16:50:19.741573	00:00:00:00:00:00:30:4d	00:00:00:00:00:00:58:54	IEEE 802.15.4	36	Data,
13	16:50:20.875666	00:00:00:00:00:00:30:4d	00:00:00:00:00:00:58:54	IEEE 802.15.4	36	Data,
14	16:50:21.888539	00:00:00:00:00:00:5a:54	00:00:00:00:00:00:58:41	IEEE 802.15.4	23	Data,
15	16:50:21.890577	00:00:00:00:00:00:30:41	00:00:00:00:00:00:58:4d	IEEE 802.15.4	52	Data,
16	16:50:21.904579	00:00:00:00:00:00:32:41	00:00:00:00:00:00:58:4d	IEEE 802.15.4	52	Data,
17	16:50:22.380601	00:00:00:00:00:00:30:4d	00:00:00:00:00:00:58:41	IEEE 802.15.4	23	Data,
18	16:50:22.395624	00:00:00:00:00:00:5a:54	00:00:00:00:00:00:58:41	IEEE 802.15.4	23	Data,
19	16:50:22.397587	00:00:00:00:00:00:30:41	00:00:00:00:00:00:58:4d	IEEE 802.15.4	52	Data,
20	16:50:22.412585	00:00:00:00:00:00:32:41	00:00:00:00:00:00:58:4d	IEEE 802.15.4	52	Data,

Figure 5.21: Overview of the Packages a Potential Attacker Could Sniff from the Active TDOA Localization System.

At the beginning of Figure 5.21, the pairing messages of the master anchor were intercepted. The messages can be found at line 10 to 13 marked in light blue. After package 13, an acknowledge package by the tag was sent to the master anchor to start the localization exchange. This package was not captured. Instead, the next package received at line 14 is the location package of the tag. This message marks the beginning of a localization exchange. It is followed by the data packages of the anchors. Again, the package of anchor 'A1' was lost. After receiving the data package from anchor 'A0' in yellow and from anchor 'A2' in red, the synchronization message from the master anchor in light blue was received. This localization exchange pattern was infinitely repeated throughout the rest of the capture file.

When it comes to the structure of the packages, the first thing noticeable compared to the passive localization system is that the active system uses 64-bit long addresses. In addition, the data packages from the anchors are encrypted and thus contain a supplemental security header. Note that Wireshark offers the function to decrypt the payload of packages following the IEEE 802.15.4 standard. First, the key that was used for encryption needs to be provided. Then, Wireshark decrypts the package's payload and shows the result in a separate window in the package overview. An example of this is shown in Figure 5.22. In the decrypted package body, one can find the temporary tag ID followed by the sequence number of the localization message and the time of arrival at the specific anchor.

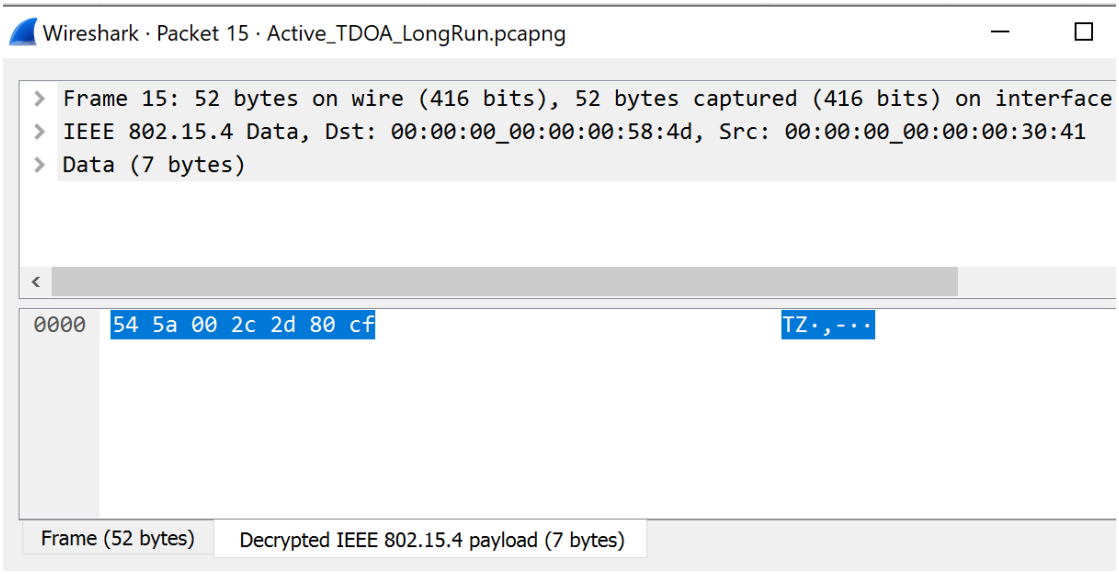

Figure 5.22: Overview of the Decryption Function in Wireshark. The payload of a data package is decrypted, revealing the temporary Tag ID, the sequence Number of the tag's localization package, and the time of arrival at anchor 'A0'.

5.7.2 Standard Settings

The same experiments that were performed for the passive TDOA localization system were repeated for the active TDOA localization system. The results of the first three experiments can be found in Figure 5.23. Upon a visual inspection, multiple findings emerge: First of all, the experiments for positions one and two show multiple point clouds for the calculated tag location. Compared to Section 4.3, the diameter of the point clouds is smaller. Moreover, for the third experiment, the border of the point cloud lies at least two meters apart from the actual tag position. Additionally, the shape of the point cloud is elongated, stretching from the middle of the northern border of the room nearly all the way to the solver limit at $y = 20$.

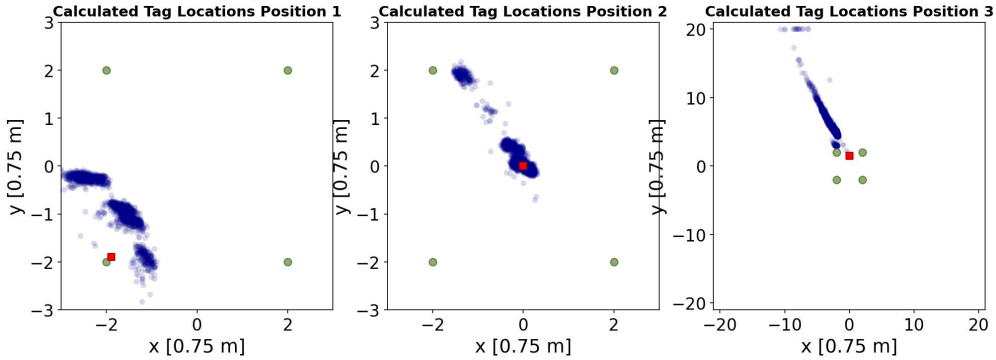

Figure 5.23: Recorded Locations for the Active System. The axes for the first Two graphs are set to (-3,3) and for the third graph to (-20,20).

When further examining the histograms of the TDOA-values of the experiments, distinctively different patterns compared to the histograms in Figure 5.17 become evident. First of all, the histograms of the active system show more outliers than the histograms of the passive system. Next, the distributions of the histograms are narrower than the histograms in Figure 5.17. Lastly, some of the histograms show a multimodal distribution, such as the examples shown in Figure 5.24. The observed patterns in the TDOA-value histograms coincide with the patterns observed in the calculated tag locations in Figure 5.23.

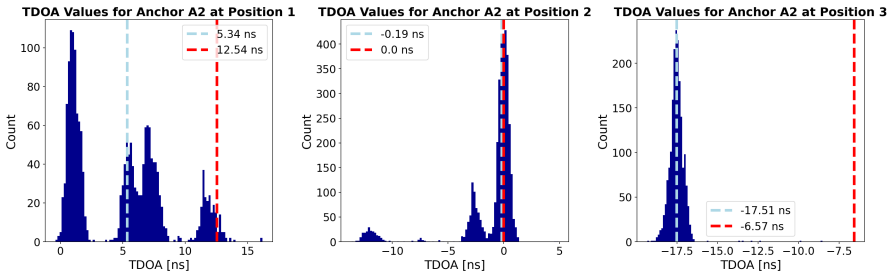

Figure 5.24: A Selection of Histograms for the TDOA-Values at a Specific Anchor for the Active TDOA system. The red line shows the expected TDOA value and the blue line the median of the collected TDOA values.

As one of the possible explanations for the unusual pattern of the calculated tag locations in Figure 5.23 is the presence of multipath effects, the experiments were performed again, this time placing the DKs onto the carton boxes they were delivered in. Visualizations of the devices on top of the carton boxes can be found in the Appendix in Figure A.4 and Figure A.5.

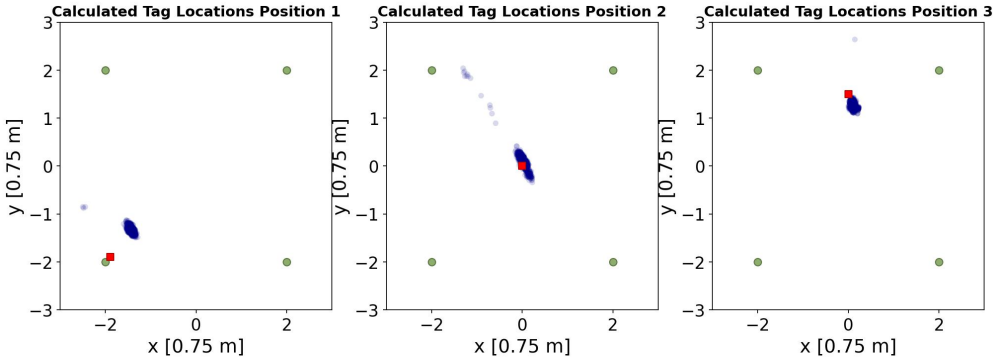

Figure 5.25: Recorded Locations for the Active System with Boxes.

The results of the experiments with boxes are shown in Figure 5.25. Compared to the results without boxes in Figure 5.23, there is just one point cloud per experiment, with the exception of a few outliers in experiment two. In addition, the diameter of the point clouds is significantly smaller than the diameter of the point clouds in Figure 5.16 and the diameter is also marginally smaller than the diameter of the point clouds in Figure 5.23. Lastly, for the first position, the point cloud lies northeast compared to the actual tag position. For the second experiment, the actual and the measured tag locations seem to match. For the third experiment, the point cloud lies slightly north compared to the actual tag location.

Position	**AP**	**Median**	**Distance Median/AP**	**MAD CP**
Position 1	x -1.9 y -1.9	x -1.43 y -1.35	0.54	0.04
Position 2	x 0.0 y 0.0	x 0.02 y 0.10	0.08	0.09
Position 3	x 0.0 y 1.5	x 0.10 y 1.28	0.18	0.04

Table 5.7: Metrics for the Experiments with the Active System and Boxes. Distance Median/AP and MAD CP in [m].

When looking at the distances between the median and the actual tag location in Table 5.7, the observations about the point cloud locations are confirmed. It is moreover noticeable that the distance for the first experiment is much bigger with 0.54 m compared to the distances of the second and third experiments with 0.08 and 0.18 m respectively.

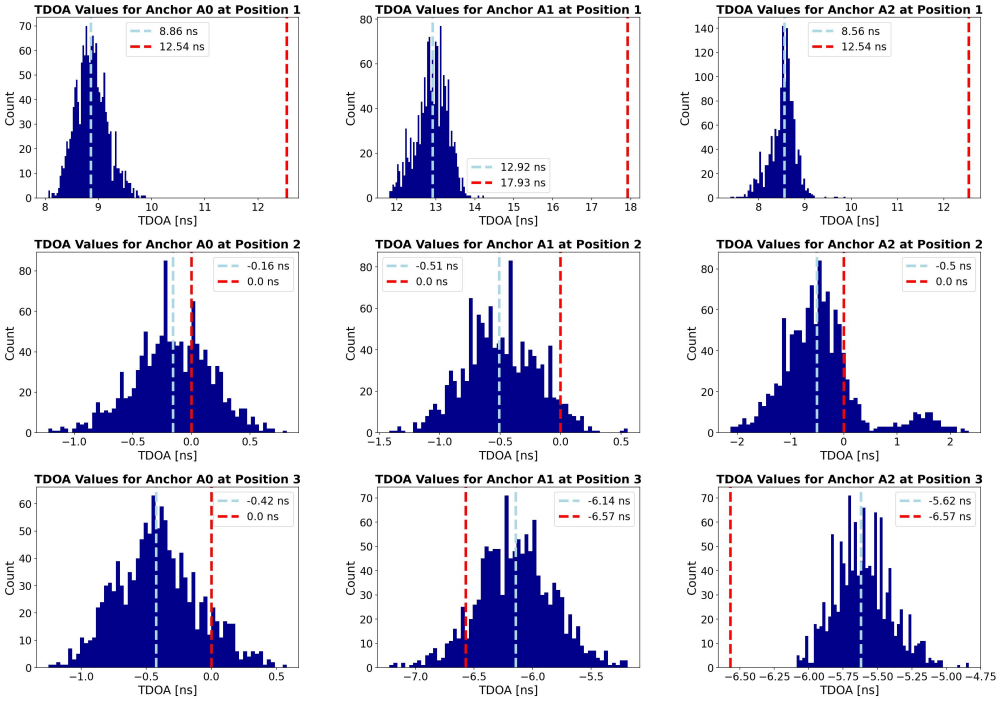

Figure 5.26: Histogram of TDOA Values for Experiments with the Active TDOA Localization System and Boxes in [ns]. The red line shows the expected TDOA value and the blue line the median of the collected TDOA values.

Finally, the TDOA values for the experiments with boxes of the active system are visualized in Figure 5.26. Compared to the histograms of the passive system in Figure 5.17, the distributions have a much smaller width of roughly 1.5 to 3 ns compared to the ten nanoseconds of the passive system. In addition, the TDOA-value distributions for the passive system show more of a triangular shape compared to the uniform shape of the TDOA-value distributions of the active system. What is in addition notable is that the plot of anchor 'A2' at position 2 shows a second, much smaller peak. All the other distributions are unimodal.

When it comes to the calculated medians and expected TDOA values, the first experiment shows the biggest difference between the two values. For all anchors, the expected TDOA value lies four to five nanoseconds higher than the calculated one. For the second experiment, the all expected TDOA values lie about 0.2 to 0.5 ns higher than the medians. For the last experiment, the direction switched for anchor 'A1' and 'A2'. Here, the median is about -0.5 to 1 ns bigger than the expected TDOA value. For anchor 'A0', the median is about 0.5 smaller than the expected TDOA value.

5.7.3 Five Anchors

Similarly to the passive localization system, an experiment suite with five anchors was performed with the active localization system too. The fifth anchor was placed at the coordinates (-2,0). Again, the devices were placed onto the boxes they were shipped in to reduce multipath effects. The results of the experiment suite are shown in Figure 5.27.

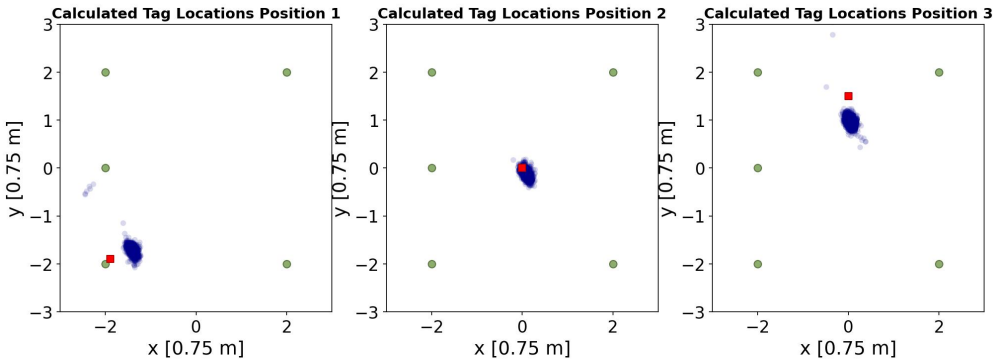

Figure 5.27: Collected Locations for the Five Anchor Active TDOA Localization System.

Upon visual examination, the recorded tag location clouds lie at most thirty centimeters apart from the actual tag location. For experiment one, the calculated point cloud lies more northeast than the actual tag location. For experiment three, the recorded tag location cloud lies more north than the actual tag location. For experiment two, the location matches closely.

Compared to the results of the active with four anchors and boxes, the calculated locations visually lie roughly at the same locations, even though the system with four anchors exhibits less of an error at positions one and three. Another difference that can be observed is that the shape of the point clouds of the experiment suite with five anchors is more round compared to the oval shape of the point clouds of the experiment suite with four anchors. Next, the metrics are shown in Table 5.8.

Position	**AP**	**Median**	**Distance Median/AP**	**MAD CP**
Position 1	x -1.9 y -1.9	x -1.4 y -1.7	0.41	0.11
Position 2	x 0.0 y 0.0	x 0.08 y -0.11	0.11	0.08
Position 3	x 0.0 y 1.5	x 0.03 y 0.97	0.4	0.08

Table 5.8: Metrics of the Experiments with the Active TDOA Localization System with Five Anchors. Distance Median/AP and MAD CP in [m].

Orientation	**A0**	**A1**	**A2**	**A3**
Position 1	-3.1	-3.83	-2.12	-1.9
Position 2	-0.66	0.29	0.8	-0.4
Position 3	-0.88	2.48	1.7	0.6

Table 5.9: Difference Between Median TDOA an Expected TDOA for Experiments with Active Five Anchor TDOA Localization System in [ns].

When looking at the metrics in Table 5.8, the observations of the visual analysis concerning the position of the point clouds are confirmed. Additionally, compared to Table 5.7, the average MAD of the system with five anchors is bigger with 0.09 m compared to 0.06 m of the system with four anchors. When it comes to the distance between the median of the point cloud and the actual position, the system with five anchors has a shorter distance for positions one and two. For position three, the system with four anchors has a shorter distance.

Lastly, the metrics for the TDOA-values can be found in Table 5.9. The differences show the same size and direction as for the experiment suite with four anchors except for the results at position two. Whereas for the system with four anchors the TDOA for anchor 'A1' and 'A2' are negative, they are positive for the experiment suite with five anchors.

5.8 Long Duration Measurements, Blocking Objects and Further Experiments.

Apart from the experiments presented above, additional experiment suites were conducted to further evaluate the active and passive TDOA localization system. These experiment suites focus on topics such as further tag positions, longer experiment durations, turning of the tag and anchors, adding anchors at different positions, and introducing interference by blocking UWB signals with objects. An overview of the experiment suites performed can be found in Appendix A.4.

Two examples of these datasets are visualized in Figure 5.28. The Figure on the left stems from an experiment with a passive four TDOA localization system. The tag was placed at the coordinates (0,1.5) and data was collected for more than twelve hours. The goal of this experiment was to see how the system behaves over a long experiment duration. The Figure on the left in Figure 5.28 shows the TDOA values over time collected for anchor 'A1'. The red line marks the rolling mean per 200 TDOA values. As the Figure on the left in Figure 5.28 shows, the average of the TDOA values seems to be more or less constant over time.

Another example of the collected datasets is shown on the right in Figure 5.28. Here, the collected locations are visualized when the signals of a four-anchor passive TDOA localization are partially blocked by a human sitting northeast of the tag. The Figure

shows that partially blocking the signal causes more outliers inside the collected point cloud. These experiment suites have been performed during the course of this work but their thorough evaluation is out of scope for this thesis.

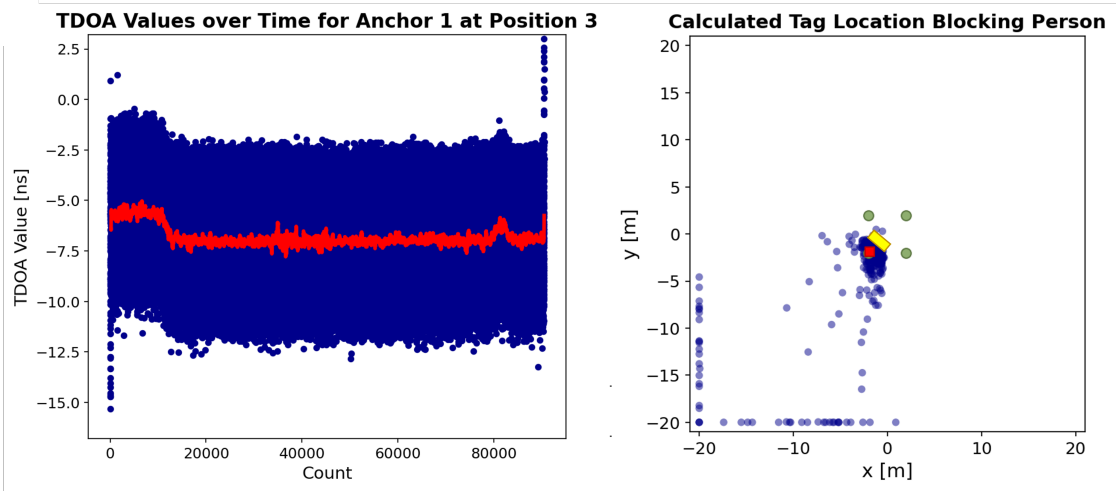

Figure 5.28: Example Selection of the Results of Further Experiments. Left: TDOA Values [ns] over Time for Anchor 'A1' During a Long-Term Collection (more than 8 hours). Right: Location Estimation for Experimental Setup with a Person Blocking Parts of the UWB Signals. The Tag is Placed in the Southwest Corner, and the Person Sits Northwest as Close as Possible to the Tag. The Tag is Represented by a Red Square, and the Human by a Yellow Square.

6

Analysis

This Chapter is split into two parts, a technical analysis and a privacy analysis of the TDOA localization systems. First, both the passive and the active TDOA localization systems are evaluated based on their capability of locating the tag in an accurate and precise manner. In the second part of this Chapter, the systems are thoroughly evaluated based on their privacy-preserving characteristics with the help of the privacy criteria selected in Subsection 4.2.3.

6.1 Technical Evaluation of the Passive TDOA System

In this Section, it is evaluated whether the passive TDOA localization system works as intended. Additionally, it is checked how accurate and precise the system performs. First of all, when visually examining Figure 5.14, it is confirmed that the passive TDOA localization system is capable of localizing the tag. When placing the tag at known locations, the system is able to roughly determine the tag's position.

Secondly, when it comes to accuracy, the distance between the median of the measured tag locations and the actual tag location is taken as the evaluation metric. The results in Table 5.2 show that out of the three experiments performed, the accuracy of the active system is the highest for position two. This is not further surprising as the system was configured at this location (see also Subsection 5.5.2). However, it is notable that there is still a considerable discrepancy of 14 cm between the actual tag position and the calculated median, even though 200 localization exchanges were used to find the configuration values.

One possible explanation for this discrepancy is that the TDOA values change over time. However, this hypothesis is not confirmed when observing the average of the TDOA values of the anchors over multiple hours (see for example the right Figure in Figure 5.28). Another explanation is, that 200 localization exchanges are not enough data points to adequately cancel out the noise inside the system.

	A0		**A1**		A2	
Position	min	max	min	max	min	max
0	-1086	-1037	-2298	-2198	-3112	-3040
1	-1118	-1036	-2553	-2455	-2712	-2537
2	-1070	-1004	-2575	-2514	-2939	-2869

Table 6.1: Minimum and Maximum Correction Values Per Anchor in [dtu] Based on Eight Batches of 200 Localization Exchanges for the Experiments in Subsection 5.6.2.

This hypothesis is supported by the values in Table 6.1. For this table, the collected TDOA values of the first fifteen minutes of the experiments in Subsection 5.6.2 were taken from each dataset and grouped into eight batches of 200 localization exchanges. After that, the optimal correction value per anchor was determined for each batch. Lastly, the minimum and maximum correction values of these batches were taken for each anchor and added to the table in Table 6.1.

The table shows that the correction values can vary between 100 DTUs per anchor which is equivalent to one or two nanoseconds. As the light travels roughly 30 cm in a nanosecond, these results confirm that the random noise inside the localization system can potentially introduce an error in the magnitude of 20 cm.

The next observation made is that there is a large error between the median of the calculated positions and the actual tag position for positions one and three. To analyze this behavior, the first step was to check whether there are any patterns visible inside the histograms of the TDOA values of the passive TDOA localization system in Figure 5.17.

One thing notable based on the histograms is that some anchors show discrepancies in the range of multiple nanoseconds between the expected TDOA value for an anchor and the median of the measured TDOA values. This is for example the case for anchor 'A1' and 'A2' for experiment 1 and anchor 'A2' for experiment three in Figure 5.17. These discrepancies can potentially have many causes: changes over time due to changes in temperature of the environment or changes in antenna constants such as the used energy level or antenna delay, the dilution of precision based on the geometry of the anchor placement, effects caused by the environment such as non-line-of-sight (NLOS) effects and multipath effects, interference with other communication technologies, antenna orientation effects and many more.

As already mentioned in this Section, the TDOA values seem to be constant over time when not moving the tag. Therefore, a difference caused by temperature changes or antenna constant changes over time is excluded as a potential explanation.

Another factor that could potentially cause a decrease in accuracy is a phenomenon known as the geometric dilution of precision [128]. However, as the tag moves inside a square with the four anchors placed at the corners, the effects of the phenomenon are negligible.

The next possible cause of the discrepancies could be multipath or NLOS effects. Generally, these effects are caused by objects blocking or deflecting UWB signals. However, in the conducted experiments, no other objects were present except the UWB devices themselves, the floor they were sitting on, and a wall at the north of the square described by the anchors. Additionally, multipath and NLOS effects are often characterized by multiple peaks in the histograms of the collected TDOA values. This is not the case for the TDOA histograms of the performed experiments (see Figure 5.17). Therefore, multipath and NLOS effects are unlikely to be the primary cause of the observed discrepancies.

What cannot be excluded as a possible explanation for the inaccuracies is potential interference with other technologies. For example, there is a WI-FI router in the same room the experiments took place. However, in case of interference, the effect should cause disturbances for all three experiments instead of just experiment one and three. Additionally, it is unclear in which way WI-FI signals would specifically affect UWB localization accuracy.

Based on the principle of exclusion, the most probable cause for the differences in the TDOA values is thus antenna orientation effects. As shown in the datasheet of the DWM3000 transceiver module, the radiation pattern of the UWB antenna changes depending on the antenna orientation [129]. According to Qorvo, when two antennas are oriented perpendicular relative to each other, the polarisation of the antennas changes and can introduce location inaccuracies.

This claim by Qorvo is further supported by the results of the second experiment suite, where the tag was turned around in the horizontal plane at a static location in the middle of the room. Here, the difference between the median of the TDOA values and the expected TDOA value at a specific anchor varies greatly depending on the antenna orientation (see Table 5.4). As moving the tag between different positions in the square automatically also changes the antenna orientation between the tag and the anchors, systematic errors are introduced. This can also be partially confirmed by the results of the experiment suite with five anchors, as both for anchor 'A1' at position one and anchor 'A2' at position two, the difference between expected TDOA value and calculated median follows a similar trend as observed in the first experiment suite with four anchors. However, the trend for anchor 'A2' at position one, where the median was bigger than the expected TDOA value for the experiment with four anchors, could not be observed for the experiment with five anchors. It is therefore concluded based on the results in Table 5.4 and the explanations by Qorvo in [129] that the systematic errors were introduced by antenna orientation.

One consideration concerning accuracy was to see if adding a supplementary anchor increases the accuracy of the TDOA localization system (see Subsection 5.6.4). For the experiment suite with four anchors, the distance between the median of the calculated

tag locations and the actual tag location for experiments one to three were 0.44, 0.14, and 0.6 meters. For the experiment suite with five anchors, the distances were 0.38, 0.34, and 0.77 m (see also Table 6.2 for a summary of the accuracy values). Therefore, the accuracy of the system decreased when adding a supplemental anchor.

Possible explanations for this behavior are, that the correction values of the anchors were, due to the natural variability inside the TDOA values, not as accurate as for the first experiment suite. Another explanation is, that adding a supplemental anchor causes the Nelder-Mead solver to diverge more often. Visually, this could not be confirmed as there were a lot of outliers for experiment one of the suites with four anchors but also a lot of outliers for experiment three for the experiment suite with five anchors (see Figure 5.16 and Figure 5.19). Therefore, it is unclear where exactly this decrease in accuracy is coming from.

Apart from accuracy, precision is another important factor to benchmark the performance of a localization system. The metric chosen to approximate precision here is the MAD between each data point and the median of the collected point cloud. Averaged over the three experiments, the MAD for the first experiment with four anchors is 0.40 m and 1.65 m for the experiment suite with five anchors. Note however that for the suite with five experiments, the MAD for the third experiment was 4.31 m, distorting the overall average MAD to be much higher than the average between experiments one and two, which was 0.40 m.

These results show that the precision of the system is relatively low, with estimated tag locations being located easily within one or two meters of each other. This is explained by the fact that the passive TDOA localization system uses the functionality of delayed transmissions provided by the DW3xxx SDK to send out the localization messages. More information about the delayed transmission functionality can be found in Subsection 5.3.4. Delayed transmissions only allow specifying the time of transmission within a range of eight nanoseconds, resulting in TDOA values with high variability.

6.2 Technical Evaluation of the Active TDOA System

For the technical evaluation of the active TDOA localization system, the same points as for the passive system are examined. First of all, it is checked if the system can perform the basic localization of the tag. After that, its accuracy and precision are evaluated.

When it comes to the basic localization capabilities of the active TDOA localization system, Figure 5.23 and Figure 5.25 display distinct results. Figure 5.23 shows that when not using the nRF52840 DK with a DWM3000EVB as the tag but rather a DWM3001CDK device, the active TDOA localization system performs very poorly. This is particularly the case when placing the DWM3001CDK directly on the floor. For the first two experiments, the system estimated the position of the tag at multiple locations, which is shown by the existence of multiple point clouds in Figure 5.23. For the third experiment, the system estimates the tag's location along a north-south running line multiple meters apart from the actual tag's position.

In contrast, when placing all devices, especially the tag, onto the carton boxes they were shipped in, the accuracy and precision of the system's results visibly increase. All plots in Figure 5.25 show a single point cloud with a narrow diameter. Additionally, the point clouds lie very close to the actual tag positions.

Based on these observations, it is concluded the active TDOA localization can perform the localization of the tag. However, the system is susceptible to localization errors when the DWM3001CDK is placed directly on the floor.

To explain the inaccurate results of the first experiment suite in Figure 5.23, several similar explanations as mentioned in Section 6.1 come into play. The most likely cause for the observed behavior is the presence of multipath and NLOS effects.

When looking at the histograms of the first experiment suite in Figure 5.24, it is notable that some histograms show multiple peaks per histogram. These peaks lie between two and ten nanoseconds apart from each other. This pattern strongly suggests the presence of multipath effects.

For experiment three, it is clearly visible in Figure 5.23 from the calculated positions that nearly none of the packages arrived at the estimated time. By analyzing the histogram for position three in Figure 5.24, it is visible that for anchor 'A2' at position 3, the median of the measured TDOA values lies nearly ten nanoseconds lower than the expected TDOA value. This difference is too big to be explained by antenna orientation effects alone and therefore strongly suggests the presence of NLOS effects.

In comparison, by placing the UWB devices on carton boxes, the antennas are lifted from the floor, which potentially reduces multipath and NLOS effects. As a result, the results depicted in Figure 5.25 exhibit higher accuracy and show only a single point cloud per experiment.

Next, for the evaluation of the accuracy and precision of the system, only the results of the experiments with boxes are considered. When it comes to the accuracy of the active system, the distance between the actual tag position and the calculated median of the measured tag positions is again chosen as the evaluation metric. For the experiment suite with four anchors, the results can be found in Table 5.7. Similar to the passive system, the distance at position two is the shortest, indicating the highest accuracy, compared to positions one and three with longer distances and thus lower accuracies. Similarly to the passive TDOA localization system, the systematic errors in the results of the active system for positions 1 and 3 are attributed to antenna orientation effects based on the same line of reasoning.

When adding a supplemental anchor to the experiment setup, the results of the metrics change as follows (see Table 5.8): for experiment one, the distance reduces by 0.13 m, for experiment two, it slightly increases by 0.03 m, and for position three there the distance augments by 0.22 m. The average of these three results is 0.30 m, compared to an average of 0.26 m for the experiment suite with four anchors. Again, no concrete explanation could be found for the decrease in accuracy by adding a supplemental anchor. Possible explanations are non-optimally chosen correction values and more outliers, but further research in this regard is necessary to confirm the hypotheses.

To assess the precision of the active TDOA localization system, the MAD between the collected locations and the median of the respective point cloud is chosen as the evaluation metric (see Table 5.7). For the experiment suite with four anchors, the MAD is 0.04 m, 0.09 m, and 0.04 m for experiments 1, 2, and 3, respectively. For the experiment suite with five anchors, the MAD augments to the values 0.11 m, 0.08 m, and 0.08 m. These small MADs are achieved by profiting from the very accurate timestamping capabilities of the DW3000 during message reception.

6.3 Comparison Between Active and Passive TDOA System

When comparing the results of experiments performed with the active and passive TDOA localization systems, several points are notable. First of all, both systems are capable of performing the basic localization of the tag. Regarding accuracy, both systems show systematic errors when comparing the collected point clouds with the actual tag position. For both systems, the most probable explanation for this behavior is the presence of antenna orientation effects.

Experiment Suite	**Position 1**	**Position 2**	**Position 3**	**Mean**
Passive 4 Anchors	0.44	0.14	0.60	0.40
Passive 5 Anchors	0.38	0.34	0.77	0.50
Active 4 Anchors	0.54	0.08	0.18	0.26
Active 5 Anchors	0.41	0.11	0.40	0.30

Table 6.2: Distance Between the Median of the Calculated Positions and the Actual Tag Position in [m] for Several Experiment Suites.

When further looking at the metric that describes the accuracy, the distance between the actual tag position, and the median of the collected TDOA values, there are several noteworthy observations. Note that the distance values for all experiment suites are summarized in Table 6.2. First of all, both systems show the highest accuracy at position two compared to lower accuracies at positions 1 and 3. This is not surprising as the systems were configured at position two.

Next, when looking at the average distance per experiment suite summarized in Table 6.2, it becomes apparent that the passive system has a slightly lower accuracy, with the average distances of 0.40 m and 0.50 m compared to the average distances of the active system with 0.26 m and 0.30 m. Several factors potentially contribute to these outcomes.

Firstly, one possible explanation is that the observed differences were simply due to the inherent variability of the experimental conditions. Secondly, incorrect correction values determined during the configuration process could also contribute to the observed results. Thirdly, the capabilities of the different antennas might play a role in the ob-

served results. For the passive system, the sole antenna responsible for the collection and timestamping of the localization messages is the antenna of the tag. If this antenna has lower capabilities due to chance compared to the other antennas, it might cause the system to perform worse compared to the active system where four different antennas are used for the collection and timestamps. The last explanation found is that the overall execution of the code introduces more errors in the passive system, resulting in lower accuracy. This could be due to the use of functionalities such as the delayed transmission, the time between the arrival of the synchronization message and the transmission of the localization message, and so on. It is important to note that the aforementioned explanations are not an exhaustive list, and other factors may also contribute to the observed variations.

What can be also observed when analyzing the average distances in Table 6.2 is that adding a fifth anchor actually decreases the accuracy both of the active and passive TDOA localization system. Apart from factors like random variability and the choice of correction values, the most likely explanation may be that adding just one further anchor on the west side of the anchor square augments the systematic error introduced by antenna orientation effects. However, further research is necessary to explain this behavior.

Experiment Suite	**Position 1**	**Position 2**	**Position 3**	**Mean**
Passive 4 Anchors	0.38	0.41	0.42	0.40
Passive 5 Anchors	0.35	0.46	4.13	1.65
Active 4 Anchors	0.04	0.9	0.04	0.06
Active 5 Anchors	0.11	0.08	0.08	0.09

Table 6.3: MAD for all Experiments in [m].

The second metric the systems can be compared by is precision. The values of the MAD for the experiment suites of the active and passive system are summarized in Table 6.3.

The first prominent observation is that the passive system has much higher MAD values for all experiments than the active system. This is consistent with the plots in Figure 5.16 and Figure 5.25, where the diameter of the point clouds of the passive system are much bigger than the diameters of the active system.

This fact can be explained by the usage of the delayed transmission functionality of the DW3000 for the passive system. The delayed transmission can only be timed with an accuracy of eight nanoseconds. Contrary, the approach taken by the active system is to send out a single localization message that is caught by multiple anchors. Here, the accuracy is much higher in the range of less than a nanosecond. This explains the much higher precision of the active system compared to the passive system.

In summary, both the active and the passive systems can perform TDOA localization. However, the active system exhibits higher precision than the passive system because it does not use the delayed transmission functionality. Additionally, the active system also demonstrates higher accuracy. However, this factor could be attributed to the natural variability of the results. Further experiments are needed to confirm the difference in accuracy.

6.4 Privacy Analysis Overview

For the privacy analysis, the active and the passive TDOA localization systems are evaluated based on the selected criteria in Subsection 4.2.3. For the active system, a conceptual distinction is made between a minimum TDOA localization system and an improved TDOA localization system. The minimum system can perform active TDOA localization but does not include the three privacy-enhancing features of dynamic addressing of the tags, encryption of the data package's payload, and the usage of the STS. The minimum system was not actively constructed and analyzed but is rather a theoretical concept designed for this Section to better highlight how additions like encryption and dynamic addressing improve the privacy-preserving characteristics of the improved system.

Based on this decision, three systems, the passive TDOA localization system, the minimum active TDOA localization system, and the improved active TDOA localization system were evaluated with the help of the privacy criteria selected in Subsection 4.2.3. The results of the evaluation are summarized in Table 6.4.

Criterion	**Passive**	**Active Minimum**	**Active Improved**
Need-to-Know	Fulfilled	Partially Fulfilled	Fulfilled
Purpose-of-Use	Fulfilled	Fulfilled	Fulfilled
Anonymity	Fulfilled	Not Fulfilled	Partially Fulfilled
Unlinkability	Fulfilled	Not Fulfilled	Partially Fulfilled
Unobservability	Fulfilled	Not Fulfilled	Partially Fulfilled
Minimization	Fulfilled	Fulfilled	Fulfilled
Authentication	Partially Fulfilled	Partially Fulfilled	Partially Fulfilled
Authorization	Fulfilled	Fulfilled	Fulfilled

Table 6.4: Summary of the Results of the Privacy Evaluation.

In the following sections, the personal information involved in the experiments will be further refined based on the specific implementation of the localization systems. Subsequently, the privacy analysis of both the passive and active localization systems will be presented. In the case of active systems, the evaluation of both the minimum and improved versions will be kept together, and any differences will concerning a specific criterium will be mentioned for this criterium.

6.4.1 Refinement Personal Information

After implementing the TDOA localization systems, personal information involved inside these systems can be identified on a more fine-grained level.

For both the passive and the active systems, three pieces of personal information are involved in the localization process: the tag IDs, the arrival timestamps of localization messages, and the sequence numbers of localization messages. The tag ID is a type of personal information because it reveals information about the identity of the user of the tag. The arrival timestamps can be used to find the location of the data subject, which is also a type of personal information [17]. The sequence number on its own does not fit the definition of personal information, but they are used for both systems to link the different times of arrivals to one localization message and can be thus also used to find the data subject's location. Therefore, they are included here for the sake of completeness as a type of personal information.

For the passive system, the pieces of information are generated and kept on the tag except for the sequence number, which is embedded in the localization messages sent by the anchors. However, the tag also extracts the sequence number of the localization messages upon their arrival.

For the active system, the tag ID is embedded in the code of the tag. When the tag sends out its localization messages, it also transmits its tag ID as the source address of the package. When using the improvements mentioned in Section 6.4, the tag uses a dynamic ID instead of its original ID. The dynamic tag ID is also embedded inside the localization messages sent by the tag is the sequence number.

These localization messages are captured by the anchors which store the (temporary) tag ID and the sequence number. Additionally, they generate the arrival timestamps of the localization messages at the respective anchor. After that, The anchors transmit the arrival times via UWB to the master anchor. In the improved version of the active system, the body of these data packages is encrypted. The master anchor extracts the sequence numbers, timestamps and tag IDs from the data packages and thus knows about all pieces of information involved in the localization process. Additionally, it can also associate temporary with constant tag IDs as explained in Subsection 5.4.5.

6.4.2 Privacy Evaluation of the Passive TDOA System

Based on the selected criteria in Subsection 4.2.3, the passive TDOA localization system is evaluated. This Subsection gives an explanation for the assessment of each criterion found in Table 6.4.

Need to Know: In the passive system, the tag is the only actor that processes personal information. It does so in order to localize itself and on top monitor its movement patterns in order to inform caretakers in case of irregular behavior of the patient wearing the tag. Therefore, the need-to-know requirement is fulfilled.

Purpose of Use: The tag processes the arrival timestamps for the purpose of localizing itself. This is the main purpose of the system and thus the passive system validates the criterion.

As both criteria Need-to-Know and Purpose-of-Use are fulfilled, it is assumed that the confidentiality criterium is thus also fulfilled.
Anonymity: The only piece of information that is related to personal information that is sent out via the air and thus revealed are the sequence numbers. However, without the timestamps, no inference can be made about the identity of the wearer of the tag. All other pieces of personal information are generated on the tag itself, where they are also processed. Therefore, the criterium of anonymity is fulfilled.
Unobservability: As the tag does not send out any signals, it is not observable from the outside that there are any tags and that they are performing localization. Unobservability is thus fulfilled.
Unlinkability: As no personal information is transmitted via UWB except the sequence number, there is also no information that can be linked back to the data subject. Unlikability is hence fulfilled too.
Minimization: In the current iteration of the passive localization system, the tag collects the timestamps, tag ID and the sequence number of one localization exchange, forwards the pieces of information to the host computer and then empties its buffer. On the host computer, the timestamps and tag locations are collected for research purposes. As only the timestamps of the current localization exchange are kept on the tag, it can be argued that the tag itself follows the principle of minimization. When it comes to the host computer, the tag's location as well as intermediate steps of the localization process such as the content of the packages transmitted via the UART connection from the tag and the TDOA values are stored long-term to be evaluated in the context of this thesis. It can be thus argued that the principle of minimization is fulfilled in the context of this thesis. When looking at the context of the derived use case, however, long-term storage of intermediate processing steps of the localization process that contain a lot of personal data does not serve the purpose of localizing the patient and tracking its movements. Thus, these intermediate collection steps need to be removed before a real-world deployment, otherwise, the principle of minimization is violated.
Authentication: Both the anchors and the tags verify whether an actor is who they claim they are by checking the source address of the received package. However, this source address can easily be faked by a potential attacker. Therefore, authentication is only partially fulfilled.
Authorization Authorization is fulfilled as only anchors are allowed to provide localization messages.

Transparency is only partially given as authentication is only partially fulfilled. Authorization is fulfilled, however.

All in all, the passive TDOA localization system fulfills most of the criteria defined in Subsection 4.2.3. The only criterium it does only partially fulfill is **Authentication**, as no mechanism such as a MIC is used to verify the source of packages before processing them, introducing a potential point of attack. Additionally, intermediate steps of the localization process were collected for the creation of this thesis. As some of these steps contain personal information and are not strictly necessary for the main purpose of tracking the elderly patient, the collection needs to be stopped in a production environment to fulfill the minimization principle.

6.4.3 Privacy Evaluation of the Active TDOA System

The active TDOA localization system is also evaluated based on the criteria in Subsection 4.2.3. For the criteria where a difference can be observed, it is differentiated between a minimum active localization system, which does not use dynamic addressing, encryption or the STS, or an improved active localization system, which profits from this improvement. The goal of this distinction is to highlight the improvements the additions bring to privacy.

Need-to-Know: For Need-to-Know, each of the components of the active TDOA localization system can be evaluated: The tag knows about its original source address and its own temporary source address. It is necessary for the tag to have these pieces of information in order to send out the localization message. The localization message is necessary to perform the localization of the tag, the prime use case of the localization system. The anchors process the localization messages of the tag which contain personal information. Firstly, the anchors know about the arrival time of the localization message at their specific position. In order to forward this piece of information to the master anchor, it is necessary for the anchor to know about the arrival time. The anchor also knows about the (temporary) tag ID. The main task of the anchor is to determine the arrival timestamp of the localization message to forward it to the master anchor. In order to do this, the anchor does not need to know about the original tag ID. Therefore, a system with temporary tag ID fulfills need-to-know whereas a system without temporary tag ID does not fulfill Need-to-Know.

Purpose-of-Use: In general, all components of the system use the pieces of personal information to localize the patient. This is congruent with the Purpose-of-Use.

Anonymity: The active localization system transmits several pieces of personal information via the air. When using the static tag ID, the information can be easily linked to the patient, and anonymity is not fulfilled. The dynamic tag ID and the usage of the STS improves anonymity. However, as there is only a single or maybe a few tags involved in such a system, anonymity is only partially fulfilled, as linking the information is still possible, just harder.

Unobservability: Based on the messages used by the active TDOA localization system and the fact that UWB technology is used, an experienced attacker can identify that the system is performing localization. Moreover, it can also identify the number of tags involved. Thus, unobservability is not given. Nonetheless, considering the relative scarcity of UWB technology in practical applications, a less experienced attacker might not readily recognize the presence of UWB traffic. Also, using an STS makes it harder to receive UWB packages and thus also increases the effort the attacker has to exert to eavesdrop on the system. These factors slightly enhance the system's level of unobservability. Nevertheless, it is important to acknowledge that eavesdropping on the system remains possible.

Unlinkability: When static tag IDs and no encryption of the data package's bodies is used, a potential attacker can easily link the timestamps to the timestamp of each anchor embedded in the data packages back to the tag and it's original owner. Unlinkability is hence not given for the minimum localization system. Using dynamic tag IDs and

particularly encrypting the bodies of the data packages increases the unlinkability of the personal data back to its originator. However, some of the personal information such as the existence of the localization packages itself containing the temporary tag ID can still be linked for example to the wearer of the tag, especially when only a few tags are used. Thus, additions such as encryption and temporary tag IDs greatly improve unlinkability but still do not guarantee complete unlinkability.
Minimization: The active system resembles the passive system when it comes to minimization. In general, the master anchor only keeps the data of one localization exchange inside its buffer, fulfilling the principle of minimization. On the side of the host computer, all packages transferred from the master anchor as well as all tags' positions stored long-term to provide the necessary data for this work. Similarly to the passive system, a production system should not store the intermediate results to fulfill minimization.
Authentication: Analogously to the passive localization system, the active localization system uses the source address of a package to verify its author. While this approach satisfies a basic level of authentication, it can be easily circumvented by an attacker. The improved active system however also uses the MIC in combination with the encrypted data packages as the IEEE 802.15.4 does not allow the generation of encrypted packages without a MIC. This greatly augments the security level of the active system, but as the tag does not use the MIC for its localization messages, there is still room for improvement when it comes to authentication.
Authorization Authorization is fulfilled as only localization messages coming from tags are processed by the anchors and the master anchor only processes data packages coming from anchors.

As authentication is only partially fulfilled, transparency is also only partially fulfilled for the active localization system.

All in all, the active localization system does not fulfill as many privacy criteria as the passive localization system. Especially when only setting up a minimum active localization system without including privacy-enhancing features, only three out of the eight privacy criteria are fully fulfilled. Using privacy-enhancing features such as dynamic tag IDs, encryption, MICs and the STS, the privacy-preserving character of the active localization system can greatly be improved, now completely fulfilling four of the privacy criteria and partially fulfilling the other four. However, including these features takes additional time and effort during the implementation of the active localization system.

7

Discussion and Limitations

This Chapter first discusses the findings of the technical and privacy evaluation from Chapter 6. After that, the technical and privacy-related limitations of this work are highlighted.

7.1 Discussion

For the technical analysis, three key factors were examined: the first factor was, whether or not the localization systems are capable of performing basic localization. Additionally, it was also examined what level of accuracy and precision the systems excerpt. Furthermore, unexpected observations about the behavior of the systems were documented.

Both the active and the passive TDOA localization systems are able to perform the basic localization of the tag. However, the active system demonstrated greater accuracy in localization compared to the passive system, as detailed in Section 6.3. However, these observations might be due to natural variability on a hardware level inside the experiment suites. However, further experiments are necessary to confirm these observations.

It was also observed that both the active and passive localization systems showed systematic errors in accuracy for experiments performed further away from the center of the room. This observation is highly likely caused by antenna orientation effects, which should be corrected for future experiments.

When it comes to precision, the active system performed significantly better than the passive system (see Section 6.3). This can be explained by the fact that the passive system uses the function of delayed message transmission, which introduces a variance of roughly eight nanoseconds in the transmission time of the localization messages. The active system on the other hand avoids this by just sending out a single localization message that is captured by multiple tags, fully profiting from the accurate timestamping capabilities of the DW3000 transceiver.

By considering these findings in the context of existing literature, it could be shown that constructing a TDOA localization system similar to the architecture of [72; 73; 74] can be achieved with reasonable effort. It could also be shown that the localization algorithm from [24] delivers reliable results while keeping a fast processing speed.

As far as the technological side of the localization system goes, no new findings were produced. The main focus of this work lies on privacy improvements, which will be discussed from here on out.

When it comes to the findings of the privacy considerations and the privacy analysis, several points need to be mentioned.

Firstly, the first big effect on privacy was discovered when evaluating the different localization architectures in Section 5.1. Based on considerations guided by the literature, it was concluded that the passive TDOA and PDOA localization systems should, in theory, achieve higher levels of privacy than the active TDOA and PDOA systems as well as a TOA localization system. This is based on the fact that the tag in a passive TDOA or PDOA setting is passively listening to the localization messages coming from the anchors and does thus not transmit personal information via UWB during the localization process.

Secondly, it was decided to confirm these theoretic considerations by building a passive and an active TDOA localization system and evaluating it based on selected privacy requirements from [1]. In addition, it was also decided to differentiate between a minimum and an improved version of the active TDOA localization system specifically for the privacy analysis. The results of the evaluation are summarized in Table 6.4. The results show, that the passive system indeed is more privacy-preserving than both versions of the active system. Only the authentication requirement is partially fulfilled by the passive system.

Thirdly, the evaluation also showed that adding the privacy-preserving improvements mentioned in Section 5.4 actually improves the privacy rating in Table 6.4. Namely, the improved active TDOA localization system performs better at the requirements of anonymity and unlinkability than the minimum active TDOA system.

When comparing the results of the privacy analysis to the literature, the first key observation is that including privacy as a requirement during the design process poses significant benefits for privacy. This design approach is called privacy by design and is highly recommended by [1]. For example, the choice of the localization approach has considerable effects on privacy and it is very difficult to switch between approaches after the implementation of the localization system.

The next connection to the literature shows, that implementing the mentioned privacy principles from [17; 12; 18] increased the level of privacy of the active system. Namely, adding dynamic addressing based on [15] and using encryption for the payload of data packages containing personal information as mentioned in [17; 12; 18] improved the rating of the privacy evaluation of the active system.

Another significant finding is that the chosen privacy criteria cover most of the privacy requirements mentioned in other works. This becomes clear when comparing the criteria used for the privacy analysis in Section 6.4 to the privacy policies mentioned in [16]. The privacy concerns in [16] that apply to the different phases of the IoT data flow and the privacy policies mentioned to tackle these concerns match relatively closely with the privacy criteria and privacy improvements encountered in this work. For example, the

concern of the revelation of sensitive user information concerning the user's activities in [16] is one of its key concerns in the data collection phase. This concern is also addressed by the unobservability criterion used in the privacy analysis of this work. Additionally, some of the other privacy policies mentioned in [16] such as data minimization are addressed in this work as well.

Next, when it comes to the IEEE 802.15.4 standard, the most useful privacy tool provided by the standard is the standardization of the encryption process [32]. This is done in the chapter about security which supports the security services of data confidentiality, data authenticity, and replay protection. For the UWB localization systems, these mechanisms are implemented with the help of the AES engine of the DW3xxx SDK (see Section 2.4 and [37]). With regards to privacy, the AES engine helps to implement privacy requirements such as anonymity and unlikability by encrypting packages that contain personal information during transmission and by helping to generate dynamic addresses.

When looking at the IEEE 802.15.4z amendment, the only contribution that can potentially improve privacy is the introduction of the STS. The main ability of the STS with regards to privacy is that it makes eavesdropping on UWB communication harder. However, further research is necessary in order to showcase how to design a reliable UWB localization system that uses the STS.

All in all, this paper found that the architecture of UWB localization systems indeed has an impact on privacy. First of all, using an architecture that uses a tag that only passively sends out localization messages increases privacy. When for other reasons, it is decided to use an architecture where personal information is for example shared by having the tag send out localization messages, the privacy of such a system can be increased by using dynamic addressing, encryption, and the STS, Encryption and the STS are tools that are provided within the IEEE 802.15.4 standard.

Therefore, there are two key contributions of this work: The first one is the results of the privacy analysis. This work is by the author's knowledge one of the first works that goes a step further than just considering UWB technology a privacy-preserving technology and also analyses the impact of UWB architectural choices. To do this, this work performed its second contribution, which is that it took COPri V2, an already existing ontology for privacy requirements, and adjusted it to use it for a privacy analysis on a network level.

7.2 Limitations

The limitations of this work can be grouped into two categories, technical and privacy-related limitations. First, the technical limitations are highlighted, followed by the privacy-related limitations.

When it comes to the technical limitations of this work, the first limitation noticeable is the systematic error in accuracy introduced by the antenna orientation effects. Based on the datasheet of the DW3000 module, these effects are particularly strong when the

antennas are oriented horizontally to each other [129]. Antenna orientation effects make it difficult to accurately track a patient's location in a real-world scenario as the patient's movements turn the tag around.

The next observation made is that the passive TDOA localization system shows a much lower precision than the active TDOA localization system. This is because the passive system uses the delayed transmission functionality. Hardware constraints limit the accuracy to which the transmission time can be specified when delaying a transmission. This introduces an error in the range of eight to twelve nanoseconds. In comparison, reception timestamps can be determined to the nanosecond accurate, giving the active localization system a much higher precision.

Further, one of the most important limitations is the usage of a deterministic STS for the active localization system. Due to the scope of this work, it was decided to use the deterministic STS mode provided by Qorvo to augment the privacy of the active TDOA system with reasonable effort. The addition of the deterministic STS allowed for an analysis of the working of the STS and the behavior of the sniffer when confronted with messages that contain an STS. However, such a system only provides limited security as skilled attackers can easily circumvent the deterministic STS.

There are many more limitations in the current implementation of the localization systems such as limited support for multiple tags, no mechanisms that deal with multipath and NLOS conditions, and lack of an optimization of the overall performance speed of the system. However, all these points have been tackled by previous work of research (see Chapter 3).

When it comes to privacy-related limitations, the most prominent limitation of this work is that only the localization system itself is analyzed based on its privacy performance but not the processes involved after the localization. This includes steps such as storing the location data on local servers or forwarding it to medical personnel (see Subsection 4.2.1). As mentioned by [16] and several other works (see Chapter 3), the steps after the data collection and the data processing are just as important in preserving privacy as the previous steps.

Another limitation that was noticed during the creation and performance of the privacy analysis is, that the used privacy requirements as well as the whole ontology in [1] do not specifically focus on the network layer of an application but rather on higher processing levels such as the application level. Therefore, the criteria provided by [1] did not ideally fit to analyze the UWB localization systems on a network level.

Lastly, authentication is currently handled in a way where an attacker can easily circumvent the authentication mechanism. At the moment, both the passive and the active localization systems check where a package is coming from by looking at the source address of a package. In particular, the first byte of the source address specifies whether a package is coming from a tag, indicated by a 'T', from an anchor 'A', or from a master anchor 'M'. An attacker with knowledge about this process can thus easily infiltrate the localization systems by providing fake messages starting with the symbols.

8

Conclusion and Future Work

In this Chapter, the key findings of this thesis are summarized and set into the broader context of this work. In addition, an overview of starting points for future work is given based on the limitations mentioned in Section 7.2.

8.1 Conclusion

UWB technology has established itself as one of the go-to technologies for indoor localization applications, many of which deal with a great amount of personal information [7]. Thus, there is a pressing need for developing UWB localization systems that safeguard user data.

Hence, this work designed, implemented, and evaluated two different UWB localization architectures based on their privacy characteristics. It became apparent that the type of UWB localization architecture used has indeed an impact on privacy. It was found that the UWB TDOA localization system that uses a passive tag shows a higher level of privacy compared to a TDOA localization system where the tag actively sends out localization messages. Secondly, it was discovered that even when opting for an active TDOA localization system, its privacy-preserving characteristics could be enhanced by using dynamic addressing, encryption for all packages that contain personal information in their body, as well as using an STS. These improvements could be achieved by using the tools provided by the IEEE 802.15.4 standard, which encompass standardized encryption methods as well as the inclusion of the STS through the newest IEEE 802.15.4z amendment.

All in all, this work highlighted the significance of incorporating privacy requirements during the early phases of the UWB localization system design, which makes it possible to identify design choices with a big impact on privacy early and to implement them with low effort. This work also showed that the current version of the IEEE 802.15.4 standard provides several tools to increase the privacy-preserving characteristics of a localization system. Lastly, this work found that there is a further need to develop a privacy framework that can be used for the data collection and processing level of all kinds of networks.

8.2 Future Work

Based on the limitations presented in Section 7.2, several starting points for future work can be identified. These are split into points that focus on technical aspects and ideas that tackle privacy-related improvements.

The first limitation mentioned in Section 7.2 is the systematic error introduced by antenna orientation effects. When building future localization systems with the DW3000 transceiver, it should be ensured that the anchor antennas are placed in a way that minimizes antenna orientation effects. Concretely, this means that when using the DWM3000, the boards should be oriented vertically to each other [129]. Additionally, the radiation patterns of the DW3000 antennas are known. Therefore, using the antennas in combination with an inertial measurement unit can help to further increase the accuracy.

The second limitation mentioned in Section 7.2 is the usage of a deterministic STS that can be easily circumvented by an attacker. However, in a real-world scenario, a fully-fledged non-deterministic STS solution should be implemented to guarantee security and privacy. Based on a brief investigation, no TDOA UWB systems were found that implement the STS for the exchanged packages. Designing such a system would be important to show how to overcome several challenges such as keeping the counters of all devices involved synchronized as well as still delivering reasonable performance and complexity.

In terms of privacy-related starting points for future work, one point to focus on is the later steps in the monitoring pipeline. When looking at the scope of this work defined in Subsection 4.2.1, later steps in the patient monitoring pipeline such as the storage of the patient's location on servers or the transmission to medical personnel have been excluded from the scope of this work. However, privacy-preserving treatment of data during the later stages of the monitoring pipeline is just as important as privacy-preserving treatment during earlier steps. As a starting point for future work, [16] provides a list of privacy concerns and privacy policies that can be for the design of the system's architecture as well as the privacy analysis. Additionally, the criteria of [1] can also be reused for the privacy analysis of later steps of the patient monitoring pipeline.

Future work could also focus on designing a more robust privacy analysis framework to perform the privacy analysis on a network level. The current framework is based on the privacy requirements of [1], which were not specifically designed to analyze an application on the network level. Hence, future work could focus on further adapting the privacy framework to fit the privacy requirements on a network level.

Another point that can be improved is the authentication process used by the localization systems. Currently, the origin of a message is identified via the source address field which can be easily tampered with. A future localization system could use a MIC for all messages to avert tampering. As the MIC is generated with the help of the AES engine and a nonce, it is very difficult for an attacker to generate a valid MIC without knowing the secret key.

Lastly, only a limited amount of architecture and use cases were analyzed. UWB applications are on the rise on so are a multitude of UWB architectures, applications, and use cases. Further studies could for example focus on evaluating UWB applications that are currently on the market and that are used in environments where a lot of personal information is generated. For example, the UWB beacons by Estimote briefly mentioned in Subsection 4.1.3 as well as the Estimote cloud could be thoroughly analyzed based on their privacy characteristics.

Bibliography

[1] M. Gharib, P. Giorgini, and J. Mylopoulos, "Copri v.2 — a core ontology for privacy requirements," *Data & Knowledge Engineering*, vol. 133, pp. 1–20, 2021.

[2] J. Tiemann, J. Friedrich, and C. Wietfeld, "Experimental evaluation of IEEE 802.15.4z UWB ranging performance under interference," *Sensors*, vol. 22, no. 4, pp. 1643–1661, 2022.

[3] Samsung, "Specs galaxy s21 ultra 5g." *https://www.samsung.com/uk/smartphones/galaxy-s21-ultra-5g/specs/*, 2023. Accessed: 2023-06-26.

[4] Apple, "iphone 11." *https://www.apple.com/by/iphone-11/specs/*, 2023. Accessed: 2023-06-26.

[5] BMW, "What's the deal with ultra-wideband?." *https://www.bmw.com/en/innovation/bmw-digital-key-plus-ultra-wideband.html*, 2023. Accessed: 2023-06-26.

[6] S. Newsroom, "Samsung announces ultra-wideband chipset with centimeter-level accuracy for mobile and automotive devices." *https://news.samsung.com/global/samsung-announces-ultra-wideband-chipset-with-centimeter-level-accuracy-for-mobile-and-automotive-devices*, 2023. Accessed: 2023-06-26.

[7] Fira, "Uwb use cases." *https://www.firaconsortium.org/discover/use-cases*, 2023. Accessed: 2023-06-26.

[8] K. Ota, Y. Ota, M. Otsu, and A. Kajiwara, "Elderly-care motion sensor using uwb-ir," in *Proceedings of the IEEE Sensors Applications Symposium*, (San Antonio, TX), pp. 159–162, IEEE, 2011.

[9] Y.-H. Liu, S. Sheelavant, M. Mercuri, P. Mateman, J. Dijkhuis, W. Zomagboguelou, A. Breeschoten, S. Traferro, Y. Zhan, T. Torf, C. Bachmann, P. Harpe, and M. Babaie, "9.3 a680 μw burst-chirp uwb radar transceiver for vital signs and occupancy sensing up to 15m distance," in *Proceedings of the IEEE International Solid- State Circuits Conference - (ISSCC)*, (San Francisco, CA), pp. 166–168, IEEE, 2019.

[10] K. Jimi, H. Seto, and A. Kajiwara, "Bathroom monitoring with fast-chirp modulation millimeter-wave uwb radar," in *Proceedings of the IEEE Radio and Wireless Symposium (RWS)*, (San Antonio, TX), pp. 134–137, IEEE, 2020.

[11] F. M. Noori, M. Z. Uddin, and J. Torresen, "Ultra-wideband radar-based activity recognition using deep learning," *IEEE Access*, vol. 9, pp. 138132–138143, 2021.

[12] S. Holcer, J. Torres-Sospedra, M. Gould, and I. Remolar, "Privacy in indoor positioning systems: A systematic review," in *Proceedings of the 2020 International Conference on Localization and GNSS (ICL-GNSS)*, (Tampere, FIN), pp. 1–6, IEEE, 2020.

[13] K. Järvinen, H. Leppäkoski, E.-S. Lohan, P. Richter, T. Schneider, O. Tkachenko, and Z. Yang, "Pilot: Practical privacy-preserving indoor localization using outsourcing," in *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*, (Stockholm, SWE), pp. 448–463, IEEE, 2019.

[14] J. Yan, Y. Meng, X. Yang, X. Luo, and X. Guan, "Privacy-preserving localization for underwater sensor networks via deep reinforcement learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1880–1895, 2021.

[15] L. Brilli, T. Pecorella, L. Pierucci, and R. Fantacci, "A novel 6lowpan-nd extension to enhance privacy in ieee 802.15.4 networks," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, (Washington, DC), pp. 1–6, IEEE, 2016.

[16] M. Sarrab and F. Alshohoumi, "Privacy concerns in iot a deeper insight into privacy concerns in iot based healthcare," *International Journal of Computing and Digital Systems*, vol. 9, no. 03, 2020.

[17] B. Liu, W. Zhou, T. Zhu, L. Gao, and Y. Xiang, "Location privacy and its applications: A systematic study," *IEEE Access*, vol. 6, pp. 17606–17624, 2018.

[18] H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar, "Location privacy-preserving mechanisms in location-based services: A comprehensive survey," *ACM Comput. Surv.*, vol. 54, no. 1, pp. 4–41, 2021.

[19] D. K. Alferidah and N. Jhanjhi, "A review on security and privacy issues and challenges in internet of things," *International Journal of Computer Science and Network Security IJCSNS*, vol. 20, no. 4, pp. 263–286, 2020.

[20] T. Shu, Y. Chen, J. Yang, and A. Williams, "Multi-lateral privacy-preserving localization in pervasive environments," in *Proceedings of the INFOCOM 2014 - IEEE Conference on Computer Communications*, (Toronto, CAN), pp. 2319–2327, IEEE, 2014.

[21] Z. Yang and K. Järvinen, "The death and rebirth of privacy-preserving wifi fingerprint localization with paillier encryption," in *Proceedings of the INFOCOM 2018*

- *IEEE Conference on Computer Communications*, (Honolulu, HI), pp. 1223–1231, IEEE, 2018.

[22] ITU, "Characteristics of ultra-wideband technology." *https://www.itu.int/dms_pubrec/itu-r/rec/sm/R-REC-SM.1755-0-200605-I!!PDF-E.pdf*, 2006. Accessed: 2023-01-27.

[23] R. S. Data, "Ultra wide-band technology." *https://www.rtsmartdata.com/technology/uwb*, 2023. Accessed: 2023-06-26.

[24] Z. Sahinoglu, S. Gezici, and I. Guvenc, *Ultra-wideband positioning systems: Theoretical Limits, Ranging Algorithms, and Protocols*, vol. 1. Cambridge, UK: Cambridge University Press, 2008.

[25] D. Coppens, E. De Poorter, A. Shahid, S. Lemey, and C. Marshall, "An overview of ultra-wideband (UWB) standards (IEEE 802.15. 4, FiRa, Apple): Interoperability aspects and future research directions," *arXiv preprint arXiv:2202.02190*, 2022.

[26] M. Yavari and B. G. Nickerson, "Ultra wideband wireless positioning systems," technical report tr14-230, Dept. Faculty Computer Science, University New Brunswick, Fredericton, CAN, March 2014.

[27] "802.15.4-2015 IEEE standard for low-rate wireless networks," IEEE, C/LM - LAN/MAN Standards Committee, 2015.

[28] "IEEE standard for low-rate wireless networks–amendment 1: Enhanced ultra wideband (UWB) physical layers (PHYs) and associated ranging techniques," IEEE, C/LM - LAN/MAN Standards Committee, 2020.

[29] Apple, "Nearby interaction with UWB." *https://developer.apple.com/nearby-interaction*, 2021. Accessed: 2023-02-16.

[30] C. C. Consortium, "Digital key release 3.0 specification download." *https://carconnectivity.org/digital-key-release-3-0-specification-download/*, 2022. Accessed: 2023-02-16.

[31] L. Alkama and L. Bouallouche-Medjkoune, "Ieee 802.15.4 historical revolution versions: A survey," *Computing*, vol. 103, no. 1, pp. 99–131, 2021.

[32] "802.15.4-2020 IEEE standard for low-rate wireless networks," IEEE, C/LM - LAN/MAN Standards Committee, 2020.

[33] FiRa Consortium, "Our vision." *https://www.firaconsortium.org/*, 2022. Accessed: 2023-02-16.

[34] FiRa Consortium, "Technical specifications." *https://www.firaconsortium.org/index.php/certifications/technical-specifications*, 2023. Accessed: 2023-02-16.

[35] FiRa Consortium, "Fira certified devices." *https://www.firaconsortium.org/index.php/certifications/certified-devices*, 2023. Accessed: 2023-02-16.

[36] Omlox, "The omlox hub and its api." *https://omlox.com/omlox-explained/omlox-hub-and-api*, 2023. Accessed: 2023-02-16.

[37] Qorvo, *DW3000 FAMILY USER MANUAL, version 1.1.* Qorvo, 2019.

[38] M. Poturalski, M. Flury, P. Papadimitratos, J.-P. Hubaux, and J.-Y. Le Boudec, "Distance bounding with ieee 802.15.4a: Attacks and countermeasures," *IEEE Transactions on Wireless Communications*, vol. 10, no. 4, pp. 1334–1344, 2011.

[39] Qorvo, "DWM3000EWB SDK." *https://www.qorvo.com/products/d/da007992*. Accessed: 2023-06-11.

[40] "Advanced encryption standard (aes)," FIPS PUB 197, National Institute of Standards and Technology (NIST), 2001.

[41] J. Daemen and V. Rijmen, "Aes proposal: Rijndael," 1999.

[42] N. Penchalaiah and R. Seshadri, "Effective comparison and evaluation of des and rijndael algorithm (aes)," *International journal of computer science and engineering*, vol. 2, no. 05, pp. 1641–1645, 2010.

[43] N. Semiconductor, "Ccm — aes ccm mode encryption." *https://infocenter.nordicsemi.com/index.jsp?topic=2Fcom.nordic.infocenter.nrf52832.ps.v1.12Fccm.html*, 2021. Accessed: 2023-12-04.

[44] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa, "Ultra wideband indoor positioning technologies: Analysis and recent advances," *Sensors*, vol. 16, no. 5, pp. 707–743, 2016.

[45] S. Gezici, "A survey on wireless position estimation," *Wireless Personal Communications*, vol. 44, p. 263–282, 2008.

[46] R. F. Brena, J. P. García-Vázquez, C. E. Galván-Tejada, D. Muñoz-Rodriguez, C. Vargas-Rosales, and J. Fangmeyer, "Evolution of indoor positioning technologies: A survey," *Journal of Sensors*, vol. 2017, 2017.

[47] M. Ridolfi, J. Fontaine, B. V. Herbruggen, W. Joseph, J. Hoebeke, and E. D. Poorter, "UWB anchor nodes self-calibration in nlos conditions: a machine learning and adaptive phy error correction approach," *Wireless Networks*, vol. 27, no. 4, pp. 3007–3023, 2021.

[48] C. L. Sang, M. Adams, T. Hörmann, M. Hesse, M. Porrmann, and U. Rückert, "An analytical study of time of flight error estimation in two-way ranging methods," in *Proceedings of the 2018 International Conference on Indoor Positioning and Indoor Navigation*, (Nantes, FRA), pp. 1–8, IEEE, 2018.

[49] M. Khalaf-Allah, "Novel solutions to the three-anchor toa-based three-dimensional positioning problem," *Sensors*, vol. 21, no. 21, pp. 1–27, 2021.

[50] S. Subedi and J.-Y. Pyun, "A survey of smartphone-based indoor positioning system using rf-based wireless technologies," *Sensors*, vol. 20, pp. 1–32, 2020.

[51] O. Bialer, D. Raphaeli, and A. J. Weiss, "Efficient time of arrival estimation algorithm achieving maximum likelihood performance in dense multipath," *IEEE Trans. Signal Process.*, vol. 60, no. 3, pp. 1241–1252, 2012.

[52] J. J. Caffery, *Wireless location in CDMA cellular radio systems*. Cincinnaty, OH: Springer Science & Business Media, 2000.

[53] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, 1952.

[54] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, pp. 308–313, 01 1965.

[55] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 1–32, 2019.

[56] I. Dotlic, A. Connell, H. Ma, J. Clancy, and M. McLaughlin, "Angle of arrival estimation using decawave dw1000 integrated circuits," in *proceedings of the 14th Workshop on Positioning, Navigation and Communications (WPNC)*, (Bremen, DEU), pp. 1–6, IEEE, 2017.

[57] H. J. Smith, T. Dinev, and H. Xu, "Information privacy research: An interdisciplinary review," *MIS Quarterly*, vol. 35, no. 4, pp. 989–1015, 2011.

[58] S. D. Warren and L. D. Brandeis, "The right to privacy," *Harvard Law Review*, vol. 4, no. 5, pp. 193–220, 1890.

[59] F. Bélanger and R. E. Crossler, "Privacy in the digital age: A review of information privacy research in information systems," *MIS Quarterly*, vol. 35, no. 4, pp. 1017–1041, 2011.

[60] D. J. Solove, "A taxonomy of privacy," *University of Pennsylvania Law Review*, vol. 154, no. 3, pp. 477–564, 2006.

[61] M. Palmirani, M. Martoni, A. Rossi, C. Bartolini, and L. Robaldo, "Pronto: Privacy ontology for legal reasoning," in *proceedings of the Electronic Government and the Information Systems Perspective* (A. Kő and E. Francesconi, eds.), EGOVIS, (Cham, CHE), pp. 139–152, Springer International Publishing, 2018.

[62] M. F. Arruda and R. F. Bulcão Neto, "Toward a lightweight ontology for privacy protection in iot," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, (New York, NY, USA), p. 880–888, Association for Computing Machinery, 2019.

[63] A. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46–55, 2003.

[64] S. Hayward, K. van Lopik, C. Hinde, and A. West, "A survey of indoor location technologies, techniques and applications in industry," *Internet of Things*, vol. 20, pp. 1–19, 2022.

[65] Z. Xiao, Y. Hei, Q. Yu, and K. Yi, "A survey on impulse-radio uwb localization," *Science China Information Sciences*, vol. 53, no. 7, pp. 1322–1335, 2010.

[66] G. Shi and Y. Ming, "Survey of indoor positioning systems based on ultra-wideband (uwb) technology," in *Proceedings of Wireless Communications, Networking and Applications* (Q.-A. Zeng, ed.), (New Delhi, IND), pp. 1269–1278, Springer India, 2016.

[67] P. Leu, G. Camurati, A. Heinrich, M. Roeschlin, C. Anliker, M. Hollick, S. Capkun, and J. Classen, "Ghost peak: Practical distance reduction attacks against HRP UWB ranging," in *Proceedings of the 31st USENIX Security Symposium*, (Boston, MA), pp. 1–19, USENIX Association, 2022.

[68] T. Wang, K. Hu, Z. Li, K. Lin, J. Wang, and Y. Shen, "A semi-supervised learning approach for uwb ranging error mitigation," *IEEE Wireless Communications Letters*, vol. 10, no. 3, pp. 688–691, 2021.

[69] S. Chen, D. Yin, and Y. Niu, "Research and implementation of improved ss-twr toa positioning method based on uwb," in *Proceedings of 2021 International Conference on Autonomous Unmanned Systems (ICAUS 2021)* (M. Wu, Y. Niu, M. Gu, and J. Cheng, eds.), (Singapore, SGP), pp. 3424–3434, Springer Singapore, 2022.

[70] S. Wang, G. Mao, and J. A. Zhang, "Joint time-of-arrival estimation for coherent uwb ranging in multipath environment with multi-user interference," *IEEE Transactions on Signal Processing*, vol. 67, no. 14, pp. 3743–3755, 2019.

[71] A. Poulose and D. S. Han, "Uwb indoor localization using deep learning lstm networks," *Applied Sciences*, vol. 10, no. 18, pp. 6290–6313, 2020.

[72] J. J. Pérez-Solano, S. Ezpeleta, and J. M. Claver, "Indoor localization using time difference of arrival with UWB signals and unsynchronized devices," *Ad Hoc Networks*, vol. 99, pp. 1–32, 2020.

[73] S. Bottigliero, D. Milanesio, M. Saccani, and R. Maggiora, "A low-cost indoor real-time locating system based on tdoa estimation of uwb pulse sequences," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[74] W. Zhao, A. Goudar, and A. P. Schoellig, "Finding the right place: Sensor placement for uwb time difference of arrival localization in cluttered indoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6075–6082, 2022.

[75] Y. Zhang and L. Duan, "A phase-difference-of-arrival assisted ultra-wideband positioning method for elderly care," *Measurement*, vol. 170, pp. 1–8, 2021.

[76] F. Ge and Y. Shen, "Single-anchor ultra-wideband localization system using wrapped pdoa," *IEEE Transactions on Mobile Computing*, vol. 21, no. 12, pp. 4609–4623, 2022.

[77] R. Fujiwara, K. Mizugaki, T. Nakagawa, D. Maeda, and M. Miyazaki, "Toa/tdoa hybrid relative positioning system using uwb-ir," in *Proceedings of the 2009 IEEE Radio and Wireless Symposium*, (San Diego, CA), pp. 679–682, IEEE, 2009.

[78] Y.-Y. Li, G.-Q. Qi, and A.-D. Sheng, "Performance metric on the best achievable accuracy for hybrid toa/aoa target localization," *IEEE Communications Letters*, vol. 22, no. 7, pp. 1474–1477, 2018.

[79] A. Chugunov, E. Zakharova, A. Mitic, V. Semenov, A. Boldyrev, D. Tsaregorodtsev, and N. Petukhov, "Integration of local ultrawideband toa/aoa phase difference of arrival system and inertial navigation systems," in *Proceedings of the 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, (St. Petersburg, RUS), pp. 1–8, IEEE, 2020.

[80] S. Monica and F. Bergenti, "Hybrid indoor localization using wifi and uwb technologies," *Electronics*, vol. 8, no. 3, pp. 334–348, 2019.

[81] X. Guo, N. Ansari, L. Li, and L. Duan, "A hybrid positioning system for location-based services: Design and implementation," *IEEE Communications Magazine*, vol. 58, no. 5, pp. 90–96, 2020.

[82] Y. Gao, H. Jing, M. Dianati, C. M. Hancock, and X. Meng, "Performance analysis of robust cooperative positioning based on gps/uwb integration for connected autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 790–802, 2023.

[83] W. C. Chung and D. Ha, "An accurate ultra wideband (uwb) ranging for precision asset location," in *Proceedings of the IEEE Conference on Ultra Wideband Systems and Technologies*, (Reston, VA), pp. 389–393, IEEE, 2003.

[84] T. Otim, A. Bahillo, L. E. Díez, P. Lopez-Iturri, and F. Falcone, "Impact of body wearable sensor positions on uwb ranging," *IEEE Sensors Journal*, vol. 19, no. 23, pp. 1–10, 2019.

[85] E. Ghanem, K. O'Keefe, and R. Klukas, "Testing vehicle-to-vehicle relative position and attitude estimation using multiple uwb ranging," in *Proceedings of the 92nd Vehicular Technology Conference (VTC2020-Fall)*, (Victoria, CAN), pp. 1–5, IEEE, 2020.

[86] Kinexon, "Industry." *https://kinexon.com/manufacturing/*, 2023. Accessed: 2023-03-12.

[87] Kinexon, "World's most trusted digital solution to mitigate the spread of covid-19." *https://kinexon.com/safezone/*, 2023. Accessed: 2023-06-26.

[88] Estimote, "What use cases are possible with proximity?." *https://community.estimote.com/hc/en-us/articles/360042185711-What-use-cases-are-possible-with-Proximity-*, 2023. Accessed: 2023-03-12.

[89] N. Macoir, J. Bauwens, B. Jooris, B. Van Herbruggen, J. Rossey, J. Hoebeke, and E. De Poorter, "UWB localization with battery-powered wireless backbone for drone-based inventory management," *Sensors*, vol. 19, no. 3, pp. 467–485, 2019.

[90] M. Hämäläinen, L. Mucchi, S. Caputo, L. Biotti, L. Ciani, D. Marabissi, and G. Patrizi, "Ultra-wideband radar-based indoor activity monitoring for elderly care," *Sensors*, vol. 21, no. 9, 2021.

[91] Kinexon, "Sports." *https://kinexon.com/sports/*, 2023. Accessed: 2023-06-26.

[92] Qorvo, "Bringing the bible to life with ultra-wideband." *https://www.qorvo.com/innovation/customer-stories/museum-of-the-bible*, 2021. Accessed: 2023-03-12.

[93] W. Sun, Z. Cai, Y. Li, F. Liu, S. Fang, and G. Wang, "Security and privacy in the medical internet of things: A review," *Security and Communication Networks*, vol. 2018, pp. 1–10, 2018.

[94] M. M. Ogonji, G. Okeyo, and J. M. Wafula, "A survey on privacy and security of internet of things," *Computer Science Review*, vol. 38, pp. 1–19, 2020.

[95] M. Bulenok, I. Tunaru, L. Biard, B. Denis, and B. Uguen, "Experimental channel-based secret key generation with integrated ultra wideband devices," in *Proceedings of the IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, (Valencia, ESP), pp. 1–6, IEEE, 2016.

[96] J. Miri, B. Nsiri, and R. Bouallegue, "Privacy group distance bounding protocol on th-uwb based ntru public key cryptosystem," in *Proceedings of the Sixth International Conference on Communications and Networking (ComNet)*, (Hammamet, TUN), pp. 1–7, IEEE, 2017.

[97] A. Alanwar, V. Gaßmann, X. He, H. Said, H. Sandberg, K. H. Johansson, and M. Althoff, "Privacy-preserving set-based estimation using partially homomorphic encryption," *European Journal of Control*, 2023. Preprint.

[98] D. S. Ha and P. R. Schaumont, "Replacing cryptography with ultra wideband (uwb) modulation in secure rfid," in *2007 IEEE International Conference on RFID*, (Grapevine, TX), pp. 23–29, IEEE, 2007.

[99] J. Misic, "Enforcing patient privacy in healthcare wsns using ecc implemented on 802.15.4 beacon enabled clusters," in *Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, (Hong Kong, CHN), pp. 686–691, IEEE, 2008.

[100] M. F. Sadikin and M. Kyas, "Rfid-tate: Efficient security and privacy protection for active rfid over ieee 802.15.4," in *Proceedings of the IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications*, (Chania, GRC), pp. 335–340, IEEE, 2014.

[101] B. Mao, Y. Kawamoto, J. Liu, and N. Kato, "Harvesting and threat aware security configuration strategy for ieee 802.15.4 based iot networks," *IEEE Communications Letters*, vol. 23, no. 11, pp. 2130–2134, 2019.

[102] G. Kalyani and S. Chaudhari, "Data privacy preservation in mac aware internet of things with optimized key generation," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 2062–2071, 2022.

[103] M. Stocker, B. Großwindhager, C. A. Boano, and K. Römer, "Towards secure and scalable uwb-based positioning systems," in *Proceedings of the IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, (Delhi, IN), pp. 247–255, IEEE, 2020.

[104] N. Semiconductor, "Nordic semiconductor - empowering wireless innovation." *https://www.nordicsemi.com/*, 2023. Accessed: 2023-03-12.

[105] N. Semiconductor, "Bluetooth low energy." *https://www.nordicsemi.com/Products/Bluetooth-Low-Energy/Development-hardware?lang=en#infotabs*, 2023. Accessed: 2023-03-12.

[106] N. Semiconductor, "nrf5 sdk." *https://www.nordicsemi.com/Products/Development-software/nRF5-SDK/Download#infotabs*, 2023. Accessed: 2023-03-12.

[107] N. Semiconductor, "Segger embedded studio." *https://www.nordicsemi.com/Products/Development-tools/segger-embedded-studio*, 2023. Accessed: 2023-03-12.

[108] Qorvo, "Ultra-wideband." *https://www.qorvo.com/products/wireless-connectivity/ultra-wideband*, 2023. Accessed: 2023-06-11.

[109] Qorvo, "Dwm1000." *https://www.qorvo.com/products/p/DWM1000*, 2023. Accessed: 2023-03-13.

[110] Qorvo, "Dwm3000." *https://www.qorvo.com/products/p/DWM3000*, 2023. Accessed: 2023-03-13.

[111] Qorvo, "Dwm3000evb." *https://www.qorvo.com/products/p/DWM3000EVB#documents*, 2023. Accessed: 2023-03-13.

[112] Qorvo, "Dwm3001cdk." *https://www.qorvo.com/products/p/DWM3001CDK#documents*, 2023. Accessed: 2023-03-13.

[113] Estimote, "Uwb beacons - spatial awareness for your mobile apps." *https://estimote.com/uwb-beacons*, 2023. Accessed: 2023-03-12.

[114] Qorvo, "Estimote." *https://www.qorvo.com/innovation/ultra-wideband/partners/estimote*, 2023. Accessed: 2023-03-12.

[115] Estimote, "Spacetimeos is a special kind of operating system." *https://estimote.com/*, 2023. Accessed: 2023-03-12.

[116] Estimote, "Swift 5.6 estimoteuwb beta 0.1.9." *https://drive.google.com/drive/u/0/folders/1E5q4x80pfOIau51IBLoi1Em4TP7EblrV*, 2023. Accessed: 2023-03-12.

[117] Qorvo, "Dw3120." *https://www.qorvo.com/products/p/DW3120*, 2023. Accessed: 2023-03-13.

[118] J. I. Hong, J. D. Ng, S. Lederer, and J. A. Landay, "Privacy risk models for designing privacy-sensitive ubiquitous computing systems," in *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, DIS '04, (New York, NY, USA), p. 91–100, Association for Computing Machinery, 2004.

[119] H. Yan, H. Huo, Y. Xu, and M. Gidlund, "Wireless sensor network based e-health system - implementation and experimental results," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2288–2295, 2010.

[120] E. Y. Baafi, N. A. Schofield, and I. G. Congress, *Geostatistics Wollongong '96 / edited by E.Y. Baafi and N.A. Schofield.* Kluwer Academic Dordrecht ; Boston, 1997.

[121] *Mean Absolute Deviation*, pp. 336–337. New York, NY: Springer New York, 2008.

[122] Qorvo, "Dw3110." *https://www.qorvo.com/products/p/DW3110*, 2023. Accessed: 2023-03-13.

[123] Qorvo, "Ultra-wideband." *https://www.qorvo.com/products/wireless-connectivity/ultra-wideband*, 2023. Accessed: 2023-04-08.

[124] Qorvo, "Getting back to basics with ultra-wideband." *https://forum.qorvo.com/t/how-to-program-or-get-angle-of-arrival-from-dwm3000evb-to-iphone11-or-other-dwm3000evb/10948*, 2021. Accessed: 2023-04-08.

[125] Qorvo, *DWM3000 Product Brief Rev. B.* Qorvo, 2021.

[126] Qorvo, "Dw1000 ieee802.15.4-2011 uwb transceiver — product data sheet." *https://www.qorvo.com/products/p/DW1000#documents*, 2017. Accessed: 2023-12-04.

[127] Qorvo, "Dw3000 uwb transceiver — product data sheet." *https://www.qorvo.com/products/p/DW3000#documents*, 2020. Accessed: 2023-18-04.

[128] J. Zhu, "Calculation of geometric dilution of precision," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 3, pp. 893–895, 1992.

[129] Qorvo, "Dwm3000 — ieee 802.15.4-z uwb transceiver module." *https://www.qorvo.com/products/p/DW3000#documents*, 2021. Accessed: 2023-12-06.

Glossary

2D	Two dimensions.
AAL	Ambient assisted living.
AEAD	authenticated encryption with associated data.
AES	Advanced encryption standard.
AOA	Angle of arrival.
AP	Actual position.
ASN	Autonomous system number.
CCM*	cipher block chaining - message authentication code.
cm	Centimeters.
COM	Communication.
CP	Calculated position.
CTS	Clear to send.
dBm	Decibel-milliwatts.
DES	Data encryption standard.
DK	Development kit.
DS-TWR	Double-sided two-way ranging.
DTU	Device time unit.
FCC	Federal communication commission..
FCS	Frame checking sequence.
FiRa	Fine-ranging. Refers to the FiRa consortium.
GHz	Gigahertz.
GPS	Global positioning system.
GUI	Graphical user interface.
HRP	High rate pulse.
IC	Integrated circuit.

ID	Identifier.
IE	Information elements.
IEEE	Institute of electrical and electronics engineers.
IoT	Internet of things.
IR	Impulse radio.
kb/s	Kilobits per second.
kHz	Kilohertz.
LRP	Low rate pulse.
m	Meter.
MAC	Medium access control.
MAD	Mean absolute deviation.
Max	Maximum.
Mb/s	Megabits per second.
MHR	Medium access control header.
MHz	Megahertz.
MIC	Message integrity code.
Min	Minimum.
NLOS	Non-line-of-sight.
Nr	Number.
ns	Nanosecond(s).
OSI	Open system interconnection.
OTP	One-time pad.
PAC	Preamble acquisition chunk.
PAN	Personal area network.
PcapNg	Package capture next generation dump file format.
PDOA	Phase difference of arrival.
PHR	Physical header.
PHY	Physical.
PLL	Phase locked loop.
RAM	Random-access memory.
RSS	Received signal strength.
RTLS	Real-time location services.
RTS	Request to send.
RX	Receiver.
SDK	Software development kit.
SECDED	single error correct, double error detect.

SFD	Start of frame delimiter.
SoC	System on a chip.
SPI	Serial peripheral interface.
SS-TWR	Single-sided two-way ranging.
STS	Scrambled timestamp sequence.
SYNC	Synchronization.
TDOA	Time difference of arrival.
TOA	Time of arrival.
TOF	Time of flight.
TSCH	Time slotted channel hopping.
TX	Transmitter.
UART	Universal asynchronous receiver transmitter.
USA	united states of America.
USB	Universal serial bus.
UTF-8	Universal coded character set 8.
UWB	Ultra-Wideband.
WI-FI	Wireless fidelity. Wireless local area network, based on the IEEE 802.11 standard.
WPAN	Wireless personal area network.

A

Appendix

All supplementary material used for this thesis can be found in a Github repository under the following link: *https://github.com/cheder456/UWB_Master_Thesis*. This includes the code used for the active and passive TDOA localization, the localizer program, the raw data during the experiments, the cleaned data and the jupyter notebooks used for analysis as well as further helper files. Additionally, a .zip version of the repository is handed in together with this thesis.

A.1 COPri V.2

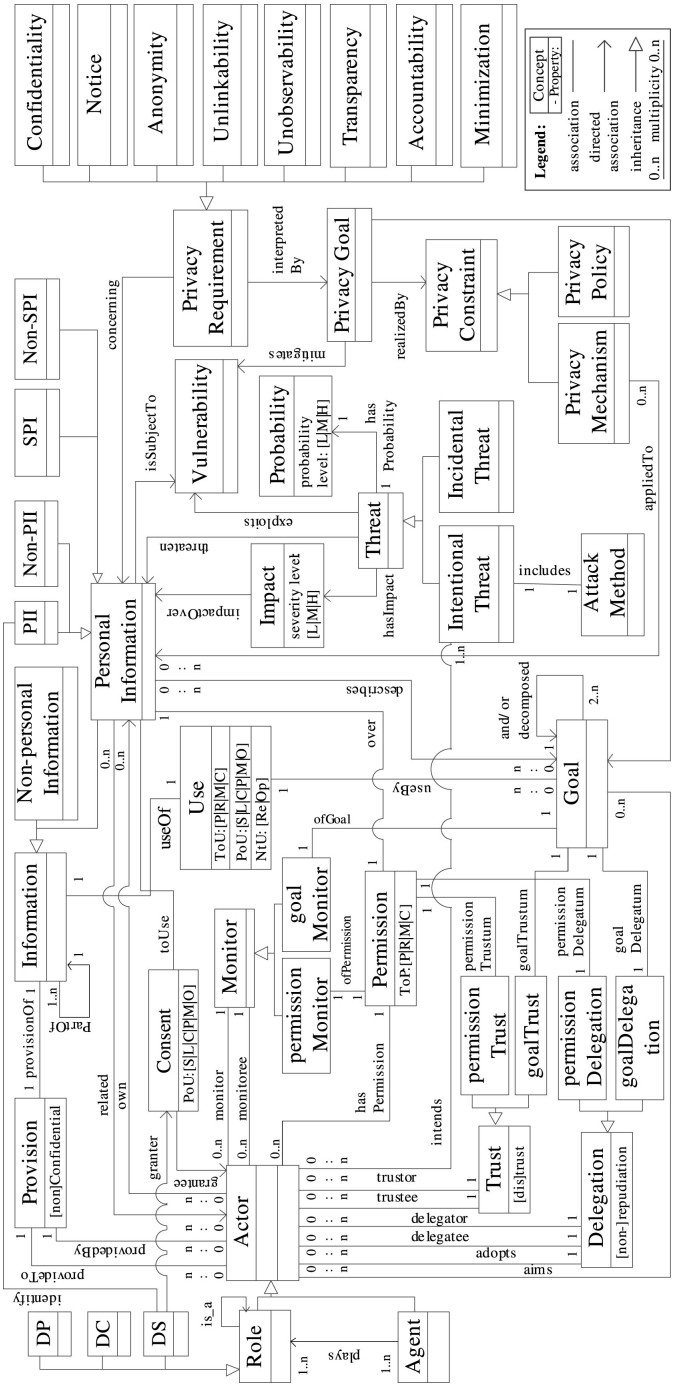

Figure A.1: Conceptual Model of COPri V2 [1].

A.2 Devices Used for the Experiments

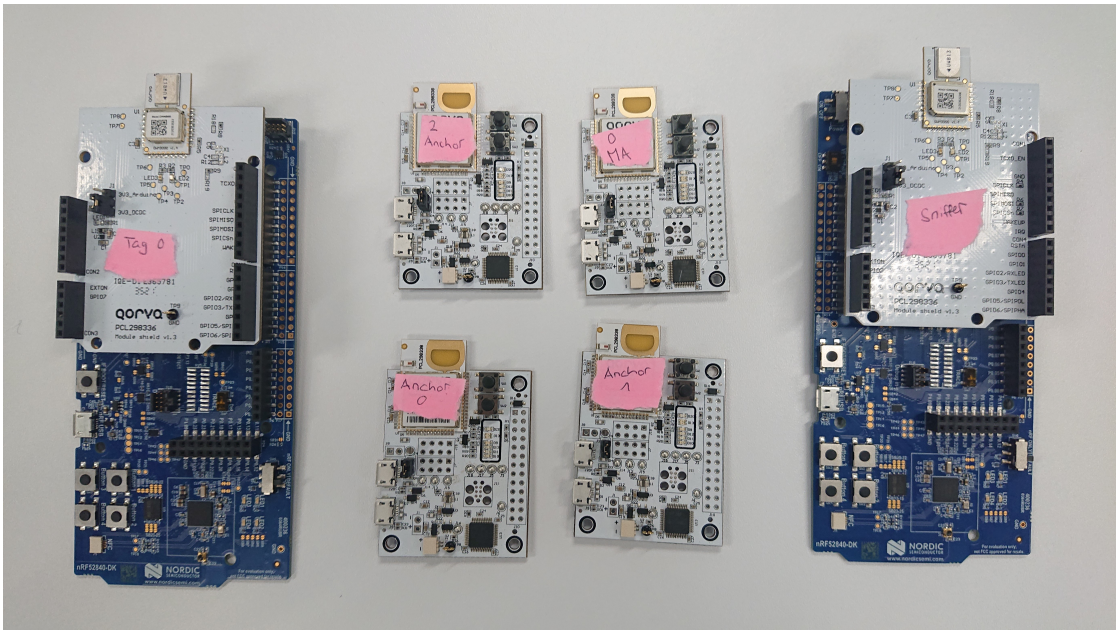

Figure A.2: These are the Devices Used for the Passive TDOA Localization System.

Figure A.3: These are the Devices Used for the Active TDOA Localization System.

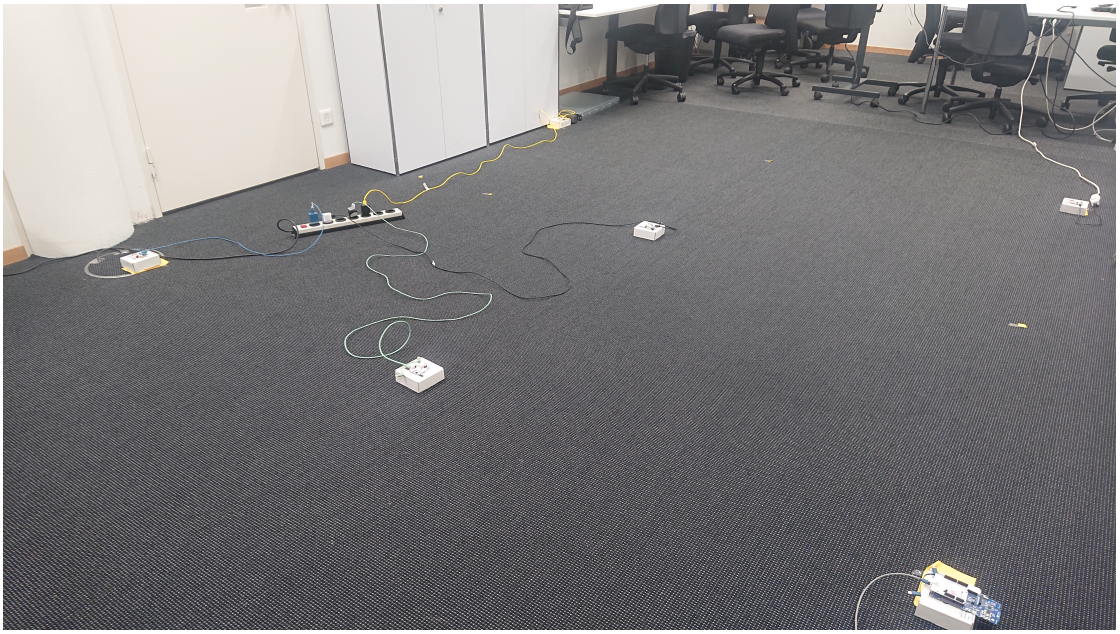

Figure A.4: These are the Devices Used for the Active TDOA Localization System.

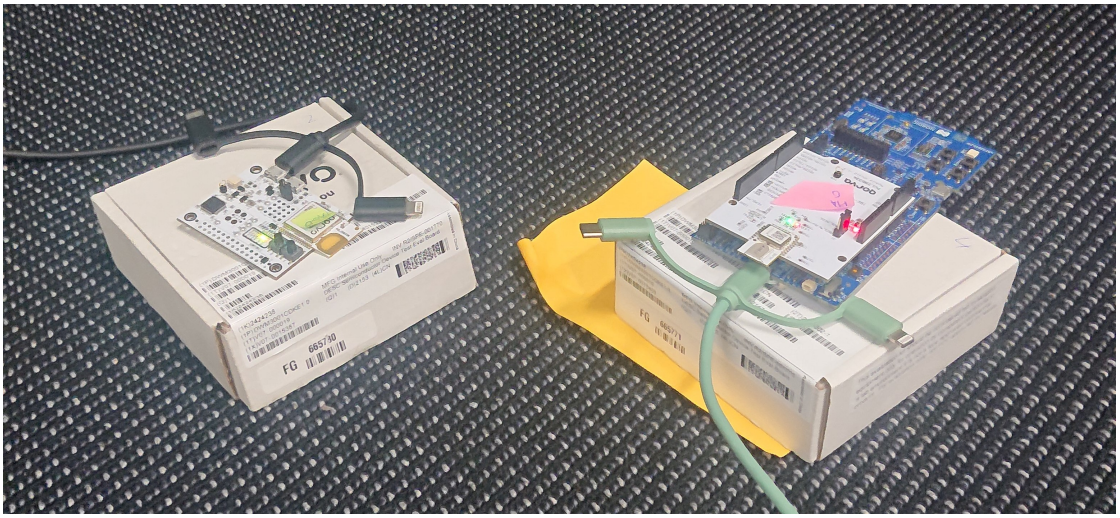

Figure A.5: These are the Devices Used for the Active TDOA Localization System.

A.3 Histograms of the Experiments

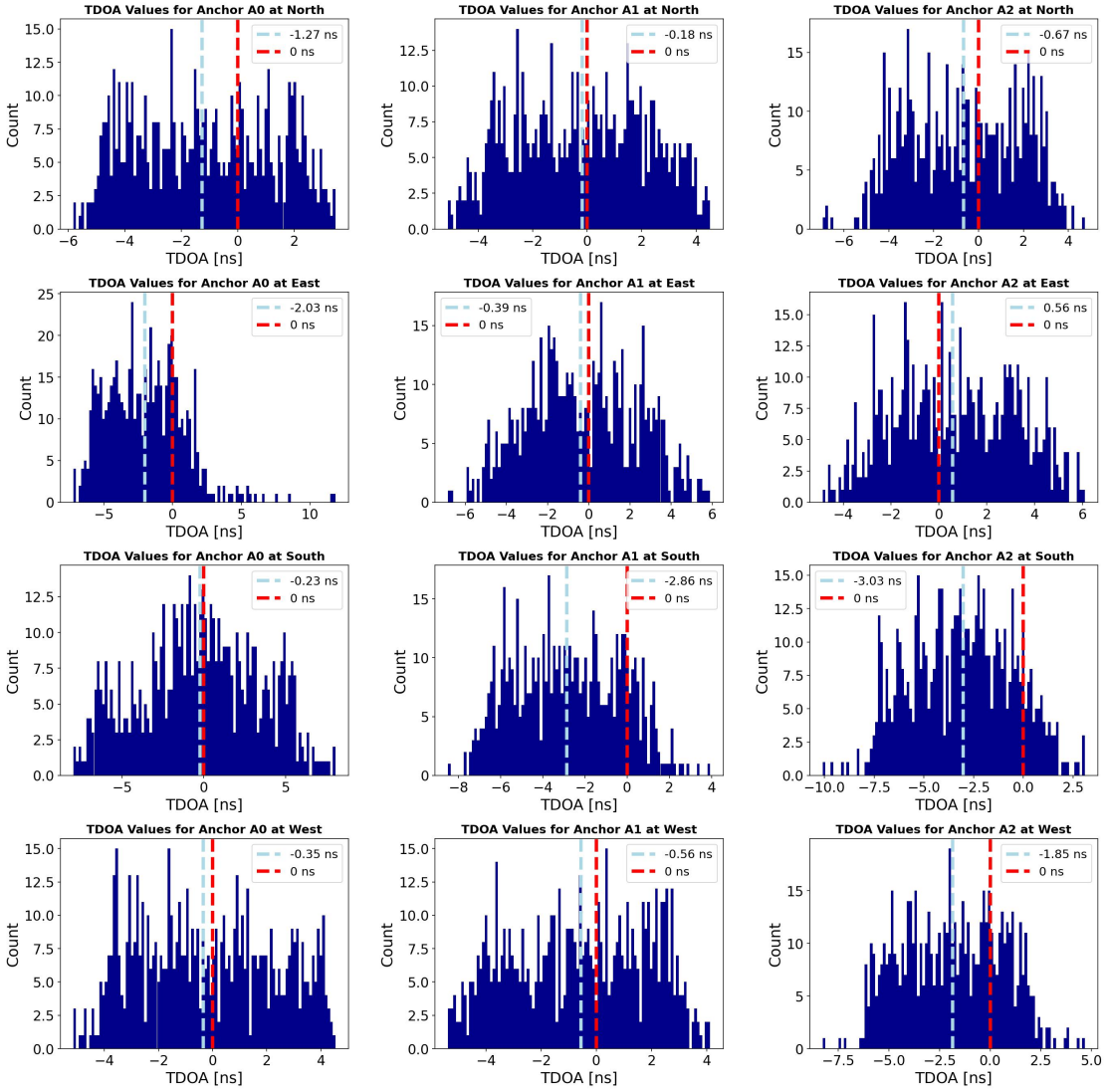

Figure A.6: Histogram of the TDOA Values in [ns] for the Experiment Suite with the Turning Tag and the Passive System Described in Subsection 5.6.3.

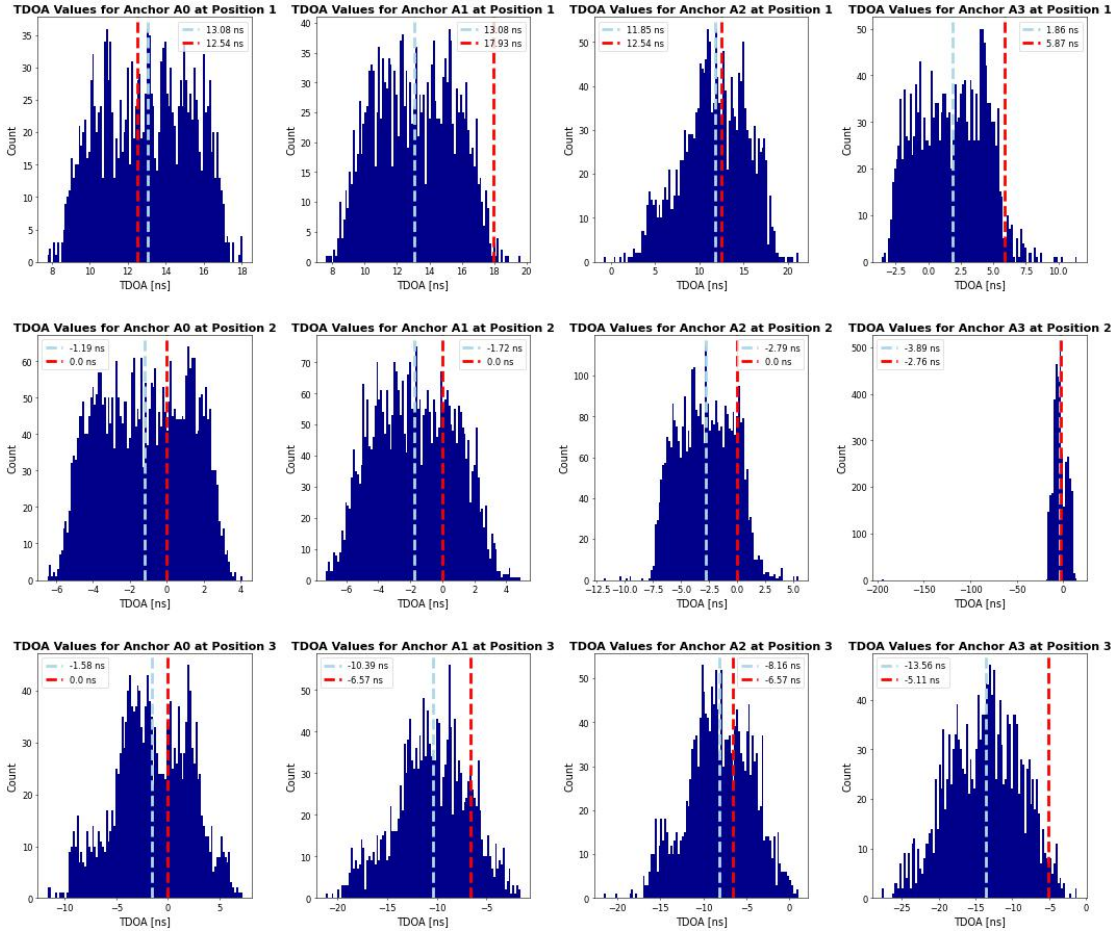

Figure A.7: Histogram of the TDOA Values in [ns] for the Experiment Suite with Five Anchors and the Passive System Described in Subsection 5.6.4.

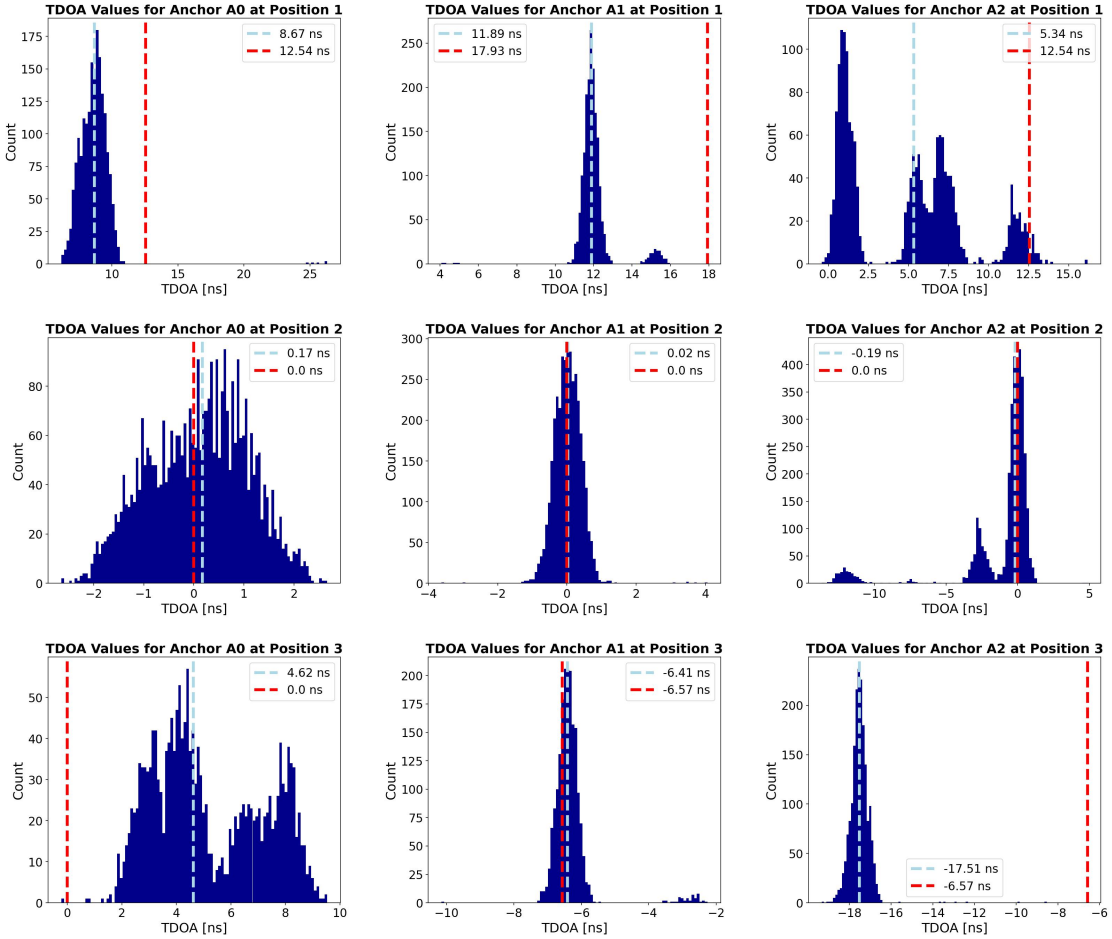

Figure A.8: Histogram of the TDOA Values in [ns] for the Experiment Suite with Four Anchors and No Boxes with the Active System Described in Subsection 5.7.2.

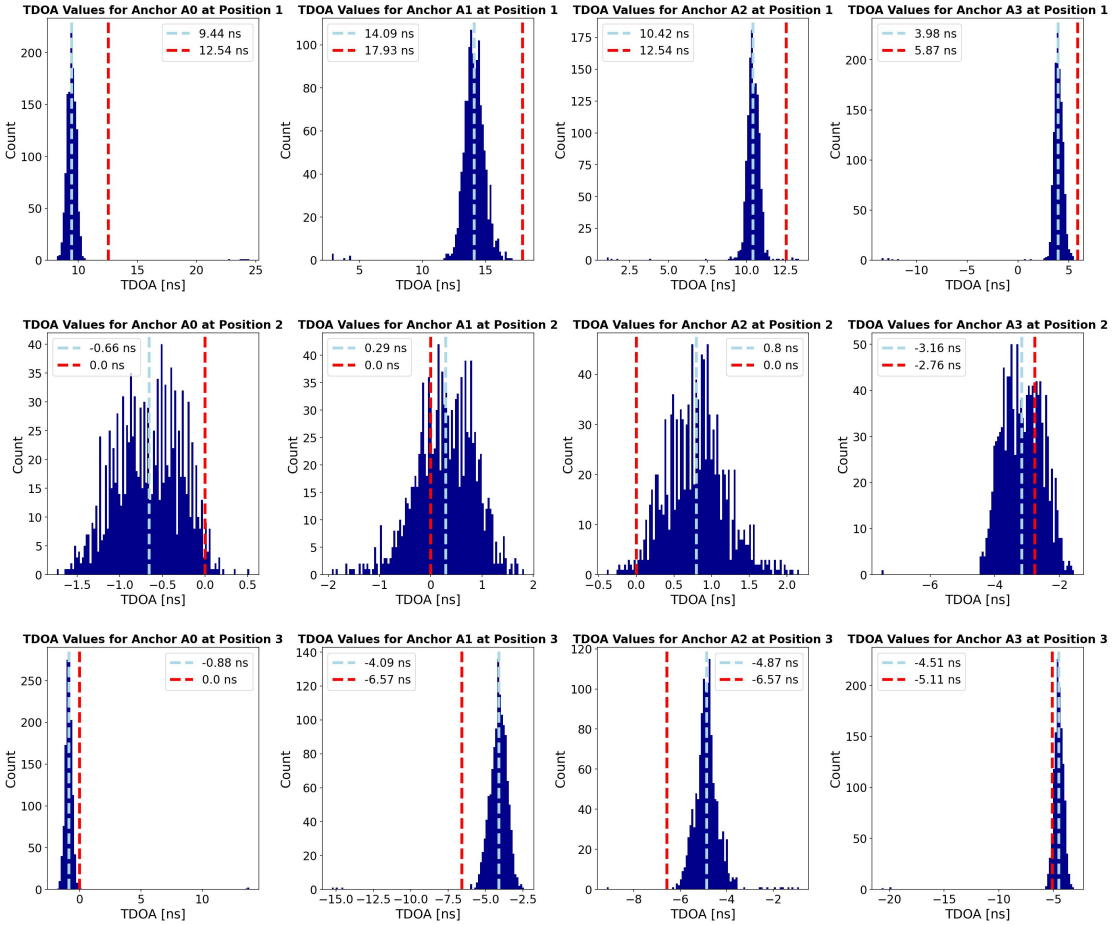

Figure A.9: Histogram of the TDOA Values in [ns] for the Experiment Suite with Five Anchors and Boxes with the Active System Described in Subsection 5.7.3.

A.4 Overview Experiments Performed

For this thesis, a series of localization experiments was performed. For the passive localization system, the following experiments were performed:

- Four Anchors
  - Standard Positions Used in this Thesis
  - Further Positions
  - Turning Tag at Multiple Locations
  - Blocking Objects
  - Long Run
- Five Anchors Left
  - Standard Positions
  - Turning Tag
- Five Anchors Middle
  - Standard Positions
  - Turning Tag

The raw data for these experiments can be found in the Github repository inside the folder *data*. Standard Positions refers to the positions used in thesis ((-1.9,-1.9), (0.0) (0.1.5)). All folders containing datasets to these experiments start with the keyword *passive* and were performed on the 24. of April 2023 and the 04. of May 2023. For the active system, the experiments start with the keyword *active* and were performed on the 19. of May 2023. The following experiments were performed:

- Four Anchors
  - Standard Positions Without Boxes
  - Further Positions Without Boxes
  - Standard Positions With Boxes
  - Further Positions With Boxes
  - Turning Tag With Boxes
  - Long Run With Boxes
- Five Anchors Left
  - Standard Positions Without Boxes
  - Standard Positions With Boxes
- Five Anchors Middle
  - Standard Positions Without Boxes
  - Standard Positions With Boxes

A.5 Overview Code Used

The code used for this master thesis can be found in the Github repository. The following code parts were used for this thesis:

- Active and Passive Localization System
  - Programs for UWB Modules: \core
  - Localizer Frontend: \Localizer
- Sniffer
  - Program for UWB Module: \core\further_files
  - Python Pipe: \Sniffer_PythonPipe
- Experiment Analysis: \Cleaned_Data_and_Analysis\UsedForMasterThesis
- PDOA Analysis: \pdoa_analysis

List of Figures

List of Tables

List of List of Codes