Here, There, and Everywhere: Security Analysis of Wi-Fi Fine Timing Measurement

Domien Schepers
Northeastern University
Boston, Massachusetts, USA
schepers.d@northeastern.edu

Mridula Singh
ETH Zurich
Zurich, Switzerland
mrsingh@inf.ethz.ch

Aanjhan Ranganathan
Northeastern University
Boston, Massachusetts, USA
aanjhan@northeastern.edu

ABSTRACT

Today, an increasing number of applications rely on location and proximity information to deliver services. With the introduction of Wi-Fi Fine Timing Measurement (FTM) in the IEEE 802.11-2016 standard, Wi-Fi derived location and proximity information will play a key role in many safety- and security-critical applications. For example, Wi-Fi FTM is adopted in Wi-Fi Aware where it enables geo-fencing and mobile identification. In this paper, we perform the first security analysis of Wi-Fi FTM and analyze its security guarantees across the logical and physical layers. We find various weaknesses that enable an attacker to introduce distance reductions and enlargements to any arbitrary attacker-chosen value, requiring commodity hardware only. We perform an evaluation using commercial access points, smartphones, and off-the-shelf Wi-Fi cards, and show that an attacker can manipulate distances with meter-level precision. Furthermore, we highlight the distance manipulation attacks which are independent of any higher-layer cryptographic protection, exposing fundamental limitations to achieving secure distance measurements in the current standard. Finally, we present security recommendations for the design and implementation of Wi-Fi FTM and next-generation positioning protocols.

CCS CONCEPTS

• **Networks → Mobile and wireless security**.

KEYWORDS

IEEE 802.11, Wi-Fi, Fine Timing Measurement, Security

ACM Reference Format:
Domien Schepers, Mridula Singh, and Aanjhan Ranganathan. 2021. Here, There, and Everywhere: Security Analysis of Wi-Fi Fine Timing Measurement. In *Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '21), June 28–July 2, 2021, Abu Dhabi, United Arab Emirates.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3448300.3467828

1 INTRODUCTION

In recent years, numerous wireless ranging and localization systems have been developed; they differ in positioning methodology (e.g., received signal strength, time-of-flight, time-of-arrival), accuracy, operating environment (e.g., indoors, outdoors), communication channel (e.g., radio frequency, ultrasound), and reliability. With the pervasiveness of Wi-Fi networks and increasing need for accurate location information, the IEEE standardized Wi-Fi Fine Timing Measurement (FTM) in IEEE 802.11mc, which is incorporated in IEEE 802.11-2016 [7]. Wi-Fi FTM uses round-trip time-of-flight measurements to determine the distance between two stations, and allows for meter-level ranging accuracy [26]. It is expected to enable new, feature-rich, safety- and security-critical applications [33], providing benefits to a variety of stakeholders. For example, it enables improved asset and personnel management, geo-fencing with trigger actions (e.g., access control and authentication) [4], network management, navigation [27], and emergency support. Today, Wi-Fi FTM is already used in security-sensitive applications, for example, it is used as a metric to discriminate a neighbor from an attacker as part of onboarding Internet of Things (IoT) devices [33]. Furthermore, the Wi-Fi Alliance adopted Wi-Fi FTM as a key feature in Wi-Fi Aware, where use cases include geo-fencing and mobile identification in airport security and autonomous vehicles [4], and recently Google released an example application for Wi-Fi Aware (*WifiNanScan*) [40], highlighting the increasing usage of Wi-Fi FTM. Moreover, the IEEE formed a task group to standardize a new positioning standard named Wi-Fi Next Generation Positioning (Wi-Fi NGP; IEEE 802.11az [28]) largely based on the fine-timing measurement mechanism introduced in Wi-Fi FTM. Wi-Fi NGP is currently under development and scheduled to be published in 2023. Until then, the usage of Wi-Fi FTM is expected to increase (e.g., Google supports Wi-Fi FTM since Android 9 [6]). Given Wi-Fi FTM's usage in security-sensitive applications and its influence in the design and development of next-generation wireless positioning standards, it is imperative to evaluate its security guarantees as researchers have repeatedly demonstrated the severe implication of distance modification attacks on ranging and localization systems (e.g., gain access to restricted areas [17], make fraudulent purchases [22, 23]).

In this paper, we present the first security analysis of Wi-Fi FTM by systematically analysing the security guarantees across the logical and physical layer. We find the protocol is vulnerable to a wide variety of distance manipulation attacks, allowing an adversary to introduce distance reductions and enlargements to any arbitrary attacker-chosen value. For example, we present a novel benchmarking technique which allows an adversary to spoof distance measurements irrespective of the station's true position. We evaluate our attacks on commercial products supporting Wi-Fi FTM, covering a wide variety of access points, smartphones, and off-the-shelf Wi-Fi cards, and show an attacker can introduce distance manipulations with meter-level accuracy. The attacker can achieve these attacks using commodity hardware only, and is not restricted by any physical location requirements. Notably, our analysis revealed attacks beyond the injection of spoofed frames which are

WiSec '21, June 28–July 2, 2021, Abu Dhabi, United Arab Emirates

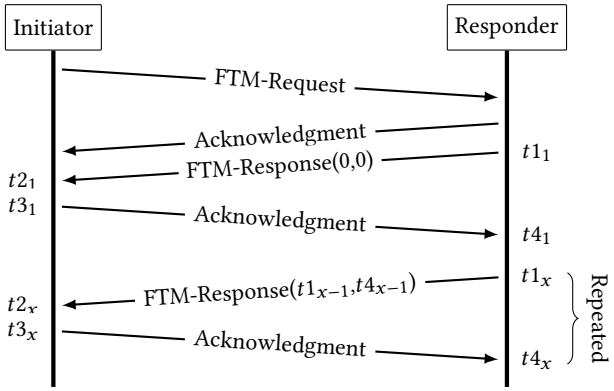

Figure 1: Example Wi-Fi Fine Timing Measurement session with ASAP=1, and x-number of measurements per burst.

successful even if the protocol would be protected by the network (e.g., WPA3). For example, we present a novel mixed-layer replay attack where frames are transmitted under unexpected physical-layer parameters (e.g., a narrower bandwidth), causing incorrect distance measurement calculations on the logical-layer. Furthermore, we present several strictly physical-layer attacks which build upon fundamental limitations in the wireless signal and receiver design (e.g., overshadow and earlier-path injection attack), and can lead to a distance manipulation of several tens of meters.

We hope our findings discourage the usage of Wi-Fi FTM for security-sensitive applications, and contribute to the development of next-generation protocols. To this end, we recommend changes to increase the security robustness of firmware implementations. Furthermore, we discuss the limitations of the physical-layer signal and receiver design of Wi-Fi FTM, and discuss approaches to secure future standards (e.g., the currently under development Wi-Fi NGP).

2 WI-FI FINE TIMING MEASUREMENT

Wi-Fi Fine Timing Measurement (FTM), also referred to as Wi-Fi Round-Trip Time (RTT) and branded as Wi-Fi Location by the Wi-Fi Alliance, is defined in the IEEE 802.11mc amendment and incorporated in IEEE 802.11-2016 [7]. Wi-Fi FTM enables stations to determine their physical distance and their precise location (e.g., using multilateration) by measuring the round-trip time of frames exchanged between them; thereby making it beneficial to several applications such as indoor navigation, asset tracking, network management, access control and authentication [4].

2.1 Round-Trip Time Measurements

A Wi-Fi FTM distance measurement session consists of three phases: a negotiation, measurement exchange, and termination phase. In Figure 1, we present an overview of two stations (initiator and responder) executing a session. The standard allows a station to act as both an initiator and responder, however, only the initiator derives the distance. If the responder wishes to determine the distance, the stations need to switch roles and start a new measurement session.

Negotiation. The initiator starts the negotiation phase by transmitting a request frame. This frame contains a trigger field used to request the initiation or termination of a measurement procedure. Additionally, the frame includes configuration parameters and vendor-specific information elements. Configuration parameters include, e.g., scheduling, channel, and operational parameters. To accommodate station constraints (e.g., concurrent sessions, higher-priority traffic), stations may request a preferred time window allocation, referred to as a burst instance. For example, a station may request the burst instance to start As Soon As Possible (ASAP) or request a later time window. The responder, often configured as an access point, responds with a status code indicating success or failure for the requested parameters. Upon successful negotiation, the responder will commence the measurement exchange phase.

Measurement Exchange. The measurement exchange phase is time-critical. In this phase, the stations measure the time elapsed for a response frame transmitted by the responder to be received, processed, and acknowledged by the initiator. The stations timestamp every transmission and reception during the measurement exchange, as shown in Figure 1. Specifically, a timestamp corresponds to when the frame's preamble appears at the antenna connector. An implementation may capture a timestamp at another point in time and correct for the expected time differences [7, §6.3.58.1]. These timestamps are expressed in picoseconds and have a resolution of 0.1 ns. Since radio waves travel at the speed of light (30 cm in 1 ns), the measurement procedure has a theoretical accuracy of around 3 cm. However, the accuracy of the measurements in part depends on the wireless signal's bandwidth [26]. The responding station shares its timestamps of each measurement round ($t1_x$ and $t4_x$ in Figure 1) in the response frame of the next measurement round. Consecutive response frames are spaced apart by a minimum time interval, as defined in the Min Delta FTM parameter. It is defined in units of 100 μs and allows for stations to prepare for the arrival of measurement exchange frames. Upon receipt of response frames, the initiator calculates the average round-trip time using the equation:

$$RTT = \frac{1}{n}\sum_{x=1}^{n}((t4_x - t1_x) - (t3_x - t2_x)) \quad (1)$$

where n is the total number of distance measurements. From the calculated RTT value and knowing that radio signals travel at the speed of light, the initiator derives the distance. For the initiator to track which timestamps correspond to its measurements and account for re-transmissions, the responder includes dialog tokens in its response frames. The first response frame contains a non-zero dialog token and increases sequentially over consecutive response frames within the session. The follow-up dialog token is set to the dialog token of the previous response frame. Setting the dialog token to zero indicates the last transmission for the burst instance.

Termination. The session terminates implicitly after the last burst instance as defined by the session configuration parameters. Additionally, the specification defines three methods for stations to terminate a session [7, §11.24.6.6]. The initiator may transmit a new request with the trigger field set to zero or request a new session with modified configuration parameters. Similarly, the responder may transmit a response frame with its dialog token set to zero.

Table 1: Overview of our distance reduction (◐) and enlargement (◑) attack types, listed with their expected resolution.

Layer	Attack Type	Effect	Resolution
Logical	Inject Response	●	1 ps
Logical	Replay Session	●	1 ps
Logical	Replay Final Response	◐	4 μs
Logical	Terminate Session	○	—
Mixed	Replay PHY-Modif. Response	●	100 ns
Physical	Replay Overshadow	●	1 ps
Physical	Earlier Path Injection	◐	1 ps

2.2 Physical-Layer Configuration

An important physical-layer parameter impacting ranging precision is the signal bandwidth [26], and Wi-Fi FTM allows for a variety of frequency channel and bandwidth configurations. When available, stations use IEEE 802.11ac due to its support for wide bandwidths, implying the usage of Orthogonal Frequency-Division Multiplexing (OFDM) with Binary Phase Shift Keying (BPSK) modulation and a long Guard Interval (GI). The initiator requests its desired configuration in the FTM parameters field and is confirmed by the responder in its first response frame. If the responder agrees to a wider bandwidth, it switches to the respective channel and bandwidth before transmitting its first response frame. An initiator can learn the supported capabilities by inspecting the beacon frame of a responder (e.g., the Very High Throughput (VHT) field for IEEE 802.11ac). Prior research performing an accuracy evaluation has shown meter-level precision using an 80 MHz bandwidth channel within the 5 GHz spectrum [26]. The standard specifies operation for up to 160 MHz in bandwidth; however, at the time of writing no commercially available product supports 80+80 or 160 MHz.

2.3 Wi-Fi FTM Implementation and Support

Wi-Fi FTM is implemented in proprietary firmware and loaded onto the SoC of a Wi-Fi card, and since implementations are proprietary, their source code is not available for public analysis by security researchers. After completing a measurement session, the firmware passes the result to its respective driver in kernel space. As a result, only the averaged measurement result is available to applications.

Support for Wi-Fi FTM was introduced in Android 9 [6], and Android 10 introduced the capability for mobile devices to request the AP location on supported stations, thereby enabling seamless positioning. Additionally, Google provided an example application [5] to aid and inspire developers to build positioning, navigation, and context-aware applications. The Wi-Fi Alliance lists certified devices [3], and includes manufacturers like Qualcomm, Broadcom, and Intel. Furthermore, researchers find growing support for Wi-Fi FTM, including APs from popular network device manufacturers such as Google, Nest, ASUS, Linksys, Eero, and Netgear, though they may not advertise the capabilities in their beacon frames [25].

3 ATTACKS ON WI-FI FTM RANGING

In this section, we present our security analysis, revealing numerous weaknesses allowing an adversary to execute a variety of distance reduction and enlargement attacks. Table 1 lists all attacks by their layer, and whether it allows for distance reduction, enlargement, or session termination. Additionally, we list their resolution, that is, the granularity with which a distance modification can be introduced.

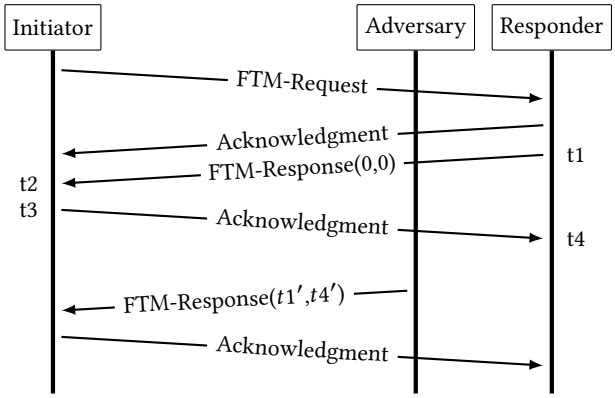

Figure 2: An adversary can reduce or enlarge the distance by spoofing response frames with modified $t1'$, $t4'$ timestamps.

Threat Model. The attacker's goal is to manipulate the distance measured between two stations, as derived by the initiator, without physically displacing the stations. The attacker intends to either enlarge or reduce the measured distance and, in some cases, terminate the execution of a successful measurement. We assume the attacker has complete knowledge of the IEEE 802.11-2016 [7] measurement protocol. We assume a standard Dolev-Yao [13] attacker commonly used to assess the security of wireless protocols which can transmit, eavesdrop, intercept, record, and replay arbitrarily strong radio frequency signals. In this paper, we demonstrate that most attacks can be executed using commercial off-the-shelf hardware (e.g., a cheap Wi-Fi adapter supporting monitor mode and frame injection). We assume the attacker executes attacks without physically tampering with the legitimate stations or modifying their firmware. Finally, we do not restrict the attacker's location unless we note otherwise.

3.1 Spoofing FTM Responses

An attacker can spoof response frames and modify their timestamps, thereby altering the measured distance to *any* attacker chosen value, as shown in Figure 2. Notably, an attacker can introduce either relative or absolute distance manipulations, that is, account for the victim's current position and inject relative distance modifications, or falsify the distance measurement independent of the victim's true position. It is straightforward to introduce false measurements relative to the victim's true position. For example, the attacker can capture timestamps from a previous measurement session, modify timestamp t_4 with a relative change, and then replay the results. However, falsifying the distance independent of the victim's current position, and targeting an arbitrary distance, comes with challenges.

3.1.1 Introducing Absolute Distance Manipulations. To introduce accurate absolute distance manipulations, we need to craft timestamps that account for the initiator's processing time i.e., we need to estimate the difference between the time-of-arrival t_2 and time-of-departure t_3. The timestamps are directly impacted by (i) the

selected physical-layer parameters for radio transmission, (ii) the length of the frame, and (iii) firmware implementations. First, different physical-layer parameters (e.g., bandwidth, modulation scheme, coding rate, guard interval) yield a different duration for frame transmission. Second, the response frame's length determines the number of symbols needed for transmission, thereby impacting its duration. Even though theoretically these durations are constant, the Wi-Fi SoC's processing times introduce certain variance. To put in perspective, an additional processing time of merely 1 ns will result in a distance measurement change of 30 cm. In order to estimate an initiator's processing time (i.e., $t_3 - t_2$), we present a novel benchmarking technique using Wi-Fi FTM as a side-channel.

3.1.2 Benchmarking Wi-Fi System-on-Chips. We present a benchmarking technique which uses Wi-Fi FTM as a side-channel to estimate the initiator processing time (i.e., $t_3 - t_2$). Our technique is independent of physical properties such as the attacker's distance to the initiator and environmental channel effects, and is stable across reboots of the Wi-Fi SoC. To benchmark a device's processing time, we transmit spoofed response frames containing a set of static timestamps. We can then record the reported distance measurement and recover the time-of-flight (i.e., $(t_2 - t_1) + (t_4 - t_3)$). Using the recorded distance and static set of timestamps, we can estimate the average Initiator Processing Time (IPT) with equation:

$$AverageIPT = \frac{1}{n}\sum_{x=1}^{n}((t_4{}' - t_1{}') - ts(2d_x)) \tag{2}$$

where $t_1{}'$ and $t_4{}'$ are our pre-defined timestamps, and $ts()$ converts a distance d (in meters) into the picosecond time-unit at the speed of light. Having measured the AverageIPT, an attacker can construct spoofed timestamps t_1 and t_4 targeting a distance d using equation:

$$t_1 = 0, t_4 = AverageIPT + ts(2d) \tag{3}$$

The AverageIPT's standard deviation, which is dependent on the evaluated Wi-Fi SoC, is a good indication of our ability to accurately craft timestamps. In Section 4.2.2, we show how a low standard deviation yields spoofed distances achieving meter-level accuracy.

3.2 Replaying FTM Sessions and Responses

The standard does not provide secure replay protection, and due to its dialog token design, is vulnerable to a variety of replay attacks. Recall from Section 2.1, the responder chooses a non-zero value for the initial dialog token and the follow-up dialog token is the value of the previous dialog token. However, if dialog tokens are constructed insecurely (e.g., each session starts with a dialog token value of one, or is consecutive), then they are predictable and allow for response frames to be replayed in a future session. Notably, an attacker can now introduce distance modifications without altering the response frame, highlighting the need for secure replay protection in next-generation protocol designs (e.g., using a nonce value, session ID).

3.2.1 Replaying the Session. An attacker can replay any response frame from an earlier session. When stations are mobile, this allows an adversary to trivially reduce or enlarge the measured distance. For example, one can reduce the distance by capturing response frames when the stations are close to each other, replaying them later, and forcing the same distance measurement independent of the stations' new location. As such, an attacker achieves meter-level accurate distance manipulation without modifying any timestamp.

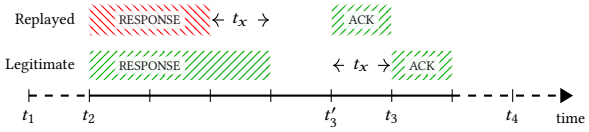

Figure 3: An adversary reduces $t_3 - t_2$ by replaying a response frame resulting in an earlier t_3'. Since the responder reports t_4, and not some t_4', the initiator measures an enlarged RTT.

3.2.2 Replaying the Final Response. When stations are stationary, an attacker can still reduce the measured distance by replaying the final response frame. Recall the first response frame contains FTM parameters and optional tagged parameters (e.g., vendor-specific elements). Consequently, its frame length increases, and therefore the transmitter requires an additional number of OFDM symbols depending on the chosen bandwidth. As a result, the round-trip time of the first measurement exchange is larger than any of the following. We can exploit this to reduce the measured distance, by replaying the final response frame. Since the final response frame has a dialog token of zero, it terminates the session resulting in a single-shot measurement. The attack granularity is directly related to the physical transmission parameters and the original first response frame's length. Say, the first response frame needs one extra OFDM symbol to carry the session parameters, then its transmission time increases with 4 μs. If we then replay a response frame without this symbol, we reduce the round-trip time by 4 μs, or a one-way distance of close to 600 m. Although we have less granular control over the distance reduction, such an attack will have significant implications in a scenario where results are augmenting other positioning systems such as GPS. Furthermore, this attack proves successful even when response frames would be cryptographically protected by the network (e.g., WPA3-Personal), that is, an adversary can jam and acknowledge intermediate response frames such that only the final response is received and (incorrectly) processed.

3.3 Replaying PHY-Modified FTM Responses

To allow for finer granularity in replay attacks, we present an attack that leverages physical-layer modifications. The attack builds upon a combination of two findings. First, IEEE 802.11mc does not provide any guidance on the management of timestamps (i.e., in anticipation of receiving results from the responder) and how to proceed upon the receipt of retransmissions. Second, the specification does not require physical-layer header verification by the receiving station, and therefore an adversary can replay frames using different physical-layer parameters (e.g., a different bandwidth).

Attack Outline. Upon receiving a response frame, the initiator derives t_2 as the preamble's time-of-arrival and initiates the transmission of its acknowledgment at time t_3. However, an attacker can force the acknowledgment to be transmitted at an earlier, or later, time t_3' if the attacker replays the response frame with modified physical-layer parameters. For example, choosing a short guard interval instead of a long guard interval reduces the transmission time by 400 *ns* for each symbol. We denote this total time difference

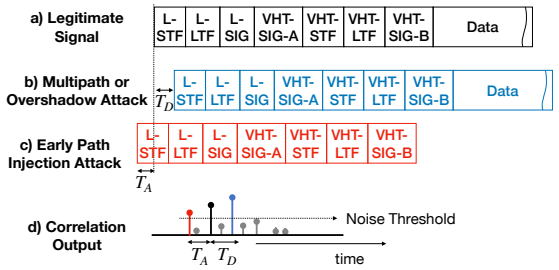

Figure 4: Physical-layer attacks against IEEE 802.11ac result in an earlier (T_A) or delayed (T_D) time-of-arrival estimation.

as t_x and illustrate a timeline as seen by the initiator in Figure 3. However, in order to have a successful attack, we must trick the responder in sending the expected t_4 timestamp. For this purpose, we exploit the mismanagement of temporarily saved timestamps. That is, upon receipt of retransmission (i.e., a response frame with the same dialog tokens), the initiator discards the latter yet acknowledges its receipt. An adversary can exploit this behavior (i.e., acknowledging a retransmission) and transmit its modified response frame right before the responder sends its legitimate response frame. Since the initiator already received a frame using this set of dialog tokens, it will discard the legitimate response, keeping its previous timestamps t_2 and t_3'. However, the responder receives an acknowledgment for its response frame and derives t_4. Unaware the initiator discarded its earlier response, the responder shares its t_1 and t_4, which will be matched to our falsified t_3' timestamp. Then, the initiator falsely derives the RTT as follows:

$$RTT = (t_4 - t_1) - (t_3' - t_2) \tag{4}$$

The granularity by which an attacker can introduce distance manipulations depends on the impact of the modified physical-layer parameters (i.e., the difference between t_3 and t_3', depicted as t_x). Often physical-layer modifications change the number of symbols needed to transmit the data, or the guard interval between them. As a result, they have a low granularity (e.g., one OFDM symbol results in a modification of 4 μs). Fortunately, the impact is averaged out across all measurements within the session. Additionally, an adversary may replay multiple response frames in a session, using a combination of modifications. Our evaluation in Section 4.2.4 shows we can achieve fine-grained manipulations of up to 100 ns. Furthermore, this attack proves successful even when response frames would be cryptographically protected by the network (e.g., WPA3-Personal), that is, similar to Section 3.2.2, an adversary can jam the initiator in order to obtain and replay fresh response frames.

3.4 Physical-Layer Distance Manipulations

Physical-layer distance manipulation attacks exploit a station's signal processing chain to control its Time-of-Arrival (ToA) estimation. An attacker who is capable of manipulating the ToA estimation can then introduce distance enlargements and reductions.

3.4.1 Receiver Design. The IEEE 802.11ac standard is often referred to as Very High Throughput (VHT), and in Figure 4(a) we show its simplified physical-layer frame format. The header's Training Fields (TF) are used for gain control, packet detection, and clock synchronization. Usage of VHT-LTF for ToA estimation is the preferred choice in IEEE 802.11ac receiver designs, though a receiver can combine samples from multiple fields for ToA estimation [20]. The receiver uses either a correlation-based approach or a Channel Impulse Response (CIR) technique for ToA estimation, both of which are threshold-based. Although the highest correlation peak can be used for packet detection and providing correct payload data, it will not provide accurate distance estimation in multipath and NLoS scenarios. As shown by the correlation output in Figure 4d, the multipath component arriving at the receiver with a delay of T_D has higher power than the direct path. Using the highest peak would give an incorrect distance estimation. Therefore, receivers typically search for the direct path in a back-search window (i.e., a peak with power above some noise threshold). From [26], we learn that Wi-Fi FTM's ranging accuracy in multipath scenarios can be off by more than 5 m, giving us an indication of the maximum distance manipulations we can introduce. For example, if the measured distance is more than the actual distance, it implies the receiver has used a multipath for ToA estimation. Similarly, an underestimated distance measurement indicates the receiver used a side peak detected during the back-search procedure.

3.4.2 Replay Overshadow Attack. An attacker can achieve distance enlargement by replaying the legitimate frame with a higher power, after a delay of T_D (Figure 4b). Wi-Fi FTM uses long OFDM symbols ($\approx 4\ \mu s$ per symbol) to represent both header and payload data at the physical layer. Using $T_D = 100\ ns$, an attacker enlarges the distance by 30 m. Both the attacker and legitimate signal overlap, and the receiver cannot distinguish between their arrival time. Even though the initial samples collected in the time T_D are unaffected by the attack signal, they are not sufficient to perform ToA estimation and therefore are discarded as noise. The receiver performs cross-correlation or CIR estimation using the entire LTF sequence for ToA estimation. The attack succeeds if the attack signal's correlation peak is the highest, and the receiver does not detect the legitimate peak during back-search, either because the delay T_D is more than the back-search window or the power of the legitimate peak is below the noise threshold. Additionally, the receiver detects correct data for the attacker's peak as it is a copy of the legitimate signal.

We note replay attack detection techniques have been presented for ultra-wideband ranging systems [52], where the duration between two pulses is more than the maximum time a signal takes to travel from one device to another, and the presence of extra energy in the channel indicates the presence of the adversary. We cannot apply such approaches to the OFDM symbol since a significant proportion of the legitimate symbol is hidden beneath the attack signal, preventing legitimate signal's detection at the receiver. Therefore, it is required to redesign the physical layer to enable secure ranging.

Spoofing Acknowledgments. A special case of an overshadow attack is when an attacker takes advantage of the acknowledgment's static and known data. An attacker can transmit a spoofed acknowledgment earlier or later and with higher power than the legitimate frame to modify the round-trip time estimate and, as a result, manipulate the distance. The attack succeeds as the receiver *locks on* to the higher power peak or one of its side peaks for ToF estimation since the LTF sequence of both legitimate and attack signal overlap.

3.4.3 Earlier Path Injection Attack. If the back-search algorithm of a receiver uses a side peak for ToA estimation, the reported distance measurement may be an underestimate, with an error exceeding the imprecision of the system [26]. Since these frames report correct data, it suggests the receiver uses the strongest peak's arrival time for packet detection and data recovery, as using a lower-power side peak's arrival time results in incorrect data. As such, payload detection and ToA estimation in Wi-Fi FTM are non-binding; the receiver uses the highest correlation peak for data detection and an earlier peak for the ToA estimate. The back-search procedure is critical for a distance measurement system; otherwise it could not operate under multipath and NLoS scenarios. An attacker can exploit the receiver design to perform a distance reduction attack, that is, insert a peak within the back-search window (Figure 4c). Since Wi-Fi FTM's physical-layer receiver design optimizes for both ToA estimation and data detection, this attack cannot be prevented.

Attack Outline. In order to inject an earlier peak, it is sufficient to transmit only the header part of the frame, and carefully control its arrival time and power. Since the preamble of a frame is static and known, it can be replayed or transmitted early. We consider the earlier path injection attack successful only when the attacker introduces a peak within the back-search window, the peak's power is between noise threshold and highest peak, and the data is detected correctly. As such, we have to accomplish two main goals. First, the attack signal should arrive T_A time earlier than the legitimate frame at the receiver, and T_A should be smaller than the back-search window. The signal will not be used for ToA estimation if it arrives too early or too late. The attacker should know the distance between the devices and use benchmarking to estimate the transmission time of acknowledgment frames. This information is sufficient to predict the arrival time of the legitimate acknowledgment frame, and the attacker can then transmit the attack signal accordingly. Second, the attacker should control the attack signal's power. If the power is higher than the legitimate signal, the receiver *locks on* to the attacker's peak for packet detection and ranging fails due to incorrect data. If the power is below the noise threshold, the signal is not detectable during the back-search procedure. Thus, the attack signal's power should be higher than the noise threshold but lower than the legitimate signal. Though a receiver can choose a noise threshold in advance, the legitimate signal's received signal strength varies depending on the channel condition. Research [38] has shown the feasibility of predicting received signal strength at a receiver location in the context of channel-based key establishment.

4 EXPERIMENTAL EVALUATION

In this section, we present the experimental setup and evaluation.

4.1 Experimental Setup and Implementation

We evaluate all commercially available products advertising support for Wi-Fi FTM, covering a wide variety of smartphones, APs, and off-the-shelf Wi-Fi cards. As a responder, we test Google Wi-Fi (Qualcomm IPQ4019), Google Nest (Qualcomm QCS404), Compulab WILD (Intel AC-8260), and ASUS RT-ARCH13 (Qualcomm IPQ4018), and as initiator a Google Pixel 4 XL (Qualcomm Snapdragon 855) smartphone, and Compulab WILD. Figure 5 shows an overview of these devices. In total, it allows for six distinct setups, all of which are configured to a 5 GHz channel with 80 MHz bandwidth (the default configuration for most devices and enables meter-level accuracy [26]). We placed all devices in an office environment, within line-of-sight of each other. For the adversary, we use a TP-LINK AC600 Archer T2UH Wi-Fi dongle supporting IEEE 802.11ac with 80 MHz in bandwidth. Physical-layer features such as carrier sensing remain enabled, and thus we do not require firmware or driver modifications. As a result, we can implement and execute all attacks using commercial off-the-shelf hardware. In Appendix A, we provide detailed implementation and configuration information.

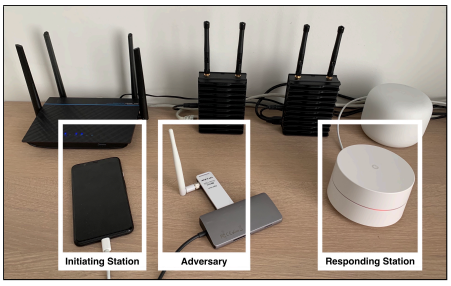

Figure 5: Experimental setup, depicting a Google Pixel 4 XL and Google Wi-Fi AP as an initiating and responding station.

Table 2: Benchmarking results for each setup, listing their default response frame size in bytes and standard deviation.

Initiator	Responder	Size	Mean IPT	σ
Pixel 4 XL	Google Nest	85	85 807 ns	4 ns
Pixel 4 XL	Google Wi-Fi	85	85 816 ns	5 ns
Pixel 4 XL	Compulab WILD	74	81 906 ns	5 ns
Co. WILD	Google Nest	62	77 478 ns	36 ns
Co. WILD	ASUS RT-ACRH13	62	77 433 ns	36 ns
Co. WILD	Compulab WILD	74	81 433 ns	36 ns

4.2 Protocol-Layer Attack Evaluation

4.2.1 Injecting FTM Responses for Relative Distance Manipulations. We evaluate relative distance manipulations by placing the stations 20 m apart, 60 cm off the floor, in direct line-of-sight. We then capture and replay a response frame from a previous session and reduce timestamp t_4 by 100 000 ps, representing a one-way distance of 15 m. We note t_4 is changed by an arbitrary value, and can be chosen with 1 ps granularity. Evaluating 1 000 sessions against each setup, we obtain distance reductions between 14.20 m and 15.26 m. It proves to be a highly effective method to modify the distance, requiring minimum knowledge about a target station. Similarly, an adversary can enlarge the distance by increasing the t_4 timestamp.

4.2.2 Injecting FTM Responses for Absolute Distance Manipulations. In order to make absolute distance manipulations, we first have to benchmark the Wi-Fi card. As we force a single-shot measurement, we need to benchmark the first response frame. This frame includes optional fields such as vendor-specific information elements, and thus has a variable length. We present our benchmarking results in Table 2 and make a few observations. First, we find that a larger

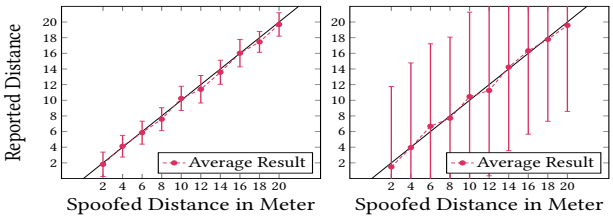

Figure 6: An adversary is able to modify the distance with meter-level precision, as demonstrated with the Google Pixel 4 XL and WILD AP (left) and WILD and Nest AP (right).

frame size results in a higher mean IPT. This is expected, since one additional OFDM symbol requires an additional 4 μs. For example, with Google Nest we obtain an IPT of 77.4 μs, and similarly, 81.4 μs using Compulab WILD, which requires an additional symbol for its larger frame size (hence a 4 μs difference). Second, we find differences in standard deviation between both Wi-Fi SoC vendors. Qualcomm results deviate less from the average and thus allow our attacks to have a higher level of accuracy. Next, we benchmarked both WILD devices having an Intel AC-8260 card and obtained similar benchmarking results. Some deviation is expected due to chip fabrication inaccuracies, as well as the naturally large standard deviation on the Intel Wi-Fi SoCs. Specifically, we find a maximum offset of 5 ns or a one-way distance of well under 1 m. Similarly, we benchmarked two Google Pixel 4 XL smartphones with a Qualcomm Wi-Fi SoC, indicating an offset of merely 1 ns, which is a better result due to the low standard deviation present on the Wi-Fi SoC.

We then evaluate the accuracy of our attack, and spoof the distance of an initiating station between 2 m and 20 m. In Figure 6, we present our results, showing averages which are off by no more than 75 cm, having a standard deviation matching the respective benchmarking results (i.e., the standard deviation of our distance modification is directly related to the standard deviation of the mean IPT). As such, we achieve meter-level accuracy using a Google Pixel 4 XL as initiator (i.e., an offset of no more than 5 ns or 1.50 m). Next, we evaluated the impact of a moving target station using the Google Pixel 4 XL and Google Wi-Fi. The results vary slightly within the expected standard deviation, confirming that there are no location restrictions for the adversary, initiator, and responder.

4.2.3 Replaying FTM Sessions and Responses. We evaluate the session replay attack between the Google Pixel 4 XL and the Google Wi-Fi AP. The session was captured with the devices placed 4 m and 20 m apart in line-of-sight with the Pixel 4 XL reporting 3.82 m and 20.27 m respectively. We then replay all measurement results of the captured session, where the position of the adversary is arbitrary, since frames are replayed without modifying their timestamps. Additionally, the legitimate response frames of the responder are not impacting our results, as its frames are discarded by the initiator due to replay detection. When replaying the captured sessions, we obtain a result of 1.72 m and 19.40 m respectively. Some offsets are expected due to the standard deviation of the Wi-Fi SoCs, and proves successful with both Intel and Qualcomm-based stations.

We then evaluate replaying the final response frame. The Google Pixel 4 XL using a Qualcomm SoC is not suitable for this attack since it discards large negative measurement results and instead reports a failed session. Therefore, we evaluate stations using Intel Wi-Fi SoCs. Specifically, we use the Compulab WILD as an initiator and the Compulab WILD and ASUS RT-ARCH13 as responders. Their response frames have a different size and require a different number of symbols for transmission. We evaluate the attack by running 1 000 sessions on each of the devices. Using the ASUS RT-ARCH13 and Compulab WILD, we see the round-trip time decrease by 3.99 μs and 8.02 μs respectively i.e., a distance reduction of 598.62 m and 1,201.58 m respectively. This round-trip time reduction is expected as the results map to the duration of one and two OFDM symbols.

Table 3: Replaying a response frame with physical-layer modified parameters results in falsified round-trip times, and allows for various distance reduction and enlargements.

Modified PHY Parameter	Avg. RTT	Avg. Distance
Baseline	104 ns	15.61 m
20 MHz Bandwidth	-25 848 ns	-3 877.20 m
40 MHz Bandwidth	-9 841 ns	-1 476.09 m
Short Guard Interval	921 ns	138.14 m
QPSK 1/2 Modulation	4 113 ns	616.92 m
16-QAM 1/2 Modulation	6 105 ns	915.73 m
256-QAM 5/6 Modulation	6 102 ns	915.29 m

4.2.4 Replay Attack with Physical-Layer Modifications. We configure the Compulab WILD and Google Nest to run distance measurements using ASAP=1 and SPB=3 on a 80 MHz bandwidth channel. In Table 3, we present a baseline using the default physical-layer configuration of a long guard interval, BPSK 1/2 modulation, and a single spatial stream, where the measured distance is 15.61 m. Under this configuration, transmission of the first response frame requires 4 OFDM symbols since it is 62-bytes in size (as listed in Table 2). We now evaluate 1 000 sessions where we replay a response frame using a modified physical-layer configuration. For example, when we select a short guard interval, t_3 reduces by a value of $4 * 400\ ns = 1600\ ns$ increasing the estimated in-flight time of the wireless signal, thereby enlarging the measured distance. We observe an additional 800 ns to the RTT value, which is expected, since our modification is averaged over two measurements (i.e., SPB=3). Similarly, a change in bandwidth or modulation will require a different number of symbols and result in a distance modification of a multiple of 4 μs. For example, QPSK 1/2 requires 2 OFDM symbols, and thus enlarges the RTT by 8 μs (or 4 μs averaged).

We note that, in order to be successful, the spoofed response frame has to be injected within a small time window before the legitimate response frame. Empirical results show the time window is around 1.5 ms. Achieving this level of accuracy using commodity hardware is hard, though can be solved by repeatedly transmitting the spoofed frame, such that one is likely to fall within the window. This trick has no negative impact on the overall attack success ratio.

Increasing Granularity. Multiple physical-layer parameters can be modified to increase the granularity of the modified distance. Similarly, replaying one response frame in a session with a higher SPB will increase the granularity since the impact is averaged over

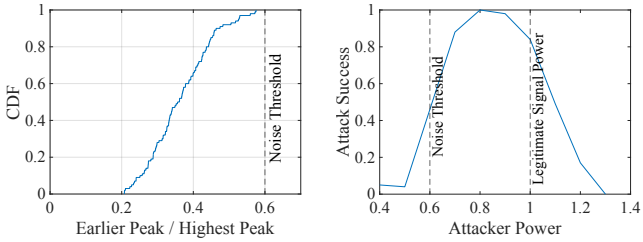

Figure 7: Receivers use a noise threshold higher than any side peaks (a). An earlier path injection requires the injected peak's power to be within a threshold (b).

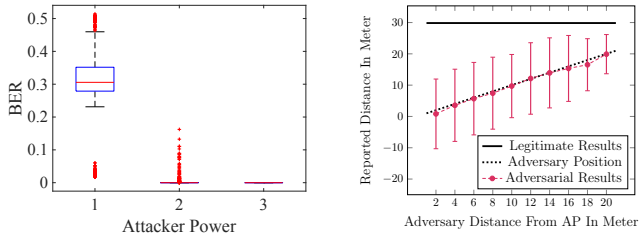

Figure 8: In an overshadow attack, a weak attacker power level leads to increased bit errors (a). An attacker spoofing acknowledgments achieves accurate distance reductions (b).

several measurements, and thereby yield more realistic results. For example, we placed a Google Pixel 4 XL and Google Wi-Fi 20 m apart, using a SPB=8 and 80 MHz default configuration. We now replay a frame with a short guard interval, 20 MHz bandwidth, and 16-QAM 1/2 modulation, and successfully reduce the distance from an average 20.24 m to a more fine-grained 2.40 m.

4.3 Physical-Layer Attack Evaluation

In physical-layer attacks, an attacker interferes with the legitimate signal at the sample level and exploits ToA estimation techniques implemented in Wi-Fi FTM receivers. For accurate ToF estimation, the receiver typically uses the highest correlation peak for data detection and an earlier peak with power above some threshold for ToF estimation. To access sample-level information, we perform a MATLAB simulation of the IEEEs 802.11ac VHT waveform with a TGac fading channel (Model-B). We use VHT-LTF training sequences for ToF estimation and a back-search window of 100 *ns*. Results are shown for BPSK modulation and a legitimate signal-to-noise ratio of 20 *dB*, a typical channel condition for LoS communication. Figure 7a shows side peaks within the back-search window to the highest peak power distribution. The receiver has to choose the noise threshold's value to differentiate the direct path signal from the side peaks based on the power distribution. In this particular receiver design, setting the noise threshold to 0.6 minimizes the false positives. Using lesser values trigger the detection of side peaks as an earlier path, and setting a higher value misses the direct path; in both conditions, the distance estimation is incorrect.

4.3.1 Replay Overshadow Attack. In a replay overshadow attack, the attacker replays a signal with a higher power, after a delay of T_D. The attacker succeeds in a distance enlargement if the receiver uses its signal for ToA estimation and data detection. As shown in Figure 8a, the attack signal has no bit error if its power is three times (≈ 4.8 *dB*) higher then the legitimate signal. The attack signal should have sufficient power to overshadow the legitimate signal to prevent its detection; it does not need to jam or saturate the receiver to prevent legitimate signal detection. If the attack signal has enough power, the back-search window does not have any bearing on this attack, and the attacker can choose the delay T_D to be smaller or higher than the back-search window.

4.3.2 Spoofing Acknowledgments. An attacker can spoof acknowledgment frames, since it contains static data, and the MAC address of the responder is known in advance. We configure the Compulab WILD initiator to the same MAC address as our adversarial Wi-Fi dongle. As such, the adversary as well as legitimate station acknowledge response frames. If both signals arrive at the receiver with a certain time difference, then the strongest signal is used for ToF estimation. Using WILD as initiator, we can evaluate the attack against the Google Nest, WILD, and ASUS APs. In Figure 8b, we plot results for the ranging session between the WILD and Google Nest AP. In absence of an adversary, the initiator reports a distance of 29.84 m and a standard deviation of 0.25 m. As the adversary starts acknowledging frames, and moves towards the AP, we find the reported distance decreases accordingly. Specifically, we obtain distances which deviate at most 1.47 m from the true distance between the adversary and AP (Figure 8b). As both the legitimate and attacker signals have the same transmit power, the receiver uses both signals, side peaks, and peaks due to the constructive multipath components for the ToF estimation; therefore, we observe a notably higher standard deviation of up to 11.58 m. These findings expose a fundamental flaw in the protocol design, whereby an adversary capable of acknowledging response frames can effortlessly introduce distance modifications. Wi-Fi FTM MAC address randomization can prevent a weak adversary from acknowledging response frames in a timely manner, however, a fundamental solution would require the protection of acknowledgment frames.

4.3.3 Earlier Path Injection. An earlier path injection is successful if the attacker injects a signal (i.e., frame header) within the back-search window, with its power under certain thresholds. Figure 7b shows the success probability for different signal strengths, assuming the signal arrives within the back-search window. A low power signal is discarded as noise, and a higher power distorts channel estimation and prevents data detection. The attack is successful only when an earlier peak inserted by the attacker is used for distance estimation, and the legitimate signal is used for data recovery.

4.4 Discussion

Throughout the paper, we presented and evaluated a wide-variety of attacks against Wi-Fi FTM. We now discuss firmware-specific findings, attack success ratios, detection techniques, and how an adversary can target several access points in a localization system.

4.4.1 Firmware-Specific Findings. Throughout our analysis, we evaluated a variety of firmware versions, and observed firmware-specific findings. In Table 4, we present an overview of all initiators. First, some filter distance measurement results that are not within

Table 4: Overview of Wi-Fi FTM initiators and supported firmware, accepted measurement range, and common flaws.

Wi-Fi Card	Firmware	Range	Terminate	PHY-Verif.	Delta Verif.	Retrans.
Qual. Snap. 855	Unknown	[-22.5,+∞]	●	○	○	●
Intel AC-8260	Ver. 31	[-∞,+∞]	●	○	○	●
Intel AC-8260	Ver. 36	[-10,+100]	●	○	○	—
Intel AC-8265	Ver. 34, 36	[-10,+100]	●	○	○	—
Intel AX-200	Ver. 53	[0,+∞]	●	○	○	●
Intel AX-200	Ver. 55	[-∞,+∞]	●	○	○	●
Intel AX-200	Ver. 57, 58	[0,+100]	●	○	●	—

predefined bounds. For example, the Google Pixel 4 XL (Qualcomm Snapdragon 855) discards distances below -22.5 m and marks the session as failed. Without filter, an adversary can spoof RTTs up to the limit of its underlying representation (e.g., a 4-byte signed integer). Even when bounds are put on the RTT, our evaluation shows numerous attacks remain successful. Second, we found none perform physical-layer verification and all allow for retransmissions (Section 4.2.4). Third, we found only the latest Intel AX-200 firmware versions perform Min Delta FTM window verification (i.e., a response frame can only be sent within the expected time window), though does not limit us since frames can be sent repeatedly until one is sent within the window. Finally, all stations are vulnerable to denial-of-service attacks, as discussed in Appendix B.

4.4.2 Success Ratio. In our protocol-layer attacks, we often force a single-shot measurement session, discarding legitimate results. It requires we inject a response frame before the first legitimate measurement result, which is sent after a Min Delta FTM time window. As this value increases, an adversary receives a larger time window to inject frames, and hence has greater odds for a successful attack. As such, the true distance between stations, and the adversary location, becomes arbitrary. Put differently, the precise arrival time of the spoofed response frame has no impact on the derived distance. We find that a Min Delta FTM of 1.80 ms is sufficient to achieve over 98% success ratio using commodity hardware, where any failures are due to our NIC not capturing all requests (thereby not initiating the attack) and variable processing times (e.g., host machine). Said time window is present in all evaluated devices and configurations.

4.4.3 Detection Techniques. The victim may detect an attack for a variety of reasons, for example, two response frames for the same dialog token are received. In order to prevent this, an adversary can jam the legitimate frame using commodity hardware [58]. Next, Channel State Information (CSI) may be used to detect spoofed frames, for example, RSSI [15]. However it is not foolproof since the legitimate station can still be mimicked [17]. In any event, without fundamental security enhancements, the victim remains unable to determine which of the frames is legitimate, and therefore may be forced to discard the session, effectively causing a denial-of-service.

4.4.4 Localization Systems. Wi-Fi FTM-based localization systems can be built using, for example, a multilateration algorithm [30]. Assuming a victim performs distance measurements with several APs, its position can still be trivially spoofed. That is, an adversary introduces relative distance measurements with each client-AP pair. Practically, there is no increase in complexity, as each measurement session occurs non-concurrently on already-known radio channels.

4.4.5 Cryptographic Protection. Prior works on secure ranging systems demonstrated that protecting exchanged data is insufficient, in addition, one needs to protect the measurement exchange [11, 53]. As a result, using typical network protection (e.g., WPA3-Personal) is insufficient to fundamentally secure the protocol. We highlighted this in several of our attacks, for example, in Section 3.2 and 3.3 we presented replay attacks which succeed even if frames are protected. Furthermore, a typical pre-shared key network configuration would be futile since an adversary can recover the encryption key if the passphrase is known (e.g., an indoor shopping center network). Finally, in Section 3.4, we presented physical-layer attacks which are effective even when the data-layer would be secured. Therefore, in Section 5.2, we discuss the physical-layer limitations in securing Wi-Fi FTM, and present approaches for securing next-generation distance measurement protocols (e.g., secure channel estimation).

5 APPROACHES AND LIMITATIONS FOR SECURING THE WI-FI FTM PROTOCOL

In this section, we present recommendations to increase Wi-Fi FTM security, and discuss the physical-layer limitations and challenges.

5.1 Firmware Recommendations

The following recommendations improve the resilience of Wi-Fi FTM to distance modification attacks and can be implemented in the firmware without any specification modifications. First, the initiator needs to *verify the received response frames' MAC and PHY header fields* i.e., the fields influencing frame demodulation, decoding, and processing times (e.g., bandwidth, modulation scheme, coding rate, and guard interval) and discard measurements that do not match the negotiated session parameters. Second, the initiator can *add a random time delay before transmitting an acknowledgment*, artificially enlarging the RTT by a value known only to the initiator. It prevents benchmarking of Wi-Fi SoCs and renders replayed sessions and responses inaccurate. Third, an initiator is incapable of differentiating replayed frames from a responder or a potential attacker. As a safety precaution, *the measurement result from replayed response frames should be discarded, even for legitimate retransmissions*, resulting in the protection against (physical-layer modified) replay attacks. Fourth, the initiator should *put a lower and upper bound for valid measurement results* (for example, [-10,100] m), and filter out those exceeding them. Specifically, each individual measurement result should be filtered, such that large distance modifications are no longer averaged out over a session. Fifth, *MAC address randomization should be mandatory*, at least within a sufficiently large subnet to avoid predictability. This forces an adversary to perform real-time attacks since these addresses can no longer be predicted or re-used. Finally, the responder should *respond using a random initial dialog token.* Sequential dialog tokens are insufficient, as an adversary can send an arbitrary number of anonymous requests to increase the local token counter, quickly forcing it to roll

over. Though tokens are only one byte in size, a random value will lower its predictability and tighten any response time requirements.

5.2 Physical-Layer Limitations and Challenges

Since cryptographic protection of data does not prevent physical-layer attacks, a system's security depends on its radio receiver design and implementation. For example, the overshadow and early path injection attacks presented in Section 3.4 take advantage of the receiver's inability to estimate the channel correctly and securely. When stations have perfect knowledge of the channel between them, the receiver can eliminate the attacker signal from its time-of-arrival estimation, since the attacker signal will appear to be a multipath. Similarly for an early path injection attack, the attacker may succeed in injecting an earlier path in the header fields, however will fail to inject the correct payload without distorting the physical layer's energy distribution. Since Wi-Fi FTM OFDM symbols have repeating sequences (e.g., a cyclic prefix), an attacker still needs to guess the symbol's remaining part without making any detectable energy distortions. Only when perfect channel estimation is possible for an OFDM-based system, can it be secured against the physical layer attacks we presented within this paper. To summarize, systems remain fundamentally vulnerable in scenarios where multipath cannot be estimated securely and accurately.

Wi-Fi Next Generation Positioning. The IEEE formed a task group to build and standardize a new positioning standard, referred to as Wi-Fi Next Generation Positioning (Wi-Fi NGP; IEEE 802.11az [28]). To date the standard remains under development, is not publicly available, and is expected to be published no earlier than 2023. Based on documents published on the task group website, a technique for secure time-of-arrival estimation is to be adopted. Specifically, they propose that time-of-arrival estimation is performed using BPSK modulated OFDM symbols, which are constructed using cryptographically generated bits, adapted to remove any repeating sequences; for example, the cyclic prefix is replaced with zero-padding. Attack detection then depends on the receiver's ability to estimate the channel correctly and securely. For example, if the residual energy is higher than a threshold after removing power coming through multiple paths, an attack is detected. As Wi-Fi FTM and Wi-Fi NGP's security may depend on secure channel estimation, we recommend next-generation protocols to explore the possibility of secure channel estimation under different channel and attacker models. Channel estimation is a well-studied topic; however, secure channel estimation is not yet possible opening the space for further research. For example, a MitM adversary can control the channel [14], especially when one or more devices can move and channel behavior is unpredictable for the receiver.

6 RELATED WORK

The recent increase in the number of ranging application requirements has resulted in the development and deployment of various wireless ranging and localization technologies, including safety- and security-critical applications. The majority of these ranging technologies were shown to be vulnerable to attacks ranging from simply relaying the signal between honest nodes to injecting messages at the physical layer with severe implications, e.g., make fraudulent contactless payments [19, 49], steal a car [18], and spoof entire locations [56, 60]. Clulow et al. [12] introduced physical-layer attacks such as early detect and late commit, demonstrating the importance of symbol structure in realizing secure ranging systems. Several studies evaluated the feasibility of these attacks on ISO 14443 RFID [16], UWB-IR [41, 42], chirps [45], and low-power standards such as ZigBee [39].

The ubiquitous availability of Wi-Fi networks prompted the design and development of several Wi-Fi-based ranging and positioning techniques [31, 32, 34–36, 50, 59, 61, 63]. After the introduction of Wi-Fi FTM into IEEE 802.11-2016 and several studies [10, 26, 62] indicating meter-level accuracy in low-multipath environments [24, 26], Wi-Fi FTM has seen adoption in numerous systems [8, 21, 27, 29, 48, 51, 64], with notable applications such as indoor positioning [10, 30, 37, 62] and vehicular positioning [27]. Furthermore, positioning requirements for next-generation wireless networks [28] are currently being defined based on Wi-Fi FTM. The lack of a thorough security analysis of Wi-Fi FTM has paved way for its use in security-critical scenarios such as network onboarding of Internet of Things (IoT) devices [33] and is also a key feature of Wi-Fi Aware, a neighbor-aware networking protocol said to provide benefits such as geofencing [4]. This highlights the importance of our work, since to our knowledge, there is no prior security evaluation of Wi-Fi FTM. In this work, we take the first step to realizing a secure Wi-Fi ranging system by understanding the security issues of Wi-Fi FTM and proposing measures to address them in next-generation systems. Although attacks on other ranging techniques described previously apply to Wi-Fi FTM, there are attacks specific to the Wi-Fi design that we demonstrate in this work. For example, Wi-Fi supports multiple bandwidth and guard interval configurations, resulting in distance modification attacks, as shown in our work. Such an attack does not apply to secure-UWB ranging systems, as these generally use fixed bandwidth and use secure symbol structure [1]. Secure ranging systems [43, 44, 46, 47, 52, 53, 55] proposed in prior work are often based on the concept of distance bounding [9] and impose tight processing timing constraints. They also require the use of specialised signals that are largely unsuitable for integration with Wi-Fi, further motivating our study. Finally, we note a few industry patent applications have proposed secure out-of-band channels to share unique dialog tokens and nonces [54, 57], or share timestamps in protected range reports [2].

7 CONCLUSION

In this paper, we performed the first security analysis of Wi-Fi Fine Timing Measurement (Wi-Fi FTM). We identified weaknesses on both the logical and physical-layer, and presented a wide variety of attacks which allow an adversary to introduce distance reductions or enlargements without physically displacing the stations. We evaluated commercial access points, smartphones, and off-the-shelf Wi-Fi cards from various vendors, and demonstrated we can modify the distance to any attacker-chosen value with meter-level precision. Finally, we presented security recommendations for protocol designers and firmware developers to aid in the development and implementation of secure next-generation positioning protocols.

ACKNOWLEDGMENTS

This work was partially supported by NSF grant 1850264.

REFERENCES

[1] 3db Access. 2020 (Accessed 20 December 2020). UWB Secure Ranging-3dB. https://www.3db-access.com/technology.

[2] Carlos Horacio Aldana, Alireza Raissinia, Santosh Vamaraju, and Kumar Anand. 2018. Secure fine timing measurement exchange. US Patent 10,064,057.

[3] Wi-Fi Alliance. 2020 (Accessed 2 April 2020). Product Finder | Wi-Fi Alliance. https://www.wi-fi.org/product-finder.

[4] Wi-Fi Alliance. 2020 (Accessed 3 December 2020). Wi-Fi Aware | Wi-Fi Alliance. https://www.wi-fi.org/discover-wi-fi/wi-fi-aware.

[5] Android. 2020 (Accessed 17 June 2020). Connectivity Samples Repository. https://github.com/android/connectivity-samples.

[6] Android. 2020 (Accessed 18 June 2020). Wi-Fi location: ranging with RTT. https://developer.android.com/guide/topics/connectivity/wifi-rtt.

[7] IEEE Standards Association et al. 2016. IEEE Std 802.11-2016, IEEE Standard for Local and Metropolitan Area Networks—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2016.

[8] Leor Banin, Uri Schatzberg, and Yuval Amizur. 2016. WiFi FTM and map information fusion for accurate positioning. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*.

[9] Stefan Brands and David Chaum. 1993. Distance-bounding protocols. In *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 344–359.

[10] Markus Bullmann, Toni Fetzer, Frank Ebner, Markus Ebner, Frank Deinzer, and Marcin Grzegorzek. 2020. Comparison of 2.4 GHz WiFi FTM-and RSSI-Based Indoor Positioning Methods in Realistic Scenarios. *Sensors* 20, 16 (2020), 4515.

[11] Jolyon Clulow, Gerhard P. Hancke, Markus G. Kuhn, and Tyler Moore. 2006. So Near and Yet So Far: Distance-bounding Attacks in Wireless Networks. In *Proceedings of the Third European Conference on Security and Privacy in Ad-Hoc and Sensor Networks* (Hamburg, Germany) *(ESAS'06)*. Springer, 83–97. https://doi.org/10.1007/11964254_9

[12] Jolyon Clulow, Gerhard P Hancke, Markus G Kuhn, and Tyler Moore. 2006. So near and yet so far: Distance-bounding attacks in wireless networks. In *European Workshop on Security in Ad-hoc and Sensor Networks*. Springer, 83–97.

[13] Danny Dolev and Andrew Yao. 1983. On the security of public key protocols. *IEEE Transactions on information theory* 29, 2 (1983), 198–208.

[14] Simon Eberz, Martin Strohmeier, Matthias Wilhelm, and Ivan Martinovic. 2012. A Practical Man-In-The-Middle Attack on Signal-Based Key Generation Protocols". In *Computer Security – ESORICS 2012"*. "Springer Berlin Heidelberg".

[15] Daniel B Faria and David R Cheriton. 2006. Detecting identity-based attacks in wireless networks using signalprints. In *Proceedings of the 5th ACM workshop on Wireless security*. 43–52.

[16] Manuel Flury, Marcin Poturalski, Panos Papadimitratos, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. 2010. Effectiveness of distance-decreasing attacks against impulse radio ranging. In *Proceedings of the third ACM conference on Wireless network security*. 117–128.

[17] Aurélien Francillon, Boris Danev, and Srdjan Capkun. 2011. Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars. In *Network and Distributed System Security Symposium (NDSS)*.

[18] Aurélien Francillon, Boris Danev, and Srdjan Capkun. 2011. Relay attacks on passive keyless entry and start systems in modern cars. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. Eidgenössische Technische Hochschule Zürich, Department of Computer Science.

[19] Lishoy Francis, Gerhard P Hancke, Keith Mayes, and Konstantinos Markantonakis. 2011. Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones. *IACR Cryptol. ePrint Arch.* 2011 (2011), 618.

[20] A. Gaber and A. Omar. 2012. A study of TDOA estimation using Matrix Pencil algorithms and IEEE 802.11ac. In *2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS)*. 1–8.

[21] Guangyi Guo, Ruizhi Chen, Feng Ye, Xuesheng Peng, Zuoya Liu, and Yuanjin Pan. 2019. Indoor Smartphone Localization: A Hybrid WiFi RTT-RSS Ranging Approach. *IEEE Access* 7 (2019), 176767–176781.

[22] Gerhard Petrus Hancke. 2005. A Practical Relay Attack on ISO 14443 Proximity Cards.

[23] Gerhard P. Hancke. 2006. Practical Attacks on Proximity Identification Systems (Short Paper). In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 328–333.

[24] Berthold KP Horn. 2020. Doubling the Accuracy of Indoor Positioning: Frequency Diversity. *Sensors* 20, 5 (2020), 1489.

[25] Berthold KP Horn. 2020 (Accessed 22 February 2020). Indoor positioning using time of flight with respect to Wi-Fi access points. http://people.csail.mit.edu/bkph/ftmrtt_aps.

[26] Mohamed Ibrahim, Hansi Liu, Minitha Jawahar, Viet Nguyen, Marco Gruteser, Richard Howard, Bo Yu, and Fan Bai. 2018. Verification: Accuracy evaluation of WiFi fine time measurements on an open platform. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 417–427.

[27] Mohamed Ibrahim, Ali Rostami, Bo Yu, Hansi Liu, Minitha Jawahar, Viet Nguyen, Marco Gruteser, Fan Bai, and Richard Howard. 2020. Wi-Go: accurate and scalable vehicle positioning using WiFi fine timing measurement. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*. 312–324.

[28] IEEE. 2021 (Accessed 25/03/2021). IEEE P802.11 - NEXT GENERATION POSITIONING STUDY GROUP. http://www.ieee802.org/11/Reports/tgaz_update.htm.

[29] Shazal Irshad, Eric Rozner, Apurv Bhartia, and Bo Chen. 2020. Rethinking Wireless Network Management Through Sensor-driven Contextual Analysis. In *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications*. 92–97.

[30] Nicolas Jathe, Michael Lütjen, and Michael Freitag. 2019. Indoor Positioning in Car Parks by using Wi-Fi Round-Trip-Time to support Finished Vehicle Logistics on Port Terminals. *IFAC-PapersOnLine* 52, 13 (2019), 857–862.

[31] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM computer communication review*, Vol. 45. ACM, 269–282.

[32] Steven Lanzisera, David Zats, and Kristofer SJ Pister. 2011. Radio frequency time-of-flight distance measurement for low-cost wireless sensor localization. *IEEE Sensors Journal* 11, 3 (2011), 837–845.

[33] Byung Moo Lee, Mayuresh Patil, Preston Hunt, and Imran Khan. 2018. An Easy Network Onboarding Scheme for Internet of Things Networks. *IEEE Access* 7 (2018), 8763–8772.

[34] Ahmed Makki, Abubakr Siddig, Mohamed Saad, and Chris Bleakley. 2015. Survey of WiFi positioning using time-based techniques. *Computer Networks* 88 (2015).

[35] Ahmed Makki, Abubakr Siddig, Mohamed Saad, Joseph R Cavallaro, and Chris J Bleakley. 2015. Indoor localization using 802.11 time differences of arrival. *IEEE Transactions on Instrumentation and Measurement* 65, 3 (2015), 614–623.

[36] Andreas Marcaletti, Maurizio Rea, Domenico Giustiniano, Vincent Lenders, and Aymen Fakhreddine. 2014. Filtering noisy 802.11 time-of-flight ranging measurements. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*. ACM, 13–20.

[37] Israel Martin-Escalona and Enrica Zola. 2020. Passive Round-Trip-Time Positioning in Dense IEEE 802.11 Networks. *Electronics* 9, 8 (2020), 1193.

[38] Suhas Mathur, Wade Trappe, Narayan Mandayam, Chunxuan Ye, and Alex Reznik. 2008. Radio-Telepathy: Extracting a Secret Key from an Unauthenticated Wireless Channel. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MobiCom '08)*. Association for Computing Machinery.

[39] Hildur Olafsdóttir, Aanjhan Ranganathan, and Srdjan Capkun. 2017. On the security of carrier phase-based ranging. In *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 490–509.

[40] Google Play. 2021 (Accessed 24 March 2021). WifiNanScan App. https://play.google.com/store/apps/details?id=com.google.android.apps.location.rtt.wifinanscan.

[41] Marcin Poturalski, Manuel Flury, Panos Papadimitratos, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. 2010. The cicada attack: degradation and denial of service in IR ranging. In *2010 IEEE International Conference on Ultra-Wideband*.

[42] Marcin Poturalski, Manuel Flury, Panos Papadimitratos, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. 2011. Distance bounding with IEEE 802.15. 4a: Attacks and countermeasures. *IEEE Transactions on Wireless Communications* 10, 4 (2011), 1334–1344.

[43] Aanjhan Ranganathan and Srdjan Capkun. 2017. Are we really close? Verifying proximity in wireless systems. *IEEE Security & Privacy* (2017).

[44] Aanjhan Ranganathan, Boris Danev, and Srdjan Capkun. 2015. Proximity verification for contactless access control and authentication systems. In *Proceedings of the 31st Annual Computer Security Applications Conference*. 271–280.

[45] Aanjhan Ranganathan, Boris Danev, Aurélien Francillon, and Srdjan Capkun. 2012. Physical-layer attacks on chirp-based ranging systems. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*. 15–26.

[46] Aanjhan Ranganathan, Nils Ole Tippenhauer, Boris Škorić, Dave Singelée, and Srdjan Čapkun. 2012. Design and implementation of a terrorist fraud resilient distance bounding system. In *European Symposium on Research in Computer Security*. Springer, 415–432.

[47] Kasper Bonne Rasmussen and Srdjan Čapkun. 2008. Location privacy of distance bounding protocols. In *Proceedings of the 15th ACM conference on Computer and communications security*. 149–160.

[48] Maurizio Rea, Traian Emanuel Abrudan, Domenico Giustiniano, Holger Claussen, and Veli-Matti Kolmonen. 2019. Smartphone positioning with radio measurements from a single wifi access point. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*. 200–206.

[49] M. Roland, J. Langer, and J. Scharinger. 2013. Applying relay attacks to Google Wallet. In *2013 5th International Workshop on Near Field Communication (NFC)*.

[50] Ian Sharp and Kegen Yu. 2014. Indoor TOA error measurement, modeling, and analysis. *IEEE Transactions on Instrumentation and Measurement* 63, 9 (2014).

[51] Minghao Si, Yunjia Wang, Shenglei Xu, Meng Sun, and Hongji Cao. 2020. A Wi-Fi FTM-Based Indoor Positioning Method with LOS/NLOS Identification. *Applied Sciences* 10, 3 (2020), 956.

[52] Mridula Singh, Patrick Leu, AbdelRahman Abdou, and Srdjan Capkun. 2019. UWB-ED: distance enlargement attack detection in ultra-wideband. In *28th*

USENIX Security Symposium (USENIX Security 19). 73–88.
[53] Mridula Singh, Patrick Leu, and Srdjan Capkun. 2017. UWB with Pulse Reordering: Securing Ranging against Relay and Physical-Layer Attacks. *IACR Cryptology ePrint Archive* 2017 (2017), 1240.
[54] Subash Marri Sridhar and Carlos Horacio Aldana. 2019. Secure fine timing measurement protocol. US Patent 10,397,779.
[55] Nils Ole Tippenhauer and Srdjan Capkun. 2012. UWB-based secure ranging and localization. *Technical report/ETH Zürich, Department of Computer Science* 586 (2012).
[56] Nils Ole Tippenhauer, Kasper Bonne Rasmussen, Christina Pöpper, and Srdjan Čapkun. 2009. Attacks on public WLAN-based positioning systems. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*.
[57] Santosh Vamaraju and Carlos Horacio Aldana. 2019. Secure fine timing measurement protocol. US Patent 10,341,979.
[58] Mathy Vanhoef and Frank Piessens. 2014. Advanced Wi-Fi attacks using commodity hardware. In *Proceedings of the 30th Annual Computer Security Applications Conference*. 256–265.
[59] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-level localization with a single WiFi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. 165–178.
[60] Jon S Warner and Roger G Johnston. 2003. GPS spoofing countermeasures. *Homeland Security Journal* 25, 2 (2003), 19–27.
[61] Sigit Basuki Wibowo, Martin Klepal, and Dirk Pesch. 2009. Time of flight ranging using off-the-self ieee802. 11 wifi tags. In *Proceedings of the International Conference on Positioning and Context-Awareness (PoCA'09)*.
[62] Shihao Xu, Ruizhi Chen, Yue Yu, Guangyi Guo, and Lixiong Huang. 2019. Locating smartphones indoors using built-in sensors and Wi-Fi ranging with an enhanced particle filter. *IEEE Access* 7 (2019), 95140–95153.
[63] Chouchang Yang and Huai-Rong Shao. 2015. WiFi-based indoor positioning. *IEEE Communications Magazine* 53, 3 (2015), 150–157.
[64] Yue Yu, Ruizhi Chen, Liang Chen, Guangyi Guo, Feng Ye, and Zuoya Liu. 2019. A robust dead reckoning algorithm based on Wi-Fi FTM and multiple sensors. *Remote Sensing* 11, 5 (2019), 504.

A CONFIGURATION AND IMPLEMENTATION

Using Wi-Fi FTM on off-the-shelf hardware can be tedious. In order to increase the reproducibility of our work, we provide additional information on the implementation of our code, as well as the configuration of available hardware. Additionally, we publish a Github repository [1] bringing together all our Wi-Fi FTM experience. We provide instructions on how to use and customize relevant software (e.g., *iw*, drivers), how to configure Wi-Fi FTM APs, track Wi-Fi FTM support, vulnerabilities, and security patches over time.

Configuration. Wi-Fi FTM is supported on off-the-shelf Wi-Fi cards from Intel, and requires certain driver and firmware releases. In order to use the Intel AC-8260/8265, we configured an NVIDIA Jetson Nano (kernel version 4.9.140) and backported its *iwlwifi* driver to release Core 33, and used a patched *iw* version [26] to run Wi-Fi FTM commands (e.g., initiate a new session). With *iw* version 5.0 and up, there is build-in support for Wi-Fi FTM commands, however its usage requires a recent Linux kernel. In order to use the Intel AX-200, we recommend upgrading to a recent Linux kernel (version 5.0.0 and up), supporting *iw* version 5.0 and up, and backporting the driver to release Core 50, which supports firmware version 53, its first to support Wi-Fi FTM. In order to use the latest firmware versions 55, 57, 58, and 59, one must upgrade to Core 56.

Implementation. Given the need for a high response time, all implementations are written in C. The Network Interface Card (NIC) can be configured to receive and inject frames using the libpcap library, and proves to be sufficient for successfully executing the attacks. We use Android's WifiRttScan-example [5] for the Google Pixel 4 XL to inspect the measurement results reported by an initiator. For Intel SoCs, we use a patch to Linux's `iw` command [26]. Intel and Qualcomm SoCs report the final session result only, and therefore we can not inspect individual measurements. The results come with a status code, and we find at least one successful measurement will result in a successful session. As such, user-level applications will observe successful sessions, even if an adversary forces malicious single-shot measurements. When replaying frames, one can ignore IEEE 802.11 MAC header sequence numbers. Often these are checked in driver or kernel space, and since Wi-Fi FTM frames are not passed outside firmware space, we will avoid any sequence number check. Finally, we note all devices convert round-trip times to a distance using a simplified speed of light value of $3 \cdot 10^8$ m/s.

Advertisements. We find not all access points advertise support for Wi-Fi FTM (e.g., ASUS RT-ARCH13 AP), however do respond to measurement requests. Since Android only initiates a session if support is explicitly advertised, we have to bypass a restriction. Fortunately, we can trivially trick the smartphone by spoofing beacon frames sent from the AP with the Wi-Fi FTM-Responder bit set (i.e., extended capabilities information field bit number 70).

B DENIAL-OF-SERVICE ATTACKS

Wi-Fi FTM initiating and responding stations are subject to trivial Denial-of-Service (DoS) attacks. An attacker can target the initiator either by injecting a spoofed response frame with the status indication field in the configuration parameters set to three (indicating a failed request) or transmit a response frame with the dialog token set to zero. Similarly, we can target the responder by injecting a spoofed request frame with the trigger field set to zero, or request new session parameters. If new parameters are requested, the responder will terminate the session and start fresh with the new parameters. To be effective, the adversary can request only one measurement per burst (i.e., the responder transmits only the first response frame without timestamps) or request a configuration that is not supported by the initiator (e.g., a wider bandwidth). Injecting the spoofed termination frame before the responder transmits measurement results will result in a DoS. This approach is more efficient than jamming since it requires the injection of only a single frame per measurement session. Additionally, the attacker can configure parameters in a response frame with a time-out interval of up to 32-seconds in which the initiator is not allowed to make new requests. Furthermore, an attacker may terminate the session after the first measurement result, and as such force a single-shot measurement session. As such, the initiator has to rely on a single measurement only. This proves useful for our attacks, as it keeps session results from being tainted by legitimate measurement results.

Evaluation. Terminating the measurement session works successfully against all tested devices. This is an expected result, since we leverage features as defined by the specification. To initiate the attack, an adversary needs to capture only the request frame and obtain the randomized MAC address of the initiator. Then, we can immediately transmit the termination frame and have a sufficiently large time window (i.e., Min Delta FTM) to succeed. Practically, we achieve a success ratio of over 98% against all devices. However, terminating the ASUS RT-ARCH13 AP with a spoofed request frame crashes the access point and thus could not be properly evaluated. This bug was reported to the vendor and has now been patched.

[1] https://www.github.com/domienschepers/wifi-ftm