Ranging and Location attacks on 802.11 FTM

Jerome Henry, Yann Busnel, Romaric Ludinard, Nicolas Montavont

▶ **To cite this version:**

Jerome Henry, Yann Busnel, Romaric Ludinard, Nicolas Montavont. Ranging and Location attacks on 802.11 FTM. 2021. hal-03265600v1

HAL Id: hal-03265600

https://imt-atlantique.hal.science/hal-03265600v1

Preprint submitted on 28 May 2021 (v1), last revised 21 Jun 2021 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ranging and Location attacks on 802.11 FTM

Jerome Henry*, Yann Busnel†, Romaric Ludinard†, Nicolas Montavont†
*Cisco Systems, Research Triangle Park, NC 27560, USA, Email: jerhenry@cisco.com
†IMT Atlantique, IRISA, Cesson-Sévigné, France, Email: firstname.lastname@imt-atlantique.fr

***Abstract*—802.11 Fine Timing Measurement is an indoor ranging technique. Because it is unauthenticated and unprotected, our experiments indicate that an adversary can implement ranging and location attacks, by inserting one or more rogue responders and causing an unsuspecting client to incorporate forged values into its location computation. FTM clients tend to a small set of responders to range against (top 3 to 6 responders with strongest signal). Once ranges have been collected, the client can compute its location using various techniques, such as 3-sphere intersection, matrix error minimization techniques or Kalman filter. Regardless of the technique in use, we show in this paper that an attacker can cause a ranging client to deviate from its intended path. We also show that protection intended for attacks on comparable ranging techniques, like GPS, are ineffective in the case of FTM.**

***Index Terms*—FTM, 802.11az, location**

I. Introduction

While outdoor location is often possible with methods leveraging GPS, indoor location stays challenging as GPS signal is often not available inside. Among several proposed techniques for indoor settings, Fine Timing Measurement (FTM) defines a new ranging procedure based on Time of Flight (ToF). Defined in 802.11-2016 [1] and augmented in the 802.11az amendment (planned for publication in 2022), FTM enables an initiating station (ISTA, typically a Wi-Fi client) to perform ranging exchanges with a responding station (RSTA, typically a Wi-Fi system set at fixed location, *e.g.*, an access point) and also query the RSTA location. Performing such exchange with multiple RSTAs allows the ISTA to then compute its location.

This technique is conceptually similar to GPS, where a receiver computes its position from estimated ranges to satellites that also announce their position. Although the designers of FTM thought that the protocol would be immune to GPS-like attacks, this paper indicates that FTM is as vulnerable as unprotected GPS, allowing an attacker to drive an ISTA off course. Such attack could have dramatic consequences in some environments, for example factories where moving machines or robots use FTM to assess their position. Unfortunately, this paper also shows that FTM also presents fundamental properties that render mostly ineffective several mitigation techniques envisioned for GPS-attacks.

The rest of this paper is organized as follows: section II examines the field of attacks (and protection) on ranging techniques like GPS. Section III exposes how FTM computes range and location. Section IV details how GPS-like attacks can be conducted against FTM. Section V demonstrates attack vectors and practical attack measurements for FTM. Section VI concludes this paper.

II. Related Work

FTM is inscribed in a family of techniques similar in spirit to outdoor GPS, where location is computed from measured distances to objects which location is announced. The simplest attacks, against GPS, aim at completely jamming the signal [2], or preventing selected satellite signals from reaching the station, causing measurement confusion [3]. More advanced techniques spoof the identity of one or more valid sources, and provide signals which timing and timestamps cause the station to misevaluate its distance to these sources [4], thus driving the mobile station off course [5].

In some cases, these attacks can be detected, for example with Receiver Autonomous Integrity Monitoring (RAIM) [6]. With this system, the receiver compares the results obtained with various subsets of GPS signals, and alerts if any satellite contribution provides results inconsistent with the others. Additional techniques look at each individual signal fingerprint [7], or use multiple sources (e.g. GPS + internal sensors) to detect anomalies [8].

Many of the attack vectors valid for GPS theoretically also apply to FTM exchanges. However, in indoor public venues, it is common to detect many APs (10 or more). Therefore, the designers of FTM in 802.11-2016 considered that an isolated attacker would be unlikely to negatively affect the location of an ISTA that can range against multiple other RSTAs (and thus can easily spot the outlier). The original designers of 802.11az introduced a secured mode to protect the FTM exchanges and avoid malicious eavesdropping or injection, but this paper illustrates that the practicality of the implementations make these design decisions ineffective.

III. FTM Location framework

1) Ranging Techniques: 802.11 FTM focuses on the ranging exchange (not the position computation). The ISTA starts by negotiating with the RSTA some ranging session parameters (intended number of exchanges over a given time period). Then, in each exchange burst (lasting between 250 μs and 128 ms, during which the ISTA is not expected to move significantly), the RSTA first sends a frame at time t_1, which is received by the ISTA at time t_2. The ISTA replies at time t_3 with an acknowledgement frame, received by the RSTA at time t_4. In the subsequent frame, the RSTA communicates the values (t_1, t_4) to the ISTA. The ISTA can then establish its distance d to the RSTA by computing:

$$d = \frac{(t_4 - t_1) - (t_3 - t_2)}{2} c \tag{1}$$

where c is the speed of light. In a related exchange, the ISTA can request from the RSTA its Location Configuration Information (LCI), a set of geographical coordinates which logic is similar to that defined in RFC 6225 [9]. After a burst of exchanges, the ISTA then retains the smallest $[(t_4 - t_1) - (t_3 - t_2)]$ measured ToF, with the reasoning that direct (line of sight) path always produces the shortest ToF. The ISTA converts the ToF to a distance through equation (1), requests the RSTA LCI, then moves to ranging against the next RSTA. After having exchanged with different RSTAs, the ISTA can combine distances and LCIs to compute its own location.

2) FTM Location Computation Modes: The 802.11-2016 or 802.11az standards do not define a method for location computation. However, it should be clear that FTM does not expect any prior knowledge of the venue. The position is computed directly with the ToF and LCI values extracted from the frame exchanges with the local RSTAs.

Several techniques exist for such location computation. Geometric methods attempt to work from the measured distances and find intersection points. Among them, the three spheres method considers the distances to 3 RSTAs as the radii of matching spheres and finds their intersections. Three spheres intersect on two points (if all three intersect). There remains a two-way ambiguity. Naturally, if all three RSTAs are on the same floor and at the same height, the two possible intersection points are at different heights, and the ISTA may have internal additional sensors that can be used to resolve the ambiguity.

In many cases, the question of the height is also not posed by the user. In a "blue-dot" scenario, the user opens a particular floor plan in an app, and asks to display the device position on that floor (so the verticality is solved before the computation starts). The three-sphere method has the merit of being computationally simple, as the coordinates in $\mathbb{R}^3$ of a point $y = (u_y, v_y, w_y)$ which distance to three points, whose position is known as $i = (u_i, v_i, w_i), j = (u_j, v_j, w_j), k = (u_k, v_k, w_k)$ can be expressed as:

$$\begin{cases} u_y &= \frac{d_i^2 - d_j^2 + u_j^2}{2u_j} \\ v_y &= \frac{d_i^2 - d_k^2 + u_k^2 + v_k^2 - 2u_k u_y}{2v_k} \\ w_y &= \pm\sqrt{d_i^2 - u_y^2 - v_y^2} \end{cases} \quad (2)$$

In this system, one RSTA is positioned at the origin, thus $i = (0, 0, 0)$. Another RSTA is positioned along the u axis, thus $j = (u_j, 0, 0)$. The last RSTA is positioned on the same plane, thus $k = (u_k, v_k, 0)$. As the measured distances are noisy, the position determination aims at minimizing the error in the computation. This can be done with a least square error technique for example, which is computationally light in the case of 3 distances. When computation cost matters, 3 RSTAs is often seen as a simple "good enough" choice. When more than 3 RSTAs are used, a more elaborate minimization technique is needed, with the downside of a higher computation cost.

Another technique organizes the measured distances in a matrix, then attempt to minimize the error between the positions computed from the distance to each RSTA. There are three different ways to compute the solution [10]. The most common approach is to use calculus, by first finding the gradient ∇e of the error as:

$$\nabla e = 2\sum_{i=1}^{n}(\|y - i\| - \tilde{d}_i)\frac{y - i}{\|y - i\|} \quad (3)$$

where e is the sum of squares of errors in distance evaluations. This regression can be computed with standard gradient descent techniques, by moving by steps the estimated position of y based on the gradient value. Here again, computation complexity increases with the number of RSTAs and the measurements noise. Most implementations stop at 6 distances with such a model, as adding more reference points increases the noise contributors without dramatically improving the possible location accuracy [11].

A third common technique is the use of an extended Kalman filter (or an alpha-beta filter). The process is iterative, happening for each new measurement, and uses several steps to estimate a true position from an imprecise estimated position and noisy measurements. Shareef and Zhu [12] provide a good introduction to this technique and Welch and Bishop [13] give more details. Kalman filters are popular in technologies where the subject is expected to move, which is the case with FTM. For the context of this paper, one key element is that a component, the Kalman gain, is used to decide how much of the new estimated location should rely on a prediction based on the previous location and the user estimated trajectory, and how much should rely on the new (noisy) values. This aspect will take its importance as we inject invalid measurements. For this technique like the previous ones, because the measured distances are noisy [14], using more than 5 to 6 responders offers diminishing incentive, as ranging to each additional RSTA increases the energy and ranging (airtime) cost with a decreasing accuracy gain.

IV. GPS-like Attacks on FTM

A. Ranging Attacks

FTM exchanges do not require association or any link security. This approach is logical, as a station would undergo 802.11 association to only one AP at a time, but will need to range against multiple AP/RSTAs. A station can range against any AP announcing FTM support without further verification, and FTM is vulnerable to AP/RSTA impersonation. 802.11az, the 802.11 Amendment for Enhancements for Positioning that expands FTM, defines a Pre-Association Security Negotiation (PASN), allowing an ISTA to establish a secure session with an AP/RSTA without association. One mode supposes the existence of shared keying material between the ISTA and the RSTA, which would prevent spoofing attacks. However, this assumption is contrary to the the most common intended use case for FTM. In most settings, a user will need FTM to display their position on a map in an unfamiliar venue. It is not realistic to expect that the user, who is not aware of the venue yet, would yet have pre-installed keying material,

especially when accounting that, as of early 2021, there were around 400 million hotspots on the planet (and the number is increasing). If the user is familiar with the venue and has installed keying material, then the need for FTM is limited.

Therefore, only the unauthenticated PASN mode, which creates protected exchanges without authentication, is practical, and merely protects the unauthenticated exchange from view. An attacker can still provide rogue FTM services. With a fresh PASN session, the attacker could even impersonate a valid AP. Therefore, we conclude that PASN may be useful in a peer-to-peer ranging exchange for proximity (*e.g.*, FTM used to unlock a door) if keys are pre-set on both sides, but does not protect against GPS-like attacks.

Also, as the ISTA is only in control of t_2 and t_3, a vector of attack is obviously to forge custom t_1 and t_4 values, causing the ISTA to compute any arbitrary distance that the attacker would deem suitable to their mean. In the implementations we tested, some ISTAs consume the range as computed, and irrespective of its likelihood in the real world (*e.g.*, a distance of 15 km to the AP). Others use some filtering, but with limited effect. For example, in [15], negatives and ranges that are 50 percent larger or smaller than the range established in the previous burst are ignored. As long as $(t_4 - t_1) \geq (t_3 - t_2)$ and the $(t_4 - t_1)$ interval is relatively consistent from one sample to the next, the ISTA will use the returned numbers. In our experiments, the only consistently observed filter has been a "top-3" or up to "top-6" measurement ordering, where the ISTA ranges against the top-3 to top-6 loudest RSTAs (strongest RSSI). This structure seems to build on the assumption that RSTAs with strongest signals are likely to be closer, and have more chances of being in LoS, thus providing less noisy results.

B. Position Attacks

Another common point is that all methods use both the distance and the location (LCI) returned by the RSTA. Thus an attack equivalent to invalid (t_4, t_1) values, and much easier to implement, is to send an invalid LCI value for one or more RSTAs. Here again, we have not found an implementation that discards unrealistic LCIs (*e.g.*, one RSTA reporting to be in Sydney Opera House, while the others report being close to the Eiffel Tower). All tested implementations simply do their mathematical and computational best to minimize the error from these various distance and location elements.

V. FTM GPS-like attack experiments

We tested these various possible attacks in a FTM deployment, with parameters as follows. In an open space free from objects (to avoid localities related to obstacles or reflections) 5 APs are deployed along a 75-meter walk path. APs are positioned 24.8 meters from each other, at 10.74 meters from the walking path, at 2.9 meters height, in an alternating fashion represented in Figure 1. This structure allocates to each AP a 750-square-meter cell, a typical Wi-Fi density in public venues. At each 50 cm interval, the ISTA ranges against all detectable APs, and the operator collects 100 ranging samples from each point to each RSTA. In a second phase, the ISTA location is then computed using the three sphere method (method 1), the distance matrix least squares resolution method (with 4 to 6 RSTAs, method 2) and the position estimation based on a Kalman filter (method 3), in Matlab. The ranging tests are run with a Pixel 3, a Compulab ISTA and a laptop running Windows 10 (and embarking an Intel AX200 WiFi card) for the ISTA side, and a Google Wi-Fi AP, a Compulab Responder and a Cisco Catalyst 9120 access point for the RSTA side. As results are comparable for all combinations, the Pixel vs Cisco 9120 figures are presented here.

A. Attack vectors

For an attacker, the different attacks listed in the previous section present consequential feasibility differences:

- **Invalid t_1 and t_4**: in most systems, the timestamp is computed at low chipset level, in the DSP microcode. Control from the operating system is limited. Without such control, one option is to capture over-the-air an FTM exchange (*e.g.*, using Wireshark), edit the file to change the victim target MAC address and insert the t_1 and t_4 values of choice (with tools like WireEdit), then use tools like TCPreplay to replay the AP response to the ISTA. This attack requires some level of preparation. Its outcome is to mislead the ISTA on its real distance to the location (LCI) reported by the attacker's AP.
- **Invalid LCI**: the effect is also to mislead the ISTA on its distance to a reported location, this time by providing valid (t_1, t_4) but invalid RSTA location. The LCI is provided by the operating system (*e.g.*, hostapd.conf file in Linux), and is therefore easy to modify. In some implementations, changing the LCI value implies restarting the Wi-Fi service. Injecting an invalid LCI is much easier than modifying the t_1 and t_4 values, unless the attacker has access to the DSP microcode.
- **Session hijacking**: the ISTA and RSTA exchange dialog token values. An attacker inserting into a valid dialog between an ISTA and a RSTA, for example to substitute the attacker response to the valid RSTA response, would need to provide the correct token value in the response. Failure to do so would cause the ISTA to ignore the frame. However, the systems we tested do not implement a complex token system. Some use a linear suite ($1, 2, 3,$ *etc*.) Others always use 0 as the token value. Additionally, 2 of the 3 RSTAs tested allow the user to define the MAC address. A simple injection attack is therefore to program the attacker RSTA with the victim RSTA MAC address, and let the local system perform FTM (responding to the ISTA tokens). The effect is ranging confusion, as the ISTA receives different distances (and LCIs) from what the ISTA assumes to be single device.

In the experiments below, we found that t_1 and t_4 manipulation provided similar outcomes as LCI manipulation, but at the cost of a much higher implementation complexity. Thus, the LCI attack outcomes are presented. The session hijacking also provided interesting observations.

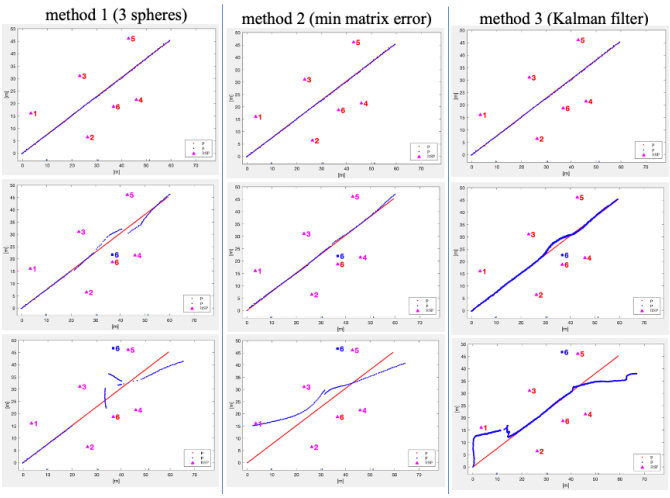

Fig. 1. Attacker AP in place, sending real LCI (top), 5-m (center row), and 35-m biased LCI (bottom), with location computed using three-sphere method (left), matrix resolution (center column) and Kalman filter estimation (right).

B. Inserting an Invalid RSTA

Throughout these experiments, the operator walks along the test path. Real position is marked p and the computed position $\tilde{p}$. 5 APs are deployed and provide their normal FTM ranging and LCI values. An attacker AP (AP6) is introduced. The experiments show that the attacker's AP physical position is not critical for this phase. When AP6 provides RSTA service and valid values, the ISTA computed position matches the actual position, regardless of the computation method, as expected (*cf.* Figure 1.top). Next, the LCI sent by the attacker AP is modified to return either an incorrect latitude value or an incorrect longitude value. Its true position is marked with a triangle, its reported position with a square.

With a small bias (*e.g.*, 5 meters), the ISTA computed position drifts by at most the bias value, and reaches its maximum near the attacker AP's position. Quite naturally, the drift increases with the bias. However, a large bias may cause a filter mechanism to detect the attacker AP as an outlier (providing values not compatible with the position determined from the other RSTAs). Additionally, large bias (*e.g.*, 35 meters) may render the position impossible to determine. With the three spheres technique, the reported positions make that the spheres do not always intersect, resulting in no location results for several points. With an extended Kalman filter, the path becomes incoherent (even if the ISTA continues to compute a position, a normal user would soon identify that one cannot walk in a straight line and yet be on such convoluted path). The reason for such incoherence lies in the way the Kalman filter technique is applied. As the goal is to resolve standard Euclidean distance equations (in the form $\tilde{d} = \sqrt{(u_i - u_y)^2 + (v_i - v_y)^2 + (w_i - w_y)^2} + b_i$) the extended Kalman filter technique seeks derivatives of the state matrix and the distance measurement matrix before applying the Kalman process, and determining the relative weight given to the observation and the prediction when computing the next likely position. The filter becomes better at predicting the next state when noise is comparable across RSTAs. But when the ISTA switches to a faulty RSTA, suddenly providing incoherent values (as it is the case with this attack model), then the measurements suddenly largely exceed the range of expected errors. The Kalman gain soon increases the weight of measurements over prediction, but the reactive process, coupled to the fact that derivatives are sought from ranging against a different contributor, causes the resulting state matrix to display sudden changes of direction as the new contributor data becomes dominant. It is worth noting that this effect is known for the Extended Kalman Filter (it is not an optimal estimator in cases when one AP provides values beyond the expected noise range).

C. Spoofing valid RSTAs

In this phase, the attacker impersonates a valid RSTA MAC address. Such action causes both the valid RTSA and the attacker RSTA to respond to the ISTA queries for ranging. A confusion attack would have the attacker RSTA respond with different parameters than the valid RSTA. Depending on the implementation, such response might cause the ISTA to ignore the valid RSTA parameters, or to fail during the ranging exchanges (as the exchanges do not match the parameters that the ISTA recorded).

A more interesting attack is to let the valid RSTA respond, then have the attacker RSTA insert FTM ranging frames within the valid exchanges. The ISTA then undergoes more exchanges than it expects (e.g. receiving 16 frames in a burst where it expects 8). On all observed ISTAs, the client considers the exchanges that can take place within the defined burst duration (and ignores RSTA FTM messages beyond the expected end of the burst), even if the burst contains more than the expected count of exchanges. On all observed RSTAs, the FTM exchanges also stop at the end of the burst duration (but each RSTA does not attempt more than the expected count of exchange within each burst). Thus, it seems that current implementations allow the ISTA to perform more exchanges than agreed upon, within the limit set by the burst duration.

The net effect is that the ISTA receives half its ranges from the valid RSTA, and half from the attacker RSTA. The ISTA retains the shortest distance in the burst, as explained in Section III. All observed ISTAs also do not consider the LCI as a fixed object. In other words, in the ISTAs we observed, each return to the channel causes the ISTA to query for the LCI again as part of the FTM exchange. In this circumstance, the ISTA receives 2 sets of LCIs. All the observed ISTAs record both received LCIs in their logs, but continue to display the second one for the burst analysis.

Therefore, in order to be preferred to the valid RSTA, the attacker first needs to make sure that the distance offered to the attacker RSTA is less than to the valid RSTA (as the ISTA retains the shortest distance in the burst). The attacker can hard code that distance in the t_1 and t_4 values, or make sure to position the attacker AP closer to the victim walking path than the real AP. In a public venue where walking paths and

valid APs positions are commonly known, choosing the right AP to target (*e.g.*, an AP away from the public walking path) makes that phase trivial. Then, the attacker can modify the LCI at each exchange to lure the victim away from the intended path. Replaying the LCI value several times can ensure that the attacker's value is received after the valid AP LCI in each burst, and thus preferred.

With these precautions in place, the effect observed is that the attacker's AP is substituted to the valid RSTA for most bursts. Although the ISTA ranges to both RSTAs, the values from the attacker are statistically retained more often, thus effectively resulting in valid AP suppression and replacement.

D. Leading the victim to a target location

It is now clear that FTM is vulnerable to GPS-like attacks that can create ranging and location confusion. We now show that careful parameter injection can be used to lead the ISTA to a location of the attacker choosing. Such possibility may have dire consequences in settings where FTM is used for business-critical navigation (computer-on-wheels in healthcare, guided robots in factories, etc.)

Naturally, each location computation technique incorporates a different set of noisy distances and parameters from which location is computed. Therefore, an efficient attack should account for the type of location equation in use by the victim device. In most cases, the victim will not choose the formula, but use the method incorporated in the operating system or the navigation app of choice. It is expected that a small set of large actors will provide the bulk of the multilateration algorithms. Thus, it is likely that knowing the victim device will allow the attacker to determine the location method in use.

To illustrate such approach, the attacker apparatus is positioned near AP3 (and marked as A in Figure 2). The apparatus is comprised of 3 Compulab RSTAs, set to AP1, AP3 and AP5 channels and MAC addresses respectively. The apparatus is closer to the walking path than AP1, AP3 and AP5 between the marks on the path in Figure 2. Because the attacker now impersonates 3 systems, and because the number of expected contributing RSTAs is limited as explained in Section III, the outlier filtering system fails (no single isolated outlier AP).

The goal of the attacker is then to provide ranging values pushing the victim toward a point between AP2 and AP4 (marked with a cross in the figure). The task becomes trivial with equation (2) and the three-sphere technique. To build LCIs with that method, any impersonated AP is chosen to assume the position at the origin and provides any arbitrary reference LCI value. Then, from the known distance from the apparatus to the point where the victim should be led (and its matching coordinates y), the system of equation can be solved to find the LCIs to be announced by the other impersonated APs. From equation (2):

$$u_j^2 - 2u_j u_y + d_i^2 - d_j^2 = 0 \tag{4}$$

The only unknown is u_j, which can be found as:

$$u_j = \frac{2u_y \pm \sqrt{4u_y^2 - 4(d_i^2 - d_j^2)}}{2} \tag{5}$$

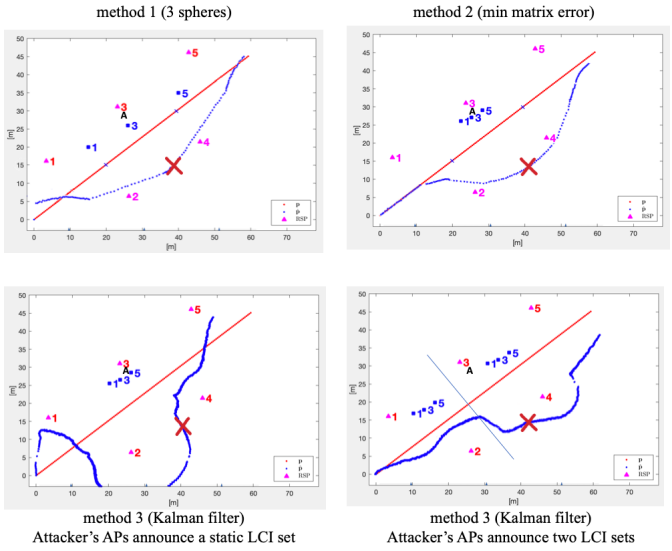

Fig. 2. Attacker APs chosen to lead a victim toward a target point, with the three-sphere method (top left), least squares (top right) and Kalman filter with one (bottom right) or two LCI set techniques).

where $(0, 0, 0)$ is the LCI of choice for the first impersonated AP. Out of the 2 possible solutions, choosing the smaller u_j limits the risk of outlier detection. Here, $v_j = v_i = 0$. Then:

$$v_k^2 - 2v_k v_y - 2u_k u_y + d_i^2 - d_k^2 + u_k^2 = 0 \tag{6}$$

Both u_k and v_k are unknown in this system, but by expressing v_k as a function of u_k:

$$v_k = \frac{2v_y \pm 2\sqrt{v_y^2 - d_i^2 + d_k^2 - u_k^2 + 2u_k u_y}}{2} \tag{7}$$

The attacker can choose an arbitrary value of u_k so that:

$$u_k^2 - 2u_k u_y - v_y^2 + d_i^2 - d_k^2 \leq 0 \tag{8}$$

In other words, u_k is an arbitrary number in the range $[p, q]$ that satisfies $pq = -v_y^2 + d_i^2 - d_k^2$ and $p + q = -2u_y$. As all other values are known to the attacker, the determination is a simple factoring exercise.

Once the attacker's APs are positioned, the victim system will measure noisy distances, but the optimal solution will lead the victim toward the intended point. As the computation includes the contribution of valid APs at some stage of the path, the location result is increasingly biased toward the attacker's APs data as the victim advances on the path, as can be seen on the top left of Figure 2.

The same logic is applicable to the least squares technique. Let's suppose the more complex case, 6 contributors, including 3 valid APs (i, j, k) and the three attacker APs (a_1, a_2, a_3). In equation (3), for each AP under the attacker control, y is the intended target destination (that we write y_i), while for each valid AP, y is the true victim position (that we write y_t), which can be known by deciding the real location of the victim when the location app computes the intended target position. The distances $\tilde{d}_i$ to all contributing APs are known from the

(t_1, t_4) values measured at that real position. Thus, we can rewrite the minimzation goal as:

$$\begin{aligned}\min[&(||y_t - i|| - \tilde{d}_i)^2 + (||y_t - j|| - \tilde{d}_j)^2 \\ &+ (||y_t - k|| - \tilde{d}_k)^2 + (||y_i - a_1|| - \tilde{d}_{a_1})^2 \\ &+ (||y_i - a_2|| - \tilde{d}_{a_2})^2 + (||y_i - a_3|| - \tilde{d}_{a_3})^2]\end{aligned} \tag{9}$$

The only unknowns are therefore the announced positions i of the attacker's APs that minimize the error at the target position. A simple solution is to insert two arbitrary positions, and let the system solve for the third. This solution is functional, but might output a LCI for the third AP far from the others. In order to minimize the risk of outlier detection, an efficient approach is to wrap this algorithm into another gradient descent structure. With this method, the first 2 APs are set at initial arbitrary positions, the third AP location is found, then a loop runs, where step-wise changes to the initial 2 AP positions are made so as to minimize the differences between all 3 resulting LCIs. Once such system is found, the victim can be led to the intended location, as displayed in the top right part of Figure 2.

The Kalman filter case is slightly different. The solution proposed for the least squares approach above is also functional, misleading the ISTA to compute its position as the target location. However, as can be seen in the lower left part of Figure 2, the effect is also to cause an incoherent trajectory (reporting a sharp turn while the user is walking along a straight line). To avoid this risk, an additional step is to identify intermediate positions where the attacker would want to smoothen the curve, determine the optimal victim position at that point, and generate a set of attacker APs LCIs accordingly. The attacker announces a first set of LCIs (or timestamps), then a next set as the victim passes key points. The effect of such modification can be seen in the lower right part of Figure 2, with an initial set of forged LCIs, then a second set announced as the victim passes the traversal (blue) line. The process needed for such attack in the Kalman Filter case is similar in concept to the Least Square cases, where a loop computes the LCI set that minimizes the differences between the spoofed APs announced positions, but is more laborious, as the range at which the announced LCIs are efficient in fooling the victim is limited. It is likely that the effort involved will match the value of leading the victim to the target point without raising suspicion.

VI. Conclusion and Future Work

In this paper, we have shown that 802.11 FTM is vulnerable to ranging and location attacks. As the client does not know in advance the APs, and as the exchanges are neither authenticated nor protected, an attacker can easily insert an additional AP that provides invalid ranging or position (LCI) information. The client has limited ability to distinguish the attacker's from valid APs, and tends to integrate the data provided by the attacker, if it is not excessively implausible, into its computation. The result is that an attacker can inject values into FTM exchanges that cause the victim to conclude on an incorrect location computation. This method can be used to make the victim's device deviate from its intended course, and potentially lead the device to a destination of the attacker's choice.

Protecting against such attack would require a mechanism for an ISTA to identify APs that are unlikely to be part of the venue deployment. In the absence of association, future work will examine how such membership proof can be provided, for example through backhaul AP-to-AP signaling.

References

[1] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 2016.

[2] K. Tugsad Seferoglu and A. Serdar Turk, "Review of spoofing and jamming attack on the global navigation systems band and countermeasure," in *2019 9th International Conference on Recent Advances in Space Technologies (RAST)*, 2019, pp. 513–520.

[3] R. Ferreira, J. Gaspar, P. Sebastiao, and N. Souto, "Effective GPS Jamming Techniques for UAVs Using Low-Cost SDR Platforms," *Wireless Personal Communications*.

[4] J. Su, J. He, P. Cheng, and J. Chen, "A Stealthy GPS Spoofing Strategy for Manipulating the Trajectory of an Unmanned Aerial Vehicle," *IFAC-PapersOnLine* .

[5] E. Horton and P. Ranganathan, "Development of a GPS spoofing apparatus to attack a DJI Matrice 100 Quadcopter," *The Journal of Global Positioning Systems*.

[6] S. Hewiston and J. Wang, "Gnss receiver autonomous integrity monitoring (raim) performance analysis," *GPS Solutions*, vol. 10, pp. 155–170, 2006.

[7] M. Foruhandeh, A. Z. Mohammed, G. Kildow, and R. Gerdes, "Spotr: GPS Spoofing Detection via Device Fingerprinting," in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec'20)*, Linz (Virtual Event), Austria, 2020.

[8] J.-H. Lee, K.-C. Kwon, D.-S. An, and D.-S. Shim, "GPS spoofing detection using accelerometers and performance analysis with probability of detection," *International Journal of Control, Automation and Systems*.

[9] Y. Polk, M. Linser, M.Thomson, and B. Aboba, "Dynamic Host Configuration Protocol Options for Coordinate-Based Location Configuration Information," Internet Requests for Comments, RFC Editor, RFC 6225, July 2011.

[10] E. Simoncelli, "Least squares optimization," September 2019.

[11] B. K. Horn, "Projective geometry considered harmful."

[12] A. Shareef and Y. Zhu, "Localization using extended kalman filters in wireless sensor networks," April 2009.

[13] G. Welch and G. Bishop, "An introduction to the kalman filter," July 2006.

[14] B. K. Horn, "Doubling the accuracy of indoor positioning: Frequency diversity." *Sensors (Basel)*, Mar. 2020.

[15] L. Banin, O. Bar-Shalom, N. Dvorecki, and Y. Amizur, "Reference-pe-and-measurements-db-for-wifi-time-based-scalable-location."