UWB Rapid-Bit-Exchange System for Distance Bounding

Nils Ole Tippenhauer
Singapore University of
Technology and Design
Singapore 487372
nils_tippenhauer@
sutd.edu.sg

Heinrich Luecken
P3 group
52070 Aachen, Germany
heinrich.luecken@
p3-group.com

Marc Kuhn
Wireless Communications
Group, ETH Zurich
8092 Zurich, Switzerland
kuhn@nari.ee.ethz.ch

Srdjan Capkun
Institute of Information
Security, ETH Zurich
8092 Zurich, Switzerland
capkuns@inf.ethz.ch

ABSTRACT

Distance bounding protocols enable one device (the verifier) to securely establish an upper bound on its distance to another device (the prover). These protocols can be used for secure location verification and detection of relay attacks, even in presence of strong attackers. The *rapid-bit-exchange* is the core of distance bounding protocols—the verifier sends single bit challenges, which the prover is expected to answer with minimal and stable processing delay. Based on the measured round trip time of flight, the verifier calculates its upper bound to the prover. Although several aspects of distance bounding implementations have been discussed in the past, no full implementation of a wireless distance bounding system has been presented so far.

In this work, we present the first full realization of a rapid bit exchange system for distance bounding. Our system consists of an Ultra-Wideband (UWB) ranging radio and of an efficient digital processing implemented on an Field-Programmable-Gate-Array (FPGA) board; it achieves a ranging accuracy of 7.5 cm and a short processing delay at the prover (< 100 ns). This minimal processing delay is the lowest reported so far for provers that demodulate the challenge before responding.

1. INTRODUCTION

In the recent years, physical-layer attacks on wireless communications have received increased attention. Examples for this are relay attacks, which have been demonstrated on current-generation automatic car locks [15]. In such relay attacks, the attacker relays signals to extend the communication range between a user and his car. As a result, the car key will unlock the car from a greater distance, without the knowledge of the user. Relay attacks could be prevented by the use of distance bounding protocols between the communicating parties. Distance bounding protocols allow the users to verify that their mutual distance is below some threshold (the distance bound). Many distance bounding protocols have been proposed, among them [3,5–7,18,20–22,25,34–36,42]. So far, the main focus of research on distance bounding protocols was on models for attackers and formal security properties [2,20,22,25]. Some works have shown that, in addition to the protocol analysis, attention needs to be dedicated to the physical-layer implementation of distance bounding protocols [9,28,29,31]. In only few works, authors discussed possible implementations of distance bounding protocols [17,18,30,32,33]. However, none of the proposals fully implement distance bounding and integrate it with a functioning ranging system. In this work, we present the design and implementation of a UWB-based distance bounding system that enables accurate ranging and secure distance bounding, even if the prover is untrusted. We discuss physical layer aspects of distance bounding implementations such as the modulation scheme, and the contributing factors for processing delay at the prover. In addition, we discuss security issues related to the preamble, and the round-trip-time (RTT) measurements.

At the core of our system is a rapid-bit-exchange (RBE) phase that provides the actual distance estimate in a secured way. Fundamentally, the RBE relies on RTT measurements taken for a challenge and response message—to minimize *distance fraud attacks*, the response must be generated with minimal processing delay. The complexity of this operation is often underestimated when only considering the protocol on an abstract layer. In particular, the resulting system needs high precision distance measurements, and very low processing delay.

In this work, we present the following contributions:

- We design a system for UWB-based distance bounding. This system can be used to implement a broad range of distance bounding protocols, including those that compute the reply based on the challenge and a shared key.
- We implement this system and show that it provides both sufficient ranging accuracy and tight security guarantees: even a strong attacker cannot reduce the distance result more than 15.6 m.

WiSec'15, June 22 - 26, 2015, New York, NY, USA
Copyright 2015 ACM ISBN 978-1-4503-3623-9/15/06 ...$15.00.
DOI: http://dx.doi.org/10.1145/2766498.2766504.

- We introduce the notion of a distance commitment and show that measuring the RTT based on interleaved preambles is secure, even if the preambles themselves are static.

This work is structured as following: in Section 2, we briefly introduce the core concepts of distance bounding protocols. We review related implementations in Section 3. In Section 4, we start by discussing security aspects of possible UWB modulation schemes and selecting a modulation for our system. In Section 5, we propose and explain our design in detail. We present the implementation of our system in Section 6, and give experimental results in Section 7. A discussion on *distance commitments* follows in Section 8. Section 9 concludes this work.

2. BACKGROUND ON DISTANCE BOUNDING

Distance bounding protocols aim to securely provide an upper bound on the distance between a verifier and a prover, even if the prover is malicious and misbehaving. In addition to preventing misbehavior of the prover, the protocols protect the measurement against an active external attacker who is able to jam arbitrary messages, eavesdrop on exchanged messages, and insert own messages. In order to achieve strong security results, strong attacker models are often used in the context of distance bounding. Thus, both the external attacker and a malicious prover are assumed to have "ideal" hardware, which allows him to reduce his processing delays to the minimal feasible.

In this setting, conventional systems to measure distance based on the time-of-flight of radio signals are not sufficient. As an example, a malicious communication partner cannot be trusted to be honest about the transmission time of an exchanged message—for example, the malicious prover could send messages earlier than claimed, effectively shortening the distance measurement result. An external attacker can also impersonate replies of a prover if these are not properly authenticated and thus shorten the measured distance (such attacks have been demonstrated on systems like GPS [40]). In particular, ranging systems based on 802.15.4a are vulnerable to a range of attacks as discussed in [30].

These and other possible attacks on distance measurement systems have been formalized as the following models in the context of distance bounding (see also [10]):

- *Mafia Fraud* attacks: the external attacker tries to shorten the distance between the verifier and honest prover [11].
- *Distance Fraud* attacks: the malicious prover tries to shorten the distance measured by the verifier [35].
- *Terrorist Fraud* attacks: the malicious prover cooperates with an external attacker to shorten the measured distance, as long as this does not compromise his private secret [11].
- *Distance Hijacking* attacks: the external attacker abuses an honest prover to shorten the distance measured between verifier and attacker [10].

2.1 Distance Bounding Protocols

As a consequence of the diverse attacker models, a wide number of distance bounding protocols have been proposed. In this work, we focus on common implementation issues and not the protocol layer, and we will only discuss the common structure of the protocols. Fundamentally, each protocol needs at least two messages: an initial challenge and a response message. For security reasons, the challenge typically contains an unpredictable challenge value c and the response message contains a value r, which is computed dynamically based on c and a shared secret. The time between sending the challenge message and the reception of the reply message determines the RTT between the verifier A (sending the challenge) and the prover B (sending the reply). While distance bounding protocols generally differ in the way the challenges are generated and how the prover computes appropriate replies in response to the challenges, all protocols use this basic concept of sending a challenge and response message pair. As such, this message exchange is the core of any such protocol, and needs special attention in security analysis and implementation.

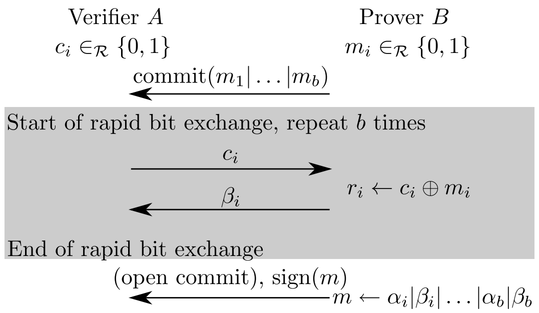

Verifier A
Prover B
$c_i \in_{\mathcal{R}} \{0,1\}$
$m_i \in_{\mathcal{R}} \{0,1\}$
commit$(m_1|\ldots|m_b)$
Start of rapid bit exchange, repeat b times
c_i
β_i
$r_i \leftarrow c_i \oplus m_i$
End of rapid bit exchange
(open commit), sign(m)
$m \leftarrow \alpha_i|\beta_i|\ldots|\alpha_b|\beta_b$

Figure 1: The distance bounding protocol of Brands and Chaum with its distinct rapid-bit-exchange phase.

Figure 1 shows the original (untrusting) distance bounding protocol from Brands and Chaum [3]. B starts the protocol by sending a cryptographic commitment to a nonce m of B's choice. A then sends a challenge c bitwise for b rounds, bit c_i at time t^A_{si}. Upon reception of c_i at time t^B_{ri}, B computes $r_i = c_i \oplus m_i$ and sends this as reply at time t^B_{si}. Finally, the replies arrive at A at time t^A_{ri} each. Following the rapid-bit-exchange phase, B opens the commitment to A. A can now validate the received replies and compute the upper bound distance $d^{\max} = \max_i((t^A_{ri} - t^A_{si})/2 \cdot v)$ (with signal propagation speed v). This distance estimate assumes that the processing time of B is constant and negligible ($(t^B_{si} - t^B_{ri}) \cdot v < \epsilon$), as this will increase the measured distance. If a non-negligible constant processing delay at B is assumed by A (and compensated in the distance measurement), a malicious prover can gain a *distance advantage* by reducing this assumed processing delay. Given the high propagation speed of wireless signals, processing delays in the range of microseconds can provide a distance advantage of several hundred meters for strong attackers (15 cm/ns).

3. DISTANCE BOUNDING IMPLEMENTATIONS

Practical implementations of distance bounding protocols have been discussed in three application areas: wired systems, near field wireless communication, and ultra-wideband (UWB) wireless communications.

3.1 Wired Distance Bounding

The first implementation of distance bounding over a *wired* channel between FPGAs was presented in [12]. As wired connection allow for a high signal bandwidth and several parallel channels, the system could be implemented with minimal effort. A 200 MHz external clock is used to input the challenge and read out the reply from the prover FPGA, allowing an accuracy of 5 ns (0.75 m), which is also the limit for mafia fraud attacks. The processing of the challenge takes 8 ns in the presented system (using the external 200 MHz clock provided by the verifier). A malicious prover with zero processing delay could at most be 13 ns or 1.95 m away to respond in time (distance fraud). The implementation was tested for cable lengths of 0.3 m, 1.0 m, and 2.0 m.

3.2 RFID Distance Bounding

Implementation ideas for wireless distance bounding for RFID tags with very short range *(near field communication)* and low accuracy appeared in [18], [26], and [34]. Their low accuracy is mainly due to the very low bandwidth of common RFID communication standards. In [34], the authors propose to use effects similar to side-channel leakage to communicate the reply to the verifier, as this out-of-band channel would not be restricted by bandwidth regulations.

In [17], the author extends the initial concept for distance bounding for RFID chips from [18]. As the author considers RFID systems as provers, his transceiver proposal has to rely on out-of-band reply signals generated by a custom logic. He experimentally shows that this logic can generate appropriate responses (near field), but does not present a solution to measure the RTT (the prover is either within 1 m distance, or not). The scheme limits mafia fraud attacks to 1 m, and distance fraud attacks to 11 m [24]. Due to the limited range of RFID tags, the maximum distance bounding verification range is around 30 cm.

3.3 UWB Distance Bounding

In [39], a commercial *UWB* ranging platform without authentication support is used to construct a distance bounding system. To authenticate the exchanged messages, custom MAC-layer identifiers are used as replacement for challenges and replies. The resulting system has an RTT measurement precision of 1 ns (15 cm) and can protect against a limited external attacker who is not able to mount an early-detect / late-commit attack [9]. Within the standard attacker model for distance bounding, a distance fraud can have an advantage of up to 56 μs, which translates to 7.5 km.

In [30], the authors discuss an adaption of the UWB distance measurement specification of the IEEE 802.15.4a standard with communication range 20-30 m. As 802.15.4a uses BPPM/BPSK data symbols, early-detection/ late-commit attacks are possible. The authors discuss the impact of such attacks with respect to rake and energy detection receivers. They also propose a set of countermeasures to improve the security of 802.15.4a for distance bounding. In this paper, we our design intentionally does not use BPPM or BPSK symbols. In [30], the authors also discuss full-duplex transmission of challenges and replies as proposed in [33]. They conclude that such a scheme would limit the effectiveness of mafia fraud and distance fraud attacks to 10 m. We note that in [33] itself, only a prover design to minimize the delay is discussed, but no details on range measurement or physical-layer protocol are provided.

In [29], the authors show that the synchronization between sender and receiver can itself be manipulated by an attacker. As part of the synchronization, in ranging systems the receiver typically selects the first (instead of the strongest) path in a multipath environment. The authors show that this can enable an attacker to trick the receiver to select a manipulated, shorter, path. That allows an attacker to shorten the ranging result between two honest nodes. In particular, the attacks are possible due to the large integration window (128 ns) of the receiver. The authors propose several countermeasures to mitigate the impact of such attacks, among them a suggestion to use a smaller integration window (referred to as EDD). In our implementation, we use an integration window of 4 ns for line-of-sight channels.

We proposed a first design of a custom UWB distance bounding scheme in [23]. Our work in this paper builds upon the concept from [23] by extending its protocol design and presenting an implementation. Based on the initial design as one integrated circuit, we estimated B's minimal processing time $\delta^B = t^B_{si} - t^B_{ri}$ for such an optimized implementation to at most 4 ns, which translates to a maximal distance advantage for the attacker of 3.6 m in the case of distance fraud. Against mafia fraud, the system is estimated to be accurate within 1.5 m. The system is designed for ranges up to 10 m.

3.4 Conclusion on existing implementations

Compared to prior work, our proposed solution in this work is fully integrated with a UWB ranging platform, can enable distance bounding in applications for ranges greater than near-field-communication, supports the widest range of processing functions and protocols (e.g., derivatives of Brands-Chaum and Hancke-Kuhn constructs), and is designed to minimize the advantage of a malicious prover.

4. UWB IMPULSE RADIO BASED RAPID-BIT-EXCHANGE

In this section, we will discuss a transceiver design for a UWB rapid-bit-exchange phase. In particular, we will study which modulation scheme is best suited to implement rapid-bit-exchange given a fixed bandwidth. We will start by arguing why UWB is well suited for distance bounding applications. Most important is the need for (i) precise distance measurements and (ii) minimal processing delay δ at the transceivers during the rapid-bit-exchange phase.

The high bandwidth of UWB enables *precise distance measurements* based on time-of-arrival measurements with resolution in the order of nanoseconds [16], which is the basis for our DB system. The reason for the high accuracy is that multipath components are separable at the receiver – which allows for precise distance measurements, even in case of multipath propagation. Narrowband localization systems, such as GPS or Wi-Fi-based positioning, fail to achieve accurate measurements in environments with multipath since they cannot distinguish the line-of-sight (LOS) signal component from reflections or scattered signal components. Moreover, UWB technology offers a huge design space with many trade-offs–from high rate transceivers with coherent receiver structures (for instance UWB based wireless USB) to UWB impulse radio transceivers with low complexity due to simple transmitters and noncoherent receivers.

The *minimal Processing time* $\delta^B = t^B_{si} - t^B_{ri}$ is the second crucial aspect for distance bounding systems, i.e. the time between the challenge bit is received, the response bit is computed, and then sent. The delay itself is not a problem regarding accurate distance measurements: As long as it is known, distances can still be computed with high precision. However, a malicious prover may exploit high expected processing delays to shorten measured distance by replying early, for instance using a sophisticated receiver structure with lower receive processing delay. Hence, it is advantageous to detect UWB pulses and transmit answers as fast as possible. The use of UWB impulse radio enables the implementation of low complexity (short delay) and low power transceivers. In particular, noncoherent receivers can be implemented very efficiently [44].

4.1 Security Aspects of Wireless Rapid-Bit-Exchange

In the following, we discuss digital modulation schemes suited for the rapid-bit-exchange phase. In particular, we use f_c to denote the carrier frequency, T_{pul} to denote the length of UWB pulses, T_s for the symbol period, and T_{int} as receiver integration window length (see Figure 2).

The security implications of symbol modulation in RTT-based secure ranging protocols have been discussed in [9]. The authors show that attacker can also gain a distance advantage by misbehaving in the (de)modulation—to mitigate this, they suggest the following principle:

> "Minimize the length of the symbol used to represent this single bit. In other words, output the energy that distinguishes the two possible transmitted bit values within as short a time as is feasible. This leaves the attacker little room to shorten this time interval further."

This suggestion is based on several attacks which can give the attacker a distance advantage up to the length of the symbol, as discussed in [9]. In early-bit-detection attacks, the attacker uses ideal receiving equipment to demodulate the symbol after receiving only a small fraction, thus yielding the data earlier then expected. In a deferred-bit-signaling attack, the attacker will send symbols such that he only has to commit to the data value of this symbol after a certain fraction of the symbol length. As a result, the attacker can reduce the RTT by up to T_{pul}.

This principle of [9] is another reason why the use of UWB communication is advantageous for ranging protocols from a security perspective. Compared to conventional narrowband systems, the large bandwidth of UWB allows very short symbol timings in the order of nanoseconds, which "minimize the length of the symbol" representing a single bit. The use of UWB impulse radio is chosen here because the rapid-bit-exchange can be implemented by an exchange of single UWB transmit pulses that have duration of few nanoseconds.

As conclusion of the discussion above, a single UWB pulse should be used to represent a single bit. This is the reason why many conventional UWB systems are not suited for distance bounding. In standard systems, a symbol often consists of more than one UWB pulse. For instance, in IEEE 802.15.4a, several pulses per bit are transmitted to achieve a sufficient energy per bit at the receiver. This increases the processing time of the receiver for decoding one bit, making the system more sensitive to the mentioned attacks. The same holds for UWB based Wireless USB, where a single OFDM symbol takes more than 300 ns [13]. For the same reason, we omit channel coding during the rapid-bit-exchange phase, as it spreads the information over multiple symbols and introduces coding delay. Without channel coding, the bit error probability of the system needs to be considered as well, as decoding errors cannot be compensated based on error correcting channel codes. This problem has been discussed in [23, 27, 36].

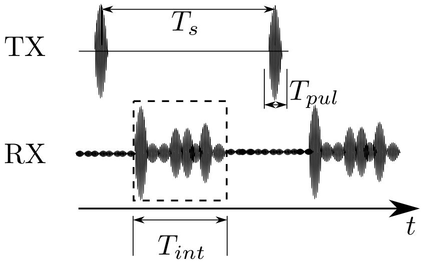


Figure 2: Overview UWB symbols transmitted (TX) and received (RX). In standard UWB receivers, the integration window is long enough to collect most multipath channel responses (T_{int}). The sender transmits only short pulses of length T_{pul}, with symbol perdiod T_s.

While the transmission of only one short UWB pulse per bit helps to reduce the processing delay, the main contributor to δ is the demodulation process delay. The dispersive wireless channel attenuates the pulse strongly and the receive energy is split between many multipath components. An optimal receiver uses all multipath components of the symbol in the decoding process. Therefore, the required receiver processing time is typically chosen such that it covers the whole symbol and the expected delay spread of the channel (see Figure 2). Depending on the environment, this delay spread can reach up to 60 ns or more [43], while the UWB pulse itself can be less than a nanosecond long.

As the receiver will collect all signal energy during this period, the attacker's maximum gain with a late bit commit or early-detection attack is limited by the observation window length T_{int}, and not the symbol length T_{pul}. Shortening this time window in the decoding process mitigates these attacks but reduces the collected symbol energy and, hence, the decoding performance. We discuss performance trade-offs related to the decoding window length in Section 4.2.

Given these prerequisites, the state-of-the-art does not offer UWB transceivers that are suitable for secure ranging with high accuracy. Therefore, we present a new framework for the development of UWB rapid-bit-exchange systems. In particular, we consider short range systems (such as WBAN, WPAN) and focus on minimum processing delay as well as on low complexity, low power and low cost designs.

Proposed receiver structure: The minimum processing delay requirement constrains the receiver processing to structures that can be implemented in an analog fashion. A receiver with analog processing is faster than digital signal processing, as it is used for conventional decoders based on OFDM or channel matched filtering. The analog processing limits the design space of the receiver since not all operations can be implemented. Due to the high bandwidth it is difficult to realize arbitrary filters that can be adapted to the channel conditions. The analog implementation of RAKE receivers requires high complexity due to the high number of fingers [8]. With the minimum processing delay requirement, coherent receivers are not the method of choice to achieve low complexity and low costs. Therefore, we propose to use non-coherent UWB communication [44] for the RBE. The energy detection receiver is well known for its efficient analog implementation, while still achieving sufficient performance for short range systems. It processes the bandpass filtered received signal $\tilde{r}_B(\cdot)$ as follows:

$$y(t) = \int_{t-T_{\text{int}}}^{t} \tilde{r}_B^2(\tau)d\tau.$$

Typically, the energy detection receiver is used in conjunction with binary pulse position modulation (BPPM). Here, the symbol is split into two time slots of length T_{slot} each. A pulse is transmitted in the first time slot for a Zero and in the second for One. The receiver estimates the energy that is received in both time slots and compares their values as follows:

$$y(iT_s) - y(iT_s + T_{\text{slot}}) \underset{\hat{c}_i=1}{\overset{\hat{c}_i=0}{\gtrless}} 0.$$

In [41], the authors presented an energy detection based ultra-low power UWB system design with a very low overall current consumption. The theoretical feasibility of the presented design respecting FCC power limits [14] together with transmission of only one pulse per bit has been shown by means of computer simulation and over the air [37]. As we will show, the security of our distance bounding solution relies on the ability to transmit a bit with only one pulse.

The security aspects of secure ranging protocols can now be discussed for the specific implementation with energy detection receivers. We agree with the general intuition of the principle in [9], but would reformulate it in the following way:

> “Minimize the length of the symbol used to present a single bit as well as the processing time the transceiver needs for demodulating a single bit challenge and sending the response bit. In other words, distinguish the two possible transmitted bit values at the receiver within as short a time as is feasible and send an answer as fast as possible. This leaves the attacker little room to shorten this time interval further.”

In [28, 29], the authors discuss attacks on the preamble synchronization in 802.15.4a. Essentially, these attacks allow the attacker to manipulate the starting time of the sampling window for the data pulses relative to the original position of the pulses. This can be achieved by manipulating the leading-edge detection schemes often used in receivers. If the sampling window is advanced to cover only the main pulse (and not the multipath components), the measured RTT is shortened without requiring the attacker to manipulate data symbols themselves. As with early-detection or late-commit attacks, the maximal gain for the attacker here is limited by the length of the integration window T_{int} at the receiver. In fact, such preamble manipulation attacks allow the attacker to influence the receiver to involuntarily perform an early-detection reception himself.

T_{int} presents an upper bound on the possible advantage for the attacker, even if late-commit and synchronization manipulation attacks are combined.

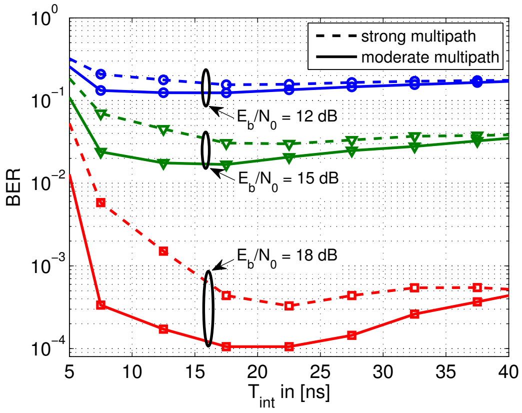


Figure 3: Bit-error-rate (BER) of energy detector vs. Integration window length T_{int} for different E_b/N_0, where E_b denotes the energy per bit and $N_0/2$ the noise power spectral density. Channels measured in indoor environment (BPPM), frequency range 3-6 GHz, see [1] for details.

4.2 Low Complexity, Minimal Delay UWB Transceiver

A minimal delay UWB transceiver is well suited to fulfill the requirements we summarized. In particular, a noncoherent energy detection receiver is able to combine very precise distance measurements (even in multipath) with low processing delay as well as low complexity and low power consumption. In the following, we will discuss two remaining important parameters of this transceiver design: (i) the minimization of the integration window of the energy detector, and (ii) the choice of the modulation scheme.

Performance trade-off related to the integration window length:: Shortening the sampling interval at the receiver (e.g., by using a smaller integration window of the energy detector) mitigates the effects of late-bit-commit or early-detection attacks. However, it also reduces the decoding performance as less signal energy can be collected. This effect can be seen in Figure 3. Increasing the observation window length leads to a decreasing bit error probability. Choosing the integration window too long increases the bit error probability again, as no additional signal energy is recovered (only noise is added). T_{int} also depends on the target channel—we are considering line-of-sight applications in this work.

Modulation scheme:: PPM shows vulnerabilities regarding early-detection attacks due to its symbol structure. To mitigate this problem, a modulation scheme called Security Enhanced Modulation (SEM) was proposed in [23]. This scheme is based on binary PPM and mitigates the effect of such attacks. The price for the increased protection is a reduced receiver SNR leading to a higher bit error probability. Similar to binary PPM, SEM uses two symbol slots, but only encodes the data content in the second slot. The first data slot is used to improve the demodulation accuracy. To demodulate such a symbol, the receiver compares the integrated energy of the second slot with the integrated energy of the first slot. If more energy is collected in the second slot, the symbol gets demodulated as One symbol, otherwise as Zero, so no fixed threshold is necessary. The polarity of both

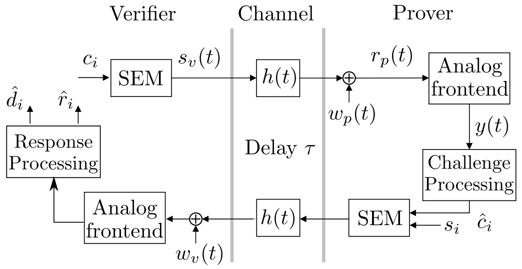


Figure 4: Abstract model of the rapid-bit-exchange system: Verifier, channel, prover, and involved information flow.

pulses, $a_i = \pm 1$, is chosen randomly to avoid discrete spectral lines in the spectrum of the transmit signal. Thus, the data part of the challenge and reply signal is given by

$$s_A(t) = \sum_{i=1}^{N} a_i b_0 p(t - iT_s) + \sum_{i=1}^{N} c_i a_i b_0 p(t - T_{\text{slot}} - iT_s),$$

with the normalized pulse shape $p(t)$, a transmission power scaling b_0, b_1 for the first and second pulse (e.g., $b_1 = \sqrt{2}b_0$), and N bits of challenge transmitted in total.

As our choice of a low complexity, minimal delay UWB receiver is based on a noncoherent energy detector, we decided to use SEM in our system design. In combination with other receiver structures, particularly coherent structures, many more modulation schemes are an option, as for example PSK, FSK, OOK etc. A more detailed analysis on optimal design parameters for the modulation (for a comparable system) can be found in [23]. In that work, we also discuss trade-offs between the achievable bit-error-rate vs. distance, and effects on false-reject/false-acceptance.

5. A MINIMAL DELAY TRANSCEIVER DESIGN

We now present the system design to implement the rapid-bit-exchange transceiver on an FPGA platform. Further details on the system can be found in [38]. While we use an analog front-end to down- and up-convert the signals to the carrier frequencies, we shifted most of the processing into the FPGA, e.g., the squaring (also possible in the analog front-end). Our basic system model is as follows (see Figure 4). The prover and verifier both use a transmitter which outputs symbols modulated using the SEM scheme. The modulated signals then pass over the wireless channel $h(t)$, get down-converted to an intermediate frequency, and are demodulated by an energy detector. The demodulated symbols are then processed digitally, and appropriate replies are sent if necessary.

In our system design, both prover and verifier use a non-coherent receiver. This allows us to reuse the receiver design from the prover for the verifier, which greatly simplified the implementation. In particular, both verifier and receiver shared the same design and most components.

In Figure 5, we show more details on the components in the prover design. In particular, we show the split between the analog front-end and digital demodulation in our design.

5.1 Component Description

We will now give an abstract overview of the design for each component to give detailed insight into the sources of processing delay. In addition, we will also present some components only required in the verifier, such as the RTT measurement setup. The final design is outlined in Figure 6 (the mapping to FPGAs is explained later).

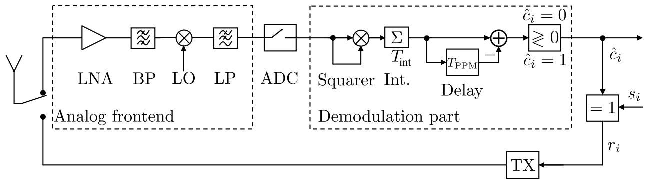


Figure 5: Prover design: On the left, the receiver front-end components. The right part is performing the actual demodulation on the FPGA. The transmit (TX) and synchronization path is not shown in detail.

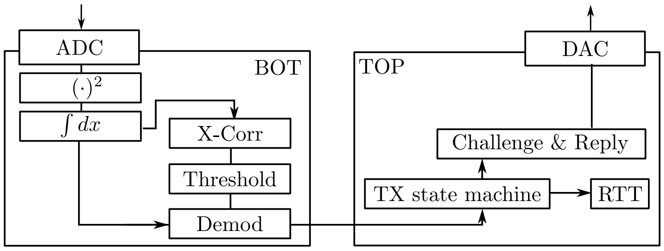


Figure 6: Mapping of the design for the FPGA platform. The design is split into two parts to fit two FPGAs (called TOP and BOT).

We assume that the ADC delivers samples with p_{ADC} bit precision, and the ADC clock is faster than the system clock by factor s_{ADC}. In our implementation, which we will present later, $s_{\text{ADC}} = 16$.

The **squaring** unit needs to compute s_{ADC} squarings of signed p_{ADC} bit numbers. As each squaring is independent of the others, we can simply design s_{ADC} parallel squaring units. Although squaring itself is a complex operation involving many gates, FPGAs usually provide special hardware units which can be integrated into the design, and which are able to compute multiplications within one clock cycle (up to a certain bit width).

The **integration** unit needs to integrate T consecutive samples. The result will be used for the symbol synchronization and demodulation. Before successful synchronization, we do not know the offset for the sampling window within the system clock period. Thus, the integration unit has to provide integration results for all possible s_{ADC} offsets. An integration over T samples requires at least T additions, to minimize the logic depth, our design uses a multi-stage approach. Each stage takes one clock cycle, but can have many parallel additions. This design allows to integrate all s_{ADC} possible offsets in parallel with minimal delay.

The **cross-correlation** unit is needed for the initial channel synchronization based on the preamble. The cross-correlation unit works similar to the integration unit. It integrates all received energy for the preamble, while applying negative or positive signs to the individual components based on the preamble's code. As subtractions are as cheap as additions, the overall effort for the preamble is as high as the integration, if the preamble has the same number of pulses since the integration window samples. As with the integration unit,

s_{ADC} parallel cross-correlation units are required because the phase offset of the incoming signal is unknown prior to synchronization. The output of the cross-correlation unit is directly connected to the threshold detection unit.

The **threshold detection** unit is also needed for the synchronization process. It performs the final decision if a correct preamble was encountered on the channel, and with which temporal offset. The threshold detection unit gets the s_{ADC} results of the cross-correlation unit per cycle, and has to find the maximum value with minimal processing delay. This maximal value is then compared to the threshold. In our design, we use a multi-stage comparison between the s_{ADC} different values, resulting in a total delay of $log_2(s_{\mathrm{ADC}})$ cycles. The threshold can be configured remotely and depends on channel conditions.

The **transmission state machine** unit is used to create the symbols for the preamble and data content of both challenges and replies. The transmission power can also be configured remotely, to switch between distance bounding over cable or wireless, for example. To adapt to the variable phase offset of the incoming signal, the transmitted pulses can also be generated with a variable phase shift. The signal generation process also uses the output of a pseudo-random number generator (an LFSR with length 32) to randomize the sign of the transmitted pulses (which does not influence the symbol data) to improve the spectral properties.

The **RTT measurement** unit is used by the verifier to measure the time between the transmission of the challenge, and the reception of the reply. The RTT measurement unit is synchronized with the TX state machine and is built around a counter which is activated when the challenge preamble is sent, and stopped when the reply preamble is received. If no reply is received with a certain timeout, the measurement is aborted. The RTT measurement unit also records the reply bits, and compares them with the expected reply. The resulting error vector and the RTT can be queried using the Ethernet interface and a client program.

The **configuration** unit allows a user to set various system parameters externally via Ethernet. For example, parts of the system can be reset and a debugging mode allows to output internal bus traffic to the DAC for real-time inspection. It is also possible to adapt the transmission power, reception threshold, pulse spacing, and others parameters.

5.2 Processing Delay Minimization

There are many factors contributing to the processing delay of the proposed transceiver. Every processing component in the critical path in Figure 5 adds its own delay, there is also communication delay due to bus connections, interface adapters such as serializers and deserializers, and finally logic to ensure correct data handover when crossing clock domains. All these components need to be optimized to achieve the minimal possible delay. All the above delays can be measured in discrete units of clock cycles, and therefore the clock frequency of the design also plays a major role to compute the final processing delay in nanoseconds.

To minimize processing delay of data with high rate, FPGA-based implementations can extensively rely on parallelization, if this is supported by the design. For example, our design will require a squaring unit for the ADC samples. If our system clock would be the same as the ADC sampling rate, we could build a (multi-stage) pipeline to square each sample individually.

In our final design the main logic frequency is 125 MHz which requires parallelization of 16 squarings.

6. IMPLEMENTATION

We now present our implementation of the proposed system design. In particular, the implementation is intended to demonstrate that (a) accurate distance measurements are possible with the proposed system, (b) the theoretically predicted processing delay at the prover can be achieved in the implementation, and (c) our design (that was optimized under security aspects) has acceptably low error rates (in our limited trials). We note that our implementation is a prototype, and for future commercial applications platforms such as ASICs should be chosen instead of FPGAs to reduce the system size and cost. We also note that, due to limitations in our testbed, we were restricted to measuring distances up to 2.5 m. We designed our system for line-of-sight channels up 10 m distance, and keep within the regulatory maximum peak power and average power constraints. T_{int} is chosen as 4 ns, using pulsewidth $T_{\mathrm{pul}} = 2$ ns and period of $T_s = 512$ ns.

Given the constraints we detailed earlier (ADC/DAC with 2 GS/s connected to an FPGA), we used two Triton VXS-5 systems by Tekmicro with 3 Virtex-5 FPGAs. Out of these 3 FPGAs, one is connected to the ADC (called bottom/BOT FPGA by the manufacturer), one is connected to the DAC (the TOP FPGA), and the third can be used to control the others. The DAC and ADC are connected to the respective FPGA using a 250 MHz DDR parallel bus (with 4 samples per cycle). We used VHDL as hardware description language and the Xilinx ISE [19] to code, synthesize and simulate the design. The Triton boards implement a prover/verifier pair.

Because the Triton has more than one FPGA available, we split the design over two FPGAs to facilitate routing and mapping by the compiler (see Figure 6). To simplify the design, we decided to integrate the baseband signal generation for challenge sequences and replies.

The Triton board also features Ethernet adapters which allow a remote control of the configuration. We implemented an API with commands to adapt the sync threshold, shared secret, challenge, RX and TX power, and more.

6.1 Block Delays

Both the ADC and DAC are clocked by an external 2 GHz clock, which is also used to derive the logic clocks for the two FPGAs. The main logic clock in each FPGA is running with 125 MHz (8 ns per cycle), a double data rate (DDR) clock is available for special logic circuits. While the Triton's development kit already provides synthesized blocks to connect to ADC and DAC, both introduce high delays of 60 ns each. We redesigned both blocks to minimize their delay, resulting in 16 ns delay each. In addition, we also implemented all blocks from Section 5.1 specifically for our target application and minimal delay. We then simulated the synthesized design to find the resulting delays, summarized in Table 1. It can be seen that half of the remaining delay stems from the ADC, DAC and inter-FPGA connection logic. The actual demodulation decision, XOR with the secret and reply symbol generation is almost instantaneous with 1 cycle each. The integration takes $log_2(8) = 3$ cycles.

6.2 Analog Front-end

All analog components were bought from Mini-Circuits, unless noted differently (component parts in parenthesis).

Block	# clock cycles	delay in ns	Block	# clock cycles	delay in ns
ADC interface	2	16	Demodulation	1	8
Squaring	1	8	FPGA bus	2	16
Integration	3	24	Reply generation	1	8
Correlation*	5	40	DAC interface	2	16
Synchronization*	5	40			
Total (critical path)				12	96

Table 1: Delay of blocks in the transceiver design, in clock cycles and nanoseconds. All elements marked with * are not in the time critical path (when processing challenges at the prover). The exact delays were determined in Xilinx ISE, and the overall delay was validated on the implementation.

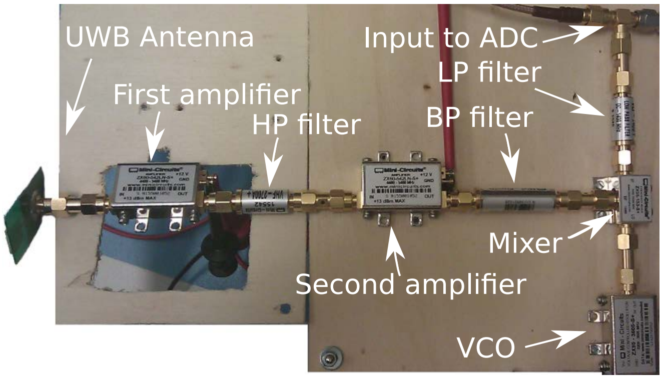


Figure 7: Overview of the analog RX frontend, with UWB antenna on the left, amplifiers, mixers, and filters.

The analog **transmission (TX) front-end** shifts the baseband signal from the DAC to the target transmission spectrum (see Figure 7). The output power of the DAC is 5 dBm, so our analog TX front-end is simply using a voltage-controlled oscillator (VCO) to generate a 4 GHz carrier (ZX95-4040-s), which is multiplied with the baseband signal using a mixer (ZX05-C60-S+). The resulting signal is transmitted by a UWB antenna (Fractus FR05-107).
Our **receiver front-end** is more complex. The signal is received by a UWB antenna, and then amplified by 23 dB (ZX60-542LN-S+). We use a high-pass filter to eliminate unwanted networks in the 2.4 GHz band (VHF-2700A+) and amplify the signal again by 23 dB (ZX60-542LN-S+). The resulting signal is band-pass filtered, and down-converted with a 3.5 GHz VCO (ZX95-3605-S+) and mixer (ZX05-153-S+) to an intermediate frequency of 500 MHz. Spectral images due to the mixing are then removed with a 1 GHz low-pass filter (VLF-1400+), and the resulting signal is sent to the ADC of the FPGA board.

6.3 Resulting Maximal Distance Advantage

We estimate the maximal distance advantage based on the obtained overall processing delay of the prover, based on analysis in [23]. The maximal distance advantage can be determined for two cases: (i) a malicious prover who minimizes his processing delay to shorten the measured distance, (ii) an external attacker trying to shorten the distance measured between an honest prover and the verifier.

The maximal distance advantage for (i) is limited by the assumed processing delay and size of the integration window T_{int} for each path. Thus, a malicious prover's maximal advantage is $\delta^B + 2T_{\text{int}} = 96\,\text{ns} + 8\,\text{ns}$ or $104\ \text{ns} \cdot 0.15\frac{m}{\text{ns}} = 15.6\,\text{m}$.

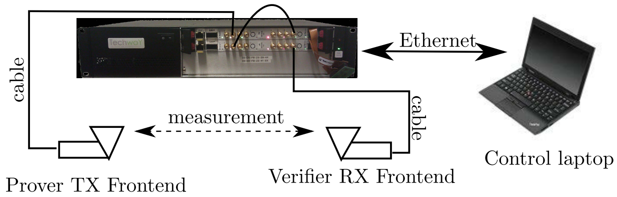


Figure 8: Our experimental setup: The FPGA boards are controlled by a host laptop via Ethernet, and are performing the wireless distance measurement using analog front-ends.

An external attacker can use attacks described in [29] to shift the integration window of both receivers. The distance advantage is thus limited by $2T_{\text{int}} = 8\,\text{ns}$ or $1.2\,\text{m}$.

7. EXPERIMENTAL RESULTS

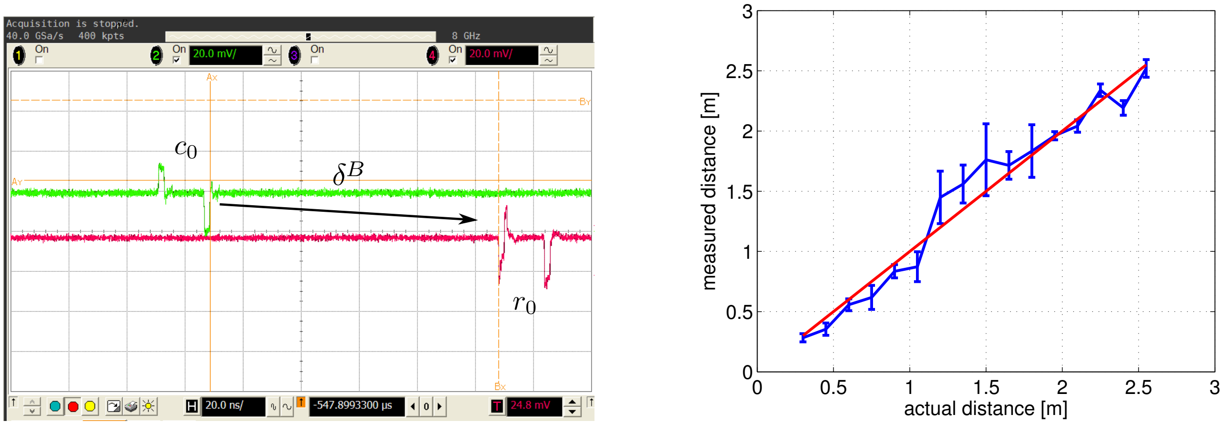
The final experimental setup is shown in Figure 8. Two Triton boards are sharing a common housing for power supply. One is acting as verifier, and the other one as prover. Both boards are supplied with a 2 GHz clock externally for the DAC and ADC components. Both boards can be configured with firmware images though a JTAG chain, the parameter configuration is possible via Ethernet. The verifier sends a continuous sequence of challenge messages, and records replies if they are encountered. The prover is continuously listening to the channel, processing any challenges received. A message is triggered when the preamble synchronization block detects energy above a certain threshold.

7.1 Example RBE Trace

Figure 9 shows such a message exchange. It can be seen how the initial preamble of the challenge message triggers the preamble of the reply message, before even the first actual challenge symbol is received. The challenge in this message is `010101010`, with a preceding first One symbol not part of the challenge, required due to implementation constraints. The shared secret in this example is `001011010`, and the correct reply `011110000` can also be seen in the figure. The time critical processing time for the challenge symbols is around 100 ns, while the initial detection of the preamble takes longer (see Figure 10a). The implementation uses a symbol period of $T_s = 512\,\text{ns}$.

7.2 Precision of Distance Measurement

Although the presented system was designed mostly from a security perspective, it is also quite accurate in distance measurements. We performed a series of experiments comparing the measured and actual distance over a wireless


Acquisition is stopped.
40.0 GSa/s 400 kpts
8 GHz
1 On
2 On 20.0 mV/
3 On
4 On 20.0 mV/
H 20.0 ns/
-547.8993300 µs
T 24.8 mV

(a)

(b)

Figure 10: Results of the implementation: (a) Details on the processing delay of the preamble and data symbols (captured by an oscilloscope). While the processing of the preamble takes more than 400 ns, the data symbols are replied to within $\delta^B \approx 100$ ns. (b) Precision of the distance measurements in line-of-sight conditions as measured in our evaluation setup.

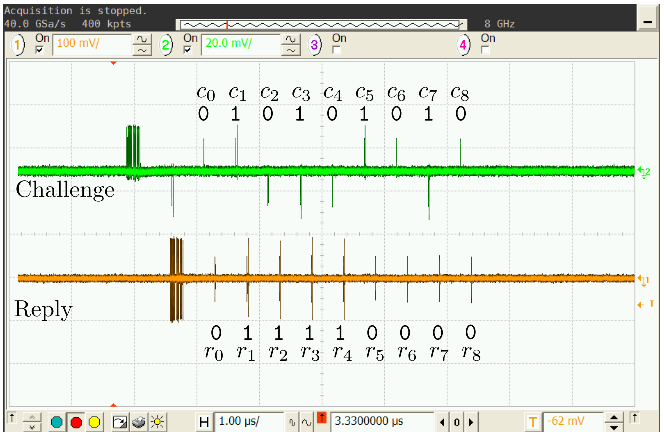

Acquisition is stopped.
40.0 GSa/s 400 kpts
8 GHz
1 On 100 mV/
2 On 20.0 mV/
3 On
4 On
H 1.00 µs/
3.3300000 µs
T -62 mV
c_0 c_1 c_2 c_3 c_4 c_5 c_6 c_7 c_8
0 1 0 1 0 1 0 1 0
Challenge
Reply
0 1 1 1 1 0 0 0 0
r_0 r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8

Figure 9: Oscilloscope trace of example messages exchanged in the implementation, and the values of the data symbols.

distance in cm	Correct in %	BER in %	estimate in cm	variance in ns
30	83	2.33	28	0.05
45	98	0.25	35	0.12
60	95	0.69	56	0.11
75	96	0.51	62	0.44
90	51	7.91	84	0.13
105	99	0.01	87	0.68
120	50	15.01	145	2.11
135	77	2.67	156	1.10
150	93	1.00	176	3.96
165	88	1.25	171	0.59
180	55	5.93	183	2.12
195	29	5.61	196	0.05
210	99	0.07	204	0.12
225	99	0.10	234	0.12
240	100	0.00	219	0.17
255	98	0.19	252	0.24

Table 2: Measurements of ranging performance: ratio of messages without bit error, bit-error-rate, distance estimate, measurement variance.

channel. To simplify these measurements, only the reply is transmitted wirelessly, while the challenge is transmitted over a wired link. The results of 1000 measurements for each distance point are summarized in Figure 10b. We also provide numeric results of the experiments in Table 2.

It can be observed that the (aggregated) accuracy is usually within 2 ns (30 cm), the error is on average 10.9 cm. The measurements were taken with static channels, so for each range the reported distance might also be influenced by indirect paths and general multipath effects. While there are several settings with higher BER (e.g., 90 cm, 120 cm, 135 cm), performance is recovering for larger distances again. In our experiments, we were limited to distances up to 255 cm, but larger distances should also be feasible. For us, it is most important that relatively short distances can be measured reliably, and therefore remote attackers do not have a gain by claiming a closer position than they actually have.

The bit-error-rate (BER) experienced in our experiments was different for each channel setup (see Table 2). In half of the measurement settings, the BER was below 1%. In another 25% of the measured distances, the BER was below 5%. Overall the mean of the BER was 2.72%. Even if these bit errors are not corrected (see [23, 27, 36]), we still have a fair number of successful exchanges without any bit errors, see the column labeled "correct". For 9 out of the 16 measured settings, more than 90% of the exchanged messages were received without any bit error.

We note that the measurement accuracy and BER can potentially both be improved by optimizing the experimental setup further. We leave this for future work.

8. DISCUSSION

We now briefly discuss the security features of our system, and introduce the novel notion of *distance commitments*.

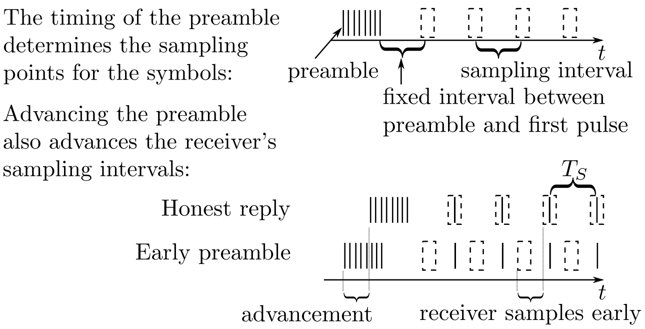

The timing of the preamble determines the sampling points for the symbols:
preamble
fixed interval between preamble and first pulse
sampling interval
t
Advancing the preamble also advances the receiver's sampling intervals:
T_S
Honest reply
Early preamble
t
advancement
receiver samples early

Figure 11: Distance Commitment: The preamble determines the sampling intervals for the data symbols of the reply. Advancing only the preamble (without advancing the data pulses) also advances the sampling intervals, leading to invalid received replies. Thus, the attacker cannot advance the preamble more than its processing advantage.

8.1 Security Guarantees of the System

We summarize the security features of our system and implementation as following (more details in Section 6.3):

- Short symbol length. For both challenge and response, the receiver integrates over a window of 4 ns. This limits the distance advantage through late-commit and early detection to $< 4ns$, less than 1 m (per direction).
- Short processing delay. In our implementation, the prover demodulates the incoming challenge pulse and sends the reply pulse in less than 96 ns. This is the fastest reported processing delay for such a system. This effectively limits the distance advantage by an attacker with 0 ns processing delay to < 15 m.

Although the maximal total distance advantage of 15.6 m is arguably larger than the maximal distance we tested our system for, the presented system provides the following guarantee: *If the prover is measured and verified to be within n meters, even the strongest attacker must be within $n + 15.6$ meters to attack successfully.*

We provide a more in-depth discussion of distance advantage possible based on the used modulation in [23].

8.2 Distance Commitment

While simultaneous transmission of prover and verifier preamble have been mentioned before [30, 33], we want to highlight the following observation: *In distance bounding, the timing of the transmitted RBE preamble acts as distance commitment by the prover.*

Why do we need a distance commitment?: Because the verifier uses the preamble of the reply message to measure the RTT, it might seem that this system is insecure. As the preamble does not (necessarily) contain secret data, the attacker could start sending the preamble of the reply message early, before actually receiving the full preamble of the challenge message. This would then shorten the measured RTT, and thus the distance estimate. This attack could be done by a malicious prover who wants to shorten the distance, or by an external attacker who wants to shorten the distance measured to an honest prover [29].

What is a distance commitment?: It is not possible to gain an advantage in the outlined attack, because the preamble is effectively a *distance commitment.* We use the novel term distance commitment to describe the communications concept of a preamble with cryptographic terms. In cryptography, a commitment (such as a hash value) can be used to prove the possession of a secret which is only revealed later [4]. For collision-free hash functions the computation of the hash requires the secret, and does not allow a later change of the secret. A distance commitment works in a similar way – it allows a sender to claim to be in a certain distance (by the transmission time of the preamble), which he has to prove later by supplying the correct secret at the correct time on the channel.

As we discussed before, the exact transmission timing of the preamble tells the receiver when to sample the channel to extract symbols to demodulate. In this sense, it is a commitment by the sender to send the data at exactly those times, as defined by the physical-layer protocol (see Figure 11). The receiver will start sampling the channel for each symbol with timing as defined by the arrival of the preamble. In other words, the insecure distance measurement based only on the preamble simplifies and enables the secure distance measurement by providing correct synchronization.

If the preamble is sent early by the attacker, the receiver will expect the data pulses earlier as well and thus start the sampling intervals earlier. If the attacker cannot provide correct data pulses in these earlier sampling intervals (e.g., because the honest prover did not send them yet), the receiver will demodulate random data.

Thus, the attacker must adhere to the correct timing between preamble and data symbols and cannot gain an advantage by sending the preamble early. At most, the attacker could advance the sampling point by up to T_{int} (forced early detection), leading to greatly increased bit error rate (see Figure 3). A similar attack was also discussed in [30], focusing on synchronization in multipath environments.

9. CONCLUSION

In this work, we presented the design and implementation of a distance bounding system. We started by discussing the lack of implementations of distance bounding protocols, in particular due to the requirements of the rapid-bit-exchange phase. We then investigated how a system for the rapid-bit-exchange should be designed from both a security and communications perspective. We argued that an UWB pulse radio system with noncoherent receivers fits our requirements, in particular due to low processing effort and short data pulses. We designed the transceiver logic for the FPGAs and the analog frontend, and implemented the system. We showed the system performance in line-of-sight experiments, and derived system parameters such as the maximal distance advantage from our system. Our prototype is able to measure distances with mean error of 10.9 cm, and has an average BER of 2.7%. To the best of our knowledge, this is the first full system which performs rapid-bit-exchange (including automatic distance estimate and response authentication) wirelessly over distance up to 2.5 m. Based on this rapid-bit-exchange system, most of the proposed distance bounding protocols can finally be build as real systems.

Acknowledgments

We would like to thank the anonymous reviewers, and our shepherd Panos Papadimitratos, for their valuable comments.

10. REFERENCES

[1] F. Althaus, F. Troesch, T. Zasowski, and A. Wittneben. STS measurements and characterization. *PULSERS Deliverable D3b6a*, IST-2001-32710 PULSERS, 2005.

[2] G. Avoine, M. A. Bingöl, S. Kardaş, C. Lauradoux, and B. Martin. A framework for analyzing RFID distance bounding protocols. *Journal of Computer Security*, 19:289–317, April 2011.

[3] S. Brands and D. Chaum. Distance-bounding protocols. In *Proceedings of the Workshop on the theory and application of cryptographic techniques on Advances in cryptology (EUROCRYPT)*. Springer, 1993.

[4] G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156 – 189, 1988.

[5] L. Bussard and W. Bagga. Distance-bounding proof of knowledge to avoid real-time attacks. In *Proceedings of IFIP Advances in Information and Communication Technology*, volume 181, 2005.

[6] S. Capkun, L. Buttyan, and J.-P. Hubaux. Sector: Secure tracking of node encounters in multi-hop wireless networks. In *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, Oct. 2003.

[7] S. Capkun and J.-P. Hubaux. Secure positioning in wireless networks. *IEEE Journal on Selected Areas of Communications*, 24(2):221–232, Feb. 2006.

[8] J. Choi and W. Stark. Performance of ultra-wideband communications with suboptimal receivers in multipath channels. *IEEE Journal on Selected Areas of Communications*, 20(9):1754–1766, Dec. 2002.

[9] J. Clulow, G. P. Hancke, M. G. Kuhn, and T. Moore. So near and yet so far: Distance-bounding attacks in wireless networks. In *Proceedings of European Conference on Security and Privacy in ad-hoc and Sensor Networks (ESAS)*, 2006.

[10] C. Cremers, K. B. Rasmussen, and S. Capkun. Distance hijacking attacks on distance bounding protocols. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, 2012.

[11] Y. G. Desmedt. Major security problems with the 'unforgeable' (feige-)fiat-shamir proofs of identity and how to overcome them. In *Proceedings of the congress on computer and communications security and protection (Securicom)*, pages 147–159, 1988.

[12] S. Drimer and S. J. Murdoch. Keep your enemies close: Distance bounding against smartcard relay attacks. In *Proceedings of USENIX Security Symposium*, 2007.

[13] ECMA-368. High rate ultra wideband PHY and MAC standard. 3rd edition, Dec. 2008.

[14] FCC. Revision of part 15 of the commission's rules regarding ultra-wideband transmission systems, adopted/released Feb. 14/Apr. 22 2002.

[15] A. Francillon, B. Danev, and S. Capkun. Relay attacks on passive keyless entry and start systems in modern cars. In *Proceedings of Network and Distributed System Security Symposium (NDSS)*, 2011.

[16] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu. Localization via UWB radios. *IEEE Signal Processing Magazine*, 22(4):70–84, Jul. 2005.

[17] G. P. Hancke. Design of a secure distance-bounding channel for RFID. *Journal on Network Computing Applications*, 34:877–887, May 2011.

[18] G. P. Hancke and M. G. Kuhn. An RFID distance bounding protocol. In *Proceedings of the Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm)*, pages 67–73, 2005.

[19] ISE design suite 10. Xilinx, Inc; *http://www.xilinx.com/*.

[20] O. Kara, S. Kardaş, M. A. Bingöl, and G. Avoine. Optimal security limits of RFID distance bounding protocols. In *Proceedings of Conference on Radio Frequency Identification: Security and Privacy Issues (RFIDSec)*, pages 220–238, 2010.

[21] C. H. Kim and G. Avoine. RFID distance bounding protocol with mixed challenges to prevent relay attacks. In *Proceedings of Conference on Cryptology and Network Security (CANS)*, volume 5888, pages 119–133, Dec 2009.

[22] C. H. Kim, G. Avoine, F. Koeune, F.-X. Standaert, and O. Pereira. The swiss-knife RFID distance bounding protocol. In *Information Security and Cryptology, ICISC*, pages 98–115. Springer-Verlag, Berlin, Heidelberg, 2009.

[23] M. Kuhn, H. Luecken, and N. O. Tippenhauer. UWB impulse radio based distance bounding. In *Proceedings of the Workshop on Positioning, Navigation and Communication (WPNC)*, 2010.

[24] A. A. Mahfouz and G. P. Hancke. Distance bounding: A practical security solution for real-time location systems. *IEEE Trans. Industrial Informatics*, 9(1):16–27, 2013.

[25] C. Meadows, P. Syverson, and L. Chang. Towards more efficient distance bounding protocols for use in sensor networks. In *Proceedings of the Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm)*, pages 1–5, 2006.

[26] J. Munilla, A. Ortiz, and A. Peinado. Distance bounding protocols with void-challenges for RFID. In *Proceedings of the Conference on Radio Frequency Identification: Security and Privacy Issues (RFIDSec)*, Graz, Austria, Jul. 2006.

[27] J. Munilla and A. Peinado. Distance bounding protocols for RFID enhanced by using void-challenges and analysis in noisy channels. *Journal on Wireless Communication in Mobile Computing*, 8:1227–1232, Nov. 2008.

[28] M. Poturalski. *Secure Neighbor Discovery and Ranging in Wireless Networks*. PhD thesis, EPFL, Lausanne, 2011.

[29] M. Poturalski, M. Flury, P. Papadimitratos, J.-P. Hubaux, and J. Boudec. On secure and precise IR-UWB ranging. *Wireless Communications, IEEE Transactions on*, 11(3):1087–1099, 2012.

[30] M. Poturalski, M. Flury, P. Papadimitratos, J.-P. Hubaux, and J.-Y. Le Boudec. Distance bounding with IEEE 802.15.4a: Attacks and countermeasures. *Wireless Communications, IEEE Transactions on*, 10(4):1334–1344, 2011.

[31] A. Ranganathan, B. Danev, A. Francillon, and S. Capkun. Physical-layer attacks on chirp-based ranging systems. In *Proceedings of the ACM Conference on Wireless Security (WiSeC)*, 2012.

[32] A. Ranganathan, N. O. Tippenhauer, B. Skoric, D. Singelée, and S. Capkun. Design and implementation of a terrorist fraud resilient distance bounding system. In *Proceedings of the European Symposium on Research in Computer Security*, 2012.

[33] K. B. Rasmussen and S. Capkun. Realization of RF distance bounding. In *Proceedings of USENIX Security Symposium*, 2010.

[34] J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji. Detecting relay attacks with timing-based protocols. In *Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 204–213, 2007.

[35] D. Singelee and B. Preneel. Location verification using secure distance bounding protocols. In *Proceedings of IEEE Conference on Mobile Adhoc and Sensor Systems Conference (MASS)*, pages 834–840, Nov 2005.

[36] D. Singelée and B. Preneel. Distance bounding in noisy environments. In *Proceedings of the European Conference on Security and Privacy in ad-hoc and Sensor Networks (ESAS)*, pages 101–115, 2007.

[37] C. Steiner, H. Luecken, T. Zasowski, F. Troesch, and A. Wittneben. Ultra low power UWB modem design: Experimental verification and performance evaluation. In *Union Radio Scientifique Internationale (URSI)*, Aug 2008.

[38] N. O. Tippenhauer. *Physical-Layer Security Aspects of Wireless Localization*. PhD thesis, ETH Zurich, Switzerland, 2012.

[39] N. O. Tippenhauer and S. Capkun. ID-based secure distance bounding and localization. In *Proceedings of the European Symposium on Research in Computer Security*, 2009.

[40] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun. On the requirements for successful gps spoofing attacks. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2011.

[41] F. Troesch, C. Steiner, T. Zasowski, T. Burger, and A. Wittneben. Hardware aware optimization of an ultra low power UWB communication system. In *Proceedings of Conference on Ultra-Wideband (ICUWB)*, pages 174–179, Sept. 2007.

[42] R. Trujillo-Rasua, B. Martin, and G. Avoine. The poulidor distance-bounding protocol. In *Proceedings of Conference on Radio Frequency Identification: Security and Privacy Issues (RFIDSec)*, pages 239–257, 2010.

[43] M. Win and R. Scholtz. Characterization of ultra-wide bandwidth wireless indoor channels: a communication-theoretic view. *IEEE Journal on Selected Areas of Communications*, 20(9):1613–1627, Dec. 2002.

[44] K. Witrisal, G. Leus, G. Janssen, M. Pausini, F. Troesch, T. Zasowski, and J. Romme. Noncoherent ultra-wideband systems. *IEEE Signal Processing Magazine*, 26(4):48–66, Jul. 2009.