IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2026, No. 1, pp. 500–532. DOI:10.46586/tches.v2026.i1.500-532

FAST: Fast and Accurate Security Testing of HRP UWB Chips

Graciana Aad[1], Giovanni Camurati[1], Matteo Dell'Amico[2] and Srdjan Capkun[1]

[1] ETH Zurich, Zurich, Switzerland, name.surname@inf.ethz.ch
[2] University of Genoa, Genoa, Italy, matteo.dellamico@unige.net

Abstract. High-Rate Pulse (HRP) Ultra-Wide Band (UWB) technology is used for secure distance measurement. It was standardized by IEEE 802.15.4z in 2020 and is implemented in chips widely deployed in consumer devices. However, due to the use of proprietary signal processing algorithms, complex implementations, and subtle physical layer considerations, evaluating the security of such chips against distance-reduction attacks analytically is challenging. In this work, we investigate how to empirically evaluate the security of HRP UWB chips against random-guess attacks that were recently proven practical on real-world chips (GhostPeak). We propose `FAST`, a generic and efficient testing methodology that we use to accurately characterize the security of HRP UWB chips against random-guess distance reduction attacks. `FAST` relies on importance sampling and can accurately estimate very low success rates using a small and practical number of tests.
Using `FAST`, we characterize the security of four HRP UWB receivers having different levels of obscurity (Qorvo DWM3000EVB, PURE, NXP SR040, and NXP SR150) across different chip configurations and attack conditions. `FAST` revealed success rates ranging from 2^{-10} to as low as 2^{-128} using only tens of thousands of test samples.

Keywords: Ultra-Wide Band · Secure Ranging · Ghost Peak

1 Introduction

High Rate Pulse Repetition Frequency (HRP) Ultra-Wide Band (UWB) chips are increasingly common. They are currently deployed in the Apple iPhones for spatial awareness [app], Google's Pixel 6 Pro [Li22] for accurate ranging and spatial orientation, and Samsung smartphones as safe digital car keys [sam]. BMW launched the BMW Digital Key compatible with smartphones on both Android (Google and Samsung) or iOS (Apple) operating systems [noa]. In 2024, Apple announced that its Apple Home application in devices with iOS 18 will support hands-free unlocking for smart locks thanks to the U1 UWB chip [Tuo24]. Coppola et. al [CCA+24] recently proposed to integrate HRP UWB with the Europay Mastercard Visa (EMV) protocol to protect contactless payments from relay attacks.

These applications require HRP UWB chips to measure distances securely, i.e, an attacker should not be able to reduce the measured distance and gain unauthorized access. To prevent such distance-reduction attacks, the IEEE 802.15.4z standard defining HRP UWB introduces an unpredictable field to ranging packets, called Scrambled Timestamp Sequence (STS), used by the receiver to verify the distance measurement. Its purpose is to prevent an attacker from reducing distance by crafting valid messages earlier than they were transmitted by the legitimate devices. While the standard specifies how to securely generate the Scrambled Timestamp Sequence, it does not describe how receivers

Received: 2025-07-15 Accepted: 2025-09-15 Published: 2026-01-16

should verify it during distance measurements. Hence, current chips deploy proprietary algorithms.

These algorithms were shown to be vulnerable to distance reduction attacks [SRZ+21, LCH+22, ACC23] and notably random-guess attacks. Ghost Peak [LCH+22] attack is a random-guess distance-reduction attack that was shown to be practical and efficient, with success rates as high as 4% on Apple U1 chips found in iPhones, AirTags, and HomePods. This attack uses a standard device to reactively inject HRP packets with random STS, which are mistakenly accepted, causing a distance reduction of several meters. This sparked research into countermeasures and better receiver designs [JLJC23, CCA+24, LKZ+24], including a reception algorithm with provable security guarantees proposed by Apple [LKZ+24].

Given the closed nature of distance measurement and in particular of STS verification algorithms, in this paper, we take a different approach to UWB chip security analysis - empirical testing. We propose **F**ast **A**ccurate **S**ecurity **T**esting (`FAST`), the first testing methodology that can be used to characterize the security of real-world ranging devices. We specifically use `FAST` to characterize the security of commercial UWB chips against random-guess distance reduction attacks, represented by the Ghost Peak attacks. `FAST` is useful for evaluators who can interact with the chip but do not know its closed-form security level, for example, security researchers, system integrators, and especially manufacturers interested in testing the compliance of their chips or other chips with which theirs interoperate to defined security levels.

One of the main challenges of empirical testing is obtaining enough successful events to accurately estimate the success rate. In fact, when estimating the success of random-guess attacks, sending random attack signals can take an impractical amount of time to obtain one successful sample. There is an asymmetry between the attacker, who is only interested in obtaining one distance reduction to break access control, and the evaluator, who needs to run a successful attack enough times to estimate the success rate. With a low success rate, the latter might require an impractically high number of tests, or even be infeasible. The problem is even worse when considering that the evaluator might want to systematically analyze different attack parameters and receiver configurations.

To overcome this challenge, in `FAST` we leverage importance sampling [TK10] which ensures a quick convergence to a good estimate. Importance sampling biases the attack toward generating more successful signals and corrects this bias when calculating the success rate. Introducing this bias is possible because the evaluator knows which signal is expected by the legitimate device (i.e., knows the expected STS sequence in the chip that is under evaluation). With this method, the evaluator can accurately estimate the success rate with a limited number of samples and systematically analyze many scenarios. Interesting case studies can be constructed based on the physical characteristics of the attack signal, the environment, and the configuration of the chip. Examples are the amplitude and shape of the attacker's UWB pulses relative to the legitimate ones, the type of wireless channel, and the chip registers that configure the verification algorithm or enable additional consistency checks.

The motivation behind proposing a testing framework for random-guess attacks comes from the emergence of GhostPeak [LCH+22], a random-guess attack proven practical against off-the-shelf chips with a success rate that can be as high as 4%. This shows that this attack, which is easy to implement, makes HRP UWB chips vulnerable to distance-reduction attacks. After validating `FAST` in simulation with a receiver for which we know the ground-truth success rate, we used it to characterize real-world chips, the Qorvo DWM3000EVB, NXP Trimension SR040, and NXP Trimension SR150. Moreover, we implement the PURE receiver proposed in [CCA+24] on the QorvoDWM3000EVB chip and evaluate using `FAST`. We make interesting observations on the role of different attack and receiver parameters. For example, using only a few tens of thousands of samples,

we characterized the attacker's success rate against this chip from 2^{-10} down to 2^{-128}, depending on the configuration of the receiver. In summary, our contributions are:

- We propose `FAST`, the first practical empirical testing methodology that can be used to characterize the security of real-world receivers against random-guess distance-reduction attacks.

- `FAST` leverages well-crafted physical-layer attack signals and importance sampling to accurately estimate the attack success rates when naive testing fails. It can be broadly applied to different types of receivers. It is particularly useful to study HRP UWB receivers because they: i) use algorithms with unknown security levels, ii) are based on proprietary implementations, and iii) are vulnerable to practical random-guess distance-reduction attacks such as Ghost Peak.

- We use `FAST` to evaluate Ghost Peak attacks on a commercial HRP UWB chip, the Qorvo DWM3000EVB on three commercial HRP UWB chips, the Qorvo DWM3000EVB, NXP Trimension SR040, and NXP Trimension SR150, and a proposed receiver, PURE, implemented on top of a Qorvo DWM3000EVB chip. Using a small number of measurements, we can fully characterize the success rate of Ghost Peak attacks in many different scenarios.

The remainder of this paper is organized as follows. In Section 2, we explain relevant concepts about HRP UWB security. In Section 3, we give an overview of our problem, introduce `FAST`, and validate it on a known receiver. We then present the experimental evaluation on real-world Qorvo and NXP chips in Section 4. Then, in Section 5, we discuss the limitations of `FAST` as well as a broader application to a wide range of technologies and attacks. Finally, we discuss related work in Section 6 and conclude in Section 7.

2 Background

2.1 UWB Ranging

Ultra-Wide Band is a wireless technology commonly used for distance measurements thanks to its excellent accuracy and ability to work in challenging environments. In this paper, we focus on the HRP mode defined in the current IEEE 802.15.4z standard. It is largely available in consumer devices like smartphones [app, Li22], and is used for security-critical applications such as Passive Key-Less Entry and Start (PKES) for cars [noa].

In cases where access control is based on the distance measured between an asset and the legitimate user, an attacker able to shorten this measurement may obtain unauthorized access. To prevent an attacker from shortening the distance by transmitting malicious packets before the legitimate devices, a field named Scrambled Timestamp Sequence was introduced in the IEEE 802.15.4z standard. It consists of a sequence of 4096 unpredictable pulses generated by AES in counter mode. The legitimate transmitter and receiver share a secret key used for the generation of the STS, so the legitimate receiver knows what STS sequence to expect. Once the receiver obtains a signal from a transmitter, it checks the similarity between the received STS field and the expected one and, if the similarity is high enough, it accepts the received signal and calculates its time of arrival. The distance between two chips is calculated from the round-trip time of multiple messages, knowing that they travel at the speed of light. Since the attacker does not know the unpredictable 4096-pulse-long STS sequence, it has a low probability of generating an STS similar to the expected one.

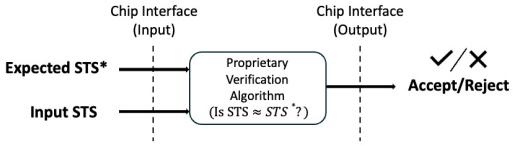


Figure 1: HRP UWB Receiver. The proprietary algorithm in the chip verifies at the physical layer whether the input STS is sufficiently similar to the expected STS* to accept the received packet or not. The security level is unknown since the algorithm is proprietary and/or uses techniques for which a closed-form security expression has not been found.

2.2 Cross-correlation-based Receivers

To find the similarity between two STSs, some receivers' algorithms rely on cross-correlating the two received signals and look for the highest peak of the cross-correlation. However, this peak is not necessarily the first peak that indicates the earliest arrival time of the STS to the receiver. For that, these receivers perform leading-edge detection; they look for the potential early peak (EP) that occured before the highest one, and if this EP passes a threshold preset by the chip manufacturer or sometimes configurable by the evaluator, the received signal is considered legitimate and accepted by the chip. The authors in [SRZ+21] present detailed explanations and analyses about cross-correlation-based receivers.

2.3 Attacks on HRP UWB

Even though HRP UWB aims at securing the integrity of ranging using an unpredictable STS field, research has demonstrated that its verification can be vulnerable to practical distance-reduction attacks [SRZ+21, LCH+22, ACC23]. Singh et al. [SRZ+21] provide the first open security analysis of HRP UWB ranging. They observe that the verification of the STS field is not standardized and likely follows a cross-correlation-based approach for which a closed-form expression of the security level does not exist. Through simulation, they show that such receivers cannot be easily configured to be both reliable in challenging environments and secure against the injection of random pulses. Leu et al. [LCH+22] demonstrate Ghost Peak, a practical distance-reduction attack on Apple U1 chips deployed in iPhones, AirTags, and HomePods. This attack is easy to implement since it consists of sending random STS sequences until succeeding in reducing the distance measured by the receiver, and HRP UWB receivers have been proven to be vulnerable against it. The attack is implemented with an inexpensive off-the-shelf UWB device that reactively injects an HRP UWB packet with random STS on top of the legitimate exchange. The success probability higher than 1 % is significantly lower than the one expected for security applications (e.g., 2^{-48} for mobile payments), and several reductions of several meters of the measured distance can be observed in a short time. Anliker et al. [ACC23] demonstrate Mix-Down, another practical distance-reduction attack on Qorvo DWM3000EVB devices used in Single-Sided Two-Way Ranging (SS-TWR) mode. The attack consists of manipulating clock-offset correction by applying a frequency shift over-the-air with a simple analog circuit, and can only be prevented by using Double-Sided Two-Way Ranging (DS-TWR). Yang et al. [YWZ+24] show practical jamming attacks on UWB ranging using off-the-shelf hardware. They rely on the observation that normalized cross-correlation is not a robust method for packet detection, as the contribution of the legitimate signal to the correlation peak decreases in presence of an interfering signal with higher power.

2.4 Improved Receiver Designs

There exist several proposals to counter Ghost Peak attacks and improve the security of HRP receivers. Joo et al. [JLJC23] suggest securing HRP UWB receivers by splitting the verification of the STS field into multiple chunks and computing various consistency checks. Coppola et al. [CCA+24] secure contactless payments from relay attacks using HRP UWB ranging. They demonstrate how using a high absolute threshold on cross-correlation can achieve both good security and reliability in the specific conditions of mobile payments, where phones are close to the terminals. The security analysis is based on the observation that pulses cannot have an arbitrary power, and each of them has a limited contribution to the total cross-correlation. This is because if the attacker sends pulses with an amplitude exceeding the physical limits of the radio front-end, their value will be clipped to the limit. The performance analysis is based on the measurement of the wireless channel between phones and terminals in typical payment scenarios. Even though this approach is effective for mobile payments, it cannot easily scale to other applications because, in the general case, a high threshold results in impractically high false rejection rates. Luo et al. [LKZ+24] propose a novel HRP UWB receiver with a proven security level against an arbitrary attacker. This design can abandon the cross-correlation approach and its limitations, thanks to the Signal-to-Noise Ratio (SNR) increase provided by the secure cancellation of multi-path interference in the time domain.

2.5 Ghost Peak

Ghost Peak [LCH+22] is the first practical distance reduction attack against HRP UWB distance measurement systems. It is done without any knowledge about the STS sequence shared between the sender and the receiver and succeeds up to 4% of the time. It consists of injecting a peak strong enough before the legitimate signal's peak and getting it accepted by the receiver. This will lead the receiver to calculate a distance smaller than the actual one, which can lead to serious security implications depending on the applications, e.g., opening a car without the knowledge of its owner.

3 FAST

Distance-reduction and position-spoofing attacks on ranging and positioning technologies often involve a guessing component because signals contain unpredictable elements. In this paper, we propose **F**ast **A**ccurate **S**ecurity **T**esting (`FAST`) as an efficient empirical testing methodology that can quickly and accurately estimate the success rate of fully random-guess attacks on a wide range of receivers. Here, to illustrate `FAST`, we consider Ghost Peak attacks, which are proven practical distance reduction attacks based solely on random guessing, on an HRP UWB receiver. An HRP UWB receiver takes as input a transmitted STS sequence and compares it to an expected STS^* with an unpredictable sequence of pulses only known to the legitimate transmitter and receiver. Then, it accepts or rejects it based on the proprietary verification algorithm of the chip that sets an acceptance condition depending on the level of similarity between the two STSs. The goal of the attacker, who does not know STS^*, is to cause a distance reduction by injecting a random STS, hoping that it is still accepted by the receiver at an earlier time than the legitimate signal. The receiver has a certain tolerance in accepting the incoming signals because channel effects can cause distortions and bit flips in natural conditions. Hence, the success rate SR of the attack might be non-negligible. The goal of `FAST` is to efficiently provide an accurate estimation $\widehat{SR}$ of the success rate SR, even when the algorithm, implementation, and security level of the receiver are unknown, as is the case for currently available chips. It could also be applied to future proposals for better receivers, should they be deployed on real devices.

Given the proprietary design of the receiver, we find SR using empirical testing by sending different STS sequences to the receiver. Each STS sequence is accepted or rejected based on an unknown algorithm. We then calculate the success rate of the sent STS sequences, as the ratio of the number of STS sequences accepted by the receiver to the total number of tested STS sequences. Despite the commercial receivers using proprietary algorithms, we know that their goal is to accept only STS sequences that are similar to a template STS, called STS^*, shared by the legitimate transmitter and receiver.

The first and most intuitive method that one could consider is naive testing, which relies on two options:

- Sending a few random STS sequences to the receiver without biasing the choice, which means that all the STS sequences have an equal probability of being drawn. Since successful STS sequences are rare, randomly sampling STS sequences will also rarely take into account successful ones. Therefore, this method might miss these sequences and give us $\widehat{SR} = 0$, creating a false sense of security.
- Sending many random STS sequences with the hope of collecting enough successes to compute the average SR. Since successful STS sequences are rare and many successes are needed, this would require a very large number of measurements.

The second option is similar to running the attack itself and waiting to get enough successful samples. So, the main difference between naive testing and the attacker's approach is that in naive testing, the goal is to gather as many successful STS sequences as possible, whereas for the attacker, the main goal is to send STS sequences to the receiver until one gets accepted.

On the contrary, `FAST` leverages the evaluator's knowledge of the expected STS^* to apply a better testing strategy. `FAST` makes successful samples, which are originally rare, more probable by biasing the sampling strategy towards STS sequences more similar to the correct STS^* value. We compensate for this bias by weighting the samples as discussed in Section 3.3, and obtain an accurate estimate of the success rate even with few samples. This approach can be repeated for different choices of physical-layer attack signal, wireless environment, and receiver configuration. Our biased sampling strategy is based on a beta-binomial distribution whose parameters, a and b, are tuned via a previous calibration phase on a small pilot set of measurements. This allows us to concentrate on the important regions of the sampling space based on the similarity to STS^*, which are those lying on the frontier between successful and failed attacks.

In the rest of this section, we will give an overview of the receiver assumptions, then we will discuss the naive testing approach and its limitations, which leads us to present `FAST` as an accurate and fast testing method. Finally, before moving to the experimental evaluation on a commercial receiver, we will validate our testing approach by simulation on a receiver with a known success rate.

3.1 Receiver Assumptions and Successful Attack

For a receiver to accept an STS sequence, its similarity with the STS^* template should be high, i.e., they should share a large number n_c of bits in common.

However, when a signal travels through a channel, it might experience a 180° phase rotation that will cause the bits to flip, i.e., bits 1 become 0 and bits 0 become 1. So, under natural conditions, there could be an ambiguity between a message and its bitwise negation (i.e., all bits are flipped). This means that an STS is likely to be accepted if it is either highly similar or highly dissimilar to STS^*. This phenomenon is observed on a real chip and is consistent with assumptions made in papers dealing with cross-correlation-based receivers [LCH+22, SRZ+21, CCA+24]. To describe the receiver, we introduce a success function $f(n_c)$ that returns 1 if an STS with n_c pulses in common to STS^* is accepted,

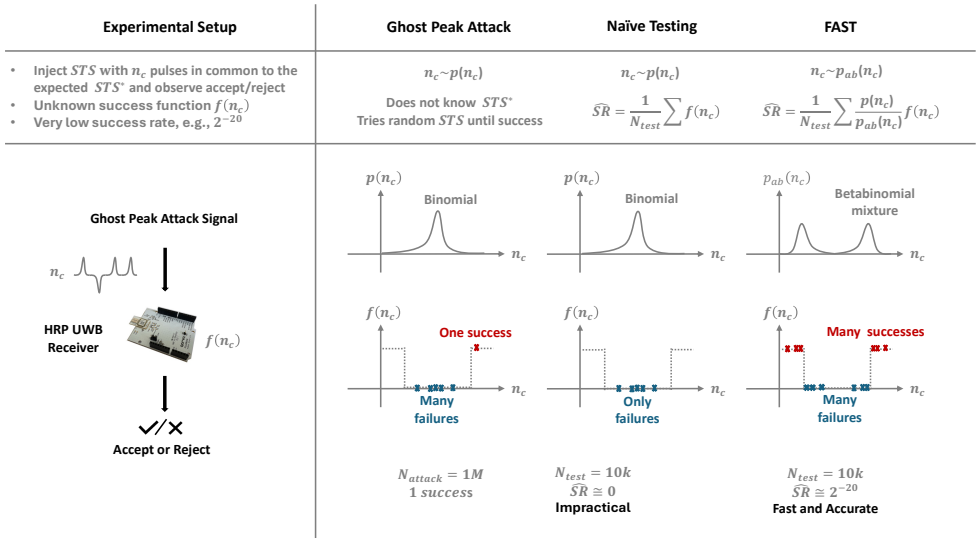


Figure 2: Overview. *Ghost Peak* [LCH+22]: The adversary injects random STSs in the chip until one is accepted earlier than the legitimate one, causing a distance reduction. The probability that a random STS is similar enough to the expected one to be accepted is low (the number n_c of pulses in common is binomial). *Naive Testing:* Observing enough successes to obtain a good non-zero estimate of the success rate requires running the attack for a long time. *FAST*: By biasing the sampling strategy such that successes are more likely (importance sampling), `FAST` obtains good estimates using significantly fewer samples.

and 0 otherwise. While f is unknown due to the proprietary nature of the receiver, we can observe its output by transmitting an STS and observing if is accepted or rejected. The function f captures the full process of receiving a physical-layer signal and validating it based on its similarity to the expected STS^*. We are only interested in observing its "accept/reject" output from real chips and not in modeling it. Hence, for clarity, we express it as a simple function of the number of correct pulses n_c. In Section 4 we show how we further control for attack parameters and receiver configurations that might influence the success rate.

To achieve a distance reduction attack, the attacker aims to send an STS sequence earlier in time and fool the receiver into accepting it as if it were legitimate. In this paper, we assume the Ghost Peak attack, a random guessing attack where an attacker exhaustively samples random STS sequences until one is accepted. The attacker cannot predict the STS sequence because it is securely generated by AES in counter mode and made of random independent pulses (see Assumption 1 by Luo et al. [LKZ+24]). Since the attacker samples STS independently from STS^*, the probability of sampling an STS with n_c bits in common to STS^* follows a binomial distribution. Successful STS sequences with very high or very low n_c are extremely rare, and there is instead a high probability of sampling STS sequences that are unsuccessful. This is illustrated in Figure 2.

3.2 Naive Security Testing

Naive testing consists of randomly sampling STS sequences, transmitting them to the chips, and counting how many succeed over the total. More formally, the estimation of the

success rate given by naive sampling is:

$$\widehat{SR} = \frac{1}{N}\sum_{i=1}^{N} f(n_{c_i}) \qquad n_{c_i} \sim p(n_{c_i}) \tag{1}$$

where n_{c_i} is the number of bits in common between a sample STS_i and the expected sequence STS^*, $p(n_c)$ is the probability of sampling an STS having n_c, and $f(n_c)$ is the output of the chip; 1 for acceptance and 0 for rejection.

The main problem of this method is that, as depicted in Figure 2, it is most probable to sample STS sequences that get rejected by the receiver, which results in $\widehat{SR} = 0$. This is because the probability of sampling an STS having a specific n_c follows a binomial distribution that is maximal at $\frac{4096}{2}$, and becomes smaller for very high and very low n_c that corresponds to the successful STS sequences. Therefore, this approach *fails to produce an accurate estimate* and results in a false sense of security.

To sample more successful STS sequences, one might think of sending STS sequences to the receiver continuously until enough successes are obtained and then estimate the SR. However, this approach takes a lot of time, especially when the true probability of success is low. Consequently, this method *fails to perform fast testing*.

We conclude that naive testing is not a practical approach for quick and accurate empirical testing. We observe that *the root problem is that sampling a successful STS is a rare event* for the evaluator as much as for the attacker. The following subsection shows how to leverage the evaluator's knowledge of the expected STS^* to overcome this problem.

3.3 Fast and Accurate Security Testing

Since successful samples are rare, drawing them is also rare, leading to inaccurate SR estimations. Our goal is to make successful samples more likely, using importance sampling.

3.3.1 Importance Sampling

Importance sampling is a family of Monte Carlo methods used to estimate the expectation of functions of random variables. It is beneficial in scientific, engineering, and financial applications requiring assessing the probability of rare events (e.g., false detection in radars). Biasing sampling toward the rare cases allows for collecting enough rare events to provide good estimates of their probability in a reasonable time. Importance sampling theory and applications were explored in an extensive corpus of specialized literature; we refer to Srinivasan [Sri14], Biondini [Bio15], and Owen [Owe13] for an introduction.

We apply importance sampling to our problem where successful STS sequences are the "rare event". We bias our sampling distribution towards regions having both successful and unsuccessful samples, and then compensate for this bias, as we will see in the rest of this section. To perform fast and accurate testing, we must appropriately choose this bias to ensure a fast convergence to the real value of SR. In other words, we need to get $\widehat{SR} \approx SR$ and make its variance as small as possible.

3.3.2 `FAST`'s Success Rate Estimator

`FAST`'s approach works as follows. When selecting which STS to transmit, instead of sampling n_c according to its original probability $p(n_c)$, we sample according to a new sampling probability distribution $p_{ab}(n_c)$ of our choice, where a and b are two tuning parameters. We first show how to correct for this bias and still obtain an estimate of the true success rate, and then explain how to select p_{ab} to optimize convergence. Intuitively, if a successful n_c is sampled because $p_{ab}(n_c)$ is higher than $p(n_c)$, then this success should

count less in the computation of the success rate. In particular, it should be weighted by the ratio between $p(n_c)$ and $p_{ab}(n_c)$. The estimate of the success rate becomes

$$\widehat{SR} = \frac{1}{N}\sum_{i=1}^{N}\frac{p(n_{c_i})}{p_{ab}(n_{c_i})}f(n_{c_i}) \qquad n_{c_i} \sim p_{ab}(n_{c_i}) \tag{2}$$

where the only differences with the naive estimate in Equation 1 are the different sampling distribution and the corresponding weighting factor. It can be proven from the theory of importance sampling that, with a sufficient number of samples, this estimate of the success rate converges to the correct value [Bio15, Sri14]. The only condition is that the new sampling probability cannot be zero when there is a non-zero probability of sampling a successful sample according to the original distribution.

3.3.3 `FAST`'s Confidence Interval

To express the accuracy of the success rate estimate $\widehat{SR}$ provided by `FAST`, we can compute a confidence interval. The smaller the confidence interval, the lower the uncertainty we have around our knowledge of the value of $\widehat{SR}$, the better we can consider our result. Since $\widehat{SR}$ is a sum of many independent random variables, we can assume that it is normally distributed, compute its standard deviation $\hat{\sigma}_{\widehat{SR}}$, and write the confidence interval as:

$$CI = \left[\widehat{SR} - z\,\hat{\sigma}_{\widehat{SR}},\ \widehat{SR} + z\,\hat{\sigma}_{\widehat{SR}}\right] \tag{3}$$

where z is a multiplier chosen to obtain the desired confidence level (i.e., 90 %).

The standard deviation is a function of both the variance of the weighted samples and their number:

$$\hat{\sigma}_{\widehat{SR}} = \sqrt{\frac{1}{N}\widehat{Var}_{p_{ab}}\left[f(n_c)\frac{p(n_c)}{p_{ab}(n_c)}\right]} \tag{4}$$

By choosing a good sampling probability p_{ab} biased towards more successful samples, we can minimize the variance and obtain a tight confidence interval using a few samples.

3.3.4 Adaptive Sampling Strategy

The main challenge in the application of importance sampling to real-world applications is crafting an effective sampling probability distribution, as it depends on knowing the importance regions of the target function f that we want to characterize in the first place. The choice involves both choosing an appropriate parametrized distribution, which requires domain knowledge about the target, and estimating the optimal values for the parameters, which requires an adaptive strategy with an efficient calibration phase before testing. Compared to purely mathematical problems, in `FAST` this is made even more challenging by the presence of noise and time required to test each sample against a real chip.

To bias our sampling toward the important regions, i.e., regions having successful samples along with the unsuccessful ones, we use a beta-binomial distribution, since it belongs to the same family as the original binomial distribution of n_c, but with parameters (a, b) that we can tune to minimize the variance of our estimation. The betabinomial is a distribution in which samples are drawn from a binomial whose probability parameter was first sampled from a beta distribution with parameters a, b. This gives us fine control over how we distribute the samples over the important region, both in terms of how biased towards a higher mean they are, and how spread around the mean they are. The latter is particularly convenient in real-world experiments with noisy signals, where there might be

Algorithm 1 Pseudo-code for FAST with Adaptive Sampling for Variance Minimization

n_c: number of bits in common between attack and legitimate 4096-bit STS.
$f(n_c)$: Function returning 1 iff the chip accepted an attack signal with n_c.
$p(n_c)$: Original binomial distribution for n_c, centered in 4096/2.
$p_u(n_c)$: Uniform distribution from 0 to 4096 used for the pilot samples.
$p^+_{ab}(n_c)$: Betabinomial distribution with parameters a, b, covering 4096/2 to 4096.
$p^-_{ab}(n_c)$: Distribution symmetrical to $p^+_{ab}(n_c)$ covering 0 to $(4096/2) - 1$.
r: Parameter controlling regions to sample: positive only (1), both (0.5).
$p_{ab}(n_c)$: Mixture $(1 - r) \cdot p^-_{ab}(n_c) + r \cdot p^+_{ab}(n_c)$ used for testing.
N_{cal}, N_{test}: Number of samples for calibration and testing.
$\widehat{SR}$, CI: Estimated success rate and confidence interval.

function FAST(N_{cal}, N_{test})
  Step 1: Calibration pilot sampling.
  Sample N_{cal} samples, $n_{c_1}, n_{c_2}, \ldots, n_{c_{N_{cal}}}$, from the uniform distribution $p_u(n_c)$.

  Step 2: Identify which regions require sampling to choose the appropriate mixture.
  if $f(n_c) = 0$ for all n_c samples in $[0, 4096/2]$ **then** $r = 1$ **else** $r = 0.5$

  Step 3: Estimate $\hat{A}_{ab}$ using importance sampling.
  for each possible candidate pair of parameters (a, b) in a desired range **do**
    Define the beta-binomial mixture $p_{ab}(n_c) = (1 - r) \cdot p^-_{ab}(n_c) + r \cdot p^+_{ab}(n_c)$.
    Initialize $\hat{A}_{ab} = 0$.
    for each sample n_{c_i} from the pilot samples **do**
      Compute the importance weight: $w_i = \left(\frac{p(n_{c_i})}{p_{ab}(n_{c_i})}\right)^2 \frac{p_{ab}(n_{c_i})}{p_u(n_{c_i})}$.
      Update the estimate: $\hat{A}_{ab} \leftarrow \hat{A}_{ab} + f(n_{c_i}) \cdot w_i$.
    Normalize the estimate: $\hat{A}_{ab} \leftarrow \frac{1}{N} \cdot \hat{A}_{ab}$.
    Store the pair $(\hat{A}_{ab}, (a, b))$.

  Step 4: Find the optimal parameters (a, b) corresponding to lowest stored $\hat{A}_{ab}$.

  Step 5: Testing phase sampling with optimal r, a, b from previous steps.
  Define the optimal beta-binomial mixture $p_{ab}(n_c) = (1 - r) \cdot p^-_{ab}(n_c) + r \cdot p^+_{ab}(n_c)$.
  Sample N_{test} samples, $n_{c_1}, n_{c_2}, \ldots, n_{c_{N_{test}}}$, from $p_{ab}(n_c)$.

  Step 6: Estimate success rate with importance sampling.
  Initialize $\widehat{SR} = 0$.
  for each sample n_{c_i} from the test samples **do**
    Compute the importance weight: $w_i = \frac{p(n_{c_i})}{p_{ab}(n_{c_i})}$.
    Update the estimate: $\widehat{SR} \leftarrow \widehat{SR} + f(n_{c_i}) \cdot w_i$.
  Normalize the estimate: $\widehat{SR} \leftarrow \frac{1}{N} \cdot \widehat{SR}$.

  Step 7: Estimate the confidence interval.
  Estimate the variance $\widehat{Var}_{p_{ab}}$ the weighted samples $f(n_{c_i}) \cdot w_i$.
  Estimate the standard deviation of the $\widehat{SR}$ as $\sigma_{\widehat{SR}} = \sqrt{\widehat{Var}_{p_{ab}}/N_{test}}$.
  Estimate $CI = \left[\widehat{SR} - z\ \hat{\sigma}_{\widehat{SR}},\ \widehat{SR} + z\ \hat{\sigma}_{\widehat{SR}}\right]$ with desired z.
  Return $(\widehat{SR}, CI)$.

a large transition region between values of n_c that always fail and values of n_c that always succeed. In contrast, simply biasing the probability parameter of a binomial distribution would not give us this control and would result in sampling very thin regions for high values of n_c. Because both high and low values of n_c are likely to be accepted, we actually use a mixture of two symmetric beta-binomial distributions, one leaning towards the high values of n_c, and one leaning towards the low values. Since high values of n_c are most likely to be accepted, we bias this distribution towards the high values of n_c. However, often the chips take into account possible phase rotations on the channel and accept STSs close to having all bits wrong (low n_c), that is, the chip behaves symmetrically around 4096/2.

In this case, we are biased towards both high and low values using a mixture of two symmetrical distributions. We design it as follows. To cover high values of n_c, we use a betabinomial distribution over the range 4096/2 to 4096 controlled by parameters (a, b). We denote the probability of sampling from this distribution as $p^{+}_{ab}(n_c)$. To cover low values of n_c, we use a symmetrical betabinomial distribution covering values from 0 to $(4096/2) - 1$. We denote the probability of sampling from this distribution as $p^{-}_{ab}(n_c)$. We then define the probability of sampling n_c as $p_{ab}(n_c) = (1 - r) \cdot p^{-}_{ab}(n_c) + r \cdot p^{+}_{ab}(n_c)$ where r is a parameter that lets us control which regions we want to sample. Specifically, with $r = 1$ we bias only towards the region with high n_c, whereas with $r = 0.5$ we bias towards high and low n_c equally.

We want to find a, b, and r that minimise the following variance:

$$\widehat{Var}_{p_{ab}}\left[\frac{f(n_c)p(n_c)}{p_{ab}(n_c)}\right] = \underbrace{\frac{1}{N}\sum\left[\frac{f(n_c)p(n_c)}{p_{ab}(n_c)}\right]^2}_{\widehat{\mathbf{A}}_{\mathbf{ab}}} - \widehat{SR}^2 \qquad n_c \sim p_{ab}(n_c) \tag{5}$$

To minimize the variance, after fixing $\widehat{SR} \approx SR$, we need to minimize the term $\widehat{A}_{ab}$ highlighted in Equation 5. The a, b, and r parameters are chosen during a short calibration phase on a small number of pilot samples. Repeating the pilot phase for different values of a, b, until good ones are found would be highly impractical even if the pilot size is small. Instead, we first sample the calibration pilot using a uniform distribution $p_u(n_c)$. We can easily choose the parameter r which controls the regions covered by the betabinomial mixture by observing whether or not a successful sample with low n_c exists. Then, we optimize the parameters a and b. We can use the importance sampling formula to estimate $\widehat{A}_{ab}$ for different values of a and b without actually resampling. In other words, we first sample uniformly and then use those samples to estimate what would have happened if we had sampled with a certain p_{ab}, without actually doing it. Starting from the uniform distribution $p_u(n_c)$ instead of the original binomial $p(n_c)$ guarantees that we broadly cover the n_c range and have some samples in the interesting regions (i.e., those where n_c succeeds). Hence, we avoid invalid corner cases in the estimates, specifically the case in which the variance is zero because no sample succeeded. The process can be formalized as follows:

$$\widehat{A}_{ab} = \frac{1}{N}\sum\left[f(n_c)\frac{p(n_c)}{p_{ab}(n_c)}\right]^2 \frac{p_{ab}(n_c)}{p_u(n_c)} \quad \forall a, b \qquad n_c \sim p_u(n_c) \tag{6}$$

$$a, b = \arg\min_{a,b}\left(\hat{A}_{ab}\right) \tag{7}$$

where N samples are drawn only once from $p_u(n_c)$. Once a and b are found, we can proceed with the testing phase during which we sample following the beta-binomial distribution. Algorithm 1 summarizes notation and provides the pseudo-code description

Table 1: Simulation. For different threshold values (Equation 3.4), the success rate $\widehat{SR}$ found by `FAST` (1000 samples for calibration and 3000 for testing) is close to the ground truth SR, which falls in the confidence interval ($\widehat{SR}_{lb}$ to $\widehat{SR}_{ub}$).

Threshold	N_{cal}	N_{test}	SR	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
81	1000	3000	$2^{-2.3}$	$2^{-2.3}$	$2^{-2.4}$	$2^{-2.1}$
122	1000	3000	$2^{-4.2}$	$2^{-4.2}$	$2^{-4.3}$	$2^{-4.1}$
204	1000	3000	$2^{-9.5}$	$2^{-9.5}$	$2^{-9.7}$	$2^{-9.4}$
368	1000	3000	$2^{-26.9}$	$2^{-26.9}$	$2^{-27.2}$	$2^{-26.7}$
450	1000	3000	$2^{-39.1}$	$2^{-39.1}$	$2^{-39.4}$	$2^{-38.8}$

of the entire `FAST` algorithm, from parameter estimation during the calibration phase to success rate estimation during the testing phase. For simplicity, and to highlight that minimization does not require resampling at each step, in this pseudocode, we represented the minimization of A_{ab} as a simple for loop over candidate values of a and b. In our (open-source) implementation, we use the Nedler-Mead minimizer from scipy.optimize with initial parameters (a=1,b=1). With these initial values, the initial p_{ab} is equivalent to the uniform distribution p_u used to take the pilot samples. We empirically found this to be a good starting point for the minimization. A comprehensive visualization can be found in Appendix A.

3.3.5 Additional Considerations

The STS is a sequence of random independent pulses generated by AES in counter mode [LKZ+24]. Since an attacker that knows the initialization vector and randomly picks the secret key could recover the STS with a probability no lower than 2^{-128}, we can safely assume 2^{-128} as a lower bound for the success rate estimated by `FAST`. For example, we can clip the results to 2^{-128} if $\widehat{SR} < 2^{-128}$. This is a corner case because we are mostly interested in success rates down to 2^{-40}, where attacks are a threat. In Section 4 we will explain how our implementation of `FAST` controls several parameters that might influence the success rate (e.g., attacker and receiver configuration) and minimises the number and effect of remaining uncontrolled variables (e.g., thermal noise, temperature).

3.4 Validation on a simple example

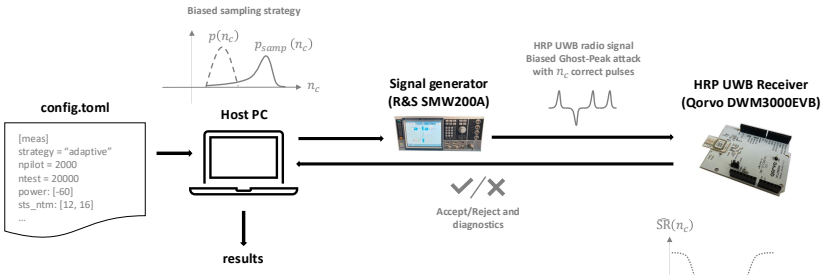
To validate the efficiency and accuracy of `FAST`, we consider the receiver proposed in [CCA+24], for which we have the following analytical formula of the success function:

$$|2n_c - N_p| > threshold \tag{8}$$

where the *threshold* is absolute. The ground truth success rate can be computed analytically [CCA+24, Eq. 3]. We estimate $\widehat{SR}$ using `FAST` and compare the result to the ground truth SR. Results in Table 1 show that, for different values of the threshold, the estimate $\widehat{SR}$ is very close to the expected SR. Moreover, the real SR falls between $\widehat{SR}_{lb}$ and $\widehat{SR}_{ub}$. This convergence requires only 3000 samples for an SR of 2^{-40}.

4 Experimental Evaluation

In Section 3.4, we have validated `FAST` by simulating a simple receiver with a known attack success rate. In this section, we leverage `FAST` to analyze real-world HRP UWB chips: a Qorvo DWM3000 [Qorc], the PURE [CCA+24] receiver implemented on a Qorvo DWM3000, an NXP Trimension SR040 [NXPa], and an NXP Trimension SR150 [NXPb]. We design several case studies to both showcase the capabilities of `FAST` and provide interesting insights into the behavior of Ghost Peak attacks against the chips.



Power (dBm)	Noise Threshold Multiplier	Number of samples	$\widehat{SR}_{LB}$	$\widehat{SR}$	$\widehat{SR}_{UB}$
-60	12	10000	2^-10.1	2^-10.3	2^-9.9
-60	16	10000	2^-14.8	2^-15.3	2^-14.5

Figure 3: Experimental Setup for Qorvo DWM3000EVB and PURE. We use FAST to estimate the success rate of different attacks on different HRP UWB transceivers. Attack signals are generated by the host computer and transmitted with an R&S SMW200A signal generator connected to the victim device via cable. The victim returns whether the signal was accepted or not, and some diagnostics. The host estimates the success rate (and a confidence interval) by counting the number of successes, appropriately weighted according to the importance-based sampling strategy that privileges successful inputs. The analysis is repeated for different scenarios specified in the configuration, for example, for different values of the threshold registers. Low success rates (e.g., 2^{-15}) can be quickly and accurately estimated with a small number of samples (e.g., 10000). The experimental setup for NXP SR150 and SR040 chips is shown in Figure 6.

Motivation: With these receivers, we demonstrate three main usage scenarios for `FAST`. First, the Qorvo chip has a proprietary verification algorithm, but the development kit provides in-depth interactions, configuration, and monitoring. This case is similar to that of a system integrator testing a chip before adding it to a more complex product. Second, the PURE receiver is a state-of-the-art proposal for secure verification with solid security arguments that can be implemented on existing cross-correlation-based chips. This case is similar to that of a designer evaluating the concrete implementation of their proposal on real hardware. Finally, for the NXP chips, we analyze a full two-way ranging demo, where we can only observe the distance reported and the over-the-air packets. This case represents that of a security analyst testing a proprietary ranging application, without low-level access to the chips.

Experimental Setup: Figure 3 depicts an overview of our experimental setup for the Qorvo DWM3000EVB and PURE receivers, whereas Figure 6 shows the setup for the NXP chips. We describe the different setups in detail in each subsection of this section. In these setups, we have an R&S SMW200A signal generator to inject attack signals into different receivers during distance measurements.

Ghost Peak: The attacker transmits its signal that is a standard HRP UWB packet that could be transmitted by a standard device, but with random STS as in [LCH+22] (we leave other strategies as future work). The attack and legitimate signals, if any, are added and transmitted at a specified power in dBm. The shape of the UWB pulses follows the Minimum-Phase Square Root Raised Cosine specified in the standard.

Sampling Strategy and Success Rate Estimation: We implement the sampling strategy and success rate estimation explained in Section 3.3, and, for comparison, the naive approach described in Section 3.2. We report the success rate estimate $\widehat{SR}$ and

the upper and lower bounds of the 90 % confidence interval ($\widehat{SR}_{lb}$ to $\widehat{SR}_{ub}$). We can also transmit signals that always have all correct pulses. `FAST` explicitly controls various parameters (e.g., attack power) that might influence the success rate. At each iteration of an experiment, their value is picked from a desired range. `FAST` can then estimate the aggregate success rate, or the success rate conditional on the controlled variables.

4.1 QorvoDWM3000EVB

4.1.1 Setup

`FAST` uses an adaptive importance sampling strategy and an R&S SMW200A signal generator to inject attack signals in the QorvoDWM3000EVB chip. Attack and receiver parameters are specified in a configuration file. Acceptance and rejection outcomes, collected from the chip via a debug monitor and other diagnostics, are then used to estimate the success rate (and a confidence interval) for each choice of the parameters. The victim receiver is directly connected to the signal generator via a coaxial cable, and the wireless signal corresponding to the full Ghost Peak attack scenario is completely generated in software by `FAST` before being fed to the chip. This approach maximizes the repeatability of the experiments and facilitates the systematic study of the factors that impact the success rate.

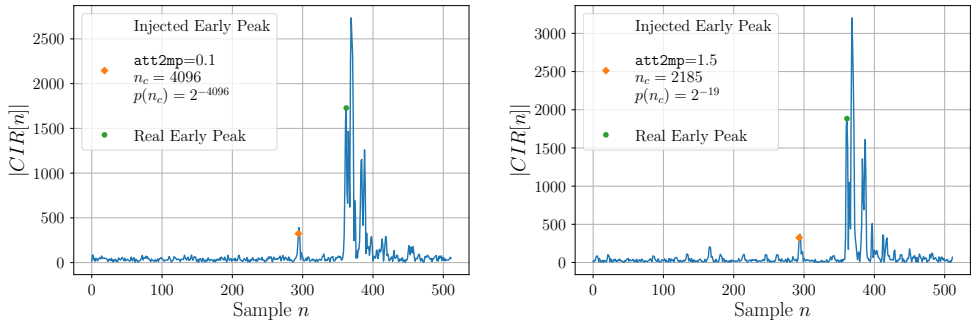
4.1.2 Ghost Peak and Attack Success

The attacker transmits either on top of a legitimate signal [LCH+22], or alone [CCA+24]. The legitimate signal is fed through a desired channel. Channels are generated in Matlab following the IEEE 802.15.4a channel models [MBC+04] as [SRZ+21]. They include several scenarios (e.g., residential, office, outdoor, industrial), each in Line-of-Sight (LoS) or Non-Line-of-Sight (NLoS) conditions. The attack signal is transmitted at a specified time before the legitimate one. The maximum amplitude of the attack signal is chosen as a specific ratio of the maximum amplitude of the legitimate signal. An attack is considered successful if all of the following three conditions are met. First, the packet is correctly received without errors or timeouts. Second, the chip has selected the Time of Arrival (ToA) corresponding to the attack signal (when the legitimate transmitter is also present, the early peak of the legitimate channel serves as reference). Third, the STS quality indicator, read with the `dwt_readstsquality` API function [Qorb], is positive. In addition, `FAST` can be configured to check the STS-related warning flags in the STS Sequence Time of Arrival Status Indicator (`STS_TOAST`) [Qorb, Qora].

4.1.3 Chip Description and Configuration

The Qorvo DWM3000EVB (v1.4) [Qorc], driven by Nordic Semiconductor nRF52DK [Nor], is an excellent target to demonstrate `FAST`. It is well documented [Qora, Qorb], and firmware to interact with it is available [Git]. The proprietary ToA estimation and verification algorithm, which is not part of the firmware, can be tuned by three configuration registers. To allow the firmware to decide whether a packet should be accepted, the chip provides an STS quality indicator and additional warning registers. The chip also provides a significant number of diagnostics, including the full Channel Impulse Response (CIR), which are very useful to analyze the results.

The chip is set to continuously receive HRP UWB packets consisting of a preamble followed by a 4096-pulse STS on channel 5. At each iteration, the expected STS is specified by setting the key, iv, and counter of the AES generator. The user manual [Qora] mentions the following three registers that configure the ToA verification algorithm. The STS Noise Threshold Multiplier (`STS_NTM`) specifies how many times higher than the noise level is the threshold for accepting early peaks in the CIR. The threshold cannot be smaller than



(a) All correct pulses, low amplitude.

(b) Few correct pulses, high amplitude.

Figure 4: Visualization of a Ghost Peak attack. We inject an early copy (marked in orange) ≈ 67 ns before the legitimate copy (marked in green). The relative amplitude between attack and legitimate signals is denoted by `att2mp`. Both a copy with all correct pulses but low amplitude (a) and a copy with fewer correct pulses but higher amplitude (b) generate a similar peak, accepted by the device. The similarity between the two CIRs shows that distinguishing attack signals from attenuated legitimate signals based on cross-correlation is hard.

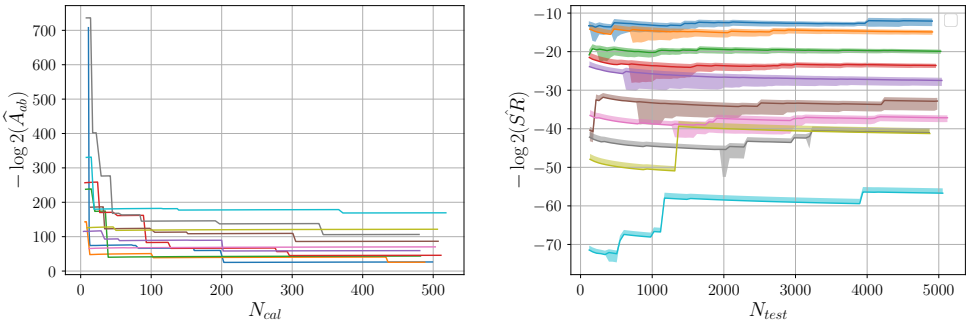
the noise peak multiplied by the STS Peak Multiplier (`STS_PMULT`). If set to 0, the early peak can be in the window used for noise estimation. In addition, the threshold cannot be smaller than the STS Minimum Noise Threshold (`STS_MNTH`). `FAST` uses either their default value or one in their allowed range. The DWM3000 chip is controlled by firmware running on the Nordic Semiconductor nRF52DK. The Python implementation of `FAST` interacts with it via a monitor based on Avatar^2 [MNFB18]. Our current implementation requires approximately 1.5 s per measurement for both naive and `FAST`.

4.1.4 Preliminary Analysis of the Chip

We analyze the Channel Impulse Response estimated by the chip in different conditions. The CIR is an estimate of the channel paths, used by the receiver to identify the earliest arrival time of the signal and compute the distance [SRZ+21]. The experiment consists of transmitting an early copy of the signal around 67 ns earlier than the legitimate signal. The first time, shown in Figure 4a, we send all 4096 correct pulses at 10 % the amplitude of the main copy of the legitimate signal. The second time, shown in Figure 4b, only 2185 of the pulses are correct, but the amplitude is 150 % higher than the legitimate signal. The two produce a similarly small early peak that is accepted by the device, resulting in a shorter distance. The two CIRs are similar, apart from a higher noise floor in the attack case. It is then hard for the receiver to use cross-correlation to distinguish between honest signals (i.e., with correct pulses, but low power) and dishonest signals (i.e., with few correct pulses resulting from random guessing, but higher power).

4.1.5 Backsearch Window

In a multi-path channel, several copies of the signal arrive at the receiver at different times and with different amplitudes. The chip first identifies the strongest copy and then performs a backsearch to find the earliest one [SRZ+21]. Before testing Ghost Peak, we study the backsearch window size. We take 5000 measurements with random channels. For each one, we inject an early copy with 4096 correct pulses and 10 % the amplitude of the main copy, placed 100 to 600 samples before the legitimate first copy. The earliest



(a) Convergence of calibration.

(b) Convergence of testing.

Figure 5: Convergence for Decreasing Success Rate. Few calibration samples are enough to find a sampling strategy that minimizes $\hat{A}_{ab}$ in Equation 6 (a), optimizing `FAST`'s convergence during testing (b). `FAST` quickly converges to an accurate estimate of the success rate. The 90 % confidence interval, denoted by the shaded area, is tight. When the lower bound is −128 it is not depicted, and only the upper side of the confidence interval is visualized. The different curves correspond to different values of the `STS_NTM` threshold, ranging from 12 to 31, which require a different tuning of the sampling strategy and lead to different success rates. In (a), the curves having higher values of - $\log_2(\widehat{A}_{ab})$ correspond to higher thresholds. In (b), curves having higher values of -$\log_2(\widehat{SR})$ correspond to lower thresholds.

successful advancement with respect to the legitimate early copy is 248 ns. The impact on distance reduction depends on the type of ranging sequence and on which packets are attacked [LCH+22]. It corresponds to a reduction of 74 m if all packets in the exchange are successfully attacked, or approximately 20 m if only one packet of a DS-TWR is targeted. These reductions are significant for access control.

4.1.6 Convergence of Calibration

As explained in Section 3.3, `FAST` uses a calibration phase to find a sampling strategy that minimizes the variance and ensures fast convergence to a good estimate of the success rate. To do so, it finds the parameters a and b that minimize $\hat{A}_{ab}$ (Equation 6) over a set of calibration measurements sampled with uniform n_c. The optimal parameters change in different scenarios, for example, for different values of the `STS_NTM` threshold. We collect around 500 measurements for values 12 and 31, and we plot the optimal $\hat{A}_{ab}$ for an increasing number of calibration samples. Figure 5a shows that 500 samples are sufficient to identify a good sampling strategy in both cases.

4.1.7 Convergence of Testing

After identifying the optimal sampling strategy, we can run a test. Figure 5b depicts the success rate and the confidence interval for increasing the number of samples, for ten values of the `STS_NTM` threshold. The numerical results are provided in Table 3. In most cases, the estimation quickly converges to an accurate estimate, and the cases are clearly distinguishable among each other. For the lowest three success rates the lower bound is $2^{-}128$, as discussed in Section 3.3.5. Since `FAST` adapts its sampling strategy to the success rate in the calibration phase, the number of samples for each case remains the same (around 5000). Instead, naive sampling would require an increasing number of samples (more than $1/SR$ to collect enough successes for an accurate estimate). Table 2

Table 2: Convergence of testing for `FAST` and baseline (naive). `FAST` converges quickly for both values of `STS_NTM`, whereas naive testing is less accurate for `STS_NTM` = 12 and does not observe even one success for `STS_NTM` = 16.

`Method`	`STS_NTM`	N_{cal}	N_{test}	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
Naive	12.00	-	10096	$2^{-9.4}$	$2^{-10.2}$	$2^{-8.9}$
Naive	16.00	-	10016	$2^{-128.0}$	-	-
`FAST`	12.00	1005	10057	$2^{-10.1}$	$2^{-10.3}$	$2^{-9.9}$
`FAST`	16.00	978	9831	$2^{-14.8}$	$2^{-15.3}$	$2^{-14.5}$

further compares `FAST` with naive testing in the same conditions. While `FAST` converges quickly for both thresholds, naive testing is less accurate with the low threshold, and it fails completely for the high threshold because 10000 samples are not enough to observe even one success. We report 2^{-128} (instead of 0) because we know that it is the lower bound for an attacker that bruteforces the key (Section 3.3.5).

4.1.8 Studying Ghost Peak Strategies with `FAST`

We set the chip to its default configuration and analyze the role of the attack and configuration parameters individually.

Presence of the Legitimate Signal: The success rate changes significantly depending on whether the attacker transmits on top of the legitimate signal or alone. If the attacker transmits alone, then the success rate is extremely low, below the 2^{-128} probability of guessing the full STS by brute-forcing the AES key used to generate the STS sequence in counter mode. This could happen because the receiver first looks for the strongest copy of the signal, then performs a backsearch to look for a possible weaker early copy. Since the main copy is expected to be strong, the chip strictly checks the validity of the STS. Instead, since the early copy can be attenuated, the receiver applies more tolerance when verifying its similarity to the expected STS. When the attacker transmits alone, it must pass the higher threshold of the main peak. Instead, when it transmits on top of a legitimate signal, it just has to pass the lower threshold of the early peak. We will only consider the second case that corresponds to the reactive injection scenario in [LCH+22].

Role of the Security Thresholds: We collect three datasets with a total of 12 000 calibration and 160 000 test measurements, in which we sweep individually the values of the `STS_MNTH`, `STS_NTM`, and `STS_PMULT` thresholds. Results in Table 3 show that increasing the thresholds can significantly decrease the success rate, with `STS_MNTH` having the largest effect. We stop increasing it when the success rate reaches the lower bound of 2^{-128}. Given the very low success rates, performing this study with naive testing would have been highly impractical. Because stricter and more secure configurations are likely to result in a much worse reliability [SRZ+21, CCA+24], their practical applicability depends on the channel conditions and performance requirements.

Role of the Attack Amplitude: The analysis of the receiver characteristics by Coppola et al. [CCA+24] shows that the receiver adapts its gain to match the overall incoming signal with the dynamic range of its analog front-end. Additionally, a signal that is too low might not be amplified to full range, and a signal that is too high is clipped, capping the contribution of the attacker pulses. The attacker does not have an interest in transmitting at powers higher than the clipping point. In this paper, we focus on the relative amplitude between the attacker signal and the main copy of the legitimate signal (denoted as `att2mp`), rather than the absolute attack power. We set the advancement to a significant value of $\approx 20\,\text{m}$, the transmit power to $-60\,\text{dBm}$ to avoid any clipping, and the receiver to the default configuration. We collect 5000 calibration samples and 50 000 testing samples for

Table 3: Qorvo DWM3000, Role of Security Thresholds. Increasing the security thresholds can significantly decrease the success rate (at the price of reliability). We fixed the following parameters: power = -50 dBm, attacker-to-main peak ratio = 0.9, advancement = 50 samples. The default values of the thresholds are the following: sts_mnth = 0x10, sts_ntm = 12, sts_pmult = 0x0.

STS_NTM	N_{cal}	N_{test}	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
12.00	502	4944	$2^{-12.1}$	$2^{-13.4}$	$2^{-11.4}$
14.00	490	4941	$2^{-14.9}$	$2^{-15.6}$	$2^{-14.4}$
16.00	483	5047	$2^{-19.9}$	$2^{-20.6}$	$2^{-19.5}$
18.00	518	4980	$2^{-23.6}$	$2^{-24.3}$	$2^{-23.1}$
20.00	485	5067	$2^{-27.5}$	$2^{-28.9}$	$2^{-26.8}$
22.00	509	5001	$2^{-32.8}$	$2^{-35.0}$	$2^{-32.0}$
24.00	504	5150	$2^{-37.2}$	$2^{-38.3}$	$2^{-36.5}$
26.00	482	4890	$2^{-41.0}$	$2^{-128.0}$	$2^{-40.0}$
28.00	507	4899	$2^{-41.2}$	$2^{-128.0}$	$2^{-39.8}$
31.00	520	5081	$2^{-56.7}$	$2^{-128.0}$	$2^{-55.5}$

STS_MNTH	N_{cal}	N_{test}	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
16.00	719	10073	$2^{-12.5}$	$2^{-12.8}$	$2^{-12.2}$
28.00	722	10038	$2^{-24.5}$	$2^{-25.4}$	$2^{-23.9}$
40.00	724	10058	$2^{-47.5}$	$2^{-128.0}$	$2^{-46.4}$
52.00	707	9845	$2^{-77.9}$	$2^{-128.0}$	$2^{-76.9}$
64.00	683	9908	$2^{-108.2}$	$2^{-128.0}$	$2^{-107.0}$
76.00	716	10141	$2^{-128.0}$	$2^{-128.0}$	$2^{-128.0}$

STS_PMULT	N_{cal}	N_{test}	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
0.00	509	2446	$2^{-12.8}$	$2^{-13.8}$	$2^{-12.2}$
1.00	523	2518	$2^{-12.7}$	$2^{-13.2}$	$2^{-12.3}$
2.00	467	2503	$2^{-13.5}$	$2^{-14.3}$	$2^{-12.9}$
3.00	501	2533	$2^{-12.8}$	$2^{-13.8}$	$2^{-12.3}$

Table 4: Qorvo DWM3000, Role of Attack Amplitude. The `att2mp` parameter denotes the ratio between the amplitude of the attacker signals and the main copy of the legitimate signal. Above 0.75 the success rate does not visibly increase. We fixed the following parameters: power = -50 dBm and advancement = 50 samples. The default values of the thresholds are the following: sts_mnth = 0x10, sts_ntm = 12, sts_pmult = 0x0.

att2mp	N_{cal}	N_{test}	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
0.25	502	4941	$2^{-34.0}$	$2^{-128.0}$	$2^{-32.8}$
0.50	490	5016	$2^{-17.5}$	$2^{-19.7}$	$2^{-16.7}$
0.75	483	5019	$2^{-14.8}$	$2^{-15.5}$	$2^{-14.4}$
1.00	518	4956	$2^{-11.4}$	$2^{-128.0}$	$2^{-10.3}$
1.25	485	4997	$2^{-11.5}$	$2^{-13.4}$	$2^{-10.7}$
1.50	509	5007	$2^{-12.5}$	$2^{-13.0}$	$2^{-12.1}$
1.75	504	5013	$2^{-12.0}$	$2^{-16.8}$	$2^{-11.0}$
2.00	482	4956	$2^{-14.8}$	$2^{-16.8}$	$2^{-14.0}$
2.25	507	5009	$2^{-14.4}$	$2^{-15.2}$	$2^{-13.8}$
2.50	520	5086	$2^{-17.2}$	$2^{-18.3}$	$2^{-16.6}$

`att2mp` ranging from 0.25 to 2.5. As shown in Table 4, a low relative attacker amplitude leads to a poor success rate, possibly because the attacker's peak in the CIR is smaller than the cross-correlation side-lobes caused by the strong copies of the legitimate signal.

As the attacker's amplitude increases, the success rate stabilizes at a higher value, This might be due to the stong attack signal contributing to the noise floor.

Success Function and n_c: We observe that the Qorvo DWM3000 accepts STSs sequences with all bits flipped, hence a two-sided acceptance region (high and low n_c). Additional results about channel, advancement, and warning registers are provided in Appendix B.

False Rejection Rate: Table 6 shows the false rejection rate for different values of every threshold: `STS_NTM`, `STS_MNTH`, and `STS_PMULT`. We send STSs identical to the expected STS, i.e., having 4096 common bits (n_c). These sequences undergo different channel conditions for every threshold value, and some of them are rejected by the receiver due to the alterations they can undergo, depending on the range of acceptance for each threshold. We calculate the False Rejection Rate (FRR) by counting unsuccessful STSs among all the ones we sent. We observe that at this power (i.e., -50 dBm), the effect of multi-path (and threshold values) on the FRR is visible in some cases but generally not pronounced. In these conditions configuring the chip to be secure without paying a large price on reliability is possible. To see a more pronounced and problematic security-performance trade-off, we need to consider more challenging channels at lower power, were peaks become smaller and more difficult to detect. We demonstrate this in Table 9 for PURE.

4.2 PURE

4.2.1 Setup

Since it is implemented on top of the Qorvo DWM3000EVB chip, PURE has the same experimental setup as the one described in Section 3.4.

4.2.2 Ghost Peak and Attack Success

The attack works as described in Section 4.1.2. PURE sets an absolute threshold that a peak should pass to get accepted, instead of using the proprietary STS quality metric

Table 5: PURE, Role of the Threshold. The `Th` parameter denotes the similarity threshold set by the receiver to accept or reject an incoming signal. We fixed the following parameters: power = -50 dBm, sts_mnth = 0x10, sts_ntm = 12, sts_pmult = 0x0, advancement = 50 samples.

Th	N_{cal}	N_{test}	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
300	5982	10914	$2^{-14.8}$	$2^{-15.3}$	$2^{-14.5}$
400	6025	11061	$2^{-23.7}$	$2^{-25.9}$	$2^{-22.9}$
500	5993	11025	$2^{-35.4}$	$2^{-38.8}$	$2^{-34.5}$

returned by the chip. Implementing this strategy is possible and efficient because the chip provides the full CIR, including the amplitude of the first peak in the diagnostics. Specifically, we accept the first peak as long as the amplitude of any of its first three samples (i.e., `stsF1`, `stsF2`, or `stsF3`) is above the threshold.

4.2.3 Chip Description and Configuration

PURE [CCA+24] is a protocol that protects contactless payments from relay attacks with a secure UWB receiver. We implemented it on a QorvoDWM3000EVB chip and tested it with `FAST`. The only configurable parameter is the PURE absolute threshold.

4.2.4 Preliminary Analysis of the Chip

In this case, we know how the chip works, instead of just getting results without any knowledge about why or why not they have been accepted or rejected by the receiver.

4.2.5 Studying Ghost Peak Strategies with `FAST`

We study the effect of different parameters on the security of PURE.
Role of the Threshold: As seen in Table 5 the estimated success rate $\widehat{SR}$ of the attack decreases when the threshold increases. This is expected since the higher the similarity threshold between the received STS and expected STS* sequences, the lower the chance that the attacker will pass this threshold by randomly guessing STS*.
Presence of the Legitimate Signal: Table 7 shows that, for a fixed threshold, sending the attack signal on top of the legitimate one slightly increases the success rate compared to sending it alone. This can be explained as in Section 4.1.8.
Role of Attack Amplitude: Table 8 shows how lowering the attack amplitude decreases the success rate, in line with the observations in [CCA+24]. This is because for the same n_c, the correlation peak becomes smaller, decreasing the chances of passing the absolute threshold.
Success Function and n_c: By design, PURE accepts STSs with all bits flipped (i.e., low n_c) in addition to those with all bits correct (i.e., high n_c). This is because it takes the absolute value of the cross-correlation peaks as a similarity metric.
False Rejection Rate: PURE [CCA+24] has demonstrated that the threshold can be set to a high (secure) value when operating in the good channels of mobile payments, with the phone in proximity to the terminal. In more challenging conditions, when the reception power is low (e.g., because of larger distances, less efficient antennas) and/or the channel has severe non-line-of-sight and multi-path conditions, peaks become smaller and more challenging to identify. Hence, in these cases setting a higher threshold generally increases the false rejection rate. To highlight this performance-security trade-off in challenging conditions, we set the power to -77 dBm. As done for the Qorvo DWM3000 chip, we send STSs identical to the expected STS, i.e., having 4096 common bits (n_c). Then, we get the

Table 6: QorvoDWM3000, False Rejection Rate vs. Thresholds. Same settings as in Table 3 but with signals sent by the legitimate transmitter in different channel models. Approximately 1000 samples per case for a total of 180000.

`Channel` / `STS_NTM`	12.00	14.00	16.00	18.00	20.00	22.00	24.00	26.00	28.00	31.00
Ind. LoS	0.015	0.014	0.018	0.011	0.014	0.019	0.016	0.018	0.021	0.015
Ind. NLOS	0.026	0.020	0.029	0.018	0.024	0.034	0.020	0.027	0.017	0.028
Office LoS	0.025	0.024	0.021	0.024	0.019	0.025	0.026	0.026	0.018	0.020
Office NLoS	0.039	0.029	0.024	0.029	0.025	0.025	0.028	0.024	0.031	0.035
Out. LoS	0.023	0.024	0.021	0.020	0.017	0.027	0.032	0.029	0.020	0.028
Out. NLOS	0.090	0.073	0.072	0.083	0.075	0.071	0.077	0.070	0.078	0.080
Res. LoS	0.020	0.025	0.023	0.017	0.026	0.032	0.018	0.029	0.031	0.028
Res. NLoS	0.021	0.026	0.019	0.029	0.022	0.027	0.020	0.030	0.035	0.027
Open Out. Env. NLOS	0.033	0.026	0.029	0.029	0.031	0.039	0.034	0.027	0.042	0.044

`Channel` / `STS_MNTH`	16.00	28.00	40.00	52.00	64.00	76.00
Industrial LoS	0.012	0.017	0.013	0.015	0.012	0.013
Industrial NLOS	0.020	0.016	0.010	0.013	0.020	0.019
Office LoS	0.014	0.014	0.013	0.015	0.014	0.024
Office NLoS	0.016	0.016	0.018	0.022	0.022	0.045
Outdoor LoS	0.010	0.017	0.011	0.020	0.025	0.030
Outdoor NLOS	0.069	0.073	0.058	0.060	0.086	0.117
Residential LoS	0.009	0.010	0.016	0.012	0.015	0.028
Residential NLoS	0.010	0.015	0.010	0.025	0.023	0.032
Open Outdoor Environment NLOS	0.028	0.026	0.011	0.022	0.027	0.026

`Channel` / `STS_PMULT`	0.00	1.00	2.00	3.00
Industrial LoS	0.021	0.014	0.012	0.028
Industrial NLOS	0.023	0.020	0.020	0.032
Office LoS	0.015	0.024	0.026	0.028
Office NLoS	0.018	0.040	0.036	0.033
Outdoor LoS	0.023	0.025	0.020	0.018
Outdoor NLOS	0.069	0.071	0.073	0.074
Residential LoS	0.025	0.025	0.022	0.026
Residential NLoS	0.024	0.019	0.030	0.022
Open Outdoor Environment NLOS	0.020	0.024	0.029	0.025

Table 7: PURE, Role of Legitimate Signal. The `Th` parameter denotes the similarity threshold set by the receiver. The `legit_on` parameter indicates whether the attacker signal is sent on top of the legitimate signal or not. We fixed the following parameters: power = -50 dBm, sts_mnth = 0x10, sts_ntm = 12, sts_pmult = 0x0, advancement = 50 samples.

`Th`	$legit\_on$	N_{cal}	N_{test}	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
400	$False$	4431	5499	$2^{-27.5}$	$2^{-29.0}$	$2^{-26.7}$
400	$True$	4546	5527	$2^{-23.3}$	$2^{-26.2}$	$2^{-22.4}$
500	$False$	4507	5483	$2^{-43.0}$	2^{-128}	$2^{-41.6}$
500	$True$	4516	5491	$2^{-36.1}$	$2^{-37.2}$	$2^{-35.5}$

Table 8: PURE, Role of Attack Amplitude. As PURE uses an absolute threshold (400 in this case), the attack amplitude (0.9 in previous experiments) is impactful. We fixed the following parameters: power = -50 dBm, sts_mnth = 0x10, sts_ntm = 12, sts_pmult = 0x0, advancement = 50 samples.

att2mp	N_{cal}	N_{test}	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
0.25	998	6022	$2^{-59.8}$	2^{-128}	$2^{-58.7}$
0.50	997	6087	$2^{-27.6}$	$2^{-32.2}$	$2^{-26.7}$
0.75	1019	5945	$2^{-26.0}$	$2^{-26.8}$	$2^{-25.5}$
1.25	986	5946	$2^{-25.2}$	$2^{-25.8}$	$2^{-24.8}$

Table 9: PURE, False Rejection Rate vs. Threshold. Same settings as in Table 5 but with signals sent by the legitimate transmitter in different channel models, and power reduced to −77 dBm to highlight the security-performance tradeoff in channels with low link budget. Approximately 500 samples per case for a total of 13500.

Channel / Threshold	300.00	400.00	500.00
Industrial LoS	0.01	0.02	0.02
Industrial NLOS	0.15	0.39	0.60
Office LoS	0.03	0.09	0.21
Office NLoS	0.17	0.37	0.56
Outdoor LoS	0.07	0.23	0.37
Outdoor NLOS	0.39	0.52	0.67
Residential LoS	0.04	0.10	0.22
Residential NLoS	0.09	0.21	0.43
Open Outdoor Environment NLOS	0.03	0.03	0.04

FRR by counting the ones that failed out of all the STSs we sent. Results are shown in Table 9.

4.3 NXP Trimension SR040 and NXP Trimension SR150

4.3.1 Setup

In this case study, the two NXP chips perform two-way ranging. When one transmits, the other acts as a receiver. These chips are connected to a Qorvo DWM3000EVB chip acting as a trigger: after receiving a message sent by one of the chips, the trigger triggers the signal generator after a suitable delay to transmit on top of the legitimate signal to the acting receiver, which is exactly what Ghost Peak does. Then, the generator transmits the attack signal, and the target chip receives it through an antenna (both chips have their antenna above them). We can read the distance between the two chips and notice its significant drop in case of a successful attack. For every n_c sample, we restart the ranging application and collect 50 distance measurements.

4.3.2 Ghost Peak and Attack Success

We send the attack signal on top of the exchanged signals between the two chips. We then read the resulting distance and deduce whether any attack succeeded or not. As the chip is proprietary, we only have access to the distance but no other diagnostics or knowledge about how or why the receiver accepted or rejected a STS sequence.

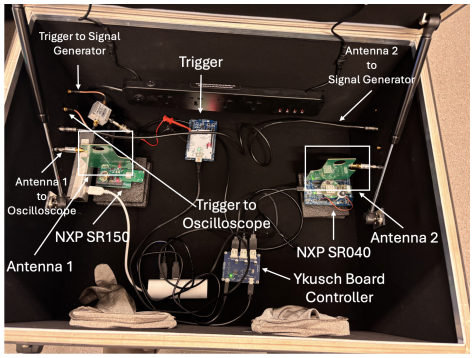


Figure 6: Experimental Setup for NXP SR040 and SR150. The two chips face each other and perform ranging. A Qorvo DWM3000 monitors the packets and acts as a trigger. When the victim chip (receiver) of the attack is the NXP SR150, we link it to the signal generator from which we send the attack signal. We link the NXP SR040 chip (transmitter) to the oscilloscope. When the NXP SR040 starts transmitting, the trigger triggers the oscilloscope to start recording the packets it is receiving from the chip via Antenna 1 so that we can decode them. Then, at our desired delay, we trigger the signal generator to send the attack signal to the victim chip via Antenna 2. When the NXP SR040 is our victim receiver, we repeat the same procedure, but linking it to the signal generator and linking the NXP SR150 to the oscilloscope. The setup is enclosed in an anechoic box to minimize interference and favor replicability.

4.3.3 Chip Description and Configuration

We use the default ranging demo from the MK UWB Kit, without any custom configuration. As testers, we can only send STS sequences into these chips and read whether they have been accepted or rejected. The two chips perform ranging, so they both alternate between transmitting and receiving. We cannot set a fixed STS.

After starting the ranging between the two chips, we decode the STS sequence set by each of them. Appendix C provides more details on our decoder. This allows us to bias our sampling when applying `FAST`. We generate the attacker signal using the signal generator. When we analyze the security of NXP Trimension SR040, we consider it as the receiver, and therefore the signal generator inputs the attacker STS to it via antenna, and sends it to the NXP Trimension SR150 otherwise.

4.3.4 Chip Insights

These two proprietary chips allow for limited interaction. When NXP Trimension SR040 transmits, we can decode its set STS, and when NXP Trimension SR150 transmits, we can decode its set STS. By observing the packets over the air, we also learn their number and timings. We can then configure the attacker to transmit at the right time.

4.3.5 Studying Ghost Peak Strategies with `FAST`

We study the role of different attack parameters on NXP chips.

Presence of Legitimate Signal and Role of Attack Amplitude:

Like in a real attack scenario, in this setup, the attacker transmits on top of the legitimate signals from the two NXP chips that are performing two-way ranging. Hence, the legitimate signal is always present, and the relative power between the attack and the legitimate signal is indirectly set by the transmit power and relative distances of the attacker and victims. Following a trial-and-error approach, we coarsely adjust the

Table 10: NXP SR040, Role of Preamble to STS Amplitude Ratio. The `pre2sts` parameter denotes the ratio between the amplitude of the preamble and the STS. We fixed the following parameters: power = -10 dBm, attacker-to-main peak ratio = 0.7, advancement = 1600 samples.

`pre2sts`	N_{cal}	N_{test}	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
0.25	40000	50000	$2^{-40.75}$	$2^{-42.87}$	$2^{-39.93}$
0.50	40000	75000	$2^{-46.33}$	2^{-128}	$2^{-45.29}$
1.00	20000	180000	$2^{-98.4}$	2^{-128}	$2^{-97.2}$

Table 11: NXP SR150, Role of Preamble to STS Amplitude Ratio. The `pre2sts` parameter denotes the ratio between the amplitude of the preamble and the STS. We fixed the following parameters: power = -10 dBm, attacker-to-main peak ratio = 0.7, advancement = 1600 samples.

`pre2sts`	N_{cal}	N_{test}	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
0.25	20400	4750	$2^{-14.95}$	$2^{-15.18}$	$2^{-14.75}$
0.50	19100	5050	$2^{-17.28}$	$2^{-17.68}$	$2^{-16.97}$
0.75	20350	5450	$2^{-13.97}$	$2^{-14.34}$	$2^{-13.67}$
1.00	20150	4750	$2^{-13.38}$	$2^{-17.35}$	$2^{-12.42}$

advancement and power of the attacker to be similar to that of the legitimate transmitter and to obtain reductions.

Role of Preamble to STS Amplitude Ratio:

Table 10 shows a noticeable decrease in the success rate of the attack for a higher preamble to STS amplitude. The chip might compute the CIR with both preamble and STS and check their consistency. For the same power, the amplitude of the peak generated by the correct preamble is likely higher than that created by a random STS. Hence, reducing the amplitude of the preamble is necessary to keep the consistency. The Ghost Peak paper [LCH+22] also observes that reducing the preamble power helps avoid jamming. However, table Table 11 shows a little and inconsistent effect of the `pre2sts` ratio. Our analyses remain speculative since the two chips are proprietary, and we do not know how they work.

False Rejection Rate: Since the STS and other thresholds cannot be set by the evaluator the case of the NXP chips, the only parameter we can change that can affect FRR is the physical channel in our setup. However, this requires physical changes to our setup, which is out of the scope of this study.

Success Function and n_c:

The SR040 accepts only STSs having high n_c, leading to a one-sided acceptance region. On the other hand, the SR150 accepts STSs with all bits flipped, hence a two-sided acceptance region (high and low n_c).

5 Discussion

5.1 Limitations

The main limitation of `FAST` is that it can only estimate the "insecurity" of a chip to a specific attack, not its security level against arbitrary attackers. While `FAST` cannot replace a formal security analysis, it is useful in several practical scenarios. First, when a closed-form expression for the security level does not exist and chips deploy proprietary algorithms, as is the case for HRP UWB. Second, to complement a theoretical analysis with

empirical tests on real hardware. Indeed, the real physical-layer behavior of the chip might deviate from the assumptions made in the analytical models, or models might depend on some unknown physical parameters. In addition, while attackers cannot leverage `FAST` against a victim device with unknown STS, they can use it to study a similar device with known STS (e.g., development kit) and identify optimal parameters.

Another limitation is that the number of parameters to control could be large, leading to an exponential increase in the number of cases to study. In practice, an iterative approach guided by experience and knowledge of the attacks and receivers could help identify the most relevant factors. In addition, `FAST` requires few samples per case.

Finally, `FAST` needs some control on the target device: knowing the secret STS to optimize the sampling strategy, and information about acceptance of the attack signal to estimate the success rate. `FAST` also needs to inject signals in the chip in receive mode, which is easier if the chip can be configured to listen for messages continuously.

5.2 Broad Application of `FAST`

Ranging and positioning systems generally embed some form of unpredictability in the signals to protect against distance-reduction/position-spoofing attacks. This aims to prevent the attacker from transmitting the signal expected by the receiver in advance. The security of the system relies on the verification of such signals [LSR+20]. Examples are: the STS in HRP/LRP UWB [IEE20a, IEE20b], randomization of the bit-to-pulse mapping in UWB-PR [SLC17], phase randomization in ODFM ranging [LKRC21], the Secure LTF field in Wifi Ranging [IEE], sounding sequence randomization for narrowband secure ranging with Bluetooth [Blu, ASR+21], encrypted spreading codes in some services of GPS, Galileo, BeiDou, and GLONASS [ZP19], spreading code puncturing in CHIMERA [ACD+17], and message authentication in Galileo [EUS, FHRSG+16].

Because of this unpredictability, attacks often involve a probabilistic guessing phase. Ghost Peak [LCH+22] randomly samples the STS before transmission. In Stretch-and-Advance [ACC23], the attacker leverages a time stretch to learn some of the unpredictable bits towards the end of the packet but has to guess the initial ones. Guess-and-Compensate [SLC17, LSR+20] adaptively decides whether to stop guessing or double the power and continue, after learning if the previous guess was correct. An advanced adaptive strategy that accounts for random pulse reordering is used to numerically evaluate the security of UWB-PR [SLC17]. In Early-Detect/Late-Commit (ED/LC) attacks [CHKM06, ZP19], the overall success rate depends on the probability of learning and committing the correct symbol. If it is randomized at the physical layer, the attack strategy is complex, and analytical expressions of the success rate might not exist [SLC17, LKRC21].

`FAST` could be applied to these technologies and attacks, by biasing the distribution of the random guesses and/or the physical variables impacting the success rate.

6 Related Work

Distance Bounding In applications requiring physical proximity between devices, distance-bounding protocols [ABB+19] can establish a secure upper bound on their distance. As distance is derived from the timings of wireless exchanges between the devices, special care is required to guarantee the integrity of the time-of-arrival measurements at the physical layer [LCH+22]. Distance measurements and distance bounding have been implemented with several wireless technologies, such as UWB [RC10, TLKC15], BLE [ASR+21], and NFC [RCN+22]. Over the last few years, chips implementing HRP UWB (IEEE 802.15.4z [IEE20a, IEE20b]) have been widely deployed in smartphones and other devices, fueled by many potential applications [FiR]. Several recent studies

have investigated the security of HRP UWB [SRZ+21], practical distance-reduction attacks [LCH+22, ACC23], practical jamming attacks [YWZ+24], applications [CCA+24], more secure receiver strategies [JLJC23, CCA+24, LKZ+24], and performance [HKPH23].

Importance Sampling Importance sampling has been applied to many scientific problems [Sri14, Bio15]. In computer security, it has been used to evaluate security against *guessing attacks*, which discover a secret (e.g., password or cryptographic key) through many guesses. These *smart guesses* are guided by a model obtained through an information leak. This could be a password generation model trained on a corpus of leaked passwords [MYLL14], or the scores assigned to key bytes by a side channel attack given a physical information leakage [VCGRS13]. Attackers generally assign a score $s(x)$ (e.g., an estimated probability) to each guess x, and perform guesses sorted by decreasing score. In such cases, attack strength is evaluated as the number of guesses g needed to find the secret; in other words, g is the *rank* of the correct guess: the number of incorrect guesses with a larger score than the correct one. The rank g can be found by simulating the guessing attack, but such an endeavor may be prohibitive for expensive attacks. This problem is very different from estimating the success rate of distance-reduction attacks. However, in this case, classic Monte Carlo methods are also inefficient and can be improved with importance sampling. A naive Monte Carlo approach would uniformly sample m guesses over the secret's $n \gg m$ possible values (assuming that n is finite), and use the scores of elements in the sample to estimate g/n and hence g. However, this procedure would require an enormous sample size if, as is often the case, $g \ll n$. Importance sampling can bias the sample generation towards higher-score guesses and result in accurate estimates with small sample sizes [DF15, CDS22]. It can be used even when the set of possible secrets is infinite, like for passwords [DF15]. Adaptive approaches are useful when scores output by the model are not well-calibrated posterior probabilities [CDS22]. Liu et al. [LBB23] provide tight bounds for rank estimates based on importance sampling.

7 Conclusion

Distance-reduction attacks like Ghost Peak threaten security applications that rely on ranging, such as seamless access control. In this paper, we have proposed `FAST`, a general methodology to efficiently and accurately characterize the security of real-world receivers by empirical testing, even when closed-form security level expressions are not available. To facilitate future research, our code is available on Zenodo https://zenodo.org/records/17295874.

Acknowledgments

This research has received funding from Fondation Botnar. This research has received funding from the Swiss National Science Foundation under NCCR Automation, grant agreement 51NF40_180545.

References

[ABB+19] Gildas Avoine, Muhammed Ali Bingöl, Ioana Boureanu, Srdjan Capkun, Gerhard P. Hancke, Süleyman Kardas, Chong Hee Kim, Cédric Lauradoux, Benjamin Martin, Jorge Munilla, Alberto Peinado, Kasper Bonne Rasmussen, Dave Singelée, Aslan Tchamkerten, Rolando Trujillo-Rasua, and Serge Vaudenay. Security of distance-bounding: A survey. *ACM Comput. Surv.*, 51(5):94:1–94:33, 2019.

[ACC23] Claudio Anliker, Giovanni Camurati, and Srdjan Capkun. Time for change: How clocks break UWB secure ranging. In Joseph A. Calandrino and Carmela Troncoso, editors, *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, pages 19–36. USENIX Association, 2023.

[ACD+17] Jon Anderson, Katherine Carroll, Nathan DeVilbiss, J. Gillis, Joanna Hinks, Brady O'Hanlon, Joseph Rushanan, Logan Scott, and Renee Yazdi. Chips-message robust authentication (chimera) for gps civilian signals. pages 2388–2416, 11 2017.

[app] Ultra Wideband security in iOS.

[ASR+21] Aysajan Abidin, Mohieddine El Soussi, Jac Romme, Pepijn Boer, Dave Singelée, and Christian Bachmann. Secure, accurate, and practical narrow-band ranging system. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(2):106–135, 2021.

[Bio15] Gino Biondini. Chapter 2 - An introduction to rare event simulation and importance sampling. In Venu Govindaraju, Vijay V. Raghavan, and C.R. Rao, editors, *Big Data Analytics*, volume 33 of *Handbook of Statistics*, pages 29–68. Elsevier, 2015.

[Blu] Bluetooth SIG. Channel Sounding CR_PR. https://www.bluetooth.com/specifications/specs/channel-sounding-cr-pr/. Accessed 2024-07-06.

[CCA+24] Daniele Coppola, Giovanni Camurati, Claudio Anliker, Xenia Hofmeier, Patrick Shaller, David Basin, and Srdjan Capkun. Pure: Payments with UWB relay-protection. In *33rd USENIX Security Symposium (USENIX Security 2024), Philadelphia, PA, USA, August 14-16, 2024*, 2024.

[CDS22] Giovanni Camurati, Matteo Dell'Amico, and François-Xavier Standaert. Mcrank: Monte Carlo key rank estimation for side-channel security evaluations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(1):277–300, Nov. 2022.

[CHKM06] Jolyon Clulow, Gerhard P. Hancke, Markus G. Kuhn, and Tyler Moore. So near and yet so far: Distance-bounding attacks in wireless networks. In Levente Buttyán, Virgil D. Gligor, and Dirk Westhoff, editors, *Security and Privacy in Ad-Hoc and Sensor Networks, Third European Workshop, ESAS 2006, Hamburg, Germany, September 20-21, 2006, Revised Selected Papers*, volume 4357 of *Lecture Notes in Computer Science*, pages 83–97. Springer, 2006.

[DF15] Matteo Dell'Amico and Maurizio Filippone. Monte Carlo strength evaluation: Fast and reliable password checking. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 158–169, New York, NY, USA, 2015. Association for Computing Machinery.

[EUS] EUSPA. Galileo Open Service Navigation Message Authentication (OSNMA) Info Note. https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OSNMA_Info_Note.pdf. Accessed 2024-07-06.

[FHRSG+16] Ignacio Fernández-Hernández, Vincent Rijmen, Gonzalo Seco-Granados, Javier Simon, Irma Rodríguez, and J. David Calle. A navigation message authentication proposal for the galileo open service. *NAVIGATION*, 63(1):85–102, 2016.

[FiR] FiRa. UWB use cases. https://www.firaconsortium.org/discover/use-cases. Accessed 2024-06-30.

[Git] GitHub user foldedtoad. Nordic nRF52-series + Decawave DWM3000 on Zephyr v2.5. https://github.com/foldedtoad/dwm3000. Accessed 2024-07-01.

[HKPH23] Alexander Heinrich, Sören Krollmann, Florentin Putz, and Matthias Hollick. Smartphones with UWB: evaluating the accuracy and reliability of UWB ranging. *CoRR*, abs/2303.11220, 2023.

[IEE] IEEE SA Working Groups. Newly Released IEEE 802.11az Standard Improving Wi-Fi Location Accuracy is Set to Unleash a New Wave of Innovation. https://standards.ieee.org/beyond-standards/newly-released-ieee-802-11az-standard-improving-wi-fi-location-accuracy-is-set-to-unleash-a-new-wave-of-innovation/. Accessed 2024-07-06.

[IEE20a] IEEE. IEEE Standard for Low-Rate Wireless Networks. *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pages 1–800, 2020.

[IEE20b] IEEE. IEEE Standard for Low-Rate Wireless Networks–Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques. *IEEE Std 802.15.4z-2020 (Amendment to IEEE Std 802.15.4-2020)*, pages 1–174, 2020.

[JLJC23] Kyungho Joo, Dong Hoon Lee, Yeonseon Jeong, and Wonsuk Choi. Protecting HRP UWB ranging system against distance reduction attacks. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, pages 622–635. ACM, 2023.

[LBB23] Peiyuan Liu, Jeremiah Blocki, and Wenjie Bai. Confident Monte Carlo: Rigorous analysis of guessing curves for probabilistic password models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 626–644, 2023.

[LCH+22] Patrick Leu, Giovanni Camurati, Alexander Heinrich, Marc Roeschlin, Claudio Anliker, Matthias Hollick, Srdjan Capkun, and Jiska Classen. Ghost peak: Practical distance reduction attacks against HRP UWB ranging. In Kevin R. B. Butler and Kurt Thomas, editors, *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pages 1343–1359. USENIX Association, 2022.

[Li22] Abner Li. Pixel 6 Pro gets settings toggle to disable Ultra-Wideband (UWB), January 2022.

[LKRC21] Patrick Leu, Martin Kotuliak, Marc Roeschlin, and Srdjan Capkun. Security of multicarrier time-of-flight ranging. In *ACSAC '21: Annual Computer Security Applications Conference, Virtual Event, USA, December 6 - 10, 2021*, pages 887–899. ACM, 2021.

[LKZ+24] X. Luo, C. Kalkanli, H. Zhou, P. Zhan, and M. Cohen. Secure ranging with ieee 802.15.4z hrp UWB. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 193–193, Los Alamitos, CA, USA, may 2024. IEEE Computer Society.

[LSR+20] Patrick Leu, Mridula Singh, Marc Roeschlin, Kenneth G. Paterson, and Srdjan Capkun. Message time of arrival codes: A fundamental primitive for secure distance measurement. In *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*, pages 500–516. IEEE, 2020.

[MBC+04] Andreas Molisch, Kannan Balakrishnan, Dajana Cassioli, Chia-Chin Chong, Shahriar Emami, Andrew Fort, Johan Karedal, Juergen Kunisch, Hans Schantz, Ulrich Schuster, and Kai Siwiak. Ieee 802.15.4a channel model - final report. 15, 01 2004.

[MNFB18] Marius Muench, Dario Nisi, Aurelien Francillon, and Davide Balzarotti. Avatar²: A Multi-target Orchestration Platform. In *Workshop on Binary Analysis Research (colocated with NDSS Symposium)*, BAR 18, February 2018.

[MYLL14] Jerry Ma, Weining Yang, Min Luo, and Ninghui Li. A study of probabilistic password models. In *2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014*, pages 689–704. IEEE Computer Society, 2014.

[noa] BMW Digital Key Plus Now Available on Compatible Android Devices. | Uncategorized.

[Nor] Nordic Semiconductor. nRF52 DK - Nordic Semiconductor. https://www.nordicsemi.com/Products/Development-hardware/nRF52-DK. Accessed 2024-07-01.

[NXPa] NXP. Trimension™ SR040: Reliable UWB Solution for IoT. https://www.nxp.com/products/wireless-connectivity/trimension-uwb/trimension-sr040-reliable-uwb-solution-for-iot:SR040. Accessed 2025-04-22.

[NXPb] NXP. Trimension™ SR150: Secure UWB Solution for IoT Devices. https://www.nxp.com/products/SR150. Accessed 2025-04-22.

[Owe13] Art B. Owen. *Monte Carlo theory, methods and examples.* https://artowen.su.domains/mc/, 2013.

[Qora] Qorvo Inc. DW3000 Family User Manual. https://www.qorvo.com/products/d/da008154. Accessed 2024-07-01.

[Qorb] Qorvo Inc. DWM3000EVB - API Software and API Guide. https://www.qorvo.com/products/d/da007992. Accessed 2024-07-01.

[Qorc] Qorvo Inc. DWM3000EVB - Qorvo. https://www.qorvo.com/products/p/DWM3000EVB. Accessed 2024-07-01.

[RC10] Kasper Bonne Rasmussen and Srdjan Capkun. Realization of RF distance bounding. In *19th USENIX Security Symposium, Washington, DC, USA, August 11-13, 2010, Proceedings*, pages 389–402. USENIX Association, 2010.

[RCN+22] Andreea-Ina Radu, Tom Chothia, Christopher J. P. Newton, Ioana Boureanu, and Liqun Chen. Practical EMV relay protection. In *43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022*, pages 1737–1756. IEEE, 2022.

[sam] Unlock a New Experience: Galaxy Users Can Now Use Secure Digital Key With the Genesis GV60.

[SLC17] Mridula Singh, Patrick Leu, and Srdjan Capkun. UWB with pulse reordering: Securing ranging against relay and physical-layer attacks. Cryptology ePrint Archive, Paper 2017/1240, 2017. https://eprint.iacr.org/2017/1240.

[Sri14] R. Srinivasan. *Importance Sampling*. Springer, 2014.

[SRZ+21] Mridula Singh, Marc Roeschlin, Ezzat Zalzala, Patrick Leu, and Srdjan Capkun. Security analysis of IEEE 802.15.4z/hrp UWB time-of-flight distance measurement. In Christina Pöpper, Mathy Vanhoef, Lejla Batina, and René Mayrhofer, editors, *WiSec '21: 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Abu Dhabi, United Arab Emirates, 28 June - 2 July, 2021*, pages 227–237. ACM, 2021.

[TK10] Surya T Tokdar and Robert E Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60, 2010.

[TLKC15] Nils Ole Tippenhauer, Heinrich Luecken, Marc Kuhn, and Srdjan Capkun. UWB rapid-bit-exchange system for distance bounding. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, New York, NY, USA, June 22-26, 2015*, pages 2:1–2:12. ACM, 2015.

[Tuo24] Jennifer Pattison Tuohy. You'll need to buy a new lock if you want Apple Home to "magically" unlock your door, June 2024.

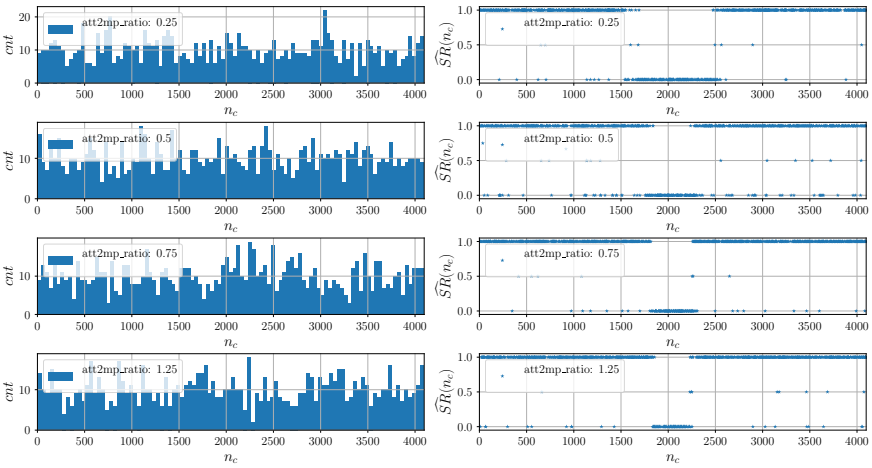
[VCGRS13] Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert. An optimal key enumeration algorithm and its application to side-channel attacks. In Lars R. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography*, pages 390–406, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[YWZ+24] Yuqiao Yang, Zhongjie Wu, Yongzhao Zhang, Ting Chen, Jun Li, Jie Yang, Wenhao Liu, Xiaosong Zhang, Ruicong Shi, Jingwei Li, Yu Jiang, and Zhuo Su. UWBAD: towards effective and imperceptible jamming attacks against UWB ranging systems with COTS chips. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS 2024, Salt Lake City, UT, USA, October 14-18, 2024*, pages 3376–3390. ACM, 2024.

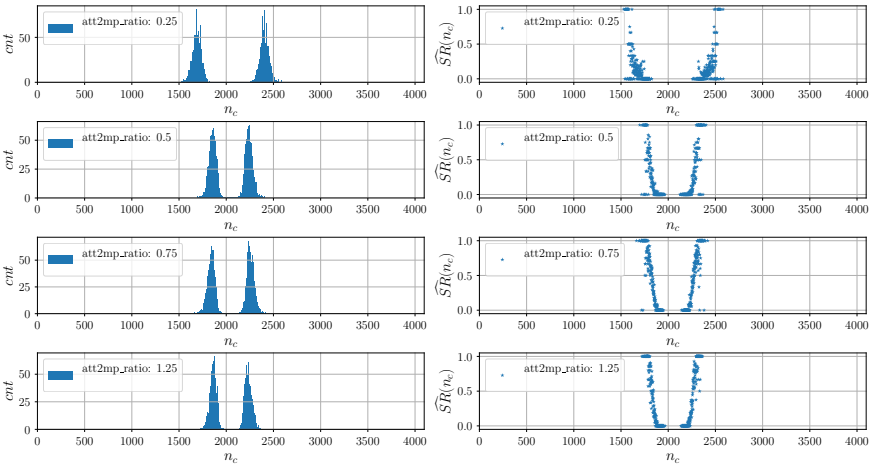
[ZP19] Kewei Zhang and Panos Papadimitratos. On the effects of distance-decreasing attacks on cryptographically protected gnss signals. pages 363–372, 02 2019.

Appendix

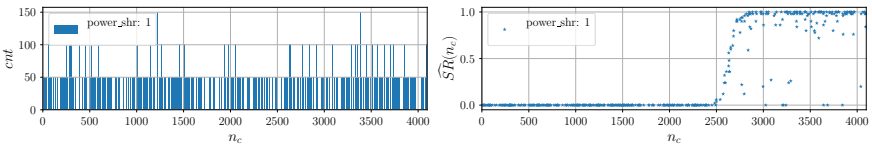
A Sampling Strategy Visualizations



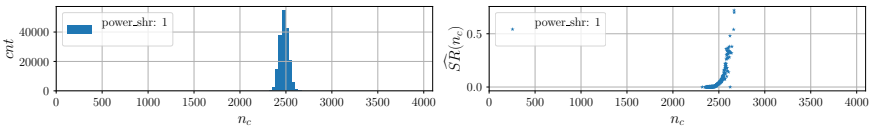
(a) Calibration with Uniform Sampling, High and Low n_c Succeed.



(b) Testing with Betabinomial Mixture, High and Low n_c Succeed.



(c) Calibration with Uniform Sampling, Only High n_c Succeed.



(d) Testing with Betabinomial, Only High n_c Succeed.

Figure 7: Sampling Strategies. The uniform sampling step is effective at identifying the important regions to sample with the betabinomial mixture during testing.

Figure 7 visualizes our sampling strategies. The left side shows the sampling distribution (`cnt` denotes the number of samples for each n_c). The right side shows the average success rate $\widehat{SR}(n_c)$ as a function of n_c, which represents the behavior of the success function $f(n_c)$ of the chips. From Figure 7a and Figure 7c (correponding to Table 8 in Section 4) we can observe that the calibration step with uniform sampling can easily identify whether a chip accepts both high and low n_c or only high n_c, and what are the important transition regions to sample depending on the attack and chip parameters. From Figure 7b and Figure 7d (Table 10, last row) we can observe the corresponding testing strategy with a betabinomial mixture that focuses the sampling to the important regions.

B Additional Results for QorvoDWM3000EVB

Table 12: Role of Wireless Channel. For a given receiver configuration, the type of channel does not have a significant impact on success.

`Channel model`	N_{cal}	N_{test}	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
Residential LoS	219	1128	$2^{-9.8}$	$2^{-10.3}$	$2^{-9.4}$
Residential NLoS	228	1069	$2^{-9.5}$	$2^{-10.3}$	$2^{-8.9}$
Office LoS	224	1145	$2^{-10.3}$	$2^{-10.8}$	$2^{-10.0}$
Office NLoS	223	1069	$2^{-9.9}$	$2^{-10.4}$	$2^{-9.5}$
Outdoor LoS	246	1147	$2^{-9.5}$	$2^{-10.4}$	$2^{-9.0}$
Outdoor NLOS	216	1080	$2^{-10.6}$	$2^{-13.2}$	$2^{-9.7}$
Industrial LoS	222	1115	$2^{-10.7}$	$2^{-12.0}$	$2^{-10.0}$
Industrial NLOS	228	1152	$2^{-10.0}$	$2^{-10.6}$	$2^{-9.5}$
Open Outdoor Environment NLOS	237	1152	$2^{-11.1}$	$2^{-12.2}$	$2^{-10.5}$

Role of the Wireless Channel Using the same dataset as for convergence with the default configuration, we now study the effect of using different channel types. Table 12 suggests that the type of channel does not have a significant impact. However, in practical use cases, one might choose the configuration (e.g., threshold) not based on its security level, but on its performance in a given type of channel. This leads to the security-performance trade-off observed in [SRZ+21], where a lower security level must be tolerated to have good reliability in challenging channel conditions.

Role of the Amount of Advancement We collect 5000 calibration measurements and 50000 test measurements, where we vary the amount by which the attack signal is advanced compared to the legitimate one. We observe in Table 13 that the success rate is high for reductions up to approximately 100 ns, corresponding to 30 m. After that, the rate decreases significantly, from around 2^{-10} down to 2^{-22}. The true acceptance rate for legitimate signals also decreases towards the extremity of the backsearch window (Section 4.1.5). In all other experiments, we consider an advancement of approximately 66.8 ns (20 m), which is meaningful in realistic scenarios (e.g., attacks on physical access control).

Role of the Warning Registers In addition to the STS quality indicator, the chip offers 9 additional warning flags. We collect 2000 calibration measurements and 4000 test measurements, where we either check the additional warnings or not. Results in Table 14 indicate that checking the registers does not significantly reduce the success rate. We attribute this to the good physical-layer design of our signal, matching the expected one.

Table 13: Role of Advancement. The success rate is stable until approximately 100 ns (30 m) of advancement.

t_{adv} (ns)	N_{cal}	N_{test}	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
−222.58	520	4964	$2^{-28.3}$	$2^{-128.0}$	$2^{-27.1}$
−200.32	507	4872	$2^{-20.1}$	$2^{-128.0}$	$2^{-18.7}$
−178.06	482	5031	$2^{-25.1}$	$2^{-30.5}$	$2^{-24.1}$
−155.80	504	5023	$2^{-24.7}$	$2^{-26.4}$	$2^{-23.9}$
−133.55	509	5140	$2^{-24.1}$	$2^{-25.3}$	$2^{-23.5}$
−111.29	485	5057	$2^{-14.3}$	$2^{-14.6}$	$2^{-13.9}$
−89.03	518	5029	$2^{-13.1}$	$2^{-13.5}$	$2^{-12.7}$
−66.77	483	5041	$2^{-13.3}$	$2^{-13.8}$	$2^{-12.9}$
−44.52	490	4906	$2^{-12.7}$	$2^{-15.4}$	$2^{-11.8}$
−22.26	502	4937	$2^{-12.7}$	$2^{-13.1}$	$2^{-12.4}$

Table 14: Role of Warning Registers. With a well-configured attack, checking the warning registers does not seem to have a significant impact on the success rate.

Warnings checked	N_{cal}	N_{test}	$\widehat{SR}$	$\widehat{SR}_{lb}$	$\widehat{SR}_{ub}$
0b0	976	1976	$2^{-9.7}$	$2^{-10.3}$	$2^{-9.3}$
0b111111111	1024	2024	$2^{-10.7}$	$2^{-11.3}$	$2^{-10.3}$

C More Details on the Decoder

Reception We collect HRP UWB packets using an R&S RTP164b oscilloscope in direct sampling at 40 GSa/s, a UWB antenna, and a band-pass filter. To trigger acquisition, we use a Qorvo DWM3000EVB in reception mode. After being awakened by the first packet in a two-way ranging sequence, it waits for the appropriate amount of time until the desired packet in the sequence, and then raises a digital pin to trigger the oscilloscope.

Preprocessing We apply the following preprocessing steps. Data are turned into their complex representation and normalized. A band-pass filter is applied to attenuate interference. A simple energy detector is used to extract the preamble and STS from the received signal. After detecting the carrier frequency, these samples are digitally down-converted to baseband, where an additional fine-grained adjustment of the frequency offset is applied.

Decoding The STS is modulated with Binary Phase Shift Keying (BPSK) pulses. After having identified each of the 4096 peaks corresponding to each pulse, we feed their samples to a Costas Loop to correct phase rotations, and finally demodulate the bits as ones and zeros. This method is neither effective nor secure for ToA verification because it works on the main copy of the signal, which does not necessarily represent the line-of-sight path and could be manipulated by an adversary. However, using the main copy is effective for decoding the STS bits with a high signal-to-noise ratio, avoiding bit errors. We confirmed that the signal-to-noise ratio in our setup is sufficient to correctly decode the full STS by testing the decoder with a transmitter with known STS. The decoding of the preamble is similar but slightly more complex because of the ternary modulation, which requires identifying possible gaps between successive BPSK peaks.