Towards Secure Multicast Ranging with Ultra-Wideband Systems

Michael Stocker, Jan Kowalczyk, Carlo Alberto Boano, and Kay Römer
Institute of Technical Informatics, Graz University of Technology, Austria
E-mails: {michael.stocker, cboano, roemer}@tugraz.at ; jan.kowalczyk@student.tugraz.at

ABSTRACT

Since the introduction of the IEEE 802.15.4z standard, compliant ultra-wideband devices support secure distance estimation: this enables a plethora of new use cases in safety-critical domains. However, the security features introduced by the standard do not apply to recent scalable and efficient distance estimation and positioning schemes in which nodes exchange multicast messages. In this work, we propose the concept of *sub-STS*, which allows multiple ultra-wideband devices that do not mutually trust each other to still derive a secure distance estimate despite the transmission of multicast messages. Compared to traditional approaches sending individual secure frames to each device, the use of multicast paired with *sub-STS* has the potential to reduce channel utilization by more than 50%. We develop a prototypical implementation of *sub-STS* on off-the-shelf UWB devices implementing the IEEE 802.15.4z standard, confirming its feasibility. We further evaluate the performance of *sub-STS* experimentally, and discuss its limitations due to the hardware limitations of current ultra-wideband platforms.

KEYWORDS

Channel impulse response, IEEE 802.15.4z, Qorvo DW3000, Scalability, Scrambled Time Sequence, Secure ranging, TDOA, UWB.

ACM Reference Format:
Michael Stocker, Jan Kowalczyk, Carlo Alberto Boano, and Kay Römer. 2022. Towards Secure Multicast Ranging with Ultra-Wideband Systems. In *Proceedings of International Conference on Embedded Wireless Systems and Networks (EWSN'22).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Ultra-wideband (UWB) technology offers outstanding time resolution and multi-path resilience compared to traditional narrow-band IoT technologies, and has thus emerged as one of the most popular choices for indoor positioning and tracking. The ability to achieve centimetre-level ranging accuracy allows to support a wide range of different applications, (e.g., asset tracking [9], robot navigation [12], or assisted living [20]), and has encouraged big industrial players such as Apple, Samsung, BMW, and VW to integrate UWB transceivers into their newest smartphones [19] and vehicles [6].

EWSN'22, October 3-5, 2022, Linz, Austria

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Support for multicast ranging. Many of the envisioned use cases require *scalable* and *efficient* communication schemes, possibly supplying thousands of mobile tags with distance estimates at high update rates while minimizing the energy expenditure, so to enable long-lasting battery operations. To this end, several techniques have been proposed as an alternative to classical two-way ranging schemes [3, 14]. For example, one-to-many (OTM) and many-to-many (MTM) ranging schemes [10] employ *multicast* messages to enable communication between multiple initiators and responders. In concurrent transmissions schemes, as proposed by Corbalán et al. [2] and Großwindhager et al. [7], multiple UWB devices transmit overlapping *broadcast* messages at the same time. Tiemann et al. [18] propose the use of Time Difference of Arrival (TDoA) to enable passive localization based on *broadcast* messages. SnapLoc [8] and Chorus [1] achieve high update rates by combining such TDoA approach with concurrent transmissions schemes.

Support for secure ranging in IEEE 802.15.4z. Many of the envisioned use-cases involving UWB-based systems entail safety-critical domains, which raises also the need for *secure* ranging schemes that prevent malicious attacks enlarging [16] or shortening [13] the estimated distances. In order to make UWB-based systems resilient against intentional or unintentional distance manipulation attempts, the IEEE standardization group has recently released the IEEE 802.15.4z amendment. The latter introduces, among others, the *scrambled timestamp sequence (STS)* field within an UWB frame, in order to prevent that any third party manipulates the reception time estimation process [10, 15] The security of IEEE 802.15.4z is built around a symmetric signal authentication and integrity verification operating on the STS field. Specifically, a pre-shared secret is distributed among all participating devices and is used to generate a unique pulse pattern in the STS field. Thus, only devices in possession of the pre-shared secret can generate and decode the STS field correctly. This, however, also means that when performing multicast ranging with a single UWB frame, the same secret needs to be shared among *all* participating devices in advance. This is often undesirable, as malicious devices may abuse pre-shared secrets to spoof the STS field and manipulate the distance estimation process of other devices. Clearly, from a security perspective, using individual messages with a dedicated STS for each device is preferable. This, however, clashes with the working principle of multicast ranging schemes recently proposed in the UWB literature, which involve the exchange of frames sent from multiple initiators or to multiple responders.

The gap to fill. What is desirable, is a solution allowing an UWB initiator to send multiple STS segments within the same frame, each generated with a different secret. The IEEE 802.15.4z standard actually foresees up to four individual STS segments within a single UWB frame, which can be used by a receiver to better estimate and validate the integrity of a signal's arrival time [10].

However, the standard neither specifies how this should be done, nor indicates whether each STS segment can be generated from different secrets [11, 17]. Moreover, state-of-the-art UWB radios such as the Qorvo DW3000 are not specifically designed to change the pre-shared secret for each STS segment. Even if they would, supporting only *four* individually-generated STS segments would be insufficient for many UWB-based systems (e.g., those based on TDoA [8, 18]), as it would limit their scalability.

We hence want to investigate whether one can support secure multicast ranging schemes by allowing the creation *in software* of an arbitrary number of STS segments, each generated with a different secret, on existing off-the-shelf IEEE 802.15.4z hardware.

Contributions. In this paper, we propose the concept of *sub-STS*, in which the STS field of a single UWB frame is created by using several different pre-shared secrets. This allows multiple devices that do not mutually trust each other to still derive secure distance estimates from the same UWB frame, towards the creation of secure and scalable positioning systems employing multicast ranging.

After presenting the key idea behind the *sub-STS*, we use simulations to quantify the theoretical improvements introduced by its usage, showing a reduction in the required airtime by more than 50%. We then develop a prototype of the *sub-STS* concept on off-the-shelf UWB devices based on the Qorvo DW3000 radio. Since the latter is not specifically designed to support such scheme, we evaluate experimentally the implications that a software-based implementation has on key metrics affecting ranging performance, such as the noise level and the received signal strength. After evaluating experimentally the feasibility and performance of our *sub-STS* implementation, we discuss our observations as well as the limitations of the proposed solution.

Paper outline. This work proceeds as follows. Sect. 2 introduces the IEEE 802.15.4z standard and the fundamentals of UWB-based secure distance estimation. Sect. 3 illustrates the working principle of *sub-STS* and discusses its potential benefits, as well as challenges in its practical realization. Sect. 4 describes our implementation on off-the-shelf UWB devices compliant to IEEE 802.15.4z, and presents the results from our preliminary evaluation. Sect. 5 concludes this work and elaborates on our next steps.

2 A PRIMER ON SECURE UWB RANGING

The simplest form of distance estimation between two UWB devices is single-sided two-way-ranging (SS-TWR), and involves the exchange of two IEEE 802.15.4z-compliant messages: *poll* and *response*. Fig. 1(a) illustrates the simplest and most common case: an initiator (A) sends a *poll* message to each of the responding nodes (in this example B and E), which answer with an individual *response* message. To calculate the distance, both nodes record the time of arrival (ToA) and the transmission time of the *poll* and *response* message. These four timestamps enable the initiating node to calculate the round-trip time and the distance between two devices. Fig. 1(b) illustrates a more efficient multicast ranging scheme, in which several responders receive the same *poll* message and answer with consecutive *response* messages. This eliminates the need for multiple *poll* messages, which significantly reduces the channel utilization. Note that there are schemes benefiting even more from multicast (broadcast) messages. For example, when using a classical

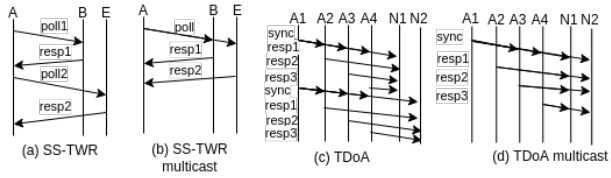

Figure 1: Exemplary distance estimation schemes. Classical SS-TWR using individual (a) and multicast (b) *poll* messages; classical TDoA system using individual (c) and multicast (d) *resp* messages.

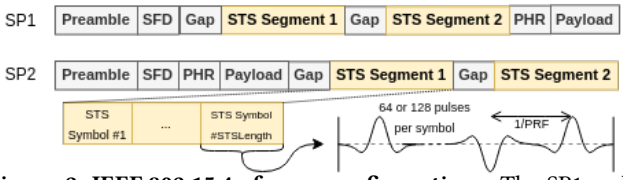

Figure 2: IEEE 802.15.4z frame configurations. The SP1 and SP2 frame formats differ in the position of the STS segments. Each STS segment is generated from 64 STS symbols (depending on the capabilities of the UWB receiver also other STS lengths are possible), which consist of 64 or 128 individual pulses.

TDoA scheme [18], a tag can self-localize by receiving the *resp* messages sent by at least four different anchors. The transmission of a multicast message coupled with TDoA [1, 8] allows to significantly reduce the number of responses, as illustrated in Fig. 1(c) and (d).

2.1 IEEE 802.15.4z Frame Format

The IEEE 802.15.4z standard specifies different UWB frame formats. Fig. 2 illustrates the SP1 and SP2 formats, which differ in the position of the STS segments. Both frame configurations feature a publicly known *preamble* sequence and *start of frame delimiter* (SFD) for frame detection and insecure ToA estimation, a *physical header* (PHR) and *payload* for data transmission, as well as up to four *scrambled timestamp sequence* (STS) segments for secure ToA estimation[1]. When using base pulse repetition frequency (BPRF)[2], each STS segment is made from 64 STS symbols that are approximately 1μs long. These symbols are constructed from 64 pulses sent at a mean rate of 64 MHz. The polarity of the pulses (i.e., -1, 1) is determined by the output of an AES-128 pseudo-random number generator. The output of the latter depends on two secrets: a 128-bit key and a 128-bit initialization vector (IV). The latter contains a 96-bit fixed part and a 32-bit variable (counter) part, which is increased by one for each 128-bit pseudo-random output. This ensures that each STS symbol is unique and that only devices in possession of the key and IV are able to generate the same STS symbols.

2.2 Secure ToA Estimation

For secure ToA estimation following the IEEE 802.15.4z specifications, the transmitting and receiving nodes must agree on the same key and IV before sending and obtaining a frame. Afterwards, the

[1]Note that Fig. 2 just illustrates two STS segments: in fact, up to four segments are foreseen by the IEEE 802.15.4z standard, but only two are mandatory.

[2]Note that the IEEE 802.15.4z standard defines also a higher pulse repetition frequency (HPRF) mode with longer STS segments, where pulses are sent at a higher rate. This mode is not supported by the Qorvo DW3000, and it is beyond the scope of this paper.

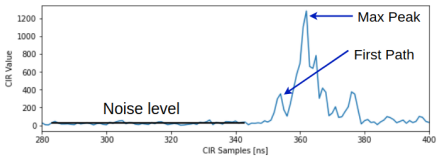

Figure 3: CIR estimate. A sufficient first peak detection algorithm must be sensitive enough to detect weak first paths, but must avoid to detect noise as first paths.

transmitting node generates and transmits an UWB frame containing pseudo-randomly generated STS symbols. A receiving device probes for incoming STS symbols by cross-correlating the received STS symbols' signal with a locally-generated template version. The output of the cross-correlation between the received signal and the expected signal produces an estimate of the channel impulse response (CIR) and multiple cross-correlation outputs are merged together to form a final estimate of the CIR as shown in Fig. 3.

First path detection. After acquiring a CIR estimate, a first path detection algorithm analyses the CIR to identify the first path. Such an algorithm typically starts by identifying the maximum peak and searches in a certain back-search window for earlier peaks passing a certain threshold. The latter is typically derived from the level of noise, but additional constraints may be applied to enhance the trustworthiness in the first peak detection, as discussed next.

Trade-off between security and sensitivity. By definition, the cross-correlation of two signals is a measure of their similarity. This property is used in secure UWB devices to validate the authenticity of the received STS symbols. Specifically, if the received signal matches with the expected STS symbols, the correlation output contains high values; otherwise, the correlation output contains comparably low values. Any artificially-introduced peaks in the CIR may confuse the first peak detection algorithm and cause wrong ToA estimates. This can happen either due to the non perfect auto-correlation property of the STS sequences, or to interfering signals (sent intentionally or unintentionally). In the case of interfering signals sent intentionally, Singh et al. [17] performed a theoretical analysis on the security of correlator-based secure UWB distance estimation, concluding that the security depends on two parameters: the first peak to average power ratio (FPAP) and the maximum peak to first peak ratio (MPFP). The FPAP defines the minimum strength of the first peak: setting it too low results in noise or artificially introduces peaks being detected as first path. Setting it too high results in a reduced sensitivity and in situations where the first path is attenuated by obstacles and may not be identified correctly. In another work, Leu et al. [13] proved that some UWB chips are indeed susceptible to this kind of attacks. As a consequence, additional vendor-specific integrity checks on the received signal must be performed to minimize the impact of artificially-introduced peaks (also called peak injection attacks).

3 *SUB-STS*: WORKING PRINCIPLE

The *sub-STS* concept allows several devices to securely estimate the ToA of multicast messages. Specifically, instead of creating the STS field of a multicast message from a single key/IV, several keys/IVs are used for different portions of the STS field. Fig. 4 illustrates the concept for three participating devices: one initiator A

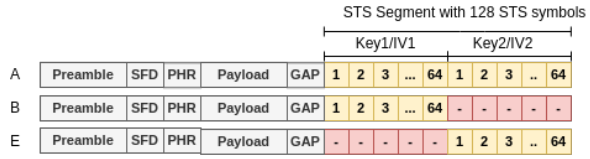

Figure 4: Working principle of *sub-STS*. Device A uses Key1/IV1 to generate the first 64 STS symbols and Key2/IV2 for the remaining ones. Receiver B uses Key1/IV1 and generates its CIR estimate from the first 64 STS symbols, ignoring the remaining ones (marked in red). Receiver E generates its CIR estimate from the second half of the STS sequence using Key2/IV2, ignoring the first half.

sending a multicast message to two receivers B and E that do not mutually trust each other. In this example, the devices make use of the SP2 frame configuration with only one STS segment (with 128 STS symbols), but the principle works in the same way for SP1 frames and multiple STS segments. Device A is in possession of both Key1/IV1 and Key2/IV2, and is able to create the first half of the STS segment (the first 64 STS symbols) using Key1/IV1 and the second half of the STS segment (the last 64 symbols) using Key2/IV2. The two receiving devices, B and E, configure their AES engine with their respective key/IV pair (i.e., device B uses Key1/IV1, whereas device E uses Key2/IV2), and instruct their receiver to create the CIR estimate from the first and the second half of the STS sequence, respectively. This way, assuming device E is evil, it cannot manipulate the ToA estimate of device B since it cannot create the STS symbols of the first half of the STS field, as Key1/IV1 != Key2/IV2. Note that, while this figure illustrates the *sub-STS* concept for only two receivers, one can scale it up to an arbitrary number of devices.

3.1 Benefits

The IEEE 802.15.4z standard foresees a preamble sequence that is $\approx 64\mu s$ long, an SFD that is $\approx 8\mu s$ long, a PHR that is $\approx 22\mu s$ long, as well as an STS field that is $\approx 64\mu s$ long in its default configuration. The total air-time per message hence amounts to at least $158\mu s$.

In case of SS-TWR, using the concept of *sub-STS* allows a secure estimation without the need of multiple *poll* messages at the cost of increasing the length of the STS field by $64\mu s$ per participating node. Hence, the amount of information sent over the air by the initiator decreases by the length of the preamble, SFD, and PHR (which amount to $\approx 94\mu s$ altogether) for each additional receiver. Fig. 5(a) shows the amount of air-time needed to complete the SS-TWR with a multicast *poll* message (Fig. 1(b)) and when using the default SS-TWR with multiple *poll* messages (Fig. 1(a)) as a function of the number of receiving devices. When having 8 receivers, the air-time of the initiator decreases by ≈26%.

In case of TDoA systems, the use of *sub-STS* allows a secure estimation without the need of multiple *resp* messages. Fig. 5(b) shows the amount of air-time needed to transmit all *resp* messages in a TDoA system when using individual responses (Fig. 1(c)) and a multicast message (Fig. 1(d)) as a function of the number of receivers. The air-time decreases by more than half when using 8 receivers.

3.2 Challenges

We discuss next the challenges to be tackled when implementing the *sub-STS* concept on currently-available UWB platforms.

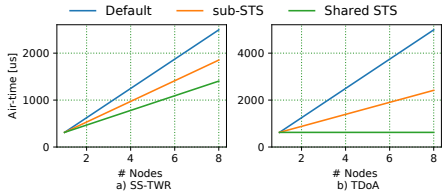

Figure 5: Air-time required by SS-TWR and TDoA schemes when sending individual secure frames, multicast frames with the *sub-STS*, and frames where all participants are sharing the same STS. When having 8 receivers, the use of multicast coupled with the *sub-STS* allows to decrease the air-time by ≈ 26% and 52% when using SS-TWR and TDoA, respectively.

(C1): Estimating the amount of received STS symbols. To verify that the *sub-STS* implementation works correctly, it is necessary to measure the number of correctly received STS symbols. While UWB receivers provide flags to indicate the integrity of received STS symbols, it is unclear to which extent these flags are suitable for estimating the number of correctly received STS symbols.

(C2): Runtime switching of Key/IV pairs. To the best of our knowledge, none of the UWB devices currently available on the market is officially capable of switching the key or IV at runtime and of selectively receiving only a portion of the STS. To implement the *sub-STS*, we hence need to exploit the radio's API to create a software-based implementation. However, since API calls are time-consuming, switching Key/IV pairs at runtime may cause delays up to several μs. Quantifying these delays and their effect on the STS and CIR quality hence needs to be evaluated.

(C3): Quantifying the impact on packet reception. The delays introduced by the runtime switching of Key/IV pairs in software and the STS portion received with a different Key/IV may cause a device to receive non-correlated STS symbols. When this happens, the receiver's ability to correctly decode the PHR and payload section is known to be limited on existing UWB devices [5]. Therefore, quantifying and mitigating this effect is necessary to meet the required performance of scalable UWB systems.

(C4): Quantifying the *sub-STS*-induced noise in the CIR estimate. Existing UWB radio estimate the CIR by correlating and merging all received STS symbols and cannot select only a portion of an STS segment. Consequently, the estimated CIR potentially contains artefacts due to non-correlated STS symbols. Therefore, it is interesting to study how non-correlated STS symbols manifest in the estimated CIR and how this impacts the ToA estimation process.

4 IMPLEMENTATION AND EVALUATION

We present a software-based implementation of the *sub-STS* using the Qorvo DW3000 IEEE 802.15.4z-compliant UWB radio (Sect. 4.1) and perform a preliminary evaluation of its performance (Sect. 4.2).

4.1 Implementation

We have observed that the DW3000 radio is able to re-configure its AES engine with new keys and/or IVs *while* transmitting messages, such that, after a key/IV is switched, any further STS symbols are generated from the new key/IV. This finding enables us to implement a prototype of a software-based implementation of the

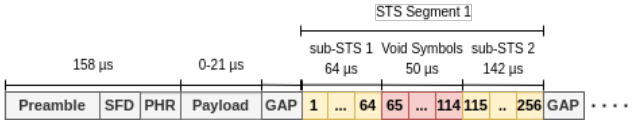

Figure 6: Key changing process. The *void STS symbols* (in red) belong neither to *sub-STS* 1 nor to *sub-STS* 2.

sub-STS concept. For this to work reliably, it is important that the MCU triggers the key or IV changing process in a timely manner. To this end, we use the DW3000's status flags, which indicate when the transmission of an UWB frame begins. Combining this timestamp with knowledge of the preamble's length, SFD, PHR, and data fields allows to infer the exact time at which the STS is sent.

Updating the key. The key register of the DW3000 is directly connected to the AES engine. Thus, changing any bits in the key register at runtime directly affects the output of the AES blocks. Due to the limited SPI speed, updating all 128 bits of the key when switching from *sub-STS* 1 to *sub-STS* 2 takes a couple of μs. During this period, the output of the AES block is generated from neither the first nor the second key, resulting in *void* STS symbols that cannot be decoded from any of the two receivers. For this reason, one needs to set a longer STS length, such that every receiver can get the full amount of expected STS symbols despite the introduction of *void* symbols. This is illustrated in Fig. 6, which shows a single STS segment of length 256 split into two *sub-STS*: the first one of length 64, and the second one of length 142. The slow key updating process introduces approximately 50 *void* STS symbols between the two *sub-STS* when using an SPI speed of 16 Mbit/s.

Updating the IV. According to the DW3000 user manual [5], updating the IV works in a two stage process: (i) one first pre-loads the IV into memory and then (ii) activates the new IV for the AES block by triggering an update *load IV* instruction. During our implementation, however, we experience a slightly different behaviour, as only the lower 32-bit (i.e., the counter part of the IV) seem to be pre-loadable. Updating the higher 96-bits of the IV yields a behaviour similar to the key update process (i.e., changing the first byte instantaneously affects the output of the AES block). For the sake of showcasing the benefits of pre-loading, we only update the pre-loadable lower 32-bits in our *sub-STS* implementation. However, updating the counter-part of the IV correctly is more challenging than updating the fixed part. This is because the counter values at the receiver and the pre-loaded counter values at the transmitter must match exactly when the *load IV* instruction is triggered. Therefore, precise timing of the *load IV* instruction and a correct prediction of the counter value are of utmost importance for this method to work reliably. To this end, we created a timing calibration application that exchanges multiple messages to estimate the best parameters to predict the counter value ahead in time.

4.2 Experimental Evaluation

To validate our *sub-STS* implementation and to evaluate its performance, we perform several experiments using three off-the-shelf DWM3000-DEV shields mounted on top of Nordic nRF52833 boards. Specifically, we study the *sub-STS* performance as a function of the length of the STS segment, the length of the *sub-STS*, the SPI communication speed, and the UWB frame configuration (focusing on SP1 and SP2 frames). For each of these variables, we collected 1000

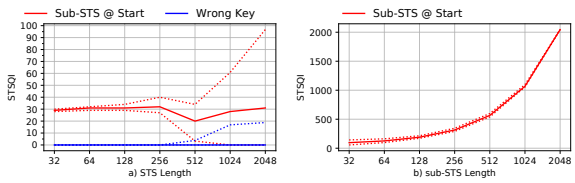

Figure 7: Suitability of the STSQI as an estimate of the amount of correctly-received STS symbols. STSQI value as a function of the STS length (a) and *sub-STS* length (b).

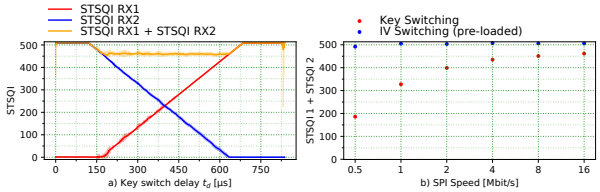

Figure 8: Runtime switching of the Key/IV pairs. STSQI value as a function of the key switching delay (a) and of the SPI speed (b).

measurements and derived four key performance indicators: (i) the STS quality indicator (STSQI), a unit-less indicator summarizing the quality of the receive STS symbols [4], (ii) the maximum peak (MAP) in the estimated CIR, (iii) the average noise (AVGN) in the estimated CIR, and (iv) the maximum noise level (MN).

4.2.1 (C1): Estimating the amount of received STS symbols. An estimate of the amount of correctly decoded STS symbols is essential for validating our *sub-STS* implementation and to measure the trustworthiness of the estimated CIR. Although it is not fully clear how the *STSQI* is derived, we believe that the *STSQI* counts the number of correlated STS symbols. In fact, we can observe a strong correlation between the STS length and the value returned by the STSQI under normal operations (i.e., when not making use of the *sub-STS*). Thus, the use of the STSQI as an indicator seems a favourable choice. Fig. 7(b) shows the STSQI value as a function of the *sub-STS* length for a fixed STS length of 2048. As expected, the median STSQI value (solid red line) correlates positively with the STS length of the *sub-STS*, indicating its suitability in identifying the number of correctly decoded STS symbols.

Fig. 7(a) shows the STSQI value as a function of the STS length when using a fixed *sub-STS* length of 32. The median value (solid red line) stays relatively constant at ≈30 regardless of the STS length, roughly matching the length of the *sub-STS*. However, we observed an increasing upper 97.5 and lower 2.5 percentile (dotted red lines) for STS lengths above 512. We further noted that, when using long STS sequences (e.g., an STS length of 2048), the STSQI occasionally returns a value above 0 even when using a wrong key throughout the entire STS segment. This is unexpected, as the STSQI value should be zero in such cases. Although the median value is indeed zero (solid blue line), we have measured values as high as 60, and an upper 97.5 percentile (dotted blue line) of ≈20. We hence conclude that the STSQI does reflect the amount of correctly-decoded STS symbols (and can be used as a metric to evaluate the correct functionality of the Key/IV switching), but there may be a few corner cases when using large STS lengths in which the returned value is unreliable. This aspect needs further investigation.

4.2.2 (C2): Runtime switching of Key/IV pairs. To evaluate if the Key/IV switching works correctly and how long it takes (i.e., how many *void* STS symbols are generated), we use three UWB devices: one transmitter and two receivers. The two receivers (RX) are configured with different keys: the first one (RX1) with *Key1* and the second one (RX2) with *Key2*. For each message transmission, the transmitter is first configured with *Key1*, and then switches to *Key2* roughly t_d μs after the beginning of the frame transmission has been signalled. Consequently, t_d influences the amount of correctly received STS symbols on both receivers. This is visible in Fig. 8(a), which depicts the STSQI value of RX1 (red) and RX2 (blue) as well as the sum of both values (orange) as a function of the key switch delay t_d. When selecting $t_d = 0$, *Key2* is set immediately after the beginning of the frame transmission and hence well before the start of the STS segment. Therefore, the STS sequence is generated solely from *Key2*, and the STSQI returned by RX2 is high (≈512). Instead, if $t_d > 675\mu s$, the key is only switched after the STS segment is transmitted, and hence the STSQI of RX1 is high. For any values in between, either RX1 or RX2 receives more symbols. Note that the exact value of t_d depends on the length of the fields prior to the STS segment. This experiment was performed with the SP2 frame configuration, a preamble length of 128, an SFD of 8, a $25\mu s$ long PHR/payload section, an STS length of 512 symbols and an SPI speed of 16 Mbit/s. Thus, the first STS symbol is sent $\approx 161\mu s$ after the message transmission flag is set: setting t_d to this value should result in a decrease of the observed STSQI of RX2. As shown in Fig. 8(a), however, the drop of the STSQI value of RX2 starts already at $\approx 120\mu s$ and not at the expected $161\mu s$. We attribute the $\approx 40\mu s$ of difference to the delay introduced by polling and evaluating the DW3000's system status register. Fig. 8(a) also shows that the STSQI of RX1 starts to raise $\approx 50\mu s$ after the drop of the STSQI of RX2. This is the amount of time the key updating procedures takes to write all key bytes into the DW3000's memory. The sum of both STSQI values is either ≈ 512 (when the key switching happens prior to or after the STS segment) or ≈ 462 (when the key switching is performed during the STS transmission). The difference of roughly 50 corresponds to the amount of *void* STS symbols, and is proportional to the SPI speed, i.e., to the amount of time it takes to transfer all 16 bytes of the 128-bit key into the DW3000's memory.

Fig. 8(b) shows the impact of different SPI speeds on the sum of the STSQI values of RX1 and RX2 (red dots) when switching key. For low SPI speeds, the amount of lost STS symbols can add up to more than 300, while at a speed of 16 Mbit/s, only 50 STS symbols are lost. Still, even at 16 Mbit/s, the amount of lost STS symbols is a limiting factor when implementing a software-based *sub-STS* scheme on the DW3000. To exploit the full potential of the *sub-STS* concept, more efficient key switching methods must be implemented, for example by allowing to pre-load keys. We demonstrate the potential of this by changing the 32-bit pre-loadable part of the IV instead of the key. Fig. 8(b) shows the sum of the STSQI of RX1 and RX2 (blue dots) when switching the IV: the value is close to 512, hinting that a fast re-configuration of the AES block via pre-loading is feasible.

4.2.3 (C3): Quantifying the impact on packet reception. Fig. 9 shows the packet error rate (i.e, the amount of packets where the payload section was not successfully decoded) as a function of the

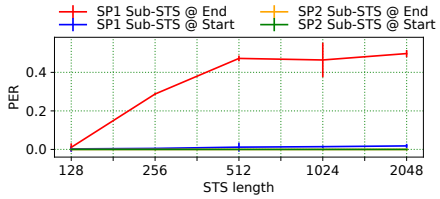

Figure 9: *sub-STS* induced PER. Fig. 9 shows the packet error rate for frame configuration SP1 and SP2 for different *sub-STS* positions. The yellow line is hidden by the green line.

STS length. We conducted this measurements for frame configuration SP1 and SP2 and for *sub-STS* at the beginning or end of the STS segment. When using the SP2 frame configuration, the PER is unaffected by the *sub-STS* concept. This was expected, as the STS segment is transmitted *after* the data portion. Instead, when using the SP1 frame configuration, the PER increases for long STS segments when the *sub-STS* is located at the very end of the segment. These results suggest preferring frame configuration SP2 over SP1 for implementing the *sub-STS* concept, and will be investigated in more depth in future work.

4.2.4 (C4): Quantifying the sub-STS-induced noise in the CIR estimate. As outlined in Sect. 2.2, the security of UWB-based distance estimates largely depends on the correct selection of the first path component in the CIR estimate. Any impact on the average noise level may result in a higher miss-detection of first path components and consequently requires a receiver to re-adjust its FPAP ratio. Therefore, we aimed to quantify the impact of the *sub-STS* concept on the CIR by studying the maximum peak (MP)[3], the average noise level (AVGN), and the maximum noise value (MN). Fig. 10(b) shows the three metrics as a function of the *sub-STS* length with a fixed STS length of 2048. The trends reveals that the AVGN stays the same regardless of the *sub-STS* length. However, the MP correlates positively with the length of the *sub-STS*. Fig. 10(a) shows the three metrics as a function of the STS length with a fixed *sub-STS* length of 32. The MP level stays similar regardless of the amount of STS symbols: this was expected as the *sub-STS* is fixed to 32. Things are different for the AVGN and MN, which consistently increase with the number of STS symbols. This indicates that the UWB receiver adds even non-correlating STS symbols to its final STS estimate. The trend is worst for an STS length of 2048, where the mean MN is already in the 90% interval of the MP level. In this extreme case, there is little room for setting a correct FPAP ratio: a slightly too low value leads to a high number of wrong first path detections. If the value is too high, the first path is not detected at all. Especially under non-optimal LOS conditions, where the first path component may be significantly weaker, it impacts the reliability of the system.

5 CONCLUSIONS AND FUTURE WORK

In this work we present a software-based implementation of the *sub-STS* concept on an off-the-shelf UWB transceiver. We show that the key and IV switching is already feasible; however, the impact on the estimated CIR is evident and its implication on the STS quality and security must be evaluated in more detail in future work. In the current implementation, the scalability of the system is primarily limited by the key switching time and noise introduced into the CIR estimate. Next generation UWB systems may incorporate these findings and provide means to efficiently switch keys/IVs and to selectively decode only a sub-set of received STS symbols. As a next step, we further aim to integrate the *sub-STS* concept into a TDoA-based localization system using multicast messages.

[3] All experiments were performed under LOS conditions, in which the first path coincided with the maximum peak.

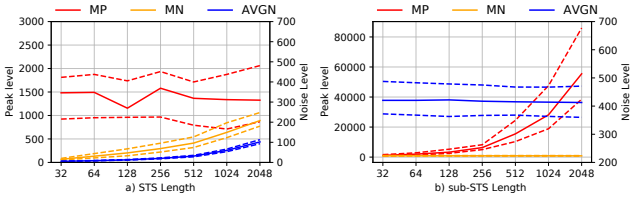

Figure 10: Maximum peak and noise level using the *sub-STS* concept. Fig. 10 (a) shows the maximum peak, the maximum noise (MN), and the average noise level (AVGN) for variable STS length and fixed sub-STS size and Fig. 10 (b) for fixed STS length but variable sub-STS.

Acknowledgements. This work was supported by the TU Graz LEAD project "Dependable Internet of Things in Adverse Environments". This work was also partially executed in the context of the SPiDR project: "Secure, Performant, Dependable, and Resilient Wireless Mesh Networks" (TII/SSRC/2120/2021).

REFERENCES

[1] P. Corbalán et al. 2019. Chorus: UWB Concurrent Transmissions for GPS-like Passive Localization of Countless Targets. In *Proc. of the 18th IPSN Conf.* ACM.
[2] P. Corbalán and G.P. Picco. 2020. Ultra-wideband Concurrent Ranging. *ACM TOSN* 16, 4 (2020).
[3] Decawave. 2014. APS013: DW1000 and Two-Way Ranging. [Online] https://tinyurl.com/2ejuhh8h – Last access: 2022-06-30.
[4] Decawave. 2020. DW3000 Datasheet, version 1.1. [Online] https://tinyurl.com/2ddkv44z – Last access: 2022-06-30.
[5] Decawave. 2020. DW3000 Usermanual, version 1.1. [Online] https://tinyurl.com/bdfd997f – Last access: 2022-07-25.
[6] EETimes. 2019. VW and NXP Show First Car Using UWB To Combat Relay Theft. [Online] https://tinyurl.com/201ybh88y5e – Last access: 2022-06-30.
[7] B. Großwindhager et al. 2018. Concurrent Ranging with UWB Radios: From Experimental Evidence to a Practical Solution. In *Proc. of the 38th ICDCS Conf.*
[8] B. Großwindhager et al. 2019. SnapLoc: An Ultra-Fast UWB-Based Indoor Localization System for an Unlimited Number of Tags. In *Proc. of the 18th IPSN Conf.*
[9] A. Gupta. 2018. *Development of UWB-IR based Low Power Asset Tracking System with Precise Location Information.* Master's thesis. NTU, Singapore.
[10] IEEE 802.15.4 Working Group. 2020. IEEE Standard for Low-Rate Wireless Networks – Part 802.15.4z-2020: Enhanced UWB PHYs and Ranging Techniques.
[11] Joerg Koepp and Nikola Serdar. 2021. UWB Reloaded: Test and Certification of UWB devices according to IEEE 802.15.4z. [Online] http://tinyurl.com/uf95kp57.
[12] A. Ledergerber et al. 2015. A Robot Self-Localization System using One-Way Ultra-Wideband Communication. In *Proc. of the IROS Conf.* IEEE / RSJ.
[13] P. Leu et al. 2021. Ghost Peak: Practical Distance Reduction Attacks Against HRP UWB Ranging. In *Proc. of the IROS Conf.* IEEE / RSJ.
[14] Ayman Naguib et al. 2020. Secure Multicast / Broadcast Ranging. European Patent Application EP3-651-410-A1.
[15] P. Sedlacek et al. 2019. An Overview of the IEEE 802.15.4z Standard and its Comparison to Existing UWB Standards. In *Proc. of the 29th RadioElektronika Conf.*
[16] M. Singh et al. 2019. UWB-ED: Distance Enlargement Attack Detection in Ultra-Wideband. In *Proc. of the 28th USENIX Security Symp.*
[17] M. Singh et al. 2021. Security Analysis of IEEE 802.15.4z/HRP UWB Time-of-Flight Distance Measurement. In *Proc. of the 14th WiSec Conf.*
[18] J. Tiemann et al. 2016. Atlas: An Open-Source TDOA-based Ultra-Wideband Localization System. In *Proceedings of the IPIN Conference.*
[19] Wired. 2019. The Biggest iPhone News Is a Tiny New Chip Inside It. [Online] https://www.wired.com/story/apple-u1-chip/ – Last access: 2022-06-30.
[20] K. Witrisal et al. 2016. High-Accuracy Localization for Assisted Living: 5G systems will Turn Multipath Channels from Foe to Friend. *IEEE Signal Processing Magazine* 33, 2 (2016).