Secure Localization of Passive Drones in UWB-based Networks
(tentative)

Mahyar Shariat
12028741
mahyar.shariat@tuwien.ac.at

20.01.2023

Abstract

Signal spoofing in drones is a well-known problem. As a countermeasure to spoofing, the European Space Agency, has adopted a broadcast authentication protocol named Timed Efficient Loss-tolerant Authentication (TESLA) in European GNSS, the Galileo. In this work, we intend to see if a similar method has ever been used to provide secure localization in other systems and whether it can be done in practice for indoor positioning systems that use ultra-wideband technology. The initial findings from our experiment show that authentication methods based on TESLA can be used to provide protection from signal-spoofing to an UWB-based indoor positioning system.

1 Introduction

1.1 Motivation

Global Positioning System (GPS) has been shown to be vulnerable to spoofing attacks both in the lab and possibly *in the wild* O'Hanlon et al. [2013]. A spoofing attack is the transmission of fake GPS signals that receivers accept as authentic ones spo. A popular theory suggests that Iran successfully captured a classified U.S. spy Unmanned Aerial Vehicle (UAV) over its territory, mostly undamaged, using a GPS-spoofing method. The cost of capturing a drone using a spoofing method can be as low as $1000 Kerns et al. [2014]. Besides UAVs, civil GPS spoofing also poses a danger to manned aircraft, maritime craft, communications systems, banking and finance institutions, and the national power grid Humphreys [2012]. Recently, spoof-resilient measures have been introduced as part of the European Global Navigation Satellite System (GNSS), Galileo, that enables authentication of the source using a protocol called Timed Efficient Stream Loss-tolerant Authentication (TESLA) osn, Khan et al. [2021], Perrig et al. [2000]. As the name suggests, the TESLA protocol is an authentication protocol (serving the I in the CIA[1] security triangle) designed exclusively for broadcast settings. When used in navigation, it can enable the authenticity of navigation messages (e.g., messages are really sent by satellites and not forged) such that

[1] Confidentially Integrity Availability

breaking it is computationally not easy. It does so by using cryptographic hash functions and delaying key disclosure. We intend to see if the same method can be applied to indoor positioning systems.

1.2 Background

What is Crazyflie? Crazyfie is an open-source mini-drone that runs FreeRTOS and is relatively easy to program. It can be extended with a wide range of extension boards (decks) to add more features. One such deck is Loco Deck, which features a DWM1000 UWB radio module.

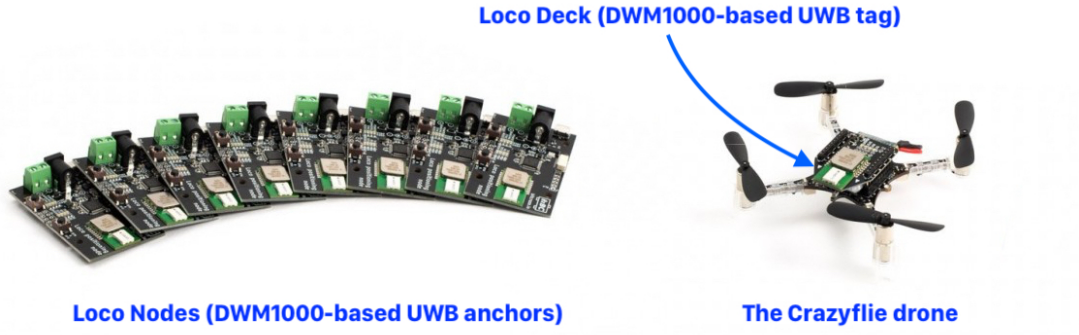


Figure 1: The devices used for UWB positioning in this research

What is Loco Positioning System (LPS)? The same company that designed Crazyflie also provides a set of anchors featuring the same UWB radio module under the commercial name Loco Positioning System. When used together with Crazyflie, the LPS enables passive localization for Crazyflie using the principle of Time Difference of Arrival (TDoA)loc. The system has already been used in many works related to autonomous flight, such as swarm [2]. It also offers two TDoA modes of operation as desired.

What is TDoA, and how does it work? The choice of LPS is rooted in our intention to model the broadcast nature of GNSS's sender-to-receiver ranging signals. LPS implements a *centralized* TDoA mode (TDoA2) and a *decentralized* TDoA mode (TDoA3). In TDoA2, all the anchors are synchronized w.r.t. one master anchor, and the TDoA measurements are expressed in the same clock. The main disadvantage of this mode is introducing a single point of failure (SPOF) in the system. In TDoA3, anchors synchronize the timescales between anchor pairs, sacrificing accurate measurements for scalability Zhao et al. [2022]. Here a basic description of the implementation of the TDoA engine in LPS is given: Suppose a situation as depicted in Figure 2,

[2] bitcraze.io/portals/research

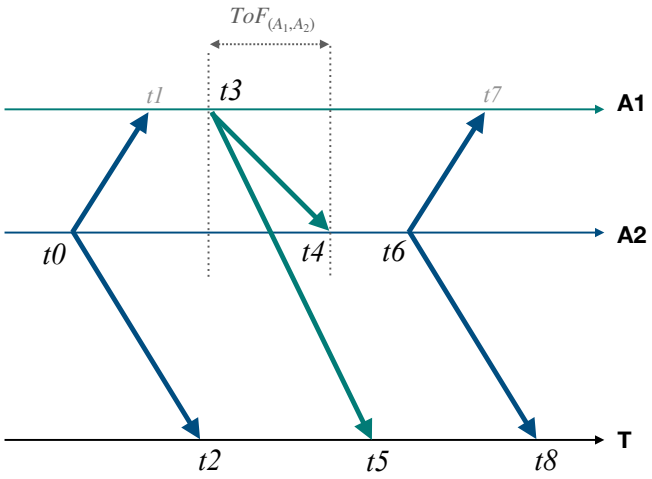


Figure 2: Example of TDoA. Time advances from left to right. Each t denotes a timestamp.

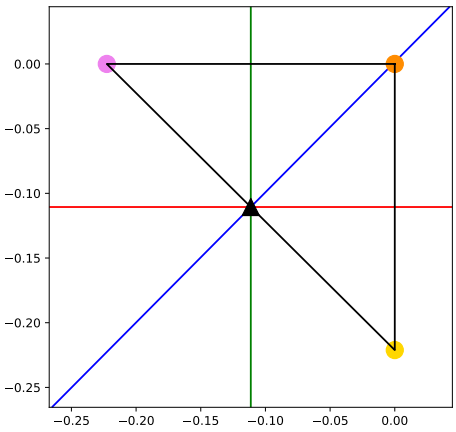
TDoA2 protocol (centralized TDoA) The TDoA2 protocol in Crazyflie is implemented as follows: There are eight anchors addresses from 0 to 7 in the system. The anchor with address 0 takes the role of the master, and other anchors are synchronized according to it their radio transmission time to avoid collisions. The transmission happens every 2ms and follows a round-robin pattern. In this way, a passive tag could listen to the transmissions and derive the distances between anchor pairs (7,0), (0,1), (1,2), (2,3), (3,4), (4,5), (5,6), and (6,7) and derives the tag's position using the TDoA measurements. As an example, consider the equation for finding the TDoA between Anchor 1 and Anchor 2:

$$TDoA_{(A_i,A_j)} = c[(t8-t5)-\alpha(t6-t4+ToF_{(A_i,A_j)})] = \|C_T - C_{A_i}\| - \|C_T - C_{A_j}\| \quad (1)$$

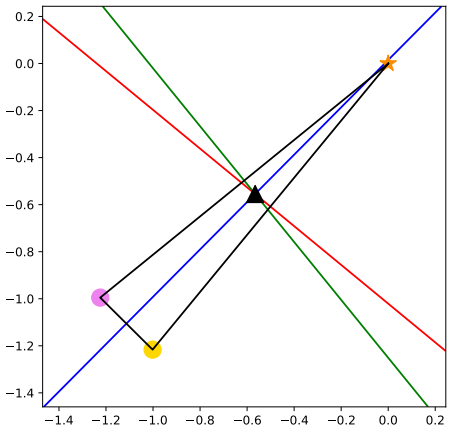
Where A_i denotes ith anchor, T is the tag, $C_X \in \mathbb{R}^3$ is the 3D coordinate for $X \in \{A_i, T\}$, the constant c is the speed of light, and $\alpha = (t8-t2)/(t6-t0)$ is the clock correction. Furthermore, $\|.\|$ denotes the ℓ_2-norm, which is defined as the Euclidean distance:

$$\|T - A_i\| = \sqrt{(x_T - x_{A_i})^2 + (y_T - y_{A_i})^2 + (z_T - z_{A_i})^2} \quad (2)$$

Signal spoofing simulation As mentioned before, signal spoofing is a fundamental problem in the navigation of drones [Chamola et al., 2021]. To give an example, figure 3 displays the simulation result of signal spoofing in two-dimensional space with three anchors. If one of the anchors is removed and replaced by a spoofer that sends fake signals about its location by an order of magnitude in the positive direction (i.e., from $(0.000001, 0.000001)$ to $(0.00001, 0.00001)$) the distance from the tag to anchors is changed from $0.1569m$ to $0.7923m$ assuming constant TDoA (e.g., $TDoA_{31} = TDoA_{12} = TDoA_{23} = 0$).



(a) Intersection of the hyperbolas under constant TDoAs without a spoofer



(b) Intersection of the hyperbolas under constant TDoAs if the orange anchor from (a) is replaced by a spoofer (the star) that lies about its location by an order of magnitude.

Figure 3: 2D geometry of passive localization using three anchors in the presence and absence of a spoofer. The hyperbolas appear like straight lines as TDoA approaches zero.

Authentication Authentication protocols can be categorized into *unicast*, *multicast*, and *broadcast*. A broadcast authentication can be used both in the network layer or in the application layer Perrig et al. [2001]. In broadcast authentication protocols, a source wants to send broadcast messages to any destination in an authenticated manner, i.e., untampered and verifiable. Broadcast authentications can be used in many applications, including navigation systems, ad-hoc networks, wireless sensor networks, emergency broadcast systems, code dissemination, and in other systems that require secure beaconing. The building block for broadcast authentication can be digital signatures or Message Authentication Code (MAC) Grover and Lim [2015]. The main ingredient for any broadcast authentication is asymmetry, such that the receivers can only verify the authentication information but not generate it. A MAC is commonly referred to as *keyed hash*. A hash function is a one-way function that maps a value of arbitrary length to a fixed-length value n, i.e.:

$$F : \{0,1\}^* \rightarrow \{0,1\}^n. \tag{3}$$

The formal definition of a Collision Resistant Hash Function (CRHF) is as follows Preneel [2011]:

1. The input X can be of arbitrary length and the result $F(X)$ has a fixed length of n bits.
2. Given $Y \in \{0,1\}^n$, it is computationally infeasible to find an X such that $F(X) = Y$ (*preimage resistance*).
3. Given X, it is computationally infeasible to find a $X' \neq X$ such that $F(X') = F(X)$ (*second preimage resistant*, also known as *target collision resistance* or *weak collision resistant*).
4. It is computationally infeasible to find any pair of inputs X, X' such that $F(X) = F(X')$ (*strong collision resistance*, or simply *collision resistance*).

2 Literature Review

Search strategy

Scopus[3] is the "largest" citation database of scientific papers. Scopus features advanced search that enables SQL[4]-like queries. To find related works, we first used the following search string `TITLE-ABS-KEY ( tdoa AND indoor AND tesla )`. However, no documents were found with those keywords. In order to broaden our search scope, we tried to be as inclusive as possible without losing accuracy and used the following search string after a few adjustments:

[3] `https://www.scopus.com`
[4] Structured Query Language

```
(gnss OR gps OR galileo OR wsn OR iot OR uwb OR ultra-wideband OR
indoor AND positioning OR localization OR tracking OR
navigation AND security OR secure OR "network security" AND
tesla OR authentication AND tdoa OR "passive localization") OR
( geoencryption AND authentication ) AND
TITLE-ABS-KEY(authentication) AND TITLE-ABS-KEY(positioning) OR
TITLE-ABS-KEY(localization) OR TITLE-ABS-KEY(tracking) OR
TITLE-ABS-KEY(navigation)
```

Where AND denotes logical conjunction, OR is logical disjunction, and search only in Title-Abstract-Keywords is denoted by `TITLE-ABS-KEY`. By default, if no constraint is defined on a word, the search engine looks for that word in all fields, more specifically, `ALL(word)` = `word`.
The search resulted in eighteen records. After reading the abstracts, four of the papers were deemed to be not related to our problem. Three out of the fourteen remaining papers were filtered out because they had been published in journals with a ranking of Q3 or lower in the year of publication according to ScimagoJR[5]. After a further examination of the remaining papers, we only found five papers to be related to our initial problem. In addition to the results from Scopus, we also used generic keywords such as `uwb` and `authentication` in Google Scholar[6] and ION[7] that resulted in two additional papers that are related to our problem definition.

We now briefly review the related works in chronological order. In Zhang et al. [2006], an authentication scheme using MIC [8] in UWB-based WSN[9]s is proposed to enable secure two-way ToA[10] localization. However, this method requires the tag to be an active node. In Qiu [2007], Qiu et al. [2007], a *geoencryption* algorithm is proposed and analyzed for LORAN[11] system. In the proposal, the same protocol is used as in Galileo OS-NMA. The results from this work suggest that signal authentication can be achieved in passive tags and protect against location spoofing. Nevertheless, the work does not involve indoor positioning or UWB radio. In Becker et al. [2009], the previous work is extended by proposing a more efficient version of TESLA for navigation systems and its application in eLORAN[12]. Guo and Zhu [2011] proposed an authentication technique for WSN using ECC[13] to enable secure ToA-based localization. The authors claim that their technique handles localization *in the presence of liars*, given a liar-honest threshold. However, this work also assumes that there is

[5] `https://www.scimagojr.com/journalrank.php`
[6] `https://www.scholar.google.com`
[7] `https://www.ion.org/publications/browse.cfm`
[8] Message Integrity Code
[9] Wireless Sensor Network
[10] Time of Arrival
[11] LOng RAnge Navigation
[12] enhanced LOng RAnge Navigation
[13] Elliptic Curve Cryptography

a bidirectional communication channel between senders and receivers. In Alawami and Kim [2020], a classification algorithm is proposed to distinguish legitimate from illegitimate nodes by characteristics of radio signals such as RSSI [14] within a small room. However, this method assumes cryptographic authentication to be already established and only provides an extra layer of classification based on user location. Lastly, in Prashar et al. [2021], a digital-signature-based method is proposed to enable secure localization in WSN. However, PKC[15]-based approaches may not be suitable for high-frequency streams [Perrig et al., 2005].
A compact comparison of the studied works can be seen in Table 1.

Table 1: Related works comparison.
Denotations: Localization Method (LOCAL), Radio Technology (RADIO), Cryptographic Primitives Used (CRYPT), System (SYS).

	LOCAL	RADIO	CRYPT	SYS
Zhang et al. [2006]	TW-ToA	UWB	MIC (MAC)	WSN
Qiu [2007], Qiu et al. [2007]	TDoA	LORAN	TESLA	Terrestrial
Becker et al. [2009]	TDoA	LORAN	TESLA	Terrestrial
Guo and Zhu [2011]	ToA	-	PKC (ECC)	WSN
Alawami and Kim [2020]	-	Wi-Fi, Bluetooth	-	WSN
Prashar et al. [2021]	Hyperbolic	-	PKC	WSN

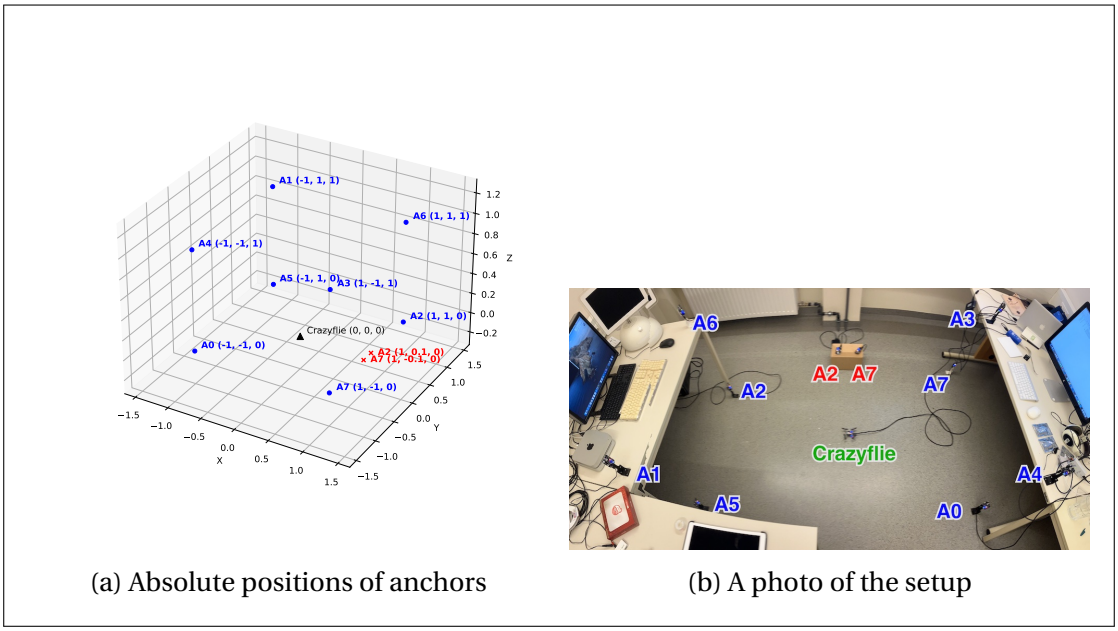
3 Data Collection

3.1 System Specification

To show the problem in indoor positioning systems, we decided to implement a basic spoofing attack in a real system that uses UWB radio and passive localization. So we used Crazyflie and Loco Positioning System (LPS) from Bitcraze.

Mode of operation Since the implementation of TDoA2 is less complicated than TDoA3, we picked TDoA2 as the starting point for our experiment. To see the effect of interference in LPS in TDoA2, we used a room set up with eight anchors around the corners to function normally and placed six anchors in the middle to broadcast false information in LPP packets (lpp) about their location. As expected, the interference caused the drone that used the system to crash. The hypothesis is that applying TESLA can address this problem. Suppose we establish that TESLA can indeed be applied to address the problem. In that case, we also need to generalize the problem for all indoor positioning systems that use TDoA, including TDoA3 in LPS.

[14] Received Radio Strength Indicator
[15] Public-Key Cryptography



(a) Absolute positions of anchors

(b) A photo of the setup

Figure 4: Anchors positions

System setup Ten anchors were used in a small room. Eight anchors transmit true signals, and two send spoofed signals. The Crazyflie was placed at the origin $(0,0,0)$. While this setup provided the line of sight for the drone, we do not make any claims about the accuracy of the system since the absolute coordinates were taken from the abstract set $\{0.0, 1.0, -1.0\}$. Nevertheless, it provides an easy approximation for the geometry of our constellation. For claims about accuracy, accurate measurements are needed, which could be obtained by a total station.

Spoofers The communication in LPS is bare-bone, and no security measure is provided. It means the drone cannot tell which signals to accept and which to discard. To exploit this weakness, we added two extra anchors to the system but didn't program the anchors with the correct coordinates. We placed an anchor with id=2 at (1.0,-0.1,0) and another with id=7 at (1.0,-0.1,0). However, we programmed both anchors to broadcast (10.0,10.0,0.0) instead of their *real* coordinates.

Sequence of human commands

We attacked the system under two scenarios:

1. The attacker joins the network at the same time as the legit nodes
   (a) Turn off everything
   (b) Turn on LPS except anchor 0
   (c) Turn on spoofers

(d) Turn on anchor 0

(e) Remotely reset Crazyflie

(f) Start sampling

2. The attacker joins the network at a later time as the legit nodes

(a) Turn off everything

(b) Turn on LPS except anchor 0

(c) Turn on anchor 0

(d) Remotely reset Crazyflie

(e) Start sampling

(f) Turn on spoofers after about 100 seconds

The spoofing devices were shut down the entire time for the normal experiment.

3.2 Sampling

We used the `logging` and `CFLib` modules provided by Bitcraze to record the position estimates in Python 3 using the following parameters:

- Sampling rate of $20ms$
- Logging delay of $20s$
- A total number of samples of 10000
- Duration = $(20 + 10000 * 20)/1000 = 220s$

The connection to Crazyflie was through a USB cable rather than Crazyradio (the nRF24-based transceiver used to communicate wirelessly between PC and Crazyflie) for a more reliable connection and independence from the 250mAh battery of Crazyflie. It's worth noting that during the entire sampling time, all nodes, including the Crazyflie, were static (i.e., acceleration=velocity=(0,0,0)).

4 Methods

4.1 TESLA

Authentication is essential when it comes to network security. Without authentication, there is no way of knowing whether messages have been tampered with. In broadcast authentication protocols, a source wants to broadcast verifiably untampered messages to everyone. Many authentication protocols have been exclusively

designed for broadcast authentication for lightweight applications. The Timed Efficient Stream Loss-tolerant Authentication (TESLA) is a well-known broadcast authentication protocol that was designed to provide low communication and computation overhead for stream purposes Perrig et al. [2000]. In μTESLA, it is assumed that the sender and receivers are loosely time-synchronized. The idea is to repeatedly use a hash function F over a pseudo-random value x; the result is a set of keys, also known as a *keychain*. The last key in this chain (K_0 or commitment) is assumed to be shared among parties. The sender will then assign each key to a timeslot within a sequence of intervals, in which D denotes the length of an interval. The sender will then use each key (in reverse order) for the MAC algorithm on message M_i to broadcast authenticated packets P_i. The most crucial phase is the so-called *delayed key-disclosure*. After disseminating each packet, the receivers will buffer fresh packets until the sender releases the key after some delay, denoted by d, measured in units of D. This way, the recipients can later verify the old packets they received. The construction of packets is as follows:

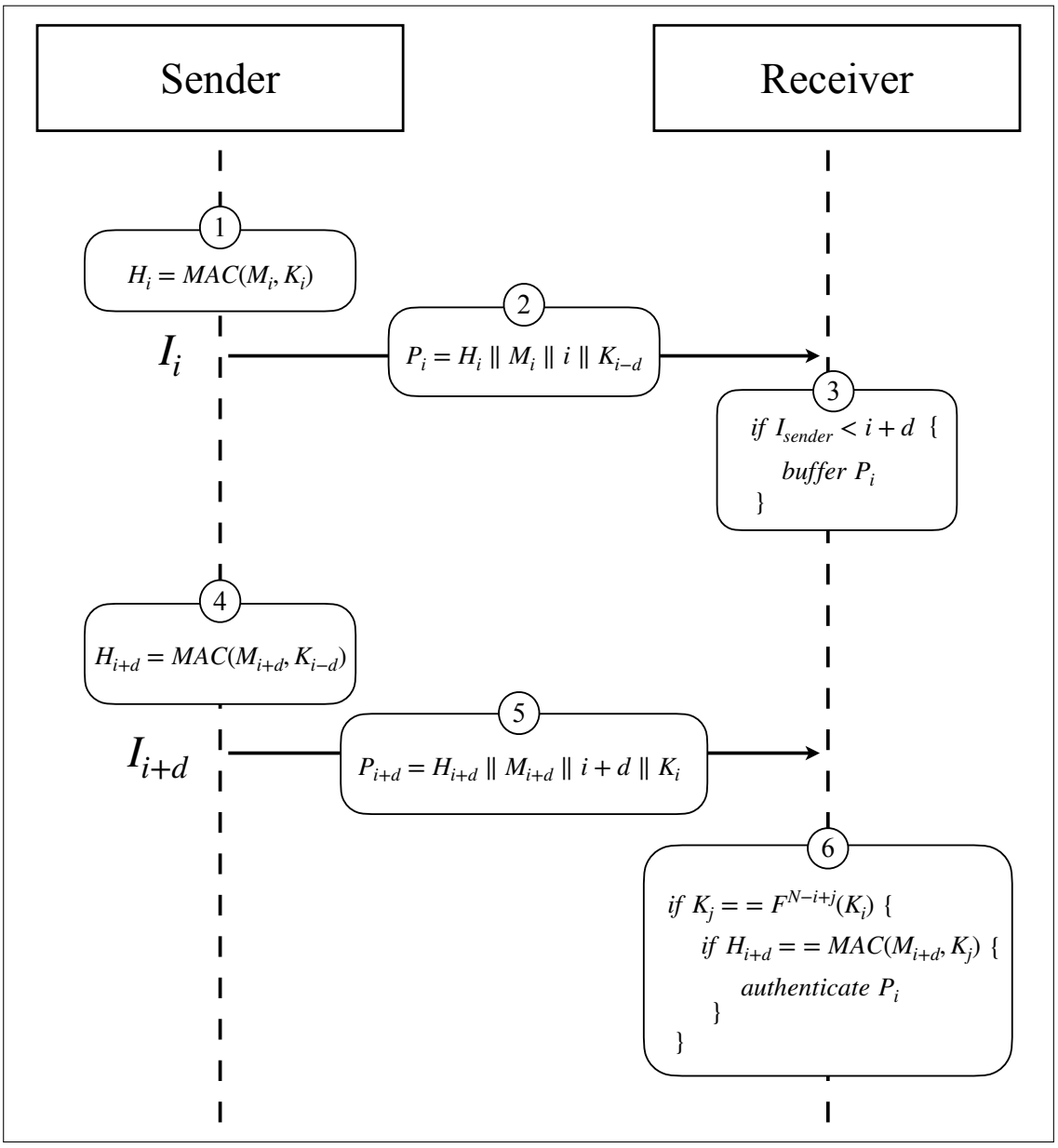
$$P_i = M_i \parallel i \parallel MAC(M_i, K_i) \parallel K_{i-d}$$

Even if some packets are lost, the keys can still be regenerated using future packets, e.g., given $d = 2$, if P_4 which contains K_{N-1} is lost, the disclosed key in P_5 (i.e., K_{N-2}) can be used to generate K_{N-1} needed to authenticate the buffered packet P_2, i.e., $K_{N-1} = F(K_{N-2})$. When the keys in the chain are exhausted, a new key chain will be generated using the same process. The general form of the broadcasting phase can be seen in Figure 5. Regarding cryptanalysis, the TESLA protocol has been formally verified both in the symbolic and computational modelBasin et al. [2011].

4.2 Implementation and limitations

Crazyflie 2.1 has 192KB RAM and 1MB of flash storage, while those numbers are respectively reduced to 16KB and 128KB for each Loco Positioning Node. Assuming that each Loco node broadcasts 50 packets per second, and assuming the size of LPP to be $12 + 16 + 16 = 44$, there will be a need of $3 * 50 * 44 = 6600$ bytes assuming a buffer of 3 seconds, which may not be available. Furthermore, if we assume the total duration of a TESLA session to be 10 minutes, then each anchor would need $60 * 10 * 16 = 9600$ bytes assuming key intervals of 1 second. There is not enough memory on the Loco node to hold the entire keychain in memory, so one solution would be to store the keychain on a micro-SD card and fetch the keys before reaching the corresponding interval. This way, it would be possible to cover longer sessions. Since navigation is a real-time service, one problem arising from using the standard TESLA would be the introduction of delays. If the key-disclosure delay is 1 second, the drone is constantly navigating 1 second behind its current position, which may not be desirable. One way to address this problem would be to modify the TESLA protocol to enable immediate authentication. This method could be particularly beneficial since the anchors have static coordinates (they do not move), so predicting future positions become trivial. It has been established that the specification of

Figure 5: Broadcasting phase in μTESLA.
The sender computes H_i in (1), and broadcasts P_i at I_i in (2). The receiver obtains P_i and buffers it if safe in (3). The sender computes H_{i+d} in (4) and broadcasts P_{i+d} at I_{i+d} in (5). Upon receipt by the receiver in (6), the receiver once again checks if the packet is safe (omitted in figure), and computes $F^{N-i+j}(K_i)$ to match with K_j, where j is the updated commitment index or $j = 0$; if matched, it can authenticate the packet using MAC.



TESLA may only be practical for some systems. To cope with these facts, we opted to use a modified version of TESLA to provide immediate authentication as described in Perrig et al. [2001]. In reality, we ran the experiment using a mock protocol version

mentioned earlier. Namely, the system could not afford a long keychain to cover the entire session, so we reused the same keychain. This practice is not secure but tries to reflect the performance of immediate TESLA in the system when enough storage is available. We call this implementation Mock Immediate TESLA (MI-TESLA). The key commitments for anchors were predetermined and hard-coded in Crazyflie. Furthermore, the interval duration was set to 1 second, and the disclosure delay (d) was set to 2. The packets that contain the coordinates for anchors are called Loco Positioning Protocol (LPP) in the Crazyflie and LPS firmware. We extended LPP to contain the authentication information needed for TESLA. In algorithm 4.1 we describe the role of anchors, and in algorithm 4.2, we describe how MI-TESLA works, where h is a cryptographic hash function and MAC are the Message Authentication Code algorithm as described before.

Algorithm 4.1: Loco Node keychain generation and its usage

Data: h, MAC, seed, interval, keychainSize, time, packet
Result: keychain for anchor, authentication information in Loco packets

```
1  struct {
2      position pos
3      string mac
4      string hash
5      string key
6      int time
7  } lpp
8  Function getKeychain():
9      array keychain = [seed]
10     for i = 0 to keychainSize do
11         keychain[i] = h(keychain[i − 1])
12     return keychain
13 array keychain = getKeychain()
14 Function TX():
15     string keyToDisclose = shouldDisclose ?
         keychain[indexToKeyForDisclosure] : nil
16     position pos = some position
17     position nextPos = some position
18     string hash = h(nextPos)
19     string keyForInterval = keychain[keychainSize - interval]
20     string mac = MAC(pos | hash | time, keyForInterval)
21     lpp p = (pos, mac, hash, keyToDisclose, time)
22     packet.payload.add(p)
23     send(packet)
```

Algorithm 4.2: Packet authentication in Crazyflie, where K_0 is the key commitment and d is the disclosure delay, x is the highest interval the sender could be.

```
Data: h, MAC, K_0, x, interval, buffer, keychainSize, time, lpp
Result: 1 if lpp is immediately authentic, 0 if lpp is unauthentic or not
        immediately authenticable, -1 if unknown
if lpp.key exists then
    if lpp.key is a descendent of K_0 then
        for p in buffer do
            if h(lpp.pos) == p.hash then
                if p.mac == MAC(p.pos|p.time|p.hash, lpp.key) then
                    return 1
else
    if x < interval+d then
        buffer lpp
        return -1
return 0
```

Loose time-syncrhonization The time synchronization method between the anchors and the tag is irrelevant for TESLA as long as it achieves the loose time synchronization prior to the actual session. A simple approach would be to use CFLib in Python to transmit the timestamps of the PC. The Crazyflie would then immediately broadcast the timestamps periodically. All nodes would reset their TESLA time to that of Crazyflie upon receipt. This would achieve the loose time-synchronization required for TESLA to work.

Extended Kalman Filter (EKF) Upon authentication, measurements are sent to the EKF given the updated H vector

$$H = [\frac{x-x_j}{d_j} - \frac{x-x_i}{d_i}, \frac{y-y_j}{d_j} - \frac{y-y_i}{d_i}, \frac{z-z_j}{d_j} - \frac{z-z_i}{d_i}, 0, 0, 0, 0, 0, 0] \tag{4}$$

where i, j are the indices of fixed anchors, (x_i, y_i, z_i) and (x_j, y_j, z_j) are the corresponding absolute positions of those anchors, and (x, y, z) is the position of the Crazyflie, and d_k is the distance of anchor k from Crazyflie.

More information about estimation using the Kalman filter in Crazyflie and its robustness can be found in Zhao et al. [2021].

5 Results

For the normal case without spoofers, we consistently observed the least deviation. In figure 6, the first blob of data (a) represents the normal (untampered) 2-dimensional

position estimates. (d) shows the same data when displayed over time. The standard deviation was unsurprisingly low ($\sigma_x \approx 8mm, \sigma_y \approx 5mm$). After rerunning the experiment in the spoofers' presence, we observed different estimator behavior. Namely, we get drastically different results when spoofers join the network in the middle of session ($\sigma_x \approx 15mm, \sigma_y \approx 37mm$) versus if they joined from the start ($\sigma_x \approx 841mm, \sigma_y \approx 844mm$).

In the case of MI-TESLA in Figure 7, it can be observed that this difference is almost negligible. While it does not show a sign of improvement in the case spoofers join the network later, it improves the precision by $778mm$ for the x-estimates and $821mm$ for the y-estimates. Table 2 provides basic statistics for the experiments. Furthermore, figure 8a and 8b display comparisons of polynomial regressions of position estimates.

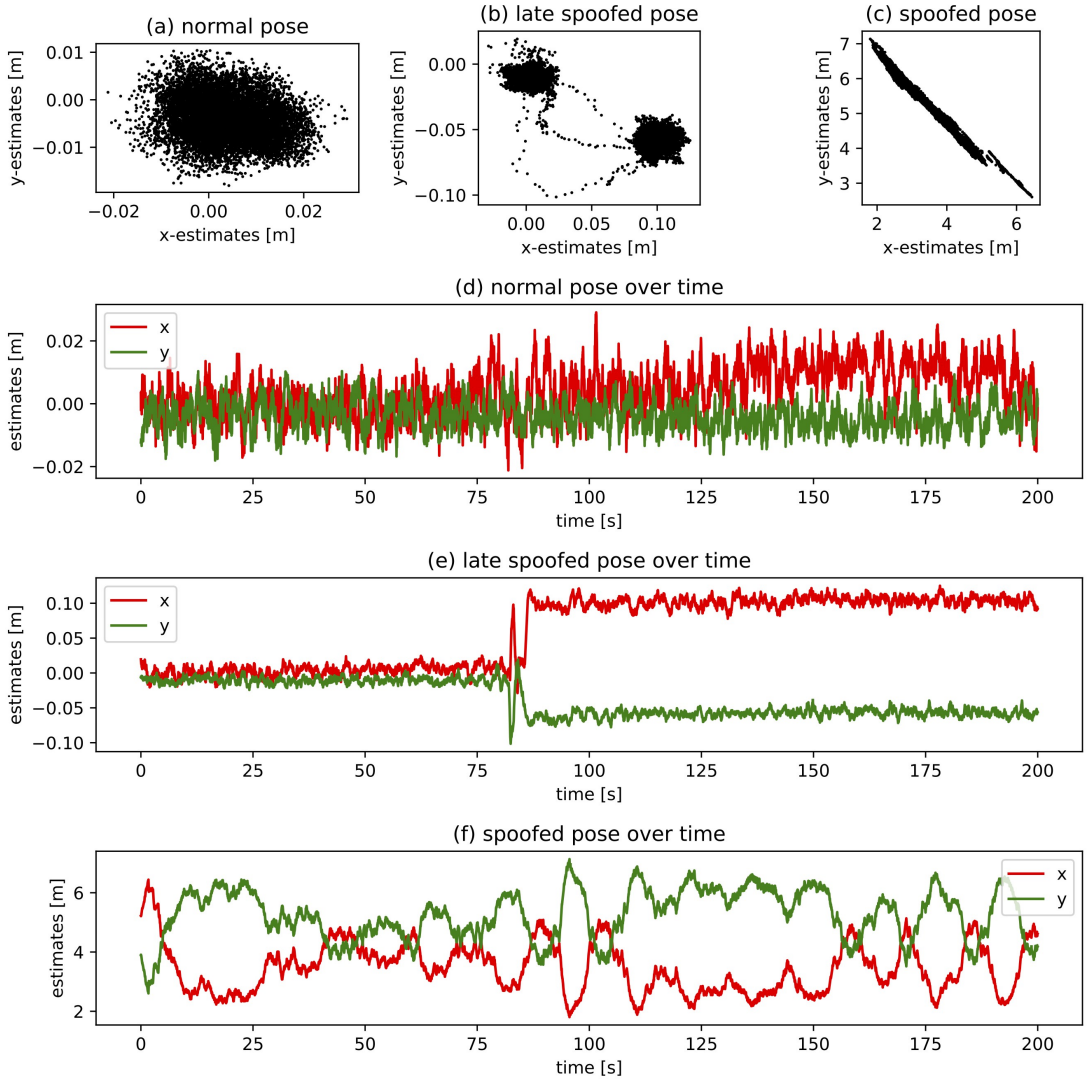


Figure 6: Visualization of position estimates without changing the firmware

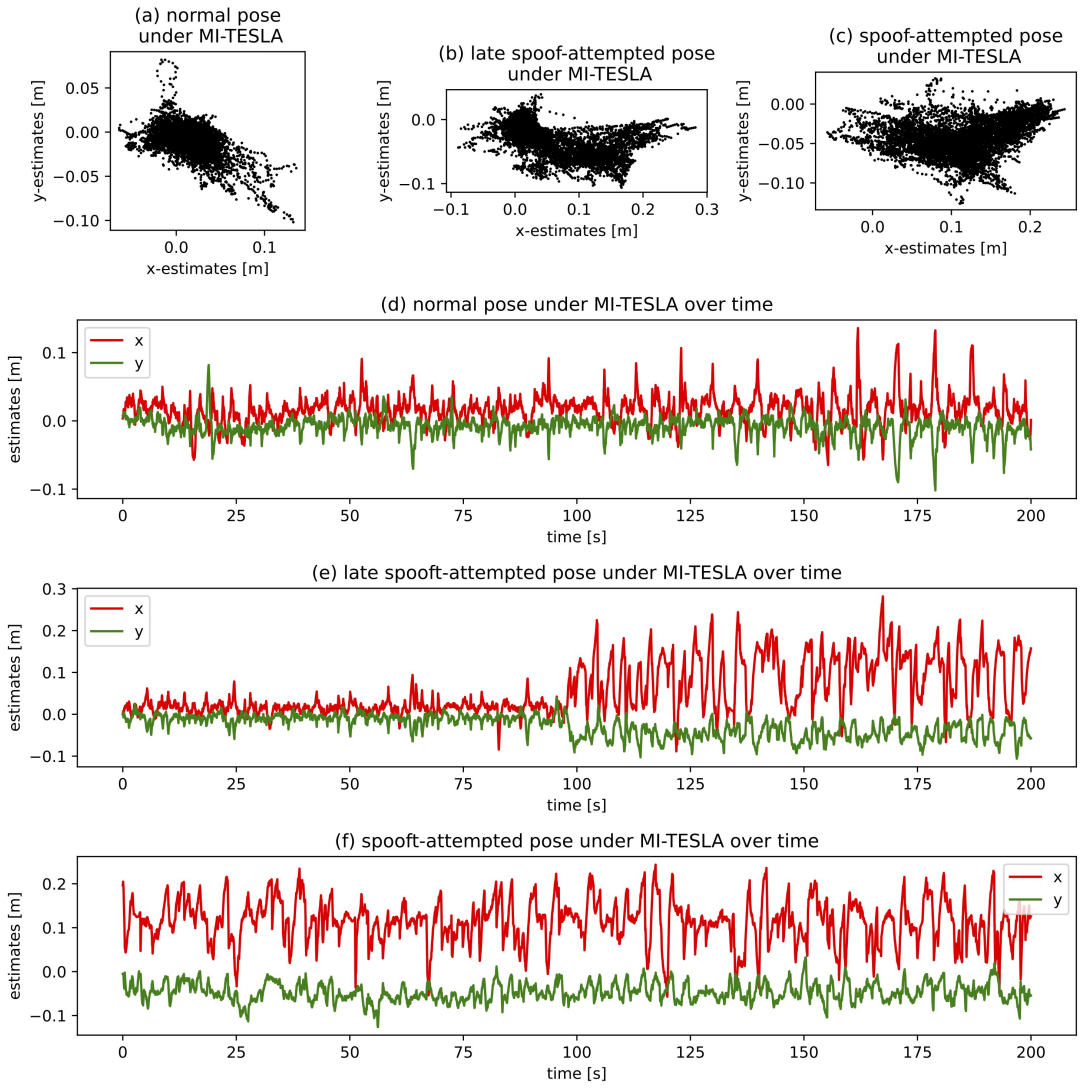
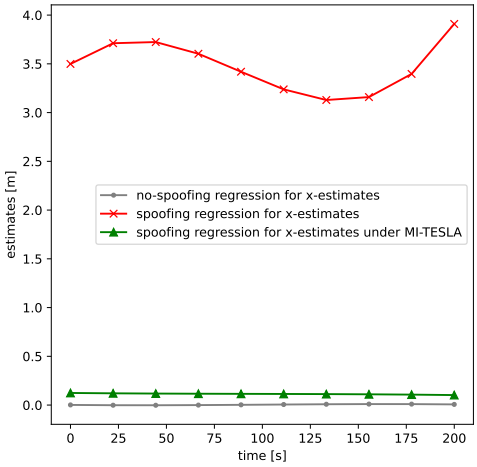


Figure 7: Visualization of position estimates under MI-TESLA

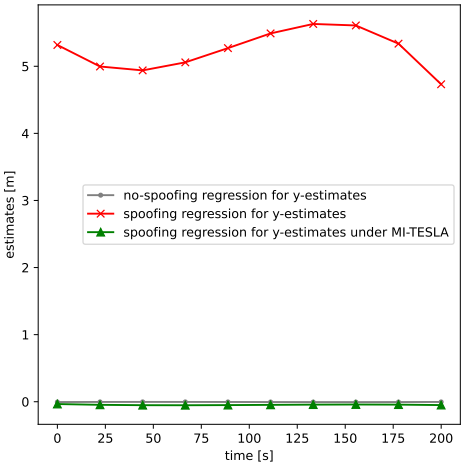
6 Conclusion

The results from this experiment show how easy it is to spoof navigation signals in an indoor positioning system and the need for more research in this area. We further established that using broadcast authentication algorithms such as TESLA that have been previously used in GNSS can be beneficial for UWB-based positioning systems. We found that using MI-TESLA, the precision of position estimates can be drastically improved depending on the parameters and scale of the signal-spoofing attack.

Future Works In this experiment, we discovered that the true-false ratio of signals could greatly impact the TDoA measurements and, subsequently, the estimates. Our guess is that there should exist a threshold function that gives the maximum number



(a) comparison of x-estimates



(b) comparison of y-estimates

Figure 8: Comparison of the normal case, spoofed data, and MI-TESLA in between.

Table 2: Basic statistics (in meters).

Denotations: σ is the standard deviation. μ is the mean.

	σ	μ	min	max
x-estimate/no MI-TESLA/normal	0.0077	0.0041	-0.0212	0.0291
y-estimate/no MI-TESLA/normal	0.0046	-0.0040	-0.0181	0.0104
x-estimate/no MI-TESLA/late spoofed	0.0146	0.0217	-0.0346	0.1039
y-estimate/no MI-TESLA/late spoofed	0.0368	-0.0496	-0.1015	0.0532
x-estimate/no MI-TESLA/spoofed	0.8410	3.4507	1.8038	6.4408
y-estimate/no MI-TESLA/spoofed	0.8443	5.2646	2.6044	7.1349
x-estimate/with MI-TESLA/normal	0.0213	0.0167	-0.0646	0.1359
y-estimate/with MI-TESLA/normal	0.0145	-0.0082	-0.1019	0.0818
x-estimate/with MI-TESLA/late spoof-attempted	0.0485	0.1141	-0.0573	0.2435
y-estimate/with MI-TESLA/late spoof-attempted	0.0208	-0.0464	-0.1261	0.0325
x-estimate/with MI-TESLA/spoof-attempted	0.0626	0.0592	-0.0746	0.2793
y-estimate/with MI-TESLA/spoof-attempted	0.0230	-0.0215	-0.1062	0.0795

of spoofers a positioning system can tolerate before becoming unusable. In this experiment, we only focused on the precision of estimates, and we want to extend our work by calculating root-mean-square deviation (RMSD) in order to make accurate claims about taking control of a drone and landing it somewhere else. For that, we need to use a new constellation of anchors with absolute coordinates measured by a total station. Here are some other interesting topics to extend this work:

- The impact of using authentication on battery usage.

- Implementation challenges regarding buffer and size of a keychain
- Dynamic analysis of estimates when using authentication
- Implications for different positioning systems and TDoA protocols
- Bootstrapping challenges of TESLA

References

Loco Positioning system. https://www.bitcraze.io/documentation/system/positioning/loco-positioning-system/. [Accessed 24-Oct-2022].

Loco Positioning Protocol. https://www.bitcraze.io/documentation/repository/lps-node-firmware/master/protocols/lpp/. [Accessed 24-Oct-2022].

Galileo Open Service Navigation Message Authentication. https://gssc.esa.int/navipedia/index.php/Galileo_Open_Service_Navigation_Message_Authentication. [Accessed 24-Oct-2022].

GPS receivers can be 'spoofed,' say researchers. https://phys.org/news/2008-09-gps-spoofed.html. [Accessed 7-Nov-2022].

M. A. Alawami and H. Kim. Locauth: A fine-grained indoor location-based authentication system using wireless networks characteristics. *Computers & Security*, 89:101683, 2020.

D. Basin, S. Capkun, P. Schaller, and B. Schmidt. Formal reasoning about physical properties of security protocols. *ACM Transactions on Information and System Security (TISSEC)*, 14(2):1–28, 2011.

G. T. Becker, S. Lo, D. De Lorenzo, D. Qiu, C. Paar, and P. Enge. Efficient authentication mechanisms for navigation systems-a radio-navigation case study. In *Proceedings of the 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2009)*, pages 901–912, 2009.

V. Chamola, P. Kotesh, A. Agarwal, Naren, N. Gupta, and M. Guizani. A comprehensive review of unmanned aerial vehicle attacks and neutralization techniques. *Ad Hoc Networks*, 111:102324, 2021. ISSN 1570-8705. doi: https://doi.org/10.1016/j.adhoc.2020.102324. URL https://www.sciencedirect.com/science/article/pii/S1570870520306788.

K. Grover and A. Lim. A survey of broadcast authentication schemes for wireless networks. *Ad Hoc Networks*, 24:288–316, 2015. ISSN 1570-8705. doi: https://doi.org/10.1016/j.adhoc.2014.06.008. URL https://www.sciencedirect.com/science/article/pii/S1570870514001632.

X. Guo and J. Zhu. Research on security issues in wireless sensor networks. In *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, volume 2, pages 636–639. IEEE, 2011.

T. Humphreys. Statement on the vulnerability of civil unmanned aerial vehicles and other systems to civil GPS spoofing. *University of Texas at Austin (July 18, 2012)*, pages 1–16, 2012.

A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys. Unmanned aircraft capture and control via GPS spoofing. *Journal of Field Robotics*, 31(4):617–636, 2014.

S. Z. Khan, M. Mohsin, and W. Iqbal. On GPS spoofing of aerial platforms: a review of threats, challenges, methodologies, and future research directions. *PeerJ Computer Science*, 7:e507, 2021.

B. W. O'Hanlon, M. L. Psiaki, J. A. Bhatti, D. P. Shepard, and T. E. Humphreys. Real-Time GPS Spoofing Detection via Correlation of Encrypted Signals. *NAVIGATION*, 60(4):267–278, 2013. doi: https://doi.org/10.1002/navi.44. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/navi.44.

A. Perrig, R. Canetti, J. Tygar, and D. Song. Efficient authentication and signing of multicast streams over lossy channels. In *Proceeding 2000 IEEE Symposium on Security and Privacy. S P 2000*, pages 56–73, 2000. doi: 10.1109/SECPRI.2000.848446.

A. Perrig, R. Canetti, D. Song, and J. D. Tygar. Efficient and secure source authentication for multicast. In *Network and Distributed System Security Symposium, NDSS*, volume 1, pages 35–46, 2001.

A. Perrig, R. Canetti, D. Song, P. D. Tygar, and B. Briscoe. Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction. RFC 4082, June 2005. URL https://rfc-editor.org/rfc/rfc4082.txt.

D. Prashar, M. Rashid, S. T. Siddiqui, D. Kumar, A. Nagpal, A. S. AlGhamdi, and S. S. Alshamrani. Sdswsn—a secure approach for a hop-based localization algorithm using a digital signature in the wireless sensor network. *Electronics*, 10(24):3074, 2021.

B. Preneel. *Hash Functions*, pages 543–553. Springer US, Boston, MA, 2011. ISBN 978-1-4419-5906-5. doi: 10.1007/978-1-4419-5906-5_580. URL https://doi.org/10.1007/978-1-4419-5906-5_580.

D. Qiu. Security analysis of geoencryption: A case study using loran. In *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*, pages 1146–1154, 2007.

D. Qiu, S. Lo, P. Enge, D. Boneh, and B. Peterson. Geoencryption using loran. In *Proceedings of the 2007 National Technical Meeting of The Institute of Navigation*, pages 104–115, 2007.

Y. Zhang, W. Liu, Y. Fang, and D. Wu. Secure localization and authentication in ultra-wideband sensor networks. *IEEE Journal on Selected areas in communications*, 24(4):829–835, 2006.

W. Zhao, J. Panerati, and A. P. Schoellig. Learning-based bias correction for time difference of arrival ultra-wideband localization of resource-constrained mobile robots. *IEEE Robotics and Automation Letters*, 6(2):3639–3646, 2021.

W. Zhao, A. Goudar, X. Qiao, and A. P. Schoellig. UTIL: An Ultra-wideband Time-difference-of-arrival Indoor Localization Dataset. *arXiv e-prints*, art. arXiv:2203.14471, Mar. 2022.