Distance bounding protocols offer an effective solution to prevent attacks on ranging systems. By combining physical layer security techniques and cryptography, these protocols enable a verifier to establish an upper-bound on the distance to an untrusted prover. In practice, most secure distance bounding protocols rely on measuring the time-of-flight during a series of fast-bit exchanges to estimate the distance between the prover and verifier. Distance bounding protocols are particularly designed to prevent relay attacks, in which a man-in-the-middle attacker wrongly convinces a verifier that a far-away prover is within its vicinity. Multiple distance bounding protocols have been proposed in the literature, often based on the design principles of Brands and Chaum [12] or Hancke and Kuhn [13].

Although all these distance bounding protocols solve the problem on the logical layer, and prevent a relay attack where the content of the messages between the prover and verifier are modified, there are still a few open research challenges. First of all, it is very hard to realize the fast bit exchanges – i.e., the core building block of any distance bounding protocol – in practice. Second, most of the distance bounding protocols proposed in the literature do not take into account physical layer attacks. Indeed, a man-in-the-middle attacker that has full control over the propagation medium between the prover and verifier, can influence the signal characteristics, such as RSS or the phase, to decrease the distance measurements made by the verifier and hence perform a relay attack. Two well-known physical layer attacks are the Early-Detect/ Late-Commit attack (ED/LC) [14], and the Cicada attack [15]. One partial mitigation approach against ED/LC is to reduce the symbol length, for example by using UWB communication. However, when using short symbols, one cannot cover large distances in non-line-of-sight (NLOS) communication, as regulations by the Federal Communications Commission (FCC) limit the power that can be used per pulse. As a result, since the received signal power is inversely proportional to the square of distance between the receiver and transmitter, the reliability of the wireless communication decreases severely when increasing the distance.

To overcome this limitation, Singh et al. [16] proposed UWB-PR as a means to increase the symbol power and cover larger distances. They use multiple UWB pulses to transmit a single bit, and apply masking and pulse reordering to prevent the aforementioned physical layer attacks. In this paper, we opt for another approach to address the problem of realising secure and reliable UWB distance bounding over

---

**Abstract**—Relay attacks pose an important threat in wireless ranging and authentication systems. Distance bounding protocols have been proposed as an effective countermeasure against these attacks and allow a verifier and a prover to establish an upper bound on the distance between them. However, secure distance bounding protocols are hard to realise in practice due to stringent implementation requirements. In this paper, we look into a yet unexplored research area and show how the security strength of Ultra Wide Band (UWB) distance bounding protocols can be significantly increased by imposing several additional security constraints during demodulation and decoding at the receiver. We demonstrate that for equal reliability metrics as in state-of-the-art UWB distance bounding protocols, our solution achieves a reduction of the success probability of a relay attack by a factor of 40. Moreover, we also argue that our security solution only needs to be combined with pulse masking and a distance commitment to achieve these security bounds and there is no need to have pulse reordering in our modulation.

**Index Terms**—UWB Communication, Distance Bounding, relay attack, Wireless Security, Physical Layer Security.

---

**I. INTRODUCTION**

The proliferation of wireless sensors and networking, such as the Internet of Things (IoT), has created a huge demand on ranging applications using wireless communications. Proximity plays a very important role in IoT applications, for example in guiding robots to work in an environment to avoid collision. Multiple security applications also implicitly rely on the distance to achieve certain security guarantees. Examples are key-less entry systems, contact-less payments, and smart access control systems. It is important that the distance in these systems can be measured securely and reliably.

There are multiple ranging techniques available, such as measuring the received signal strength (RSS) [1], [2], angle of arrival (AOA) [3], [4], phase-based ranging [5], and time of flight [6], [7]. However, all these techniques have known security vulnerabilities. The attacker can use an amplifier to increase the power of transmitted signal resulted in decreasing actual distance using RSS measurement. Also, several attacks on phase-based ranging systems has been studied in [8]. There are also several attacks that have been demonstrated on commercial applications. For example, Francillon et al. [9] showed an attack on passive key-less entry systems where one could trick a car into believing that a key fob is within vicinity of the car, while in practice both the owner of the car and the key fob were far away. A similar attack can be applied on contact-less payments too [10], [11].
large distances. We propose a novel demodulation scheme and a security layer that can be integrated into the decoding process, and show that it is possible to obtain both security and reliability at the same time for large distances. We reduce the data rate of the distance bounding protocol by increasing the number of UWB pulses needed to transmit a single bit. This results in a significant increase in reliability, and hence also communication range that can be covered. In addition, we use our demodulation scheme and security layer in combination with pulse masking to restrict the degrees of freedom of an adversary. Our proposed demodulation scheme is compatible with the 802.15.4f UWB standard, as well with the FCC and European Telecommunications Standards Institute (ETSI) regulations. We analyze the security of our solution and provide a probabilistic security bound for the optimal attack strategy. We show that for a specific configuration of our demodulation and security layer, the success probability of an optimal relay attack can be reduced from 0.5 to 0.0131 per bit. Moreover, we demonstrate that this improvement in security can be achieved without relying on pulse reordering.

The rest of this paper is organized as follows. In Section II, we provide an introduction on distance bounding protocols as well as physical layer attacks and some of our notations and variables in the rest of the paper. In Section III, we describe our novel demodulation scheme. In Section IV, we analyze the impact of this demodulation scheme on the reliability of the UWB distance bounding protocol. We model the relay attacker and attack strategies in Section V and use this security model as the basis for the security analysis in the remainder of the paper. Next, we analyze the security of our demodulation scheme in Section VI and demonstrate the need for additional security measures to prevent the strongest notion of a relay attacker. We introduce the security layer in Section VII and analyse its effect on the reliability. We show in Section VIII that the proposed security layer, in combination with our demodulation scheme, reduces the success probability of the relay attacker. In Section IX we illustrate the effectiveness of our solution via a numerical example. In Section X we further enhance the security layer. Finally, we conclude the paper in Section XI.

II. BACKGROUND

A. Distance bounding protocols and literature review

Distance bounding protocols were initially introduced by Brands and Chaum [12]. These protocols are designed to establish an upper-bound on the distance between the verifier and prover, to prevent one or more “distance-based attacks”. Examples include distance-fraud attack, where a fraudulent far-away prover wants to decrease the distance measurement between itself and the verifier, and a relay attack, where a man-in-the-middle attacker wants to decrease the distance measurement between the verifier and a far-away genuine prover. The Brands-Chaum protocol works as follows. The protocol consists of three phases: (1) commitment phase, (2) fast-bit exchange, and (3) verification phase. The actual distance measurement happens in the second phase. During this phase, n rounds of fast bit exchanges are carried out. In each round, the verifier sends a random challenge to the prover. The latter will compute and reply with a response which is a function of the challenge and a secret value it committed to during the commitment phase. In each round, the verifier measures the time between sending the challenge and receiving the response. If the authentication during the verification phase succeeds and all measurements from the n rounds are below an acceptable threshold, then the prover is successfully authenticated. Otherwise, the verifier aborts the protocol.

Another blueprint for distance bounding protocols was presented by Hancke and Kuhn [13]. The Brands-Chaum and Hancke-Kuhn protocols inspired many other researchers in the last 20 years to propose other constructions for distance bounding protocols [17]–[27]. We refer to [28] for a more comprehensive overview.

B. Physical layer relay attacks

In practice, distance bounding protocols should not only prevent logical-layer attacks, but also physical layer relay attacks. In this subsection, we study 2 important physical layer relay attacks and approaches to mitigate their impact.

1. Early-Detect/ Late-Commit (ED/LC) attack: This physical layer attack was introduced firstly in [14]. It basically relies on the observation that most of the (wireless) communication systems are designed to achieve a high reliability and therefore embed some safety margins in the modulation and demodulation schemes that are used. The attacker can exploit this property during an ED/LC attack, and rely on the predictability of the symbol shapes of the wireless communication medium to reduce the time-of-flight during the fast bit exchange phase, hence resulting in a decrease of the distance estimation. In this attack, the attacker first detects the legitimate symbol in a small fraction of symbol duration. Next, it forges the shape of its own signal during the second stage of the attack, in such a way that the demodulation of the signal will still succeed (i.e. yield the same result as the demodulation of the legitimate symbol). Detecting a symbol in a small fraction of the symbol duration is possible since the attacker can be close to the transmitter and detect the symbol with high signal to noise ratio (SNR) or when it has a good antenna. This attack can decrease the distance up to twice the symbol duration. To decrease the impact of this attack, we have to keep the symbol duration as short as possible.

2. Cicada attack: This attack is applicable when the prover and verifier have to send signals in a multi-path propagation medium. This attack relies on the search algorithm that finds the strongest path to search back and find the first path as the arrival time. In this attack, the attacker aims to mislead this search algorithm by sending specific signals. The aim is to trick the search algorithm of the receiver in finding a wrong first arrival path at an earlier time. This again results in a decrease of the distance measurement between the prover and verifier. This attack can be mitigated by decreasing the length of the search window in the receiver to find the first arrival path. Some other alternative solutions to decrease the impact of this attack are also proposed in [15].

An interesting observation is that both countermeasures (for ED/LC and Cicada) achieve better security by basically reduc-
ing the reliability of the wireless communication medium. We will come back to this later in the paper.

**C. UWB communication and ranging systems**

Impulse-radio UWB systems obtain a very good opportunity for ranging applications, since it allows to send very narrow pulses in the order of nanoseconds, which increases the accuracy of ranging systems in the order of centimeters [16].

As we stated earlier, ED/LC attack can be mitigated if we decrease the symbol duration and send one pulse per bit in the modulation scheme. Although using one pulse works when we design the distance bounding protocol for small ranges, the reliability decreases when the distance increases and we cannot increase the power for each bit. Thus, for applications that need longer ranges, we need to find a way to increase the SNR. First, we can increase the power per transmitted bit to increase the SNR, while keeping the number of pulses the same, i.e., 1 pulse per bit. This approach is highly favorable since we keep the symbol size small and can prevent the ED/LC attack. However, there are limitations to this approach because of regulations. FCC dictates that the power spectral density has to be less than -41.3 dBm/MHz, averaged over a time interval of 1 ms and the power measured in a 50 MHz bandwidth around the peak frequency has to be less than 0 dBm. The first rule limits the average power that can be used in 1 ms and the second rule limits the instantaneous peak pulse power. Moreover, there are also limitations imposed by ETSI, and by hardware constraints to produce high power for each pulse in a very short duration. Therefore, one has to rely on a second approach to increase the power per transmitted bit. This can be done by increasing the number of pulses per bit, i.e., using multiple pulses to modulate a bit.

In the 802.15.4f UWB standard, both approaches correspond to the two modes of operations that are specified. Both modes use on-off keying (OOK) modulation: i.e. send nothing if the modulated bit is 0 and send a pulse if the modulated bit is 1. The first mode in the standard only sends one pulse per bit and the second (extended) mode uses multiple pulses to modulate a bit to increase the power transfer per bit. This second mode of operation is the most popular one, as it can achieve a longer communication range. However, as explained before, this mode is vulnerable to ED/LC attacks. In this paper, we will use the second communication mode to achieve a long communication range and rely on pulse masking to mitigate ED/LC attacks.

**D. UWB distance bounding: system model**

Below, we present our system model for an UWB distance bounding protocol carried out between prover and verifier. In the protocol, we employ the extended mode of the 802.15.4f UWB standard, and assume that the prover and verifier send their nonces in packets of $N_B$ bits. Each bit is modulated by $N_P$ pulses, which are either 0 or a fixed, pre-known pulse $g(t)$ with power $\|g\|^2$, based on the regulations imposed by FCC. Thus, we have in total $N_P N_B$ pulses that are transmitted. In real experiments, we have to consider the impact of noise and distance between the prover and verifier. Therefore, we assume that we have an additive white Gaussian noise (AWGN) channel and send pulses in this channel and the power decays with multiplication $l$ for distance $x$. This means if the transmitter sends a pulse with power $\|g\|^2$, the receiver gets a pulse with power $l \|g\|^2$ at a distance $x$ from the transmitter. We design our system for a given acceptable bit error rate (BER), $P_{ber}$, for the maximum acceptable distance $x$.

We assume that the UWB distance bounding protocol consists of at least two phases: a commitment phase and the fast bit exchange. During the commitment phase, the necessary nonces and parameters are exchanged that are needed during the next phases of the protocol (e.g., computation of the responses, the pulse masks, etc.). We refer to the literature for constructions of cryptographic protocols applied in distance bounding. In the remainder of the paper, we only focus on the fast-bit exchange phase, as the demodulation scheme and security layer that we propose are particularly targeted towards this phase.

Our solution can be integrated into any UWB distance bounding protocol. It only relies on two other security components being implemented at the fast-bit exchange phase of the distance bounding protocol. The first technique is the use of a distance commitment, and the second technique is to apply pulse masking.

**Distance measurement with distance Commitment: The concept of a distance commitment was introduced by Tippenhauer et al. [27] to securely synchronize the prover and verifier. The basic idea is that the preamble, which is the first part of the packet and publicly known, is used to synchronize the transmitter and receiver. Based on the preamble, the receiver knows when to expect each pulse. The verifier also measures the round-trip time of challenge and response based on this preamble. It seems that the attacker can send the preamble earlier to decrease the distance between the prover and verifier. However, if he does this, he has to send the payload, i.e., the challenges or responses, earlier as well, which means he has to guess each pulse value. Also, if the attacker only sends the preamble earlier and does not change the payload, the result would be that the receiver loses the synchronization and cannot recover the data. This mechanism is similar to a commitment in cryptography, in the sense that if a prover sends a preamble to show that it is within a acceptable range to the verifier, it has to prove this claim by sending the correct secret values at the correct time as well.

**Pulse masking: The concept of pulse masking, also denoted as pulse blinding, was introduced by Singh et al. [16]. Note that in our UWB distance bounding protocol, a series of UWB pulses, either with power 0 or power $\|g\|^2$, are exchanged between prover and verifier during the fast bit exchange phase. Moreover, each bit is modulated by $N_P$ pulses. Without any modifications to our system, each of these $N_P$ pulses would have the same amplitude. This would make ED/LC attacks trivial, as an adversary would know the amplitude of the $N_P - 1$ pulses after it has observed the first pulse. To mitigate this, one can XOR the sequence of pulses with a random mask $M$ that is shared by the prover and verifier (based on nonces exchanged in the first phase of the protocol and the shared key). By applying this pulse mask, each of the $N_P$ pulses becomes independent of the other pulses. An attacker
observing the pulse sequence cannot predict the next pulses in a sequence based on the amplitudes of the other pulses. It is important that this mask is completely random, and changes every run of the protocol.

E. Notations

The goal of our proposed security solution is to reduce the success probability of a relay attack while still having the same reliability as the state-of-the-art in UWB long-range distance bounding, i.e., UWB-PR [16]. Therefore, in the rest of this paper, we assume that the attacker aims to carry out a relay attack. So unless mentioned differently, we use attacker to refer to a man-in-the-middle attacker that is performing a relay attack and aims to impersonate the legitimate (far-away) prover during the fast bit exchange phase.

Throughout the paper, we interchangeably use two notations for a UWB pulse. Either we refer to the digital value that this pulse represents (i.e. a 0 or a 1), or we refer to the power of that pulse (i.e. 0 or $||g||^2$). We refer to Table I for all other notations used in the paper.

III. NOVEL DEMODULATION SCHEME

In this section, we explain the modulation and novel demodulation scheme we propose for UWB distance bounding.

As we mentioned before, we use the extended mode of the 802.15.4f UWB standard. Thus, we have the following binary sequence to transmit.

$$\hat{P} = (b_1, \ldots, b_{N_P}, \ldots, b_{N_B}, \ldots, b_{N_B})$$

(1)

Where, $\forall i \in [1, N_B]$, $b_i$ is $i$-th bit of $N_B$ bits.

However, to blind the pulses from the attacker point of view, we XOR $\hat{P}$ with a random binary mask $M$ before we send $\hat{P}$. The result after masking is denoted as $\tilde{P}$:

$$\tilde{P} = \hat{P} \oplus M.$$  

(2)

The mask $M$ is completely random, so all bit sequences are equally possible, and is a shared secret between the prover and verifier. We denote $\tilde{P}_i$ as a vector with size of $N_P$ that contains all entries in $\tilde{P}$ corresponding to the pulses of a bit $b_i$. We name the $j$-th entry of $\tilde{P}_i$ as $\tilde{P}_{ij}$ and denote $S_i = \sum_{j=1}^{N_P} \tilde{P}_{ij}$.

1) Modulation

For modulation, we send a pulse $g(t)$ if the corresponding pulse in $\tilde{P}$ is 1, and if the corresponding pulse in $\tilde{P}$ is 0, we send nothing. Thus, the modulated can be written as following:

$$s(t) = \sum_{k=1}^{N_P N_B} \tilde{P}[k] g(t - k T_P).$$  

(3)

2) Demodulation

For demodulation, we apply a matched filter $h(t) = g(-t)$, where $||g|| = \sqrt{T_P g^2(t) dt}$, and calculate the convolution of the received signal with the matched filter $h(t)$ and sample at $r T_P$ as follows,

$$Y[r] = (s \ast h)(r T_P) = \sqrt{I} ||g|| \tilde{P}[r].$$  

(4)

To consider the effect of the distance between the prover and verifier, we assume that the power decays with a multiplication factor $l$. Now we form two hypotheses vector with length $N_P$ for $b_i$ as follows.

$$\tilde{P}_{H_0} = [0]_{1 \times N_P} + M_i$$

(5)

$$\tilde{P}_{H_1} = [1]_{1 \times N_P} + M_i$$

(6)

where $[0]_{1 \times N_P}$ and $[1]_{1 \times N_P}$ are vectors with size of $N_P$, where all the entries are 0 and 1, respectively. $M_i$ is a vector with size of $N_P$ that contains all entries in $M$ corresponding to the bit $b_i$. We denote the $j$-th entry of $M_i$ as $m_{ij}$. We define $\sigma^i$ as follows.

$$\sigma^i = <Y_i, \tilde{P}_{H_1} > - <Y_i, \tilde{P}_{H_0} >,$$  

(7)

where $< , >$ denotes the inner product and $Y_i$ is a vector with size of $N_P$ that contains all entries in the received signal $Y$ corresponding to the bit $b_i$. We have the following equation for $\sigma^i$,

$$\sigma^i = \begin{cases} \sqrt{l(N_P - L_i)} ||g|| & b_i = 1 \\ -\sqrt{lL_i} ||g|| & b_i = 0 \end{cases}$$  

(8)
where $L_i = \sum_{j=1}^{N_i} m_{ij}$. The optimal threshold for the decoding decision (i.e. decode to the bit 0 or 1) is the point that has the same distance from $\sqrt{\ell(N_P-L_i)||g||}$ and $-\sqrt{\ell}L_i||g||$. Therefore, the optimal threshold is $\sqrt{\ell(N_P-2L_i)||g||}$. If we set this threshold, our bit error rate (BER) is minimized and hence reliability maximized.

However, when choosing this optimal threshold, the attacker also has the most degrees of freedom to transmit fake signals into the channel, which will negatively affect security. We discuss this more in detail later. Therefore, our proposal is to make the demodulation and decoding more strict to reduce the success probability of the attacker. This will be modeled by a parameter $\alpha$. This parameter can have any value between 0 and $\frac{N_P}{2}$. The higher the value $\alpha$, the more we deviate from the optimal threshold in terms of reliability ($\alpha = 0$). To compensate this, we can increase the number of pulses per bit, i.e., $N_P$. Therefore, in this work we demodulate $b_i$ as follows,

$$\hat{b}_i = \begin{cases} 1 & \sigma^i > \sqrt{\ell(N_P-2L_i)||g||} + \sqrt{\ell}a||g|| \\ 0 & \sigma^i < \sqrt{\ell(N_P-2L_i)||g||} - \sqrt{\ell}a||g|| \\ \omega & \text{otherwise} \end{cases}, \quad (9)$$

where $\hat{b}_i$ is the demodulated bit $b_i$ and $\omega$ denotes an error. This is also shown in Figure 1. In general, we want to have $Pr(b_i \neq \hat{b}_i) < P_{\text{ber}}$.

IV. ERROR PROBABILITY AND RELIABILITY

In this section, we have a look at the error probability when applying the proposed demodulation scheme in the receiver. We compare the case $\alpha \neq 0$ with the optimal case $\alpha = 0$ in terms of reliability.

In Section III we did not consider the effect of noise. In this section, we assume that we have an AWGN channel. Thus, instead of receiving $s(t)$ we now receive,

$$\tilde{s}(t) = \sqrt{\ell} \sum_{k=1}^{N_PN_B} \tilde{P}[k]g(t-kT_P) + n(t), \quad (10)$$

where $n(t)$ is a white Gaussian noise process with power spectral density $N$. If we rewrite $Y[r]$ in equation (4), we now have,

$$\hat{Y}[r] = (\tilde{s} \ast h)(rT_P) = \sqrt{\ell}||g||\tilde{P}[r] + n[r], \quad (11)$$

where $n[r] \leq r \leq N_PN_B$ are independent and identically distributed, i.i.d., Gaussian random variables with variance $N$ and mean 0. To calculate the error probability for $b_i$, we need to know the distribution of $\sigma^i$. We can show that the distribution of $\sigma^i$ if $b_i = 1$ is $\mathcal{N}\left((-L_i||g||\sqrt{\ell}, N_PN)\right)$ and if $b_i = 0$ it is $\mathcal{N}\left(-L_i||g||\sqrt{\ell}, N_PN\right)$, where $\mathcal{N}(a, b)$ is a Gaussian distribution with mean $a$ and variance $b$. We have,

$$Pr(e_i|b_i = 1) = Pr(\hat{b}_i = 0 \lor \hat{b}_i = \omega|b_i = 1) = \quad (12)$$

$$Pr(\sigma^i \leq \alpha||g||\sqrt{\ell} + \frac{(N_P-2L_i)||g||}{2}\sqrt{\ell}b_i = 1) = \quad Q\left(\frac{(\frac{N_P}{2} - \alpha)||g||}{\sqrt{N_PN}}\right), \quad (13)$$

where $Q(.)$ is Q-function. Similarly, we can show that

$$Pr(e_i|b_i = 0) = Q\left(\frac{\frac{N_P}{2} - \alpha||g||}{\sqrt{N_PN}}\right).$$

We can conclude that the error probability is as follows,

$$Pr(e_i) = Pr(e_i|b_i = 1)Pr(b_i = 1) + Pr(e_i|b_i = 0)Pr(b_i = 0)$$

$$= \frac{1}{2}\left(Pr(e_i|b_i = 1) + Pr(e_i|b_i = 0)\right) = 2Q\left(\frac{\frac{N_P}{2} - \alpha||g||}{\sqrt{N_PN}}\right).$$

We can make the following observations:

- Although $\sigma^i$ and $\hat{b}_i$ depend on $L_i$, which means it is different from one bit to another bit, the error probability is independent from the bits and only depends on $N_P$, $\alpha$, the noise power $N$, the distance, and the transmitted power $||g||^2$, which are all the same for all $N_B$ bits.
- In distance bounding systems, there is always an upper bound on the acceptable distance between the prover and verifier. This is often denoted as a circle of trust with radius $x$. This means that as long as the distance is less than $x$, a genuine prover should be accepted. With this in mind, we can design our system for the worst case scenario, which is for a distance $x$. Therefore, the multiplication $l$ in equation (13) is the ratio of the received power divided by the transmitted power for a distance $x$. Hence, the error probability given in equation (13) is the error probability in the demodulation when the distance between the prover and verifier is $x$. It is obvious that the error probability decreases when the distance is less than $x$, since the signal power or equally the SNR increases. Therefore, we design our system for a given acceptable error probability, $P_{\text{ber}}$, for the maximum acceptable distance $x$.

Let us now study the effect of $\alpha$ by giving an example on how much we have to increase $N_P$ to compensate the effect of $\alpha$, to achieve the same BER as when $\alpha = 0$. We consider two cases where in the first, the number of pulses per bit is $N_P_1$ and $\alpha_1 = 0$, and in the second case our goal is to find $N_P_2$ such that if $\alpha_2 = \frac{N_P}{2}$, we have the same error probability as

![Diagram showing demodulation with $\alpha \neq 0$.]
the first case. Thus we have,
\[
\frac{(N_{P_2} - \alpha_2)||g||\sqrt{I}}{\sqrt{N_{P_2}N}} = \frac{(N_{P_1} - \alpha_1)||g||\sqrt{I}}{\sqrt{N_{P_1}N}}
\]
\[
\Rightarrow \frac{N_{P_2} - \alpha_2}{\sqrt{N_{P_2}}} = \frac{N_{P_1} - \alpha_1}{\sqrt{N_{P_1}}} \Rightarrow N_{P_2} = 4N_{P_1}.
\]
Thus, for the case $\alpha = \frac{N_{P_2}}{2}$ we can say that by increasing the number of pulses per bit by a factor of 4 compared to the case where $\alpha = 0$, we can achieve the same error probability. Of course, this setup was only an example and we can have different setups with different $\alpha$.

V. ATTACKER MODEL

In this section, we first model the attacker’s output through a vector, denoted as attack sequence. We then provide a condition which should hold for the attacker to be successful. At the end of this section, we define the optimal attack strategy.

A. Attack sequence

First of all, let us recap the goal of the relay attacker. It wants to impersonate the fast-aray prover during the fast-bit exchange phase of a distance bounding protocol by sending the correct responses to the verifier. If the UWB distance bounding protocol is well designed [28], the best strategy for the attacker is to guess the responses that need to be sent to the verifier.

As we send $N_{P_2}$ bits as challenge or response and only use a mask to blind the binary values of the pulses of each bit, the attacker knows which pulse belong to which bit. Thus, the best policy to guess the whole sequence of the $N_{P_2}$ bits in the response is to apply the optimal attack strategy of guessing all the bits separately, since all bits are independent of each other. Therefore, to analyze the attack success probability, we can just analyze the attack success probability when $N_B = 1$. For example, if the attack success probability when $N_B = 1$ is $p$, the attack success probability for $N_B > 1$ is $p^{N_B}$.

Any attacker can be illustrated with a vector of length $N_P$ after he transmits the whole series of guessed pulses for each bit. We denote this vector as the attack sequence. The attack sequence for $b_i$ is denoted as $A_{s_i}$ as follows,
\[
A_{s_i} = (a_{i1}, a_{i2}, ..., a_{iN_{P_2}}),
\]
where $a_{ij}$ is the square root of the power of the $j$-th pulse of the bit $b_i$ that the attacker sends the receiver. As we explained in Section III, the receiver computes $\sigma^i$ in equation (7) as a variable to detect the transmitted bit. If there is an attack, the receiver gets vector $A_{s_i}$, instead of $Y_i$, for $b_i$ since the attacker will try to impersonate the prover by transmitting its guesses for the pulses. The effect of the pulses sent by the legitimate prover can be ignored, as we assume that the prover is far away in case of a relay attack. Therefore, we have equation (7) as following for $b_i$,
\[
\sigma^i = \langle A_{s_i}, \bar{P}_{H_0^i} \rangle = \langle A_{s_i}, \bar{P}_{H_1^i} \rangle.
\]

We can calculate $\sigma^i$ as follows,
\[
\sigma^i = \sum_{j=1}^{N_{P_2}} (-1)^{g_j} a_{ij},
\]
where $g_j$ is given in the following equation,
\[
g_j = \begin{cases} 2 & \bar{P}_{H_1^i}(j) = 1 \\ 1 & \bar{P}_{H_1^i}(j) = 0 \end{cases},
\]
where $\bar{P}_{H_1^i}(j)$ is the $j$-th entry of vector $\bar{P}_{H_1^i}$.

Without loss of generality, we assume that $b_i = 1$. Based on equation [18], we can conclude that if the square root of the power of the $j$-th legitimate transmitted pulse of $b_i$ is $||g||$, i.e., $\bar{P}_{H_1^i}(j) = 1$, the attacker makes a positive contribution $a_{ij}$ to convince the receiver that $b_i = 1$. Otherwise the attacker makes a negative contribution of $-a_{ij}$. Based on equation (9), the attack is successful if the following equation holds.
\[
\sigma^i = \sum_{j=1}^{N_{P_2}} (-1)^{g_j} a_{ij} > \frac{(N_{P_2} - 2L_i)\sqrt{I}||g||}{2} + \alpha\sqrt{I}||g||
\]
\[
=(S_i - \frac{N_{P_2}}{2})\sqrt{I}||g|| + \alpha\sqrt{I}||g||.
\]
This means that the attacker is successful if the summation of all his contributions is higher than $(S_i - \frac{N_{P_2}}{2})\sqrt{I}||g|| + \alpha\sqrt{I}||g||$.

Equation (19) holds when $b_i = 0$ with a minus sign on the left side of the equation, i.e., $-\sigma^i$.

B. Optimal attack strategy

For the rest of the paper, we denote the advantage of the attacker for $b_i$ as $Ad_i$ in the following equation,
\[
Ad_i = \sum_{j=1}^{N_{P_2}} (-1)^{g_j} a_{ij},
\]
where $t_j = 2$ if the power of $j$-th legitimate transmitted pulse of $b_i$ is $||g||^2$, otherwise $t_j = 1$. Therefore, an attack on $b_i$ is successful if the following equation holds.
\[
Ad_i > (S_i - \frac{N_{P_2}}{2})\sqrt{I}||g|| + \alpha\sqrt{I}||g||.
\]
Note that if $\alpha = 0$, the minimum attacker advantage becomes $(S_i - \frac{N_{P_2}}{2})\sqrt{I}||g||$ rather than $(S_i - \frac{N_{P_2}}{2})\sqrt{I}||g|| + \alpha\sqrt{I}||g||$ for a successful attack. This illustrates that a successful attack is easier when $\alpha = 0$, as expected. Therefore the higher the value of $\alpha$, the harder the attack becomes since the attacker needs to increase its advantage $\alpha\sqrt{I}||g||$ higher to have a successful attack. The attacker only has two ways to increase its advantage. First, guessing more pulses correctly, or second, increasing the power for pulses to have a higher advantage when a pulse was guessed correctly. Let us illustrate the latter with a small example. Suppose $b_i = 1$. In theory, an attacker can remain silent for all the pulses belonging to that bit, except for only one pulse (with a legitimate power $||g||^2$). If the attacker guesses this pulse correctly and replaces it by a pulse with an extremely high power, he could already be successful. If this power is high enough, it can compensate all the other pulses belonging to that bit where the attacker wrongly remained silent (i.e. a pulse with power 0 was sent by the attacker while the legitimate pulse had power $||g||^2$).

We denote the attacker advantage on $b_i$ until the $j$-th pulse as $Ad_{i,j}$, and $S_{i,j}$ the number of pulses until the $j$-th pulse that are different from zero, in the following equation:
\[ A_d = \sum_{k=1}^{j} (-1)^k a_{ik} \]  
\[ S_i = \sum_{k=1}^{j} P_{ik}. \]

We assume that the only information that the attacker has before he sends the \( j \)-th pulse of \( b_i \) is the last \((j-1)\) legitimate pulses, i.e., \( P_{i1}, ..., P_{ij-1} \). The reason for this is that the attacker receives the legitimate pulses from the far-away legitimate prover in delay (since the prover is far away, the attacker only receives the legitimate pulse after it has sent its guess for that pulse to the verifier). This is useful information for the attacker. Since the attacker knows the last \((j-1)\) legitimate pulses, it can also compute its advantages on \( b_i \) until \((j-1)\)-th pulse, i.e., \( A_d, ..., A_{d_{ij-1}} \). However, when considering \((j)\), only the terms \( A_d_{ij} \) and \( S_{ij-1} \) are relevant for the attacker, all the other values are redundant. Using these two values, the attacker can compute the optimal value \( a_{ij} \). Therefore, the attacker can use a deterministic strategy. The attacker always transmits a fixed pulse with the square root of power \( a_{ij} \) as the \( j \)-th pulse of \( b_i \) based on values \( A_d_{ij-1} \) and \( S_{ij-1} \). It is obvious that this deterministic strategy to compute the value \( a_{ij} \) of the next pulse is always better than a probabilistic strategy where the attacker would randomly choose the next value. Thus, for the rest of the paper, we only consider the deterministic attacker described above.

VI. SECURITY ANALYSIS OF THE DEMODULATION SCHEME

In this section, we introduce two types of attackers, single-power and multi-power relay attackers, and study the security of our demodulation scheme against these attackers.

A. Single- vs multi-power attacker

We can consider two different types of attackers against our demodulation scheme: single-power and multi-power attackers.

Single-power attacker: this is an attacker that only transmits pulses with either power 0 or \( l ||g||^2 \). The probability of guessing each pulse power correctly is exactly 0.5 since both values occur with the same probability, due to the randomness of the mask \( M \).

Multi-power attacker: this is an attacker that can transmit pulses with power higher than the normal (pre-agreed) power in our design, i.e. \( l ||g||^2 \).

B. Security analysis of the single-power attacker

Since the power of all legitimate transmitted pulses are \( N_P N_B \) i.i.d. binary random variables with probability of 0.5 for each power, the attack is successful if and only if the attacker guesses at least \( \frac{N_P}{2} + \alpha \) pulses correctly for each bit according to equation (21). We define \( P_{sa} \) as the attack success probability for the single-power attacker. For the sake of brevity, we assume that \( N_P \) is even. The attack success probability for the single-power attacker is as follows.

\[ P_{sa} = \left( \frac{1}{2} \left( \frac{N_P}{2} + \alpha \right) + \sum_{f=\frac{N_P}{2}+\alpha+1}^{N_P} \left( \frac{N_P}{2} + \alpha \right) \right)^{N_B} \]  
(23)

The attack success probability for the single-power attacker has been illustrated in Figure 2 when \( N_P = 32 \) for \( \alpha = 0 \) and \( \alpha = \frac{N_P}{2} = 8 \). It can be easily seen that the gain that we have in this attack model is really significant when we have \( \alpha = 8 \) in comparison with the case \( \alpha = 0 \). This design decreases the attack success probability, in the case that the attacker does not use high power pulses and just guesses the sign of the pulses, i.e. 0 or 1. It can be easily shown that the attack success probability when \( \alpha = 0 \) is always \( \frac{1}{2^{N_B}} \) and is independent from \( N_P \) for the single-power attacker.

C. Security analysis of the multi-power attacker

As we defined, the multi-power attacker can easily transmit fake pulses with a power above \( l ||g||^2 \) to compensate its own wrong guesses. In this subsection, we derive the attack success probability for the multi-power attacker. If the attacker can increase its own power unlimited to compensate its own wrong guesses, it is enough that the attacker start sending pulses with power \( \left( \frac{N_P}{2} + \alpha \right)^2 l ||g||^2 \). If the attacker is lucky and the legitimate transmitted power of the first pulse of \( b_i \) is \( ||g||^2 \), the attacker stops and sends nothing, i.e., pulses with power 0, for the rest of the pulses that belong to \( b_i \). Otherwise, he multiplies the power of the next pulse by 4 and sends it, i.e., a pulse with power \( 4 \left( \frac{N_P}{2} + \alpha \right)^2 l ||g||^2 \). If he is lucky in the second guess, he can send pulses with power 0 for the rest of the pulses belonging to \( b_i \). The attacker can continue this strategy until he has transmitted the first correct pulse with power \( ||g||^2 \) for \( b_i \). The attacker is not successful if and only if all pulses for \( b_i \) were 0. Therefore, we can say that the attack success probability is as follows.

\[ P_{ma} = \left( 1 - \frac{1}{2^{N_P}} \right)^{N_B}, \]  
(24)

It can be easily seen from equation (24) that \( P_{ma} \) is independent of \( \alpha \). Let us illustrate equation (24) with an example where \( N_P = 32 \) and \( N_B = 10 \). The attack success probability for this case is very close to 1, which means that the attacker is always successful. Although using \( \alpha \neq 0 \) helps us to
improve security against the single power attacker, it cannot help us when we are dealing with the strongest attackers. This observation shows the need for additional security rules that need to be applied during the demodulation, and avoid that an attacker can transmit pulses with arbitrary length. This will be realised by introducing an additional security layer.

VII. INTEGRATION OF SECURITY LAYER INTO DECODING

As we discussed in Subsection VI-C, the multi-power attacker can still bypass the proposed demodulation scheme, even when $\alpha > 0$. This is true because the attacker can easily increase the power and compensate its own wrong guesses for the rest of remaining pulses. Therefore, we go one step further and propose a security layer that can be added to the decoding process. As will be shown later, this security layer is an efficient way to prevent the attacks described in the previous section.

What we basically want to achieve, is to reduce the degrees of freedom of the multi-power attacker, so that its success probability gets closer to the single-power one (which we have shown we can reduce) and still having the reliability within an acceptable range, i.e., a failure rate less than $P_{ber}$. Therefore, we want to add a security layer to the decoding process, and combine this with our proposed demodulation scheme, to help us detect if an attacker is present. From the previous section, we can observe that we have to restrict the maximum power that the attacker can use. We also want to prevent the attacker from using an attack strategy where it adaptively changes the transmission power of the transmitted pulses. These two security requirements will be respectively translated into two security rules in our security layer, that will be enforced during the decoding process at the receiver. These rules will now be discussed more in detail. It is important to stress that both security rules need to be applied at the same time to achieve the required security properties. Moreover, we also want to point out that all the parameters in the two security rules can be freely chosen, and in theory even adapted at runtime.

A. First security rule

The attacker does not know the value of the pulses in advance, i.e., before they arrive in delay from the legitimate prover. Also knowing the value of the previous pulses does not help, because all pulses are blinded with a random binary sequence length $N_P N_B$, i.e., the mask $M$. The only thing the attacker does learn, is which of its previously transmitted pulses it has guessed correctly (because it receives the correct values from the legitimate prover in delay). As we discussed in Subsection VI-C, a multi-power attacker can increase its attack success probability by increasing its power beyond $||g||^2$. However, we can exploit the fact that the attacker has to guess the value the pulses. There is a 50% chance that he transmits a wrong pulse and if this happens, we can detect it during decoding.

Recall that the output of the demodulation is a bit 0 or 1. If we now combine a group of $N_P$ pulses and want to decode it into a binary 0 or 1, we can detect some anomalies. More in particular, if we have a high power pulse when we were expecting a pulse with power 0, most likely an attacker was present. A legitimate prover would never send a high power pulse when a zero pulse was expected, they are just silent during these time slots. Thus, the only other factor that could have affected the power of a pulse is noise. However, the noise is fixed, related to the non-ideality of the receiver and not related to distance.

Therefore, we can enforce a security constraint (i.e. our first security rule) and apply a strict threshold $lk_1^2||g||^2$ as the maximum acceptable power each time we expect a pulse with power 0. This means that if an attacker transmits a fake pulse with power higher than $lk_1^2||g||^2$, we abort the protocol. Note that $k_1$ is a security parameter in our security layer. In an ideal communication system where no attacker would be present, we could have $k_1 = \infty$. The parameter $k_1$ obviously decreases the reliability, but offers a good security interchangeably. The probability that the additive white Gaussian noise increases the power of a pulse at least $lk_1^2||g||^2$ is $Q(k_1||g||\sqrt{N})$. The probability that we have at least one pulse among all the transmitted pulses with power higher than $lk_1^2||g||^2$, when a pulse with power 0 was transmitted by the legitimate prover, is upper-bounded as follows:

\[
Pr(A_{per}) =
\]

\[
\sum_{i=1}^{N_P N_B} \left( 1 - Pr(N_P N_B - \sum_{j=1}^{N_B} S_j = i) \right) Pr(N_P N_B - \sum_{j=1}^{N_B} S_j = i)
\]

\[
< \sum_{i=1}^{N_P N_B} \frac{iQ(k_1||g||\sqrt{N})}{Pr(S_j = i)} Pr(N_P N_B - \sum_{j=1}^{N_B} S_j = i)
\]

\[
= \sum_{i=1}^{N_P N_B} \frac{iQ(k_1||g||\sqrt{N})}{2N_P N_B} \left( \frac{N_P N_B}{i} \right) Pr(S_j = i)
\]

\[
= \frac{N_P N_B}{2} Q(k_1||g||\sqrt{N}).
\]

In equation (25), $A_{per}$ denotes the probability of a wrong abortion when applying our security rule on a packet of $N_B$ bits. To find the effect of the first rule on one bit, we have to calculate the error probability for a packet of $N_B$ bit based on the acceptable error probability, i.e., $P_{ber}$, and compare it with equation (25). The error probability for a packet is given in the following equation:

\[
P_{per} = 1 - (1 - P_{ber})^{N_P} \simeq N_B P_{ber}
\]

\[
\Rightarrow P_{ber} \simeq \frac{P_{per}}{N_B} \Rightarrow Pr(A_{ber}) < \frac{N_P}{2} Q(k_1||g||\sqrt{N})
\]

where $P_{per}$ denotes the error probability of a packet of $N_B$ bits, and $A_{ber}$ denotes the event of a wrong abortion of the protocol for one bit, due to the constrains imposed by the first rule.
B. Second security rule

The first security rule detects the case where an attacker would send a pulse with high power when a 0 pulse was expected. However, this does not stop a multi-power attacker that adaptively changes its power when correctly sending a 1 pulse. Therefore, we can apply the same concept and detect this pattern. The pattern of significantly changing the transmission power would not be observed when the transmitter is legitimate. The legitimate transmitter only sends pulses using a relatively fixed transmission power, so all the received pulses are expected to be similar, except for additional white Gaussian noise with fixed variance, as can be seen in equation (11). Thus, the difference of power between any two pulses with transmitted power $|g|^2$ cannot vary by a high degree.

Therefore, in our second security rule, we apply another threshold $\sqrt{ld|g|}$ for being the maximum allowed difference between the square root of the power of any two pulses (with expected transmission power $|g|^2$, i.e. corresponding to a binary value 1). Similarly as for the first rule, this decreases the reliability. If we increase $d$, we increase the reliability but leave more space for the attacker to transmit fake pulses into the channel.

For a given threshold $\sqrt{ld|g|}$, we can compute the probability $Pr_{nd}$ that the difference between the square root of the power of two legitimate pulses (with binary value 1) belong to an arbitrary bit $b_j$ is higher than the threshold. In other words, $Pr_{nd}$ denotes the probability that the protocol aborts wrongly due to the restriction imposed by the second security rule.

$$Pr_{nd} = Pr(\bigcup_{i,j \in [1,z], i < j} (S_{ij}))$$

$$\leq \sum_{z=2}^{N_p} Pr(S_j = z) \left(\frac{z}{2}\right) Pr(A_{ij})$$

$$= \sum_{z=2}^{N_p} Pr(S_j = z) \left(\frac{z}{2}\right) Pr(A)$$

$$= Pr(A) \sum_{z=2}^{N_p} \left(\frac{z}{2}\right) \frac{N_p}{2N_p} = Pr(A) \sum_{z=2}^{N_p} \left(\frac{z}{2}\right) \frac{N_p}{2N_p} - \frac{1}{2} \sum_{z=0}^{N_p} \left(\frac{N_p}{2}\right)^2$$

$$= Pr(A) \left(\frac{N_p}{2}\right)^2.$$  

(27)

In this equation, $Pr(A_{ij})$ denotes the probability that the absolute value of the difference between the square root of the power of the $i$-th and $j$-th pulses (out of $z$ pulses with legitimate transmitted power $|g|^2$) is higher than $\sqrt{ld|g|}$. Since we have additive i.i.d. Gaussian noise for all $z$ pulses, $Pr(A_{ij})$ is the same for all $i$ and $j$. Therefore, we can rename it to $Pr(A)$. If we denote $x$ and $y$ as the square root of the received power of the $i$-th and $j$-th pulses, we can calculate $Pr(A)$ as follows. Note that we use the symbol $a = \sqrt{ld|g|}$ to simplify the notation.

$$Pr(A) = \int_{|x-y| > a} \frac{1}{2\pi N} e^{-\frac{x^2+y^2}{2\pi}} dx dy = 2\int_{-\infty}^{+\infty} \frac{1}{2\sqrt{2\pi N}} Q\left(\frac{y + a}{\sqrt{N}}\right) e^{\frac{-y^2}{2\pi}} dy = 2\int_{-\infty}^{+\infty} \frac{1}{2\sqrt{2\pi N}} Q\left(\frac{y + a}{\sqrt{N}}\right) e^{\frac{-y^2}{2\pi}} dy$$

$$+ 2\int_{-\infty}^{+\infty} \frac{1}{2\sqrt{2\pi N}} Q\left(\frac{y - a}{\sqrt{N}}\right) e^{\frac{-y^2}{2\pi}} dy = 2\int_{-\infty}^{+\infty} \frac{1}{2\sqrt{2\pi N}} e^{\frac{-y^2}{2\pi}} dy + 2\int_{-\infty}^{+\infty} \frac{1}{2\sqrt{2\pi N}} Q\left(\frac{y + a}{\sqrt{N}}\right) e^{\frac{-y^2}{2\pi}} dy = 2Q\left(\frac{a}{\sqrt{N}}\right) + 2\int_{-\infty}^{+\infty} \frac{1}{2\sqrt{2\pi N}} Q\left(\frac{y + a}{\sqrt{N}}\right) e^{\frac{-y^2}{2\pi}} dy.$$  

(28)

To calculate the integral, we use the Chernoff bound $Q(x) \leq \frac{1}{2} e^{-\frac{x^2}{2}}$ given in [29]. This results in:

$$Pr(A) < 2Q\left(\frac{a}{\sqrt{N}}\right) + \int_{-\infty}^{+\infty} \frac{1}{2\sqrt{2\pi N}} e^{\frac{-(y+a)^2}{2\pi}} e^{\frac{-y^2}{\sqrt{2}}} dy$$

$$< 2Q\left(\frac{a}{\sqrt{N}}\right) + e^{-\frac{a^2}{N}} \int_{-\infty}^{+\infty} \frac{1}{2\sqrt{2\pi N}} e^{\frac{-(y+a)^2}{2\pi}} dy$$

$$= 2Q\left(\frac{a}{\sqrt{N}}\right) + e^{-\frac{a^2}{N}} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{\pi N}} e^{\frac{-(y+a)^2}{2\pi}} dy$$

$$= 2Q\left(\frac{a}{\sqrt{N}}\right) + e^{-\frac{a^2}{N}} \left(1 - Q\left(\frac{a}{\sqrt{2N}}\right)\right) < 2Q\left(\frac{a}{\sqrt{N}}\right) + e^{-\frac{a^2}{2N}}.$$  

(29)

From (27) and (29), we have,

$$Pr_{nd} < \frac{(N_p)}{4} \left(2Q\left(\frac{a}{\sqrt{N}}\right) + e^{-\frac{a^2}{2N}}\right).$$  

(30)

VIII. SECURITY ANALYSIS OF THE SECURITY LAYER

In this section, we can now study the attack success probability when we apply the proposed security rules. We start the analysis by first studying the effect of each of the rules individually, and then discuss the optimal attack strategy when both rules are applied.

A. Security analysis when only the first rule is applied

When we only apply the first security rule and ignore the second rule, the optimal attack strategy is as follows. Assuming that $k_1$ is small enough, the attacker randomly selects one pulse in each round of $N_p$ pulses. For this particular pulse, the attacker sends a high-power pulse with power $(\frac{N_p}{2} + \alpha + 1)^2|g|^2$. For all the other $N_p - 1$ pulses, the attacker just stays silent. If a 1 value was expected for the pulse that the attacker impersonated, then the attack is successful based on equation (21). Otherwise, the attack fails since the attacker transmitted a fake high-power pulse in a wrong place and the protocol is aborted due to the first security rule. Since both 0 and 1 are equally probable for each pulse, we can conclude that the attack is successful with probability
0.5 for each bit. Since all bits are independent from each other, the overall attack success probability is \(2^{-N_B}\) when the size of the response is \(N_B\) bits. Although we have improved the security against the multi-power attacker by only applying this first rule, the attacker is still powerful.

**B. Security analysis when only the second rule is applied**

If we only apply the second rule and ignore the first rule, we have a similar result. In this case, the optimal attack strategy, again assuming \(d\) is small enough, is as follows. For all the \(N_P N_B\) pulses, the attacker sends pulses with power \((\frac{N_P}{2} + \alpha + 1)^2 l ||g||^2\). At the end, the attacker wins if for each bit, the number of legitimate pulses with value 1 is higher than the number of legitimate pulses with value 0. Therefore, the attack success probability is \(\prod_{i=1}^{N_P} Pr(S_i > \frac{N_P}{2})\). \(Pr(S_i > \frac{N_P}{2})\) is close to \(\frac{1}{2}\) for large enough values of \(N_P\). Thus, we can again conclude that the attack success probability is about \(2^{-N_B}\) if we only apply the second rule.

**C. Security analysis for both security rules combined**

In this subsection, we present two important properties of the optimal attack strategy when both security rules are combined. We refer to Appendix A for a formal proof of these two lemmas.

**Lemma VIII.1.** In the optimal attack strategy, the attacker does not increase the square root of the power of the \(N_P N_B\) pulses higher than \((k_1 + d) \sqrt{l} ||g||\).

This lemma obtains an upper bound on the maximum power that the attacker can use in the optimal attack strategy. Based on this lemma, we can conclude that by finding the tightest possible thresholds \(k_1\) and \(d\), i.e. the values that do not reduce the error probability below an acceptable threshold \(P_{ber}\), we can restrict the attacker, as he cannot send fake pulses with a power higher than \((k_1 + d)^2 l ||g||^2\). The next lemma describes another property of the optimal attack strategy.

**Lemma VIII.2.** If during the attack, the attacker wants to send \(i\) pulses \((i \geq 1)\) with a power \(P_i l ||g||^2\) for which the condition \(k_1^2 < P_i \leq (k_1 + d)^2\) holds, then the optimal attack strategy is to choose \(P_i = (k_1 + d)^2\) for all these \(i\) pulses.

It is good to note that the security layer needs to be combined with the demodulation scheme presented before. If \(\alpha = 0\), an attacker can guess each bit of the protocol with probability 0.5, just by sending random pulses with either power \(l ||g||^2\) or 0 (i.e. sending nothing). This guessing attack cannot be detected by the security layer. Therefore, using the proposed demodulation scheme with \(\alpha > 0\) helps us to decrease the probability of this attack. To be successful, the attacker needs to achieve a higher advantage (see equation (21)) to compensate this. And the options to increase this advantage are limited due to the rules of the security layer. In conclusion, the security layer and the demodulation scheme with \(\alpha > 0\) complement each other, and the combination of both achieves the best security and reliability trade-offs.

**IX. ILLUSTRATION OF OUR SECURITY SOLUTION**

In this section, we will illustrate the effectiveness of our proposed demodulation scheme and security layer by some practical examples. As discussed before, we need to send multiple pulses per bit to increase the reliability and the distance that can be covered. As a benchmark, let us take the case where 1 pulse per bit is used. Let us assume that this benchmark system can achieve an error probability of \(P_{ber}\) for distances less than \(x_0\). Our goal is now to extend the range of the distance bounding system to cover larger distances, for example \(x_1 = 2x_0\) and \(x_2 = 4x_0\), and still have an error probability lower than \(P_{ber}\). In our example, we will assume that the acceptable bit error rate is \(P_{ber} = 10^{-3}\). Furthermore, we consider two cases: one where \(x_1 = 2x_0\) and another one \(x_2 = 4x_0\). For each of these two cases, we will illustrate how our demodulation scheme and security layer can achieve the predetermined reliability requirement. Next, we will also analyse the security for both cases.

**A. Reliability analysis for both examples**

For the case \(x_1 = 2x_0\), we propose to use \(N_P = 32\) pulses per bit, and set the parameter of our demodulation scheme to \(\alpha = \frac{N_P}{2}\). Moreover, we set the thresholds of our first and second security rule to \(k_1 = 1.29\) and \(d = 2.3\). First of all, let us show that we can still achieve the required reliability for distance \(x_1\) when using the proposed parameters and thresholds. It is worth to note that there are 3 cases in which a legitimate proof gets rejected (i.e. the distance bounding protocol aborts wrongly). The first case is a wrong abortion of the protocol due to the first security rule (with threshold \(k_1\)), the second case is due to the second security rule (with threshold \(d\)), and the third case is a demodulation error due to noise. Therefore, to calculate the error probability, we have to combine these 3 cases as follows:

\[
P_{e_1} = Pr(A_{ber} \cup B \cup C) < Pr(A_{ber}) + Pr(B) + Pr(C)
\]

(31)

In this equation, \(A_{ber}\) denotes a wrong abortion due to the first rule, \(B\) a wrong abortion due to the second rule, and \(C\) an error caused by the demodulation. From equations (26), (30), and (13) we have,

\[
Pr(A_{ber}) < \frac{N_P}{2} Q\left(\frac{k_1 ||g|| \sqrt{T_1}}{\sqrt{N}}\right)
\]

(32)

\[
Pr(B) = Pr_{nd} < \left(\frac{N_P}{2}\right) \left(2 Q\left(\frac{\sqrt{T_1 ||g|| \sqrt{T_1}}}{\sqrt{N}}\right) \frac{1}{4} + \frac{\sqrt{\frac{1}{12}}}{\sqrt{2}}\right)
\]

\[
Pr(C) = Q\left(\frac{\sqrt{N_P} ||g|| \sqrt{T_1}}{4 \sqrt{N}}\right).
\]

The error probability for distance \(x_0\) is \(Q(\frac{\sqrt{10} ||g|| \sqrt{10}}{2\sqrt{N}})\). Taking into account our assumption that \(P_{ber} = 10^{-3}\), we know that \(Q(\frac{\sqrt{10} ||g|| \sqrt{10}}{2\sqrt{N}}) = 10^{-3}\). Hence, we have \(\frac{\sqrt{10} ||g|| \sqrt{10}}{2\sqrt{N}} = 3.09\). Also, we have \(l_1 = \frac{\sqrt{10}}{4}\) since the received power is inversely proportional to the square of the distance between the receiver and transmitter. Combining this information in equation (32), we get \(Pr(A_{ber}) < 5.37 \times 10^{-4}\).
\[ Pr(B) < 2.87 \times 10^{-4}, \quad Pr(c) \approx 6.6 \times 10^{-6}. \] This results in \( P_{c_1} < 8.3 \times 10^{-4} < 10^{-3}. \) This means that the reliability for distance \( x_1 \) is not worse than the reliability for distance \( x_0 \) and our reliability requirement of \( P_{\text{err}} = 10^{-3} \) was met.

For distance \( 4x_0 \), we can set our thresholds and parameters to \( k_1 = 2.9, \quad d = 5, \) and \( N_P = 128. \) Thus, we have \( \sqrt{2l|||g||^2} = 1.545 \) since \( l_2 = \frac{4a}{16}. \) Using the same reasoning as above for distance \( x_2 = 4x_0, \) we get \( P_{v_2} < 7.21 \times 10^{-4} < \frac{P_{\text{err}}}{10^{-3}}. \)

The numbers above demonstrate that we can achieve a similar reliability as in the benchmark case, while having increased the distance that can be covered by the distance bounding protocol.

B. Security analysis for both examples

The attacker goal is to increase its advantage for each bit that it wants to relay to the point that equation (21) is satisfied, while also still complying with the two rules in the security layer. Recall that the attacker has two options to do this, as was discussed in Section VI. First, keep the power of the pulses lower than \( l|g|^2 \) and guesses more pulse values, which is hard and results in a single-power attacker. Second, the attacker increases the power of the pulses to have an opportunity to compensate wrong pulse guesses and hence increase the attacker advantage (see equation (21)).

Close-to-optimal attack strategy. As we proved in Lemmas VII.1 and VII.2, the optimal attack strategy is bounded by some constraints. In the optimal attack strategy the maximum square root of the power of a pulse is \( (k_1 + d)\sqrt{l|g|}. \)

Also, the optimal attacker always has to send pulses with a specific power between \( k_1^2l|g| \) and \( (k_1 + d)^2l|g| \), and for the rest of pulses the power has to be less than \( k_1^2l|g| \). Thus, the optimal attacker gets really restricted by the security layer. Although we have derived these two constraints, it is difficult to derive a close formula or algorithm for the optimal strategy. The reason is that for each pulse the optimal attacker has to send, it has to calculate the probabilities of all future cases (i.e. potential values of the legitimate pulses), and then adapt its strategy based on this to maximize its advantage. This makes it very expensive to carry out the optimal attack strategy in practice. Therefore, we will only discuss an interesting close-to-optimal attack strategy in the remainder of this section.

In the optimal attack strategy, the attacker needs to stop sending pulses with power higher than \( l|g| \) from the moment its advantage for a bit, computed by equation (22), is big enough. If the attacker would continue sending pulses with this higher power after that moment, its success probability would decrease again. Therefore, in our baseline attack we assume that the attacker sends \( t \) pulses with the highest possible amount of power, i.e., \( (d+1)^2l|g| \). We call this the first step of the attack, and \( t \) is the first parameter of our attack strategy. \( (d+1)^2l|g| \) is the highest possible amount of power rather than \( (d+k_1)^2l|g| \), taking into account the second rule of security layer, the attacker cannot decrease the square root of the power for the remaining pulses of that bit more than \( d\sqrt{t|g|} \) and the attacker needs to stop sending pulses with power higher than \( l|g| \) from the moment its advantage is high enough. We assume that after the first step of the attack, the attacker sends pulses with power \( k_1^2l|g| \) until the moment it has obtained an advantage \( ak_1 + t(d+1) \). We call this the second step of the attack, with \( a \) being the second parameter of the attack strategy. Next, after the required advantage has been reached, the attacker further decreases the power and transmits pulses with power \( l|g| \) for all the remaining pulses of that bit, to prevent that its advantage decreases too much. We call this the third step of the attack.

The basic idea behind this attack strategy is to obtain the maximum possible advantage in the first and second steps by sending pulses with the highest possible power and then protect the obtained advantage by decreasing the power for the rest of pulses in the third step to the lowest power possible. Since the attacker does not know the mask, all pulses power are equally probable from the attacker’s point of view. Therefore, changing the power in the second step of the attack does not make sense. The only way that the attacker can increase its advantage is to use the maximum possible power and hope that the legitimate transmitted pulse is \( |g| \). Roughly speaking, the multi-power attacker can gain mostly from a legitimate transmitted sequence where most of the pulses have a power \( |g| \). The more legitimate pulses are zero, the more that increasing the power is useless for the attacker, since this only decreases the advantage for that bit (see (21)).

In the attack strategy above, the attacker has to find the two parameters \( t \) and \( a \) to find the best attack performance. We know that \( 0 \leq t \leq N_P \) and \( 0 \leq a \leq N_P - t \). However, the optimal \( t \) will be much smaller than \( N_P \). If the attacker sends \( t \) pulses with power higher than \( k_1^2l|g| \), the probability that the distance bounding protocol will abort due to the first rule of the security layer is \( 1 - \frac{1}{2}T \) in the first step of the attack (when a high power pulse was sent when a 0 pulse was expected).

Numerical example. Let us now revisit the two examples discussed before, and find the optimal \( t \) and \( a \) for these two cases. We have run a simulation for different \( a \) and \( t \), the results are shown in figures 3 and 4. In the simulations, for each value of \( a \) and \( t \), we have generated \( 10^6 \) legitimate binary sequences to calculate the attack success probability.

As illustrated in Figure 3 for distance \( x = x_1 \) with \( N_P = 32, \quad k_1 = 1.29 \) and \( d = 2.3, \) if \( t = 0 \) (denoted by the blue curve) the attack success probability increases to 0.04
until $a = 11$, and then decrease again to 0.025. This means if $t = 0$, it is best to send pulses with power $1.29 l_1 ||g||^2$ until we have an advantage $1.29 \times 11 = 14.19$ in the second step of attack, and then send the remaining pulses of that bit with power $l_1 ||g||^2$. The figure also shows that if we increase $t$, the attack success probability will increase until $t = 2$. For this case, if $a = 5$ we have the best attack success probability, equal to 0.086. If we increase $t$ further, the probability of the protocol aborting increases. This results in having a lower attack success probability in comparison with $t = 2$. For example, for $t = 3$ the probability that the protocol aborts in the first step is $\frac{7}{8}$, which means there is only a probability of $\frac{1}{8}$ that attacker can continue to the second step of attack. In Figure 3 for $t = 3$ and 4, we can see that it is better to jump immediately from the first step to the third step, i.e., sending only 3 and 4 pulses with power $3.3^2 l_1 ||g||^2$, respectively, and after that sending all the other pulses with power $l_1 ||g||^2$. The reason is that the advantage after the first step is already high enough and we only need to decrease the power to save the advantage. This is different than the cases $t = 0, 1, 2$, where we do need the second step to have a high attack success probability.

As illustrated in Figure 4 for distance $x = x_2$ with $N_P = 128$, $k_1 = 2.9$ and $d = 5$, the best attack strategy is not to send any pulse with power higher than $2.9^2 l_2 ||g||^2$, i.e., $t = 0$. Also, the optimal time to stop sending pulses with power $2.9^2 l_2 ||g||^2$ is when $a = 15$, i.e. when we have obtained an advantage $a k_1 = 15 \times 2.9 = 43.5$

X. FURTHER ENHANCEMENTS OF THE SECURITY LAYER

A. Enhanced first security rule

In terms of security, we can even go one step further and enhance the first rule of our security layer even more. As we discussed, all attacks in which an attacker inserts pulses with a power higher than $k_2^2 l_2 ||g||^2$ when the power of the legitimate pulse is expected to be 0, are detectable. That means that if the attacker transmits even one pulse with a power higher than $k_2^2 l_2 ||g||^2$ when a 0 was expected, the protocol aborts and the attack fails. One could further extend this rule and have an adaptable threshold. For example, we could allow at most one pulse with power between $k_2^2 l_2 ||g||^2$ and $k_2^2 l_2 ||g||^2$, where $k_2 < k_1$. This would mean that the attacker would have to reduce the power of its transmitted pulses after one wrong guess, otherwise the protocol would abort at the second wrong guess of a legitimate pulse with power 0.

We can generalize this and consider $i$ thresholds denoted by $k_1, k_2, \ldots, k_i$. The enhanced first security rule now states that if $j \leq i$ pulses with power higher than $k_j^2 l_j ||g||^2$ are detected, when the power of each of these $j$ pulses was expected to be 0, we abort the protocol. Of course, we have to take care that we still satisfy the reliability requirements, i.e., have an error probability less than $P_{ber}$. We can rewrite the upper-bound of the probability of a wrong protocol abortion due to the enhanced first rule as follows,

$$Pr(A_{ber}) = \sum_{j=1}^{N_B} \sum_{z \in S_j} \left( Pr(A_z) \right) \leq \sum_{j=1}^{N_B} \frac{Pr(A_z)}{N_B},$$

where $A_z$ is the event of a wrong protocol abortion caused by applying the threshold $k_z$. We can calculate $Pr(A_z)$ as following,

$$Pr(A_z) \leq \sum_{j=1}^{N_B} \left( \sum_{z \in S_j} \left( \frac{j}{2^N} \right) \frac{Pr(A_z)}{N_B} - \sum_{k=1}^{N_B} S_k = j \right)$$

$$= \sum_{j=1}^{N_B} \left( \sum_{z \in S_j} \left( \frac{j}{2^N} \right) \frac{Pr(A_z)}{N_B} - \sum_{k=1}^{N_B} S_k = j \right)$$

We can now compute the upper bound on the total error probability when the first security rule is replaced by the enhanced rule. In other words, we can compute the error probability of combining the first enhanced rule, the second rule and the demodulation scheme. This probability is parameterized by the design parameters, $\alpha, k_1, k_2, \ldots, k_i, d, N_P, N_B, ||g||$ and our channel characteristics $l, N$, and is equal to:

$$P_e \leq \sum_{i=1}^{N_P} \left( \sum_{j=1}^{N_B} \left( \frac{j}{2^N} \right) \frac{Pr(A_z)}{N_B} - \sum_{k=1}^{N_B} S_k = j \right)$$

$$+ \left( \frac{N_P}{2} \right) \left( \frac{\sqrt{\sqrt{2} l_2 ||g||}}{\sqrt{N}} \right) + \frac{e^{-2\sqrt{\sqrt{2} l_2 ||g||}}}{2^N} + \frac{Q \left( \frac{N_P}{2} - \alpha ||g|| \sqrt{N} \right) }{\sqrt{N_P N}}$$

The first term of (35) is the error probability due to the enhanced first rule; the second term is the error probability due to the second rule; and the third term is the error probability caused by demodulation errors. Note that we require that this total error probability should be less than $P_{ber}$.

This enhanced first rule does not significantly impact the complexity of the system. Both in the initial first security rule as in the enhanced version, one needs to validate that for each pulse where a 0 was expected, the power of the received pulse does not exceed a threshold. We refer to Appendix B for some other practical considerations regarding the realization of the security layer.
B. Numerical example

To show the effectiveness of this enhanced security layer, let us consider the numerical example where \( i = 7, k_1 = 1.29, k_2 = \infty, k_3 = 1, k_4 = k_5 = k_6 = \infty, k_7 = 0.7 \) and \( d = 2.3 \) with \( N_P = 32, N_B = 5 \), and \( \alpha = \frac{N_P}{2} \), for the distance \( x_1 = 2x_0 \). First, from equation (55), we can easily verify that our error probability is \( 9.87 \times 10^{-4} < 10^{-3} \). Therefore, the reliability condition is satisfied.

Now let us discuss the security for this numerical example. The attacker is clearly highly restricted since he can only send pulses with power \((k_2 + d)^2t_1||g||^2 = 3^2t_1||g||^2\) instead \((1 + d)^2t_1||g||^2 = 3.3^2t_1||g||^2\). He can only make 6 wrong guesses (of a 0 pulse) using a power higher than \(k_1^2t_1||g||^2 = 0.7^2t_1||g||^2\) out of a total of \(N_PN_B = 32 \times 5 = 160\) pulses. Moreover, if he sends pulses with a power \(3.3^2t_1||g||^2\), then he can not decrease the power of any other 1 pulse below \(t_1||g||^2\), to comply with the second rule. The attacker can also make only 2 wrong guesses with a power higher than \(k_1^2t_1||g||^2 = t_1||g||^2\) among the 160 pulses. These two constraints imply that the attacker can only gain from sending multiple pulses, let us say \(t\) pulses, with a power \(3^2t_1|.|g|^2\) and hope that these pulses were expected to be 1 and send the rest of the pulses with power \(0.7^2t_1|.|g|^2\).

To find the optimal \(t\), we again run a simulation for different values of \(t\), as shown in Figure 5. Similarly as before, for each value of \(t\), we generated \(10^6\) transmitted legitimate binary sequences to calculate the attack success probability. Also, since the attacker can guess in total up to 2 pulses wrongly using a power between \(t_1|.|g|^2\) and \(1.29^2t_1|.|g|^2\) for all bits, we assume that after sending \(t\) pulses with power \(3^2t_1|.|g|^2\), the attacker sends pulses with power \(1.29^2t_1|.|g|^2\) until the first wrong guess for each bit. This provides more degree of freedom for the attacker since now it can guess in total up to 5 pulses wrongly rather than 2 pulses among 160 pulses. Then he further decreases the power to \(0.7^2t_1|.|g|^2\) for the rest of the pulses belong to that specific bit. According to Figure 5 the best attack success probability is 0.0496 and can be obtained with \(t = 3\). Note that the attack success probability in Figure 5 is the success probability per bit. To have a successful attack, the attacker has to be successful for all \(N_B = 5\) bits, which means that the attack success probability for the entire protocol becomes 0.0496\(^5\) = \(3 \times 10^{-5}\).

In Figure 6 we show different attack success probabilities for different values of \(N_B\):
- black curve: The default OOK modulation with \(\alpha = 0\) and no security layer
- Red curve: Our demodulation scheme with \(\alpha = 8\) and the default security layer
- Blue curve: Our demodulation scheme with \(\alpha = 8\) and the enhanced security layer (i.e. enhanced first rule)

It can be easily seen that our proposed demodulation scheme in combination with the security layer decreases the attack success probability compared to the conventional solution with default demodulation and no security layer in place, and that the enhanced security layer improves the security even more. One might also notice that the blue curve deviates at \(N_B = 4\). The reason for this is that we did no longer meet the reliability requirements when \(N_B < 5\), and hence had to use different parameters for \(N_B < 5\) in the blue curve. In Appendix C we revisit this numerical example and also consider the effect of noise on the attack success probability.

\[\text{Fig. 5. Attack success probability per bit for the distance } x_1 = 2x_0 \text{ with } k_1 = 1.29, k_2 = \infty, k_3 = 1, k_4 = k_5 = k_6 = \infty, k_7 = 0.7 \text{ and } d = 2.3 \text{ with } N_P = 32, N_B = 5, \text{ and } \alpha = 8 \text{ for different values of } t.\]

\[\text{Fig. 6. Attack success probability for the distance } x_1 = 2x_0 \text{ with } k_1 = 1.29, k_2 = \infty, k_3 = 1, k_4 = k_5 = k_6 = \infty, k_7 = 0.7 \text{ and } d = 2.3 \text{ with } N_P = 32, N_B = 5, \text{ and } \alpha = 8 \text{ for } N_B < 5 \text{ in the blue curve.}\]

**XI. Conclusion and Future Work**

In this paper, we have shown that the security of UWB distance bounding protocols can be improved by modifying the demodulation scheme and adding a security layer to the decoding process. The combination of both allows to extend the communication range of the protocol without reducing reliability nor security. Our proposal is compatible with the 802.15.4F UWB standard, as well as with the FCC regulations. We showed that our solution can achieve strong security bounds for both single-power and multi-power relay attackers by solely relying on pulse masking and the use of a distance commitment. It does not require any other security techniques, such as for example pulse reordering, to achieve these security claims. Therefore, our security solution can be combined with the majority of the UWB distance bounding protocols presented in the literature.
We believe that the concept of applying additional security rules at the decoding process in the receiver is an unexplored though interesting area that requires further research. One could for example explore the integration of machine learning or other decoding constraints into the verification stages of the verifier to model a legitimate prover, and by doing so restrict the degrees of freedom of an attacker at a relatively low performance cost. This will be the topic for future work.

REFERENCES

[1] P. Bahl and V. Padmanabhan, “An in-building RF-based user location and tracking system,” in IEEE INFOCOM, 2000, pp. 775–784.

[2] R. Parker and S. Valaee, “Vehicular node localization using received-signal-strength indicator,” IEEE Transactions on Vehicular Technology, vol. 56, no. 6, pp. 3371–3380, 2007.

[3] M. Li and Y. Lu, “Angle-of-arrival estimation for localization and communication in wireless networks,” in 2008 16th European Signal Processing Conference. IEEE, 2008, pp. 1–5.

[4] C. Zhou and J. D. Griffin, “Accurate phase-based ranging measurements for backscatter rfid tags,” IEEE Antennas and Wireless Propagation Letters, vol. 11, pp. 152–155, 2012.

[5] C. Zhou and J. D. Griffin, “Accurate phase-based ranging measurements for backscatter rfid tags,” IEEE Antennas and Wireless Propagation Letters, vol. 11, pp. 152–155, 2012.

[6] D. Vasish, S. Kumar, and D. Katabi, “Decimeter-level localization with a single wifi access point,” in 13th {USENIX} Symposium on Networked Systems Design and Implementation (\{NSDI\} 16), 2016, pp. 165–176.

[7] S. Lanzisera, D. T. Lin, and K. S. Pister, “Rf time of flight ranging for wireless sensor network localization,” in 2006 international workshop on intelligent solutions in embedded systems. IEEE, 2006, pp. 1–12.

[8] H. Olafsd ãottir, A. Ranganathan, and S. Capkun, “On the security of car-
door entry and start systems in modern cars,” in Proceedings of the Network and Distributed System Security Symposium, NDSS 2011. The Internet Society, 2011.

[9] S. Drimer, S. J. Murdoch et al., “Keep your enemies close: Distance bounding against smartcard relay attacks.” in USENIX security symposium, vol. 312, 2007.

[10] T. Chothia, F. D. Garcia, J. de Ruiter, J. van den Breekel, and M. Thompson, “Relay cost bounding for contactless EMV payments,” in Financial Cryptography and Data Security - 19th International Conference, FC 2015, ser. Lecture Notes in Computer Science, R. Böhme and T. Okamoto, Eds., vol. 8975. Springer, 2015, pp. 189–206.

[11] S. Brands and D. Chaum, “Distance-bounding protocols (extended abstract),” in EUROCRYPT, 1993, pp. 344–359.

[12] G. P. Hancke and M. G. Kuhn, “An rfid distance bounding protocol,” in First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM’05). IEEE, 2005, pp. 67–73.

[13] J. Clulow, G. P. Hancke, M. G. Kuhn, and T. Moore, “So near and yet so far: Distance-bounding attacks in wireless networks,” in European Workshop on Security in Ad-hoc and Sensor Networks. Springer, 2006, pp. 83–97.

[14] M. Poturlalski, M. Flury, P. Papadimitratos, J.-P. Hubaux, and J.-Y. Le Boudec, “The cicada attack: degradation and denial of service in ir ranging,” in 2010 IEEE International Conference on Ultra-Wideband, vol. 2. IEEE, 2010, pp. 1–4.

[15] M. Singh, P. Leu, and S. Capkun, “UWB with pulse reordering: Securing ranging against relay and physical-layer attacks.” in Proceedings of the Network and Distributed System Security Symposium Symposium, NDSS 2017. The Internet Society, 2017.

[16] L. Bussard and W. Bagga, “Distance-bounding proof of knowledge to avoid real-time attacks,” in IFIP International Information Security Conference. Springer, 2005, pp. 223–238.

[17] J. Munilla, A. Ortiz, and A. Peinado, “Distance bounding protocols with void-challenges for rfid,” in Printed handout at the Workshop on RFID Security–RFIDSEC, vol. 6, 2006.

[18] Y.-J. Tu and S. Piramuthu, “Rfid distance bounding protocols,” in First International EURASIP Workshop on RFID Technology, 2007, pp. 67–68.

[19] J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji, “Detecting relay attacks with timing-based protocols,” in Proceedings of the 2nd ACM symposium on Information, computer and communications security, 2007, pp. 204–213.

[20] C. H. Kim, G. Avoine, F. Koeune, F.-X. Standaert, and O. Pereira, “The swiss-knife rfid distance bounding protocol,” in International Conference on Information Security and Cryptology. Springer, 2008, pp. 98–115.

[21] J. Munilla and A. Peinado, “Distance bounding protocols for rfid enhanced by using void-challenges and analysis in noisy channels,” Wireless Communications and Mobile Computing, vol. 8, no. 9, pp. 1227–1232, 2008.

[22] N. O. Tippenhauer and S. Čapkun, “Id-based secure distance bounding and localization,” in European Symposium on Research in Computer Security. Springer, 2009, pp. 621–636.

[23] K. B. Rasmussen and S. Capkun, “Realization of rfid distance bounding,” in USENIX Security Symposium, 2010, pp. 389–402.

[24] A. Ranganathan, N. O. Tippenhauer, B. Škorić, D. Singelée, and S. Čapkun, “Design and implementation of a terrorist fraud resilient distance bounding system,” in European Symposium on Research in Computer Security. Springer, 2012, pp. 415–432.

[25] C. Cremers, K. B. Rasmussen, B. Schmidt, and S. Capkun, “Distance hijacking attacks on distance bounding protocols,” in 2012 IEEE Symposium on Security and Privacy. IEEE, 2012, pp. 113–127.

[26] N. O. Tippenhauer, H. Luecken, M. Kuhn, and S. Čapkun, “Uwb rapid-bit-exchange system for distance bounding,” in Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, 2015, pp. 1–12.

[27] G. Avoine, M. A. Bingöl, I. Bourouan, S. Capkun, G. P. Hancke, S. Kardas, C. H. Kim, C. Lauradoux, B. Martin, J. Munilla, A. Peinado, K. B. Rasmussen, D. Singelée, A. Tchamkerten, R. Trujillo-Russa, and S. Vaudenay, “Security of distance-bounding: A survey,” ACM Computing Surveys, vol. 51, no. 5, pp. 94:1–94:33, 2019. [Online]. Available: https://doi.org/10.1145/3264628

[28] H. Chernoff et al., “A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations,” The Annals of Mathematical Statistics, vol. 23, no. 4, pp. 493–507, 1952.

[29] J. G. Proakis and M. Salehi, Digital communications. McGraw-hill New York, 2001, vol. 4.

APPENDIX

A. Optimal attack strategy against the security layer

In Section [VII-C] we have presented two lemmas that formalize two properties of the optimal attack strategy against our proposed security layer. Below, we will now prove these two lemmas.

Lemma (VIII.1). In the optimal attack strategy, the attacker does not increase the square root of the power of the $N_P-N_B$ pulses higher than $(k_1 + d) \sqrt{|g|}$. Proof. We prove this lemma by contradiction. Let us assume that in the optimal attack strategy, the attacker transmits the $j$-th pulse of $b_l$ with a power for which the square root is higher than $(k_1 + d) \sqrt{|g|}$. There are two possibilities. First, the power of the legitimate transmitted $j$-th pulse of $b_l$ is 0. In this case, since the power of the pulse transmitted by the attacker is higher than $(k_1 + d) \sqrt{|g|}$, there are two possibilities. First, the power of the legitimate transmitted $j$-th pulse of $b_l$ is 0. In this case, we do not abort the protocol and the attacker is lucky. However, for each of the other pulses that belongs to $b_l$, the attacker has only two choices. The first option is, decreasing the square root of power more than the threshold $d \sqrt{|g|}$, such that the transmitted power is less than $k_2^2 |g|^2$. The second option is to continue transmitting a pulse with a power higher than $k_2^2 |g|^2$. For the first choice, the protocol aborts if the legitimate transmitted power is $|g|^2$ since there is at least one pair of pulses (with
value 1) where the difference of the square root of their power is higher than $d\sqrt{\lVert g \rVert}$, causing the protocol to abort due to the second rule. For the second choice, the protocol is aborted when the legitimate transmitted power is 0, since the attacker transmitted a pulse with power higher than $k_1^2 \lVert g \rVert^2$ when a 0 was expected. This violates the first rule. From the discussion above, we can conclude that if the attacker transmits a pulse with a power higher than $(k_1 + d)^2 \lVert g \rVert^2$ for the $j-th$ pulse of $i-th$ bit, the only way to be successful is to guess the transmitted power of all the other pulses that belong to this $i-th$ bit correctly, otherwise the protocol is aborted. This means that the attacker can only be successful with probability $2^{-N_P}$. The attack success probability for a single-power attacker is given in the following equation (see Eq. (23)).

$$
\left(1 - \frac{1}{2} \frac{N_P}{2 + \alpha} + \sum_{f=\frac{N_P}{2} + \alpha + 1}^{N_P} \left(\frac{N_P}{f}\right) \right) \frac{1}{2^{N_P}},
$$

(36)

This success probability is higher than the optimal attack strategy. This is a contradiction since the multi-power attacker is a stronger variant of the single power attacker. This contradiction proves the lemma.

**Lemma (8.2).** If during the attack, the attacker wants to send $i$ pulses $(i \geq 1)$ with a power $P_i \lVert g \rVert^2$ for which the condition $k_1^2 < P_i \leq (k_1 + d)^2$ holds, then the optimal attack strategy is to choose $P_i = (k_1 + d)^2$ for all these $i$ pulses.

**Proof.** Again we prove this lemma by contradiction. Therefore, let us assume that the optimal attack strategy for a specific sequence of legitimate pulses $G$ is denoted by $\Gamma$. We assume that this optimal attacker transmits $i$ pulses with powers $P_1 \lVert g \rVert^2$, $P_2 \lVert g \rVert^2$, ..., and $P_n \lVert g \rVert^2$, where $k_1^2 < P_1 < P_2 < ... < P_n < (k_1 + d)^2$, $n \leq i$; and $N_P - i$ pulses with powers $P_{n+1} \lVert g \rVert^2$, $P_{n+2} \lVert g \rVert^2$, ..., and $P_{N_P} \lVert g \rVert^2$, where $P_{n+1} < P_{n+2} < ... < P_{n+h} < k_1^2$ and $h \leq N_P - i$. Let us now also consider another strategy, let us call it $\Omega$, where the attacker transmits $i$ pulses exactly with power $P_i \lVert g \rVert^2$ equal to $(k_1 + d)^2 \lVert g \rVert^2$, instead of the strategy followed by $\Gamma$. For the rest of $N_P - i$ pulses, the strategy of $\Omega$ is identical to the one of $\Gamma$, i.e., $N_P - i$ pulses with powers $P_{n+1} \lVert g \rVert^2$, $P_{n+2} \lVert g \rVert^2$, ..., and $P_{N_P} \lVert g \rVert^2$ are sent, exactly in the same places as the optimal attacker $\Gamma$. If the the optimal attacker $\Gamma$ is successful for the legitimate sequence $G$, the attack is also successful for the strategy $\Omega$, since every transmitted pulse is the same as the optimal attacker except for the pulses with power $P_{n+1} \lVert g \rVert^2$. These pulses also correspond to legitimate transmitted pulses with power $\lVert g \rVert^2$, and make the advantage for the strategy $\Omega$ higher than the optimal attack strategy $\Gamma$, based on equation (21). This is the case, since the protocol would have aborted due to the first security rule if one of the legitimate transmitted pulses would have been 0. This means that the strategy $\Omega$ is successful in every cases that the optimal attack is successful $\Gamma$. Also, since the advantage of this attack is higher than optimal attack, strategy $\Omega$ outperforms the optimal strategy $\Gamma$ in all the cases we do not abort the protocol. Since it is a contradiction that strategy $\Omega$ can outperform the optimal attack strategy $\Gamma$, this proves the lemma.

**B. Practical realisation of our solution**

In this subsection, we address a few practical considerations on the realisation of our proposed solution.

**a) What happens if the legitimate prover is very close to the prover**

In this case, the power of the pulses received by the verifier might be higher than $\lVert g \rVert^2$. One might wonder if this would cause the protocol to abort, due to the application of the second security layer. Fortunately, this is not the case. The first security rule would not be violated, since the legitimate prover would not send any pulse when a 0 is expected. Also the second security rule would not be violated, as the prover transmits all its 1 pulses with the same power. In conclusion, this does not have an impact on the reliability.

**b) Is there a risk that the power of pulses gets too high due to multi-path (reflection to walls, etc.)**

We argue that there is a low probability that multi-path would decrease the reliability of our solution. First of all, the power of the pulses is only divided between some paths. Second, in UWB communication there are two modes: high rate pulse (HRP), and low rate pulse (LRP). In LRP systems, where the 802.15.4f UWB standard operates, the rate of pulses is low, and as a result pulses do not interfere with each other. Therefore, the probability of receiving a high power pulse when a legitimate pulse of 0 was expected, or having constructive interference between two pulses with value 1 such that the threshold of the second security rule is exceeded, is relatively low. The largest effect on the received power of pulses in UWB LRP systems can be expected due to background noise, and this is taken into account on the design of our solution.

**c) Is there a risk that the power of pulses gets too high due to interference with other wireless systems**

If there are other wireless systems in the vicinity of the prover or verifier, for example another UWB transmitter, there might be a risk that both these systems would interfere. This interference could be significant, as we have no control on the other devices that operate in the same bandwidth. However, we can argue that this holds for any distance bounding protocol that is proposed in the literature and that the problem is hence independent of our proposed solution. If interference could be expected, it is better to redesign the UWB system and ensure that this interference is minimized before the demodulation takes place.

**C. Noise vs attack success probability**

In our security analysis, we always considered the best-case scenario for the attacker, where he can transmit an arbitrary attack sequence and the receiver measures the power of the received pulses exactly as the attacker wants. However, this is not a realistic scenario. In practice, the attacker inserts its transmitted pulses into the wireless channel and has no control on the receiving part. Moreover, the channel between the attacker and the verifier can be expected to be noisy.

We have two broad categories of noise in wireless communications, internal noise – i.e., produced in the electrical
circuits by the receiver such as thermal noise – and external noise – i.e., the source of noise is not the receiver. The most dominant noise factor is typically the internal noise. The internal noise cannot be modified by the attacker since the source is the receiver. Also the external noise is mostly outside the control of the adversary, as external noise is mostly related to the environment, for example noise induced by the sun or the galaxy \[50\]. The attacker can only increase SNR by getting closer to the verifier and/or use higher power. But, this obviously increases the probability of the protocol aborting due to a violation of one of the security rules and has nothing to do with the noise power. So in conclusion, the attacker cannot easily decrease the noise level in the receiver by a high degree. Therefore, to have a successful attack, the attacker has to take into account the noise and a safety margin between the power at which he transmits the pulses and the power thresholds that are used by the verifier during decoding. As have been discussed before in the paper, the success probability of an attacker is reduced when he can only transmit pulses with a lower power.

Let us now illustrate with a numerical example how the attack performance decreases due to a noisy channel. Lets again consider the case where \(x_1 = 2x_0\), with the same setup as used in Section \[X\]. In the noise-free scenario, the attacker sent \(t\) pulses with power \(3^2l_1||g||^2\) in the first step of the attack, and then pulses with power \(0.7^2l_1||g||^2\) in the last step of the attack. This strategy no longer works when there is a noisy channel, as there is a 0.5 probability that the noise increases the power of the received pulses. If the power of the pulses received by the verifier increases, this could result in an abortion of the protocol. The reason is twofold. First, it could result in a violation of the first security rule, because the power of the received pulse could exceed the power threshold when a 0 pulse was expected. Second, the increase in power could cause the square root of the difference between the received power of the pulses to be larger than the threshold \(d\), hence resulting in abortion of the protocol due to the second security rule. To decrease this probability, let us now assume that the attacker decreases the power of the \(t\) pulses in the first step to \(2^2l_1||g||^2\), and transmits the other pulses with power \(0.3^2l_1||g||^2\).

To find the optimal \(t\), we have run a simulation for different \(t\), as can be seen in Figure \[7\] As before, we generated \(10^6\) legitimate pulse sequences in our simulation to compute the attack success probability for each value of \(t\). As can be seen in Figure \[7\] the best attack success probability is 0.0131 and can be obtained with \(t = 6\). Note that the attack success probability shown in Figure \[7\] is only the attack success probability per bit. To have a successful attack, the attacker has to be successful for all \(N_B = 5\) bits, which means that the success probability for successfully performing a relay attack on the distance bounding protocol is equal to \(0.0131^5 \approx 3.85 \times 10^{-10}\). In Figure \[8\] we compare the success probability of the noise-free attack scenario with the attack scenario where noise is considered. One can clearly observe that the attack success probability is reduced due to the noise.