engineering proceedings

MDPI

Proceeding Paper

UWB-Based Positioning Is Not Invulnerable from Spoofing Attacks: A Case Study of Crazyswarm †

Mahyar Shariat [1,*], Jelena Gabela Majić [2], Max Brandstätter [2] and Wolfgang Kastner [1]

[1] Institute of Computer Engineering, TU Wien, Treitlstr. 1-3/E191, 1040 Vienna, Austria; wolfgang.kastner@tuwien.ac.at
[2] Department of Geodesy and Geoinformation, TU Wien, Wiedner Hauptstraße 8/E120, 1040 Vienna, Austria; jelena.gabela@tuwien.ac.at (J.G.M.); max.brandstaetter@tuwien.ac.at (M.B.)
* Correspondence: mahyar.shariat@tuwien.ac.at; Tel.: +43-1-5880118316
† Presented at the European Navigation Conference 2024, Noordwijk, The Netherlands, 22–24 May 2024.

Abstract: Spoofing attacks pose a threat to drones, which can lead to their crash or takeover. As a countermeasure, the European Space Agency has implemented the Timed Efficient Loss-tolerant Authentication (TESLA) broadcast protocol in the Galileo Open Service Navigation Message Authentication (OSNMA) to detect such events. This study explores the application of TESLA in detecting spoofing attacks targeted at drone swarms that rely on positioning systems utilizing ultra-wideband (UWB) technology. The results of our experiments reaffirm that UWB-based positioning systems are not automatically invulnerable from spoofing attacks and that cryptographic methods such as TESLA are required to provide a layer of protection against spoofing attacks to detect them effectively.

Keywords: spoofing; UWB; security; authentication; TESLA

1. Introduction

Unmanned aerial vehicles (UAVs), or drones, have seen significant advancements in academia and industry due to technological improvements and reduced size and cost [1]. The Federal Aviation Administration predicts that the United States alone will have 3 million drones by 2028 [2]. Interest in drone swarms has grown due to their varied applications in entertainment, security, transportation, environmental monitoring, and aerial communication platforms. However, security threats such as spoofing attacks are also on the rise [3]. It is essential to tackle these spoofing threats to ensure the safety of navigation [4]. Signal spoofing in a Global Navigation Satellite System (GNSS) is a prominent example that threatens drones and communication networks [5]. Although GNSS signal spoofing has been well studied, spoofing attacks in GNSS-challenged environments, such as UAV swarms where GNSS signals may be unreliable or unavailable, have received less attention. Crazyswarm [6] is a framework for aerial swarms that enables the control of multiple Crazyflie (CF) drones. When operating in a UWB setting, drones use signals from station modules (anchors) for positioning via Loco Positioning System (LPS), with each anchor featuring a DWM1000-based transceiver module, allowing passive drones to calculate their positions using Time Difference of Arrival (TDoA). However, communication is susceptible to spoofing attacks due to the lack of mechanisms for drones to verify the authenticity of anchors [7]. In this paper, we demonstrate how an attacker can hijack a drone swarm by falsifying anchor positions and propose implementing a broadcast authentication protocol to allow the swarm to detect and resist such attacks.

check for updates

Academic Editor: Terry Moore

Published: 7 May 2025

Citation: Shariat, M.; Gabela Majić, J.; Brandstätter, M.; Kastner, W. UWB-Based Positioning Is Not Invulnerable from Spoofing Attacks: A Case Study of Crazyswarm. *Eng. Proc.* **2025**, *88*, 43. https://doi.org/10.3390/engproc2025088043

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

2. Related Works

This section reviews the body of literature in order of publication. The research described in [8] presents an authentication scheme using message authentication code (MAC) to enable secure bidirectional time of arrival (ToA) positioning in UWB-based wireless sensor networks (WSNs); however, it does not discuss passive tags. In [9], a *geoencryption* strategy suitable for a long range navigation (LORAN) system is proposed, integrating the TESLA protocol in navigation. It shows that passive tags can authenticate signals and protect against spoofing, although it is limited to LORAN. Furthermore, in [10], suggestions for enhancing TESLA for navigation systems, particularly aimed at enhanced long range navigation (eLORAN), are discussed. The study in [11] introduces a security mechanism for WSN using elliptic curve cryptography (ECC) to safeguard ToA-based positioning, claiming to be effective even in the presence of malicious actors. However, this approach presumes a two-way communication framework among participants. In [12], a position estimation algorithm suitable for passive and active UWB nodes is proposed in the presence of spoofing attacks. However, the emphasis is on the accuracy of position estimation rather than the detection of such events; therefore, the lack of cryptographic primitives is evident. The research in [13] reveals a classification system intended to identify malicious vs. truthful nodes through signal features such as the received radio strength indicator (RSSI) in restricted settings; however, it assumes that cryptographic mechanisms already are in place and only adds a layer of location-based classification. In [14], a digital signature-based method is proposed for secure positioning in WSN; however, public-key cryptography (PKC) may not be suitable for high-frequency streams [15,16]. The study in [17] proposes sub-STS, a modification of the scrambled timestamp sequence (STS) in IEEE 802.15.4z. However, this technique does not fully address the TDoA case and requires that the sender and receivers agree on a shared key and initial vector before measuring ToA, a symmetry that could allow receivers to create authentic signals [18]. In [7], a simulated TESLA model is used for Crazyflie drones as a proof of concept, but this research has some drawbacks, such as having a single point of failure in the system and the static condition of the drones rather than being airborne. Finally, in [19] a method is proposed to mitigate spoofing events within the convex hull; however, it does not use cryptographic primitives.

3. Threat Model

In this section, we explore the interactions between the networks, highlighting the functions and communication strategies of anchors and tags. Furthermore, we describe possible threats, especially those related to spoofing attacks, as well as different scenarios and their effectiveness.

The symbols used in this paper are defined in Table 1.

Table 1. Notations.

Symbol	Description
$\mathscr{A}$	UWB-based anchor
$\mathscr{T}$	UWB-based passive tag (e.g., a drone)
$\mathcal{A}$	An ordered subset of $\mathbb{R}^3$, indexed by $\mathscr{A}$ elements.
$\mathcal{T}$	An ordered subset of $\mathbb{R}^3$, indexed by $\mathscr{T}$ elements.
$\|.\|_2$	ℓ_2-norm, i.e., the Euclidean distance
$h(x)$	Collision Resistant Hash Function with x as input of arbitrarily length
K_i	ith key in TESLA keychain
$HMAC(x,k)$	Message Authentication Code based on $h(x)$ with k as the key
$\|$	Concatenation operation
N	TESLA keychain size (i.e., number of keys)
s	TESLA keychain seed, a pseudorandom value
D	TESLA interval length, measured in seconds
d	TESLA key disclosure delay, measured in units of D
I	TESLA interval index with $0 \leq I < D \times N$

3.1. Network Model

We distinguish between trusted and untrusted networks in our model. A trusted network is a fixed set of a number of active nodes that want to broadcast navigation information to an untrusted network with a large number of passive nodes. We refer to such a node in the trusted network as an anchor, denoted by $\mathscr{A}$. We refer to a passive node in the untrusted network as a tag, denoted by $\mathscr{T}$. The goal of the trusted network is to provide continuous navigation information that the tags in the untrusted network need for self-positioning. We assume a broadcast-only communication model from the trusted network to the untrusted network. While in practice, bootstrapping a broadcast scheme can involve unicast communication, we mainly focus on the threats that may arise from the one-way communication phase. Although bidirectional unicast between $\mathscr{A}$ and $\mathscr{T}$ is common in wireless networks, it can be difficult to manage as the number of nodes in the network grows; specifically, the number of connections needed in a two-way setting would be equal to the number of edges in a complete bipartite graph.

3.2. Positioning

In the proposed network model, the time of flight (ToF) between an anchor $\mathscr{A}$ and a target $\mathscr{T}$ cannot be calculated directly. However, the time of arrival (ToA) at $\mathscr{T}$ can still be measured. Given the difference in ToA, denoted as δToA, for two synchronous signals from different anchors indexed by i and j, and their corresponding positions $\mathcal{A}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ and $\mathcal{A}_j = (x_j, y_j, z_j) \in \mathbb{R}^3$, the following equation can be formulated as a hyperboloid:

$$\begin{aligned} \delta\text{ToA} \times c &= \|\mathcal{T}_* - \mathcal{A}_i\|_2 - \|\mathcal{T}_* - \mathcal{A}_j\|_2 \\ &= \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - \sqrt{(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2} \end{aligned} \tag{1}$$

where c is the speed of light and $\mathcal{T}* = (x, y, z)$ is the unknown position of $\mathscr{T}$. With four or more hyperboloids, $\mathcal{T}*$ can be estimated.

3.3. Attacker Model

We assume a computationally bounded active attacker in the network. The attacker has full access to the network channel and can manipulate the traffic in the network, for example, by eavesdropping, changing, replaying, or delaying messages. The attacker is assumed to be powerful, but does not possess unbounded resources. We also distinguish between an insider attacker and an outsider attacker. An insider attacker is part of the trusted network; it is considered a stronger notion that symmetric-key methods do not protect against [20]. The ideal goal of the attacker is to smoothly capture self-positioning tags undetected. As the attacker has full access to the network, it can block packets from arriving or being processed, for example, by jamming the signals. Furthermore, we assume that the attacker knows the true position of $\mathscr{T}$.

3.4. Spoofing Model

We define a spoofing attack as any RF-based [21] attempt by the attacker to influence the perceived position of the tags. In our model, the tags are drones, and as the drones cannot verify whether $\mathcal{A}$ originates from the trusted network, the attacker can send arbitrary positions and influence the swarm. To illustrate such a scenario, we consider a simple 2D example, as depicted in Figure 1, with three synchronous senders out of which two belong to the trusted network and one to the attacker. Each anchor broadcasts its position simultaneously with other anchors. The drones can then use these positions and TDoA measurements for positioning. However if the attacker changes the broadcast position of an anchor, it will result in a change in the perceived position of the swarm.

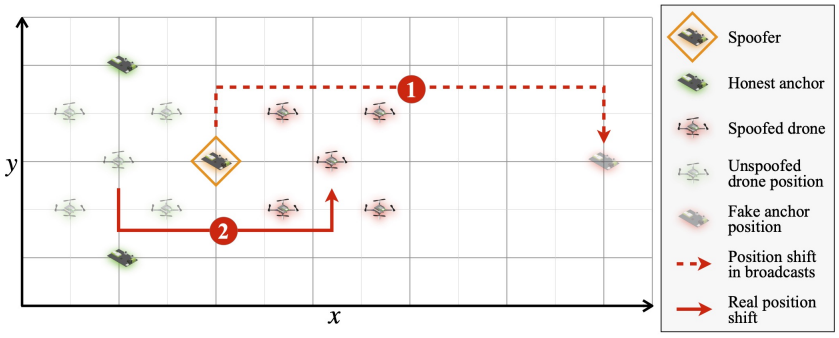


Figure 1. A simple TDoA positioning scenario with a swarm of drones and three anchors. Here, spoofing occurs when one anchor broadcasts a fake position when it is really not there (1), leading to incorrect drone positions (2).

4. Requirements

This section describes the necessary criteria for configuring a system that aligns with our threat model.

4.1. TDoA Mode

So far, we have assumed that anchor transmissions are simultaneous; however, in reality, a sensor pairing algorithm should be implemented before determining the TDoA. There are two methods to achieve this: centralized or decentralized sensor pairing [22]. We focus on the decentralized method due to its scalability in order to find suitable TDoA measurements, as described in [23].

4.2. Spoofing Detection Method

We recognize two categories for detecting a spoofing attack: with and without cryptography. We emphasize that any spoofing detection method that does not use cryptographic primitives is limited and cannot be generalized to all situations. For example, in Figure 1, the attacker gradually changes its broadcast position, making it difficult—if not nearly impossible—to detect such an event without using cryptography, especially when the spoofed position remains inside the convex hull. On the other hand, we acknowledge that relying solely on cryptography for spoofing attack detection, although necessary, is not always sufficient. Building on top of our threat model, we focus only on a method based on cryptographic primitives in this work.

TESLA

The Time-Efficient Stream Loss-tolerant Authentication (TESLA) method is a formally verified broadcast authentication protocol that prioritizes low communication and computational overhead [24,25]. It assumes time synchronization between senders and receivers. The TESLA approach involves the iterative application of a cryptographic hash function h to a pseudorandom value s to generate a set of keys, known as a keychain. Specifically, for a keychain of size N, the keys are $K_N = h(s)$ and $K_i = h(K_{i+1})$ for i, ranging from $N-1$ down to 0. As part of the bootstrapping, K_0 along with other parameters, is shared with receivers. The sender assigns each key to a time slot within a sequence of intervals, with D as the interval length. The sender uses the HMAC algorithm on the message M_i to broadcast authenticated packets P_i. After broadcasting each packet, the receivers buffer safe packets until the sender releases the key after a delay d (measured in units of D). This enables recipients to verify the packets in the buffer. A packet is constructed as follows:

$$P_i = M_i \parallel i \parallel HMAC(M_i, K_i) \parallel K_{i-d} \quad (2)$$

Even if some packets are lost, key regeneration is possible using future packets.

Although TESLA is effective and appropriate for streaming applications, it exhibits several drawbacks compared with digital signature systems, including the need for synchronization between sender and receiver, it not being easy to implement in all use cases, memory requirements, susceptibility to delay and denial-of-service (DoS) attacks, and lag in authentication [26,27]. However, TESLA is still considered appropriate for the authentication of navigational broadcasts, such as in Galileo Open Service Navigation Message Authentication (OSNMA), due to the small size of the authentication messages, the resilience to message loss, and the simple key distribution mechanism [28].

5. Experiment

In this study, a constellation of 16 anchors was deployed in the laboratory and their relative positions were accurately determined using a total station theodolite, which produced the coordinate system, as shown in our experimental setup in Figure 2. All of the measurements in the rest of this paper are expressed in meters, unless otherwise indicated.

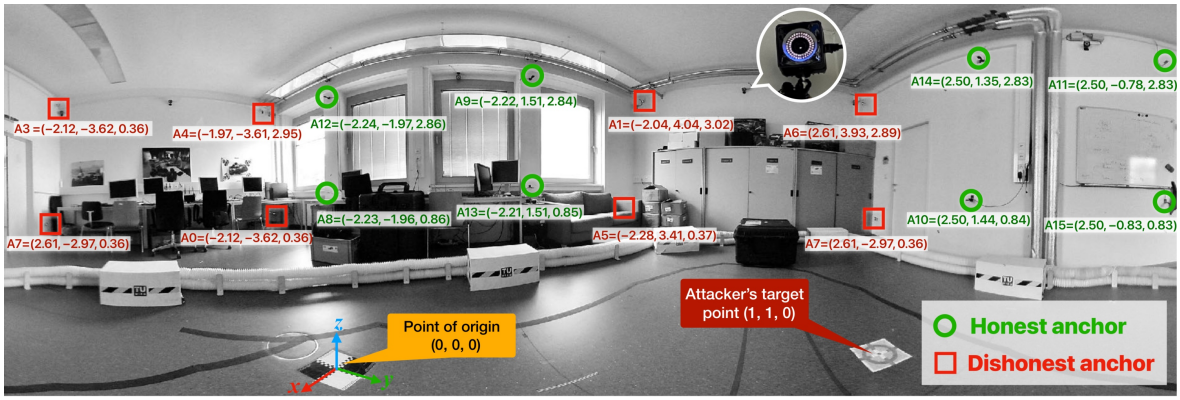


Figure 2. The coordinate system utilized in the experimental setup involves anchors with indices of 0 to 7 for broadcasting deceptive positions, while anchors 8 to 15 are part of the trusted network and broadcast truthful positions. The positions are measured using a total station theodolite.

Due to space limitations, the experiment involved two drones with initial positions of $\mathcal{T}_1 = (0, 0, 0.2)$ and $\mathcal{T}_2 = (0, 0.5, 0.2)$, as depicted in Figure 3, which were controlled using the Crazyswarm2 Robot Operating System (ROS) package. All of the drones and anchors were configured to operate in a decentralized TDoA mode (referred to as TDoA3 in the firmware). We carried out two tests: one without authentication and another with authentication. Our findings from the first test demonstrate that drones can be moved to the desired location by broadcasting fake positions through the LPS, while the perceived position of the drones did not show this change. To differentiate between truthful and deceitful packets, the TESLA protocol was used.

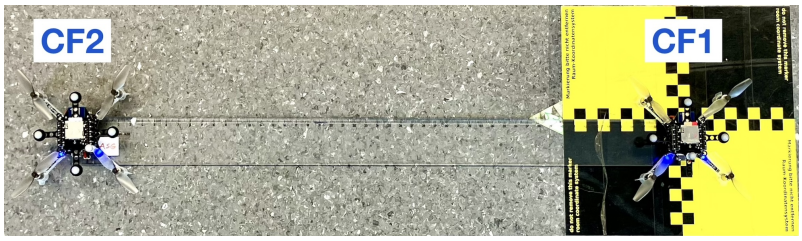


Figure 3. Crazyflie drones in the swarm.

5.1. Authentication Parameters

TESLA's need for time synchronization to work, and the fact of operating in a broadcast setting, led us to implement a similar approach to the flooding time synchronization protocol (FTSP). With the DWM1000's clock sensitivity at LSB/15.65 ps, implementing FTSP yielded a steady (root mean square error) RMSE of approximately 15 µs by Crazyflie. For simplicity, we assign $N = 200, D = d = 1$, and use a standard cryptographic hash function with an output length of 128 bits for keys and truncated HMAC outputs of 64 bits for our TESLA implementation. The information about h and HMAC algorithms, and K_0 was pre-shared. .

5.2. System Configuration

A motion capture system (MoCap) tracked each drone's trajectory, sampling MoCap and UWB-based positions at 2.5 ms intervals. The drones used an Extended Kalman Filter (EKF) for estimation, the implementation details of which can be found in [29,30]. The presence of a remote switch in each LPS helped automate the experiment, allowing events triggered such as take-offs, attacks, and landings at predetermined times.

5.3. Algorithms

We implemented Algorithm 1 for anchors and Algorithm 2 for Crazyflies, and used a single shared keychain for improved authentication. The TESLA keychain generation in Algorithm 1 was previously discussed in Section 4. The `attachAuthInfo()` method was executed just before UWB packet transmission, specifically during the Loco Positioning Protocol (LPP) packet setup in LPS. In Algorithm 2, the drone runs `bufferIfLegit()` upon packet receipt. A timer periodically invokes `timerCallback()` every $d \times D$ seconds to check the authenticity of packets in the buffer and, if verified, pushes TDoA measurements for estimation.

Algorithm 1: TESLA keychain generation in $\mathscr{A}$ and its usage

Data: I, N, s, $\mathcal{A}_*$, txpacket
Result: TESLA keychain, authentication information in txpacket
Function `getKeychain()`**:**
  array keychain[N] = [$h(s)$]
  for $i = 0$ *to* N **do**
    keychain[$i + 1$] = h(keychain[i])
  end
  return *keychain*
array keychain = getKeychain()
Function `attachAuthInfo()`**:**
  string key = I ? keychain[N - I] : nil
  string currentKey = keychain[N - 1 - I]
  string mac = $MAC((\mathcal{A}_*, I)$, currentKey)
  attach ($\mathcal{A}_*$, I, mac, key) to txpacket payload

Algorithm 2: Packet authentication in $\mathcal{T}$

Data: K_0, I, buffer, N, rxpacket
Result: 1 if authentic, 0 otherwise
array keychain[N];
Function `bufferIfLegit()`:
  if *rxpacket is safe AND rxpacket.key is a descendent of* K_0 **then**
    keychain[$N-1-I$] = rxpacket.key;
    calculate tdoa;
    add (tdoa, rxpacket) to buffer;
  end
Function `timerCallback()`:
  for *(tdoa, rxpacket) in buffer* **do**
    if *rxpacket.mac == HMAC(*($\mathcal{A}_*$, *I), keychain[N-1-rxpacket.I])* **then**
      push tdoa to EKF
    end
  end

6. Discussion

In this section, we will discuss the results of our experiment. Figure 4 shows how a swarm can be swayed away over time when $\mathcal{A}$ is fake. Although the drones were commanded to hover over their initial positions, and they believed they were, in reality, they ended up in different positions near (1, 1) on the xy plane based on the attacker's data. After implementing the TESLA protocol, the drones were able to distinguish when the signals came from trusted anchors and remained over the initial positions at (0, 0) and (0, 0.5) as commanded. This was confirmed by the kernel density estimates (KDE) in Figure 5. When authentication was used, the RMSE values for the first Crazyflie drone (CF1) and the second Crazyflie drone (CF2) were approximately (x = 0.058, y = 0.111, z = 0.124) and (x = 0.061, y = 0.164, z = 0.149), respectively, consistent with the DWM1000 specifications. Without authentication, the RMSE values were (x = 0.563, y = 0.533, z = 0.139) and (x = 0.532, y = 0.521, z = 0.133), highlighting the impact of the spoofing attack. A demo is available in Supplementary Materials. Furthermore, distance difference measurements recorded during the experiments showed a noticeable change when the attack was initiated, as illustrated in the sample in Figure 6. In contrast, the position estimates remained statistically stable over the initial positions when authentication was used, as shown in Figure 5.

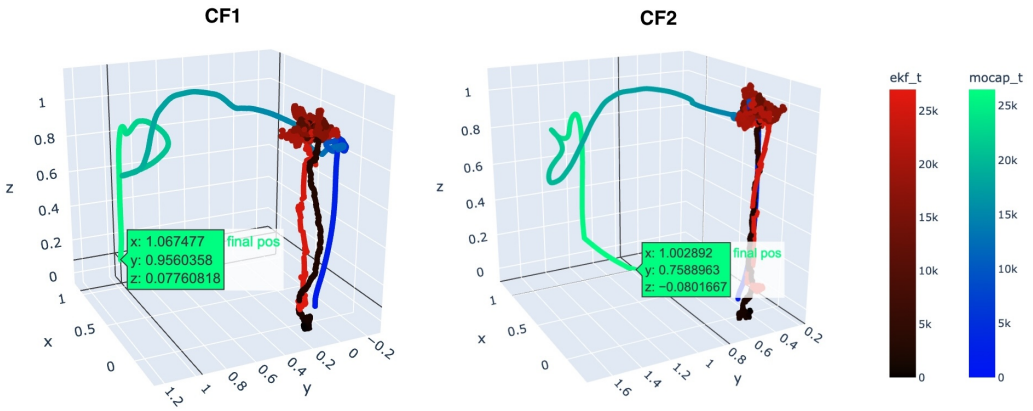


Figure 4. The false perception of the position in a swarm involving two drones (i.e., CF1 and CF2) during the spoofing attack.

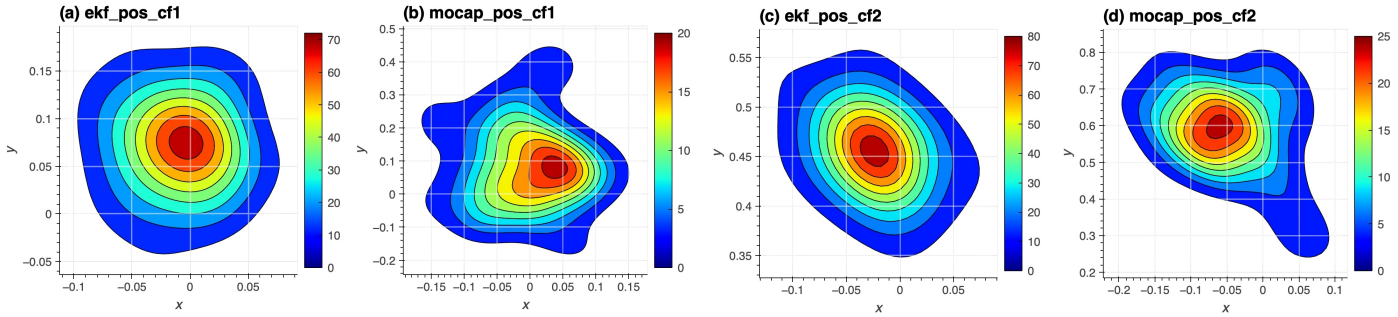


Figure 5. A comparison of the xy kernel density estimation (KDE) in the swarm vs. the ground truth when using authentication. (**a**,**c**) show the KDEs of 2D positions calculated onboard by CF1 and CF2, respectively. (**b**,**d**) show the KDEs of 2D positions of the same drones recorded by the motion capture system.

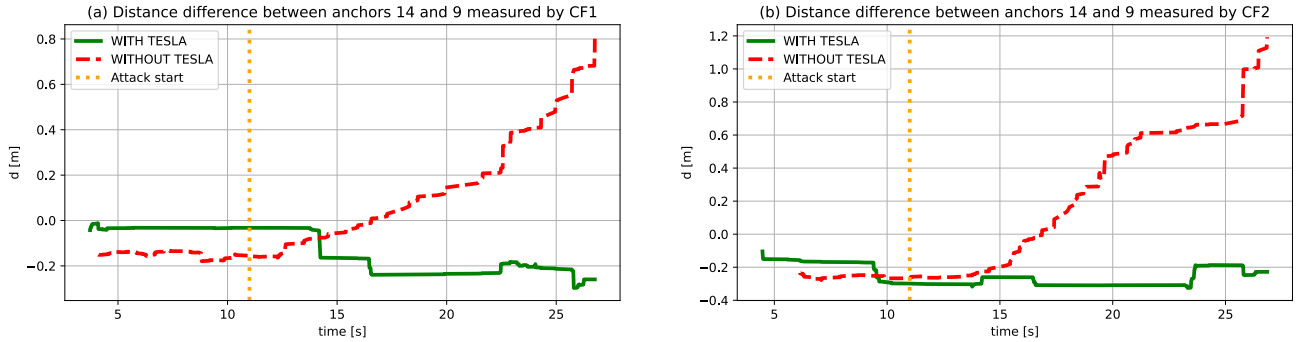


Figure 6. Example of distance difference measurements under attack with and without authentication. Measu.

7. Conclusions

We demonstrated the importance of secure communication protocols in the operation of UAV swarms, particularly in environments where GNSS signals are not available. The deployment of the TESLA protocol, in conjunction with the use of UWB technology for positioning, has proven to be effective in mitigating the risks associated with spoofing attacks. Our experiments have shown that without robust authentication mechanisms, drones are vulnerable to spoofing, which can significantly alter their perceived positions and lead to operational failures. However, when authentication was used, it ensured that drones maintained an accurate position and resisted deceptive signals, demonstrated by stable RMSE values and consistent position estimations in our experiment.

Future research should explore larger UAV swarms, more complex scenarios, and evaluate lightweight digital signatures such as Ed25519 against TESLA.

Supplementary Materials: A demo of the experiment is available at: https://cloud.auto.tuwien.ac.at/index.php/s/zPQFieJdPTRXL6k (accessed on 29 April 2025).

Author Contributions: conceptualization, M.S.; methodology, M.S.; software, M.S.; validation, M.S. and J.G.M.; formal analysis, M.S.; investigation, M.S. and J.G.M.; resources, M.S., J.G.M. and M.B.; data curation, M.S., J.G.M. and M.B; writing—original draft preparation, M.S.; writing—review and editing, M.S., J.G.M. and W.K.; visualization, M.S.; supervision, W.K.; project administration, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Raw data recorded at 400 Hz on each drone, with and without authentication, can be publicly accessed using following link: https://cloud.auto.tuwien.ac.at/index.php/s/GKmAjnWHpqkCE9A (accessed on 29 April 2025). For more information, please contact the correspondent author.

Acknowledgments: We would like to express our gratitude to our colleagues who assisted us in the laboratory and provided valuable feedback. This work has been supported by the Doctoral College Resilient Embedded Systems, which is run jointly by the TU Wien's Faculty of Informatics and the UAS Technikum Wien.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Abdelkader, M.; Güler, S.; Jaleel, H.; Shamma, J.S. Aerial swarms: Recent applications and challenges. *Curr. Robot. Rep.* **2021**, *2*, 309–320. [CrossRef] [PubMed]
2. FAA Aerospace Forecast Fiscal Years 2024–2044. Available online: https://www.faa.gov/dataresearch/aviation/aerospaceforecasts/faa-aerospace-forecast-fy-2024-2044 (accessed on 6 May 2024).
3. Eshmawi, A.A.; Umer, M.; Ashraf, I.; Park, Y. Enhanced Machine Learning Ensemble Approach for Securing Small Unmanned Aerial Vehicles From GPS Spoofing Attacks. *IEEE Access* **2024**, *12*, 27344–27355. [CrossRef]
4. Novák, A.; Kováčiková, K.; Kandera, B.; Sedláčková, A.N. Global Navigation Satellite Systems Signal Vulnerabilities in Unmanned Aerial Vehicle Operations: Impact of Affordable Software-Defined Radio. *Drones* **2024**, *8*, 109. [CrossRef]
5. Humphreys, T. Statement on the Vulnerability of Civil Unmanned Aerial Vehicles and Other Systems to Civil GPS Spoofing. Available online: https://radionavlab.ae.utexas.edu/images/stories/files/papers/Testimony-Humphreys.pdf (accessed on 29 April 2025).
6. Preiss, J.A.; Honig, W.; Sukhatme, G.S.; Ayanian, N. Crazyswarm: A large nano-quadcopter swarm. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017.
7. Shariat, M.; Kastner, W. Authenticated UWB-Based Positioning of Passive Drones. In Proceedings of the 2023 IEEE 19th International Conference on Factory Communication Systems (WFCS), Pavia, Italy, 26–28 April 2023; pp. 1–8.
8. Zhang, Y.; Liu, W.; Fang, Y.; Wu, D. Secure localization and authentication in ultra-wideband sensor networks. *IEEE J. Sel. Areas Commun.* **2006**, *24*, 829–835. [CrossRef]
9. Qiu, D.; Lo, S.; Enge, P.; Boneh, D.; Peterson, B. Geoencryption Using Loran. In Proceedings of the 2007 National Technical Meeting of The Institute of Navigation, San Diego, CA, USA, 22–24 January 2007; pp. 104–115.
10. Becker, G.; Lo, S.; Lorenzo, D.D.; Qiu, D.; Paar, C.; Enge, P. 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, Savannah, GA, USA, 22–25 September 2009; pp. 901–912.
11. Guo, X.; Zhu, J. Research on security issues in Wireless Sensor Networks. In Proceedings of the 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, Harbin, China, 12–14 August 2011.
12. Zhang, P.; Nagarajan, S.G.; Nevat, I. Secure Location of Things (SLOT): Mitigating Localization Spoofing Attacks in the Internet of Things. *IEEE Internet Things J.* **2017**, *4*, 2199–2206. [CrossRef]
13. Alawami, M.A.; Kim, H. LocAuth: A fine-grained indoor location-based authentication system using wireless networks characteristics. *Comput. Secur.* **2020**, *89*, 101683. [CrossRef]
14. Prashar, D.; Rashid, M.; Siddiqui, S.T.; Kumar, D.; Nagpal, A.; AlGhamdi, A.S.; Alshamrani, S.S. SDSWSN—A secure approach for a hop-based localization algorithm using a digital signature in the wireless sensor network. *Electronics* **2021**, *10*, 3074. [CrossRef]
15. Perrig, A.; Song, D.; Canetti, R.; Tygar, J.D.; Briscoe, B. *Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction*; Technical Report RFC 4082; Internet Engineering Task Force: Fremont, CA, USA, 2005.
16. *ISO/IEC 29192-7:2019*; Information Security—Lightweight Cryptography—Part 7: Broadcast Authentication Protocols. International Organization for Standardization: Geneva, Switzerland, 2019.
17. Stocker, M.; Kowalczyk, J.; Boano, C.A.; Römer, K. Towards Secure Multicast Ranging with Ultra-Wideband Systems. In Proceedings of the 2022 International Conference on Embedded Wireless Systems and Networks, Linz, Austria, 3–5 October 2022; EWSN '22, pp. 256–261.
18. Stocker, M.; Groswindhager, B.; Boano, C.A.; Romer, K. Towards secure and scalable UWB-based positioning systems. In Proceedings of the 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Virtual, 10–13 December 2020.
19. Chen, H.; Dhekne, A. Spoofing Evident and Spoofing Deterrent Localization Using Ultrawideband (UWB) Active–Passive Ranging. *IEEE J. Indoor Seamless Position. Navig.* **2024**, *2*, 12–24. [CrossRef]
20. Itkin, E.; Wool, A. A security analysis and revised security extension for the precision time protocol. *IEEE Trans. Dependable Secure Comput.* **2020**, *17*, 22–34. [CrossRef]
21. van der Merwe, J.R.; Zubizarreta, X.; Lukčin, I.; Rügamer, A.; Felber, W. Classification of Spoofing Attack Types. In Proceedings of the 2018 European Navigation Conference (ENC), Gothenburg, Sweden, 14–17 May 2018; pp. 91–99.
22. Meng, W.; Xie, L.; Xiao, W. Optimal TDOA sensor-pair placement with uncertainty in source location. *IEEE Trans. Veh. Technol.* **2016**, *65*, 9260–9271. [CrossRef]

23. Zhao, W.; Goudar, A.; Qiao, X.; Schoellig, A.P. UTIL: An ultra-wideband time-difference-of-arrival indoor localization dataset. *Int. J. Rob. Res.* **2024**, *43*, 1443–1456. [CrossRef]
24. Perrig, A.; Canetti, R.; Tygar, J.D.; Song, D. Efficient authentication and signing of multicast streams over lossy channels. In Proceedings of the 2000 IEEE Symposium on Security and Privacy, S&P 2000, Berkeley, CA, USA, 14–17 May 2002.
25. Basin, D.; Capkun, S.; Schaller, P.; Schmidt, B. Formal reasoning about physical properties of security protocols. *ACM Trans. Inf. Syst. Secur.* **2011**, *14*, 1–28. [CrossRef]
26. Ning, P.; Liu, A.; Du, W. Mitigating DoS attacks against broadcast authentication in wireless sensor networks. *ACM Trans. Sen. Netw.* **2008**, *4*, 1–35. [CrossRef]
27. Annessi, R.; Fabini, J.; Zseby, T. It's about Time: Securing Broadcast Time Synchronization with Data Origin Authentication. In Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, Canada, 31 July–3 August 2017; pp. 1–11.
28. Cancela, S.; Calle, J.D.; Fernandez-Hernandez, I. CPU consumption analysis of TESLA-based navigation message authentication. In Proceedings of the 2019 European Navigation Conference (ENC), Warsaw, Poland, 9–12 April 2019.
29. Mueller, M.W.; Hamer, M.; D'Andrea, R. Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015.
30. Mueller, M.W.; Hehn, M.; D'Andrea, R. Covariance correction step for Kalman filtering with an attitude. *J. Guid. Control Dyn.* **2017**, *40*, 2301–2306. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.