Securing ADS-B with Multi-point Distance-bounding for UAV Collision Avoidance

Zachary P. Languell and Qijun Gu
Texas State University, San Marcos, TX 78666

***Abstract*—The Automatic Dependent Surveillance-Broadcast (ADS-B) protocol is being adopted for use in unmanned aerial vehicles (UAVs) as the primary source of information for emerging multi-UAV collision avoidance algorithms. The lack of security features in ADS-B leaves any processes dependent upon the information vulnerable to a variety of threats from compromised and dishonest UAVs. This could result in substantial losses or damage to properties. This research proposes a new distance-bounding scheme for verifying the distance and flight trajectory in the ADS-B broadcast data from surrounding UAVs. The proposed scheme enables UAVs or ground stations to identify fraudulent UAVs and avoid collisions. The scheme was implemented and tested in the ArduPilot SITL (Software In The Loop) simulator to verify its ability to detect fraudulent UAVs. The experiments showed that the scheme achieved desired accuracy in both flight trajectory measurement and attack detection.**

I. Introduction

The Automatic Dependent Surveillance-Broadcast (ADS-B) protocol is a surveillance technology used by aircraft to share position and navigation information with surrounding aerial vehicles and ground elements. Starting January 1, 2020, aircraft must be equipped with ADS-B Out to fly in most controlled airspace [1]. The Federal regulations mandating this technology will have a direct impact on how new and developing systems may implement functionality, giving its focus on collision avoidance algorithms using this technology.

Although ADS-B will be mandated on manned aircraft at the current stage, the safety awareness enabled by this technology is gaining attention in the community of unmanned aerial vehicles (UAVs) as well [2]–[4]. Growing accidents caused by UAVs colliding with manned aircraft have raised new concerns on airspace and UAV safety [5], [6]. Recent technology advancement has made the ADS-B technology commercially available to consumer UAVs [7], [8]. We foresee that UAVs equipped with ADS-B transponders will be able to alert and detect nearby manned and unmanned aircraft to avoid collisions and improve airspace safety.

The current ADS-B specification does not have any built-in security features. ADS-B packets are broadcast in plain-text and lack any method of authenticating the ADS-B broadcasters and verifying the data in the packets. Using ADS-B alone to make flight decisions is exposed to a wide range of threats [9]–[12], such as ADS-B data injection, spoofing, modification, jamming and so on. Attackers can include false distance and velocity information in ADS-B packets to forcefully make other UAVs to change their flight trajectories for the sake of collision avoidance.

Researchers have studied broadcast authentication schemes that ensure the integrity of identity and data of the ADS-B packets [13], [14]. Nevertheless, attackers with compromised credentials and dishonest UAVs can still exploit ADS-B with incorrect information. It is more critical and challenging to verify the correctness of the data in ADS-B packets so that UAVs can take correct collision avoidance maneuvers. To address this unresolved security issue, we introduce distance-bounding to ADS-B in this work to enable the verification of the distance and velocity information carried in ADS-B packets as well as the detection of attacking UAVs.

A typical distance-bounding protocol is carried out interactively between a verifier and a prover. The verifier sends challenges to the prover and measures the time between the challenges and the responses from the prover to determine if the prover exceeds a distance threshold. A distance-bounding protocol is reliable when the distance between the verifier and the prover is small and both the verifier and the prover are stationary. But when using distance-bounding on UAVs, we must consider many more factors, such as large distances, fast-moving UAVs, GPS errors, processing time variations, and potentially lost packets.

To address these challenges, we develop a new distance-bounding scheme as a supplemental component to ADS-B. The new distance-bounding scheme utilizes the data of ADS-B and multiple points along with the flight path of UAVs to verify the correctness of the ADS-B data. We demonstrate that the new scheme can reliably detect and filter fraudulent ADS-B data in the vicinity and allow UAVs to make maneuvering decisions regarding collision avoidance with confidence. To the best of our knowledge, our work is the first to include distance-bounding in ADS-B for UAVs in a practical and effective way. Other contributions of this work include (i) a full analysis on the impacts of a variety of procedural noises in distance-bounding and ADS-B, and (ii) a new simulation component [15] contributed to the ArduPilot SITL (Software in the Loop) simulator [16].

The rest of the paper includes the following sections. We first provide the background of ADS-B and distance-bounding in Section II. Then, we present the design of the new distance-bounding protocol in Section III. We analyze the impacts of various procedural errors on distance-bounding in Section IV. We present our implementation, simulation and evaluation in Section V. We summarize the related works on the security of ADS-B and distance-bounding in Section VI. Finally, we conclude the paper in Section VII.

2155-6814/19/$31.00 ©2019 IEEE
DOI 10.1109/MASS.2019.00026

Bit	1 - 5	6 - 8	9 - 32	33 - 88	89 - 112
Field	DF Downlink Format	CA Capabilities	AA ICAO Address	ME ADS-B Message Field	PI Parity

Fig. 1. ADS-B Message Structure: Downlink Format

Bit	33 - 37	38 - 39	40	47 - 82	53	54	55 - 71	72 - 88
Field	Type Code	Surv Status	NIC Supp-B	Altitude	Time	CPR FMT	Latitude	Longitude

Fig. 2. ADS-B Aircraft Position Message Structure

II. Background

A. ADS-B

ADS-B is composed of two capabilities, ADS-B Out (transmitter) and ADS-B In (receiver). With ADS-B Out, an aircraft periodically broadcasts information regarding itself in the 1090 Mhz or 978 Mhz frequency bands. The broadcast information includes the identification, current position, and velocity of the aircraft. The broadcast information can be used by both other aircraft and ground stations. ADS-B In facilitates reception and demodulation of nearby ADS-B Out broadcasts. ADS-B In has no current mandates announced, but it is an integral part of the collision avoidance functionality desired in manned aircraft and UAVs.

Adaptations of these devices designed for small UAVs are already on the market [7], making ADS-B a viable solution. The information gathered from nearby broadcasts can be used to dynamically adjust UAVs' flight paths even when operating in the way-point mode without ground-station communication. This is a highly desired feature to address the rising concerns on the UAV-related airspace safety.

All ADS-B messages use the Downlink Format-17/18 (DF17/18) as shown in Figure 1. An ICAO address field (AA) of 24 bits identifies the transponder sending signal, and then 56 bits are used for the data (ME). A unique ICAO address is assigned to each Mode-S transponder of an aircraft. There are variety types of messages, which are identified by the type code representing the first five bits of the ME field. These types range from routine messages, like identification and position, to specialized and circumstantial ones, such as target state and status. This work is particularly interested in aircraft identification message, position message, and airborne velocity message.

The aircraft identification message gives information about the identity, size and type of aircraft. This message is sent on average once every five seconds while airborne. This information is never cleared out despite no updates from navigation, and this message never terminates its broadcast.

The position message (as shown in Figure 2) contains altitude and latitude/longitude encoded value. This encoding requires an odd and even frame sequence that is indicated by a flag bit in the message. This message is sent on average twice every second while airborne. All bits except for altitude and surveillance status are cleared after two seconds with no update from the navigation system.

Bit	33 - 37	38 - 40	41	42	43 - 45	46	47 - 56	57	58 - 67	68 - 78	79 - 80	81 - 88
Field	Type Code	Sub type	ICF	Reserved	NACv	E/W Direction	E/W Velocity	N/S Direction	N/S Velocity	Vert Rate	Reserved	Diff Bara Alt

Bit	33 - 37	38 - 40	41	42	43 - 45	46	47 - 56	57	58 - 67	68 - 78	79 - 80	81 - 88
Field	Type Code	Sub type	ICF	Reserved	NACv	Heading Status	Heading	Airspeed Type	Airspeed	Vert Rate	Reserved	Diff Bara Alt

Fig. 3. ADS-B Aircraft Velocity Message Structures

The velocity message (as shown in Figure 3) has different subtypes and provides either cardinal direction velocity or heading and airspeed. Both include a vertical rate as well as a navigation accuracy estimate. This message is sent on average twice every second while airborne.

B. Distance-Bounding

Distance-bounding is the process of timing interaction between two devices across a medium, a verifying device known as *Verifier* and a proving device known as *Prover*. Determining the distance between them is based on the speed of the signal propagation across the medium plus the processing time of the prover. The implementation [17] of distance-bounding relies on the fact that radio wave travels near the speed of light. The distance of the two devices is thus computed from the round trip time (RTT) as $d = \frac{c}{2}(t_{rtt} - t_p)$, where c is the speed of light, t_{rtt} is the total RTT, and t_p is the processing time the prover takes to calculate the response. The distance estimate calculated here is most effective when t_p and d are consistent, because the estimated error is predictable.

A variety of distance-bounding protocols have been proposed in the literature [18]–[20]. They generally include a fast process of exchanging challenge and response bits between the verifier and the prover. Then, the verifier will check that the responses are correct and do not exceed T_{max}, a determined value for the maximum distance the verifier wants to allow. Distance-bounding is common in a wide variety of systems, such as credit card readers and security badge readers.

Distance-bounding protocols are designed to counteract a variety of threats designed to defeat or improve the chances of defeating a distance-bounding process. These threats include distance fraud, mafia fraud [21], terrorist fraud [22], and distance hijacking [23].

(1) *Distance Fraud.* A dishonest prover or an adversary claims to be somewhere in the verifier's vicinity. This is a broad category that can define attacks not fitting of a more specific one. The proceeding types could be considered variations or sub-types of distance fraud.

(2) *Mafia Fraud.* In a Mafia fraud attack [21], an adversary exists between an honest prover and a verifier. The attack closely resembles a man-in-the-middle attack. The adversary tries to make the distance between these two seem shorter than it is in reality. An example of this threat, an adversary uses an RF reader to pick up your badge signal and send the communications to another location where a transmitter is

being placed near the verifier. In distance-bounding, the time it takes for the adversary to send these transmissions back and forth would exceed T_{max}. The probability of success varies on the medium and distance-bounding protocol [24].

(3) *Terrorist Fraud.* An adversary uses a dishonest prover to conduct the attack, but it must be in such a way that it does not give any assistance in future attacks. Terrorist Fraud attacks [22] are considered thwarted, when assisting an adversary would reveal the dishonest prover's long term secret or key. If the dishonest prover can assist an adversary without revealing any damaging or long term secrets, then the protocol is considered vulnerable.

(4) *Distance Hijacking.* Distance Hijacking is a recently discovered attack procedure where a dishonest prover uses an honest prover by hijacking their verification phases [23]. In this attack, an attacker will take advantage of an honest prover without their assistance or consent. This often means hijacking the interaction between an honest prover and verifier during the bit exchange phase.

III. Protocol Design

A. Challenges

Distance-bounding is used in many current systems today such as credit card verification schemes, badge protected building access, and others. These systems involve a stationary verifying device and a nearby stationary prover. They interact when held very close to each other, sometimes even in contact with another. Directly applying such distance-bounding was not suited for the UAV ADS-B verification process.

In airspace, distance-bounding needs to occur at a significantly larger distance than traditional distance-bounding operations, leaving the UAVs to take appropriate actions in the scenario where a collision is possible. Mobility is another concern when porting distance-bounding to this type of problem. We consider that the prover is a UAV while the verifier could be a ground station, a UAV or a manned aircraft. Furthermore, various noises present in the process for UAVs, such as location data noise, time measurement noise, response processing noise and so on.

With all these challenges to overcome, the standard method of distance-bounding is incapable of providing an accurate way to filter erroneous or fraudulent ADS-B messages. The protocol would be susceptible to high numbers of false positive fraud detection, because the propagation medium is noisy resulting in lost challenges and a fluctuating processing time combined with imprecise navigation information. The allowable limit of failed challenges would either be too high to filter out bad messages or too low to allow realistic message variation [25]. If an attacker was to create false ADS-B messages, they could use the allowable variance in the processing time and fault tolerance to deceive a verifier. In order to deal with this, the protocol needs to be able to mitigate both failed challenges and processing time noise.

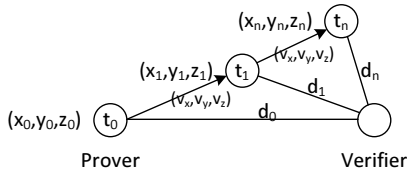


Fig. 4. Mutli-point Distance-bounding

B. Threat Model

In this work, we consider the collision threat where an attacker exploits fraudulent ADS-B messages to make itself appear as approaching a target aircraft in a collision trajectory although the attacker is actually far away from the target. The collision threat intends to force the target aircraft to take collision avoidance maneuvers so that the target aircraft changes its flight path. This kind of collision threat can result in overwhelming traffic in the target airspace [9], [26].

To send such fraudulent ADS-B messages, the attacker may have the credentials necessary to authenticate itself, or the attacker is simply a dishonest aircraft. Additionally, the attacker knows distance-bounding will be used to verify the location data in the fraudulent ADS-B messages. The attacker (as the prover) will take the fraud operations (alone or with helpers) as discussed in Section II-B to counter the distance-bounding process.

The goal of our protocol is to enable the target aircraft to verify if a UAV is actually on its flight path to approach the target and detect if the approaching UAV is malicious.

C. Multi-point Distance-Bounding

To address the aforementioned threat, we propose a multi-point distance-bounding protocol for mobile UAVs as depicted in Figure 4. The protocol makes use of multiple positions of a flying UAV (prover) to verify the truthfulness of the flight trajectory of the UAV. The prover broadcasts ADS-B messages that include its position and velocity. The verifier (another UAV, a ground control station, or a maned aircraft) then chooses a few random time points and predicts the positions of the prover on these time points. The verifier and the prover perform a session of multiple rounds of distance-bounding. Each round is executed by exchanging multiple challenge-response bits over one of the predicted positions of the prover. Since the verifier could be stationary or moving, we consider the following two models for the verifier.

1) Stationary Verifier Model: We first consider the stationary verifier model, where the verifier is fixed at the origin of a coordinate system. The verifier could be a ground station or a hovering aircraft.

In a multi-point distance-bounding session, the verifier chooses a few random time points $t_i,\ 0 \leq i \leq n$, where the session interval $(t_n - t_0)$ will be set to the ADS-B broadcast period. Let $\check{\boldsymbol{p}}_{\boldsymbol{0}}$ and $\check{\boldsymbol{v}}$ be the prover's position and velocity in the ADS-B broadcast at t_0. The verifier will consider $\check{\boldsymbol{v}}$ a constant vector during the multi-point distance-bounding

session, as $\check{\boldsymbol{v}}$ will only be updated on the next ADS-B broadcast.

Along with the multiple time points, the prover's position $\check{\boldsymbol{p}}_{\boldsymbol{i}}$ at t_i is in Eq(1).

$$\check{\boldsymbol{p}}_{\boldsymbol{i}} = \check{\boldsymbol{p}}_{\mathbf{0}} + \check{\boldsymbol{v}}(t_i - t_0) \tag{1}$$

Therefore, the distance of the prover and the verifier at t_i is $\check{d}_i = |\check{\boldsymbol{p}}_{\boldsymbol{i}}| = |\check{\boldsymbol{p}}_{\mathbf{0}} + \check{\boldsymbol{v}}(t_i - t_0)|$ as calculated in Eq(2).

$$\begin{aligned} \check{d}_i &= \sqrt{\check{\boldsymbol{p}}_{\boldsymbol{i}}'\check{\boldsymbol{p}}_{\boldsymbol{i}}} = \sqrt{(\check{\boldsymbol{p}}_{\mathbf{0}} + \check{\boldsymbol{v}}(t_i - t_0))'(\check{\boldsymbol{p}}_{\mathbf{0}} + \check{\boldsymbol{v}}(t_i - t_0))} \\ &= \sqrt{\check{d}_0^2 + 2\check{\boldsymbol{v}}'\check{\boldsymbol{p}}_{\mathbf{0}}(t_i - t_0) + \check{v}^2(t_i - t_0)^2} \end{aligned} \tag{2}$$

Without any noise in distance-bounding, the round-trip time delay between prover and verifier is in Eq(3), where t_p is the processing time of the prover to produce a response bit and c is the speed of light.

$$\begin{aligned} \check{\tau}_i &= \frac{2}{c}\check{d}_i + \check{t_p} \\ &= \frac{2}{c}\sqrt{\check{d}_0^2 + 2\check{\boldsymbol{v}}'\check{\boldsymbol{p}}_{\mathbf{0}}(t_i - t_0) + \check{v}^2(t_i - t_0)^2} + \check{t_p} \end{aligned} \tag{3}$$

2) Moving Verifier Model: When the verifier is a UAV or a maned aircraft, the verifier is flying too. We need to consider distance-bounding for a moving verifier

Let $\check{\boldsymbol{v}}_{\boldsymbol{v}}$ be verifier's velocity, and $\check{\boldsymbol{v}}_{\boldsymbol{p}}$ be prover's velocity. Same as in the stationary verifier model, we consider both $\check{\boldsymbol{v}}_{\boldsymbol{v}}$ and $\check{\boldsymbol{v}}_{\boldsymbol{p}}$ constant velocities until they are updated on the next ADS-B messages.

Along with the multiple time points, the verifier's position at t_i is $\check{\boldsymbol{p}}_{\boldsymbol{vi}} = \check{\boldsymbol{p}}_{\boldsymbol{v0}} + \check{\boldsymbol{v}}_{\boldsymbol{v}}(t_i - t_0)$ and the prover's position at t_i is $\check{\boldsymbol{p}}_{\boldsymbol{pi}} = \check{\boldsymbol{p}}_{\boldsymbol{p0}} + \check{\boldsymbol{v}}_{\boldsymbol{p}}(t_i - t_0)$. Hence, the distance between the prover and the verifier at t_i is $\check{d}_i = |\check{\boldsymbol{p}}_{\boldsymbol{pi}} - \check{\boldsymbol{p}}_{\boldsymbol{vi}}| = |(\check{\boldsymbol{p}}_{\boldsymbol{p0}} - \check{\boldsymbol{p}}_{\boldsymbol{v0}}) + (\check{\boldsymbol{v}}_{\boldsymbol{v}} - \check{\boldsymbol{v}}_{\boldsymbol{p}})(t_i - t_0)|$.

Let $\check{\boldsymbol{p}}_{\mathbf{0}} = \check{\boldsymbol{p}}_{\boldsymbol{p0}} - \check{\boldsymbol{p}}_{\boldsymbol{v0}}$ and $\check{\boldsymbol{v}} = \check{\boldsymbol{v}}_{\boldsymbol{v}} - \check{\boldsymbol{v}}_{\boldsymbol{p}}$. Then, $\check{d}_i = |\check{\boldsymbol{p}}_{\mathbf{0}} + \check{\boldsymbol{v}}(t_i - t_0)|$. The round-trip time delay between the prover and the verifier is $\check{\tau}_i = \frac{2}{c}\check{d}_i + \check{t_p}$. Therefore, the moving verifier model can be reduced to the stationary verifier model in a relative coordinate system with the verifier as the referencing point. In the rest of the paper, we will use the stationary model to present our work.

D. Full Protocol

Figure 5 illustrates our full multi-point distance-bounding protocol. The protocol consists of five steps: authentication, setup, bit exchange, verification and detection. Our work is focused on the last three steps that allow the UAVs to detect and filter fraudulent ADS-B messages in real time while adding minimum communication and computational overhead. Meanwhile, for the first two steps, we adopt the methods from existing distance-bounding protocols (for example, the Hancke and Kuhn distance-bounding protocol [27] in this paper).

1) Authentication: The authentication step is to mutually authenticate the verifier and the prover and then establish a secret symmetric key between them to use in the setup phase. This needs to occur before the distance-bounding can begin.

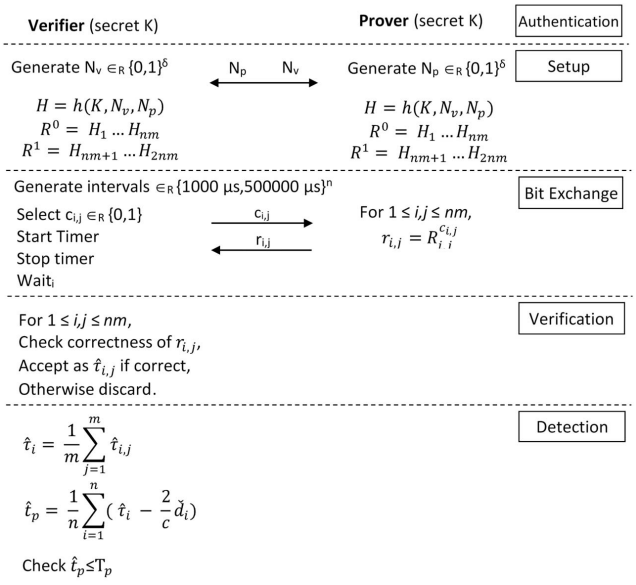


Fig. 5. Mutli-point Distance-bounding Protocol

Following the Hancke and Kuhn protocol, an authenticated Diffie-Hellman exchange is performed, and the security of the key is not compromised during this exchange.

2) Setup: The setup step is used to exchange nonces N_v and N_p that will be used to generate challenge and response bits. Before setting up the protocol, several parameters should be agreed upon by the verifier and prover: (a) the number of rounds, n, and the number of challenge bits per round, m; (b) the size of nonces, N_v and N_p; and (c) the hash function. Then, each party sets up two bit sequences of equal length to the total number of challenge bits, $n \times m$.

3) Bit Exchange: In this step, the verifier and the prover exchange challenge and response bits over multiple rounds. Each round is corresponding to the verification of the prover's position $\boldsymbol{p}_{\boldsymbol{i}}$ at t_i as discussed in Section III-C1. First, the verifier randomly generates n time points. Then, at each time point t_i, the verifier and the prover perform a round of fast m bits exchange. The verifier sends a challenge bit $c_{i,j}$ to the prover. The prover responds immediately with a response bit $r_{i,j}$, from the corresponding bit sequence determined by $c_{i,j}$. This exchange between them occurs m times (thus m bits) in each round. After all n rounds are completed, the verifier will verify the responses.

4) Verification: As in a typical distance-bounding protocol, the verifier generates the correct bits and compares them with the received prover's responses. There may be a small number of failed or lost response bits due to interference. Existing distance-bounding protocols usually have a threshold of allowable failed response bits to tolerate fault. Our work uses the successfully received response bits that pass the threshold.

Next, our protocol estimates the prover's processing time and distance from the time delays of the response bits. Because

our protocol performs one round of distance-bounding at t_i, the verifier calculates the average time delay $\hat{\tau}_i$ at t_i, where $\hat{\tau}_{i,j}$ is the time delay of the j-th response bit of the i-th round in a distance-bound session.

$$\hat{\tau}_i = \frac{1}{m}\sum_{j=1}^{m}\hat{\tau}_{i,j} \tag{4}$$

As in Eq(3), the verifier predicts $\check{\boldsymbol{p}}_{\boldsymbol{i}}$ and thus $\check{d}_i$ of the prover at t_i. Hence, the prover's processing time at t_i is estimated as $\hat{t_{p_i}} = \hat{\tau}_i - \frac{2}{c}\check{d}_i$. The average of the prover's processing time is $\hat{t_p}$ over all rounds of a distance-bound session.

$$\hat{t_p} = \frac{1}{n}\sum_{i=1}^{n}\hat{t_{p_i}} = \frac{1}{n}\sum_{i=1}^{n}(\hat{\tau}_i - \frac{2}{c}\check{d}_i) \tag{5}$$

Then, the verifier estimates the distance to the prover at t_i as

$$\hat{d}_i = \frac{c}{2}(\hat{\tau}_i - \hat{t_p}) \tag{6}$$

5) Attack Detection: When estimating $\hat{t_p}$ according to Eq(5), the verifier uses $\check{d}_i$ derived from the ADS-B message as in Eq(2). If the prover is honest, $\check{d}_i$ will be close to the prover's true position. Thereby, $\hat{t_p}$ will be fairly small, because the prover is supposed to instantly respond. We will provide an in-depth analysis of how $\hat{t_p}$ is influenced by noises in Section IV-A.

In contrast, if the prover is dishonest or malicious, the prover lies on $\check{\boldsymbol{p}}_{\boldsymbol{i}}$ so that $\check{\boldsymbol{p}}_{\boldsymbol{i}}$ appears close to the verifier. Consequently, $\check{d}_i$ will be small but $\check{\boldsymbol{p}}_{\boldsymbol{i}}$ is actually far away from the prover's true position. Hence, a large discrepancy will occur between $\hat{\tau}_i$ and $\frac{2}{c}\check{d}_i$, and result in a large $\hat{t_p}$.

We exploit this obvious anomaly of $\hat{t_p}$ to detect if the prover is dishonest or malicious with a threshold T_p. If $\hat{t_p} \leq T_p$, the prover is honest. Otherwise, the prover is dishonest or malicious. We set T_p as in Eq(7) with honest and bounded $\hat{t_p}$, where $E(\hat{t_p})$ is the mean of honest $\hat{t_p}$ and $\sigma^*_{\hat{t_p}}$ is the standard deviation of $\hat{t_p}$ with the worst noise. A more detailed analysis will be presented in Section IV-B.

$$T_p = E(\hat{t_p}) + 5\sigma^*_{\hat{t_p}} \tag{7}$$

IV. Distance-bounding Analysis

In practice, the time measurement of multi-point distance-bounding is affected by noise in location data and processing hardware. It can also be manipulated with fraudulent location data. In the following, we analyze the accuracy and resilience of our multi-point distance-bounding protocol.

A. Noise Analysis

1) Noise Model: We consider three types of noises in ADS-B and distance-bounding in Eqs(8-10), where $\boldsymbol{p_0}$, $\boldsymbol{v}$ and t_p are the true position, velocity and processing time of the prover. The noises are the location noise $\boldsymbol{\epsilon_p}$ and the velocity noise $\boldsymbol{\epsilon_v}$ in the ADS-B data and the prover's processing time noise ϵ_t. Because the three noises are from different sources, they are considered independent of each other.

$$\check{\boldsymbol{p}}_{\boldsymbol{0}} = \boldsymbol{p_0} + \boldsymbol{\epsilon_p},\ \boldsymbol{\epsilon_p} \sim N(0, \boldsymbol{\sigma_p}) \tag{8}$$

$$\check{\boldsymbol{v}} = \boldsymbol{v} + \boldsymbol{\epsilon_v},\ \boldsymbol{\epsilon_v} \sim N(0, \boldsymbol{\sigma_v}) \tag{9}$$

$$\check{t_p} = t_p + \epsilon_{t_p},\ \epsilon_{t_p} \sim U(0, \rho) \tag{10}$$

$\boldsymbol{\epsilon_p}$ and $\boldsymbol{\epsilon_v}$ reflect the inaccuracy of GPS data. We assume $\boldsymbol{\epsilon_p}$ and $\boldsymbol{\epsilon_v}$ follow a normal distribution. In distance-bounding, the computation in the prover is to produce the response bits. Therefore, we can assume the prover's processing time t_p is bounded within a range $[t_p, t_p + \rho]$ in most distance-bounding devices. ϵ_{t_p} then represents the noise of the processing time uniformly distributed in the range $[0, \rho]$.

Accordingly, the position of the prover in the ADS-B message at t_i is

$$\check{\boldsymbol{p}}_i = \boldsymbol{p}_0 + \boldsymbol{\epsilon_p} + (\boldsymbol{v} + \boldsymbol{\epsilon_v})(t_i - t_0) \tag{11}$$

The distance between the prover and the verifier at t_i is

$$\begin{aligned}\check{d}_i^2 &= |\boldsymbol{v} + \boldsymbol{\epsilon_v}|^2(t_i - t_0)^2 + |\boldsymbol{p}_0 + \boldsymbol{\epsilon_p}|^2 \\ &\quad +2(\boldsymbol{v} + \boldsymbol{\epsilon_v})'(\boldsymbol{p}_0 + \boldsymbol{\epsilon_p})(t_i - t_0)\end{aligned} \tag{12}$$

Consider that $\boldsymbol{\epsilon_p}$ and $\boldsymbol{\epsilon_v}$ are relatively small, we can derive the first order Taylor approximation of $\check{d}_i$ as in Eq(13).

$$\begin{aligned}\check{d}_i &= d_i + \epsilon_{d_i} = d_i + \tfrac{\boldsymbol{p}_i\boldsymbol{\epsilon}_i'}{d_i} \\ &= d_i + \tfrac{1}{d_i}(\boldsymbol{p_0} + \boldsymbol{v}(t_i - t_0)(\boldsymbol{\epsilon_p} + \boldsymbol{\epsilon_v}(t_i - t_0))'\end{aligned} \tag{13}$$

2) Error Estimation on $\hat{t_p}$ and d_i: According to Eq(4), $\hat{\tau}_i$ is the average of $\hat{\tau}_{i,j}$ of all m bits in the i-th round of distance bounding. $\hat{\tau}_{i,j}$ is the sum of the round trip time over the true distance and the prover's processing time as in Eq(14), where $\tau_i = \frac{2}{c}d_i$.

$$\hat{\tau}_{i,j} = \frac{2}{c}d_i + \check{t_{p_{i,j}}} = \tau_i + t_p + \epsilon_{t_p\,i,j} \tag{14}$$

Following Eq(4) and Eq(14), we have $\hat{\tau}_i = \tau_i + t_p + \xi(\hat{\tau}_i)$, where $\xi(\hat{\tau}_i)$ is the noise component of $\hat{\tau}_i$.

$$\xi(\hat{\tau}_i) = \frac{1}{m}\sum_{j=1}^{m}\epsilon_{t_p\,i,j} \tag{15}$$

Following Eq(5) and Eq(13), we have $\hat{t_p} = \frac{1}{n}\sum_{i=1}^{n}(\hat{\tau}_i - \frac{2}{c}(d_i + \epsilon_{d_i})) = t_p + \xi(\hat{t_p})$, where $\xi(\hat{t_p})$ is the noise component of $\hat{t_p}$.

$$\xi(\hat{t_p}) = \frac{1}{n}\sum_{i=1}^{n}(\xi(\hat{\tau}_i) - \frac{2}{c}\xi(\check{d}_i)) = \frac{1}{n}\sum_{i=1}^{n}(\xi(\hat{\tau}_i) - \frac{2\boldsymbol{p}_i\boldsymbol{\epsilon}_i'}{cd_i}) \tag{16}$$

Therefore, the mean and the standard deviation of $\xi(\hat{t_p})$ are

$$E(\xi(\hat{t_p})) = \frac{1}{n}\sum_{i=1}^{n}(\frac{\rho}{2} + 0) = \frac{\rho}{2} \tag{17}$$

$$\sigma^2(\xi(\hat{t_p})) = \frac{\rho^2}{12nm} + \frac{4}{n^2c^2}\sum_{i=1}^{n}(\sigma_p^2 + \sigma_v^2(t_i - t_0)^2) \quad (18)$$

Similarly, following Eq(6), we have $\hat{d_i} = \frac{c}{2}(\frac{1}{m}\sum_{j=1}^{m}\hat{\tau}_{i,j} - \frac{1}{n}\sum_{k=1}^{n}(\hat{\tau}_k - \frac{2}{c}\check{d}_k)) = d_i + \xi(\hat{d_i})$. The noise component $\xi(\hat{d_i})$ is

$$\xi(\hat{d_i}) = \frac{c}{2}(\frac{n-1}{nm}\sum_{j=1}^{m}\epsilon_{t_{p\,i,j}} - \frac{1}{nm}\sum_{k=1,k\neq i}^{n}\sum_{j=1}^{m}\epsilon_{t_{p\,k,j}} - \frac{1}{n}\sum_{k=1}^{n}\frac{2}{c}\epsilon_{d_k}) \quad (19)$$

The mean and the standard deviation of $\xi(\hat{d_i})$ are

$$E(\xi(\hat{d_i})) = 0 \quad (20)$$

$$\sigma^2(\hat{d_i}) = \frac{c^2(n-1)\rho^2}{48nm} + \frac{1}{n^2}\sum_{k=1}^{n}(\sigma_p^2 + \sigma_v^2(t_k - t_0)^2) \quad (21)$$

B. Attack Analysis

As discussed in the threat model in Section III-B, the attacker broadcasts a fraudulent location close to the target to make the target change its flight path. The attack can be modeled as in Eq(22), where Δ is the collision avoidance range. Comparing Eq(22) and Eq(13), the noise component ϵ_{d_i} is replaced by the attack component $-\delta_{d_i}$.

$$\check{d_i} = d_i - \delta_{d_i} < \Delta \quad (22)$$

Following Eq(16), the $\hat{t_p}$ under attack has an error as

$$\xi'(\hat{t_p}) = \frac{1}{n}\sum_{i=1}^{n}(\xi(\hat{\tau}_i) - \frac{2}{c}\xi(\check{d_i})) = \frac{1}{n}\sum_{i=1}^{n}(\xi(\hat{\tau}_i) + \frac{2}{c}\delta_{d_i}) \quad (23)$$

Therefore, the mean of $\xi'(\hat{t_p})$ under attack is

$$E(\xi'(\hat{t_p})) = \frac{1}{n}\sum_{i=1}^{n}(\frac{\rho}{2} + \frac{2}{c}\delta_{d_i}) = \frac{\rho}{2} + \frac{2}{c}E(\delta_{d_i}) \quad (24)$$

If the attacker is truly far away from the target, then $\delta_{d_i} > d_i - \Delta \gg \Delta$, Comparing Eq(17), Eq(22) and Eq(24), we can find that $E(\xi'(\hat{t_p})) > \frac{\rho}{2} + \frac{c}{2}(E(d_i) - \Delta) \gg E(\xi(\hat{t_p}))$. This indicates that the measured t_p under attack is much larger than the measured good t_p with noise components. Hence, we can set a threshold T_p as in Eq(7) for attack detection. In practice, the error bound and processing time bound would be predetermined as a specification required on the equipment.

V. Evaluation

We implemented the multi-point distance-bounding protocol in the ArduPilot software-in-the-loop (SITL) simulator [16]. SILT is widely used to facilitate UAV testing before actually flying with a hardware flight controller. We evaluated the accuracy of measuring distance and the quality of attack detection in SITL.

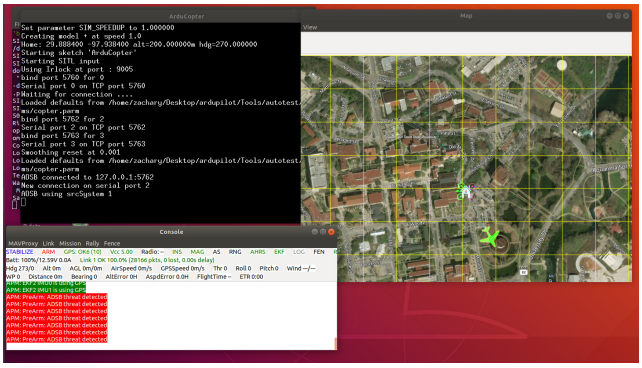


Fig. 6. Snapshot of Simulation in SITL

A. Implementation

We developed modules in SITL to create ghost aircraft, produce ADS-B broadcasts, perform distance bounding operations and log flight data during simulations. The added modules are available at [15]. A snapshot of one flight in simulation is shown in Figure 6.

The verifier is a simulated quad-copter centered in the aerial map. A ghost aircraft is created in simulation to act as the prover. The ghost aircraft broadcasts legitimate or fraudulent ADS-B messages as MAVLink packets periodically. Upon receiving ADS-B messages, the verifier records the prover's position and velocity, and then initiates the multi-point distance bounding protocol to verify the distance with the prover and detect if the prover is a threat.

To control the flight path of the ghost plane, we developed a flight configuration module to parse configuration files and obtain a variety of flight parameters. Then, SITL can set the ghost aircraft on the specified flight path according to the parameters in simulation. SITL itself provides some logging functions to record flight traces. In addition to that, we added a module that logs ADS-B and distance-bounding related flight data for more in-depth analysis.

B. Experiment Setting

As discussed in Section III-C, a moving verifier can be modeled as a stationary verifier in a relative coordinate system. Hence, in simulations, we fix the location of the verifier and only set parameters to the prover. An overview of the parameters and their values are listed in Table I.

The distance between the verifier and the prover needs to be significant enough for the verifier to make decisions based on the information received. Meanwhile, UAVs have significantly less flying range than standard or commercial aircraft. Hence, the experiments choose the range of distance up to 500 meters centering around the verifier. The prover's flight paths are set within this range.

Velocity is a unique parameter in our work that could have a significant impact on the success of distance bounding. The experiment should test for the spectrum of UAV operating speeds. The FAA dictates no UAV should exceed 100 mph

TABLE I
THE PROVER'S PARAMETERS

Parameter	Value
Distance d (m)	0 to 500
Position noise σ_p (m)	0, 5
Velocity v (m/s)	5, 10, 20, 30, 40
v noise σ_p (m/s)	0, 3
Processing time t_p (ns)	1, 10, 100, 1000
t_p's noise ρ (ns)	0.05, 0.5, 5, 50
Round interval (ms)	50, 100, 250, 500
Number of rounds	3
Challenges per round	16

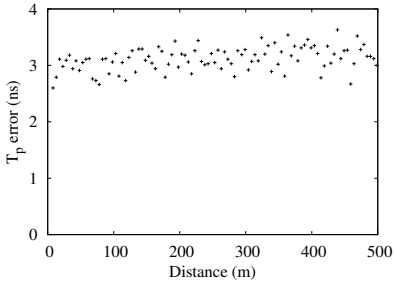


Fig. 7. T_p Error over Distance

(87 knots) [28], which is approximately 45 m/s. Using that information, velocity was given an upper bound of 40 m/s and a lower bound of 5 m/s. In practice, both position and velocity are provided by the GPS sensor. We set noises of velocity and position based on observations from our earlier field tests [29].

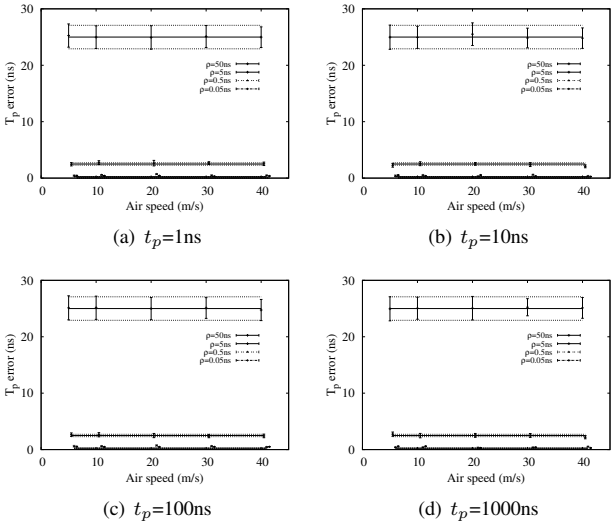
Processing time of the prover is crucial to distance-bounding in both distance verification and attack detection. In [30], the processing time at the prover is less than 1 ns and its variation is less than 0.07 ns. Hence, we choose the processing time at higher values to study how our protocol performs under worse conditions.

Round interval is another unique distance-bounding parameter in this work. The round interval is a randomly determined pause between rounds in a distance-bounding session. The randomness will provide additional security, as the prover does not know the round interval between rounds. In addition, the change in distance during one distance-bounding session will largely depend upon the round interval and the UAV speed.

The combination of the simulation parameters in Table I produces 640 flight path configurations. Along with each flight path, the prover is set on a collision trajectory towards the verifier. In all experiments, upon receiving each prover's ADS-B broadcast, the verifier performs 3 rounds of distance bounding and exchanges 16 challenge-response bits per round with the prover.

C. Experiment Results

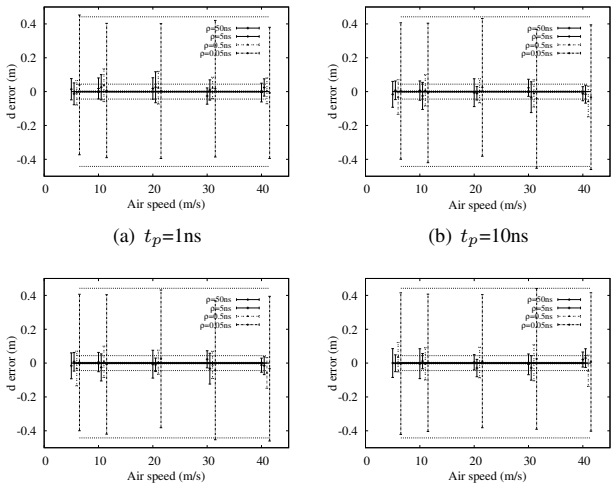
In all experiments, the verifier measures the prover's processing time and distance following the multi-point distance-bounding protocol. We then compare the results with the estimated processing times and distances according to the analysis in Section IV. In the following, we show and discuss the results with legitimate and fraudulent ADS-B broadcasts and the ability to detect fraudulent UAVs.



(a) t_p=1ns (b) t_p=10ns

(c) t_p=100ns (d) t_p=1000ns

Fig. 8. Errors in Measured Processing Time with Legitimate ADS-B



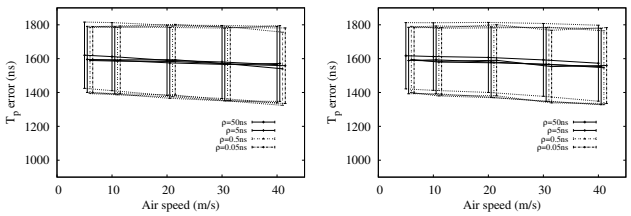
(a) t_p=1ns (b) t_p=10ns

(c) t_p=100ns (d) t_p=1000ns

Fig. 9. Errors in Measured Distance with Legitimate ADS-B

1) Distance-bounding with Legitimate ADS-B: With legitimate ADS-B broadcasts, the goal of our experiments is to analyze the key factors that determine the accuracy of distance measurement in distance-bounding. As discussed in Section III-D, our protocol can measure both the prover's processing time $\hat{t_p}$ and the distance $\hat{d}$. Because various noises exist in the process, the measured results contain errors $\xi(\hat{t_p})$ and $\xi(\hat{d})$ to the actual values.

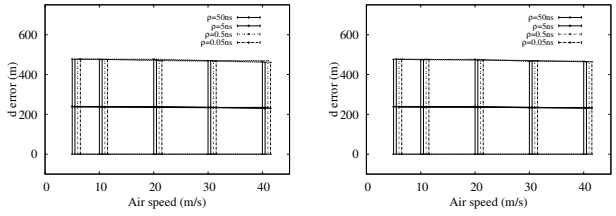
First, we examine if distance has an impact on the measurement. Figure 7 shows the errors in the measured processing time on various locations along with one flight path. The figure indicates the measurement errors are not related to the



(a) t_p=1ns (b) t_p=10ns

(c) t_p=100ns (d) t_p=1000ns

Fig. 10. Processing Time T_p Error with Fraudulent ADS-B



(a) t_p=1ns (b) t_p=10ns

(c) t_p=100ns (d) t_p=1000ns

Fig. 11. Distance Error with Fraudulent ADS-B

distance. This confirms our analysis in Eq(10). Therefore, in our following analysis, we average the errors in the measured results along with each flight path.

Figure 8 shows the means (dots) and standard deviations (error bars) of the measured processing time errors $\xi(\hat{t_p})$, i.e. the difference between the measured processing time and the actual processing time. Each sub figure is corresponding to an actual processing time t_p of the prover. We also add the means (solid lines) and standard deviations (dashed lines) of the processing time errors that are estimated according to Eq(17) and Eq(18). The figure shows that the processing time errors are not affected by the prover's speed, distance or processing time. Rather, the noises in processing time, location and velocity determine the measured processing time errors. Greater noise results in a larger error in measurement. This result well confirms our noise analysis.

TABLE II
FRAUDULENT FLIGHT DETECTION

	Sessions	Fraud	False Rate
Legitimate	33022	0	0.0% (False Positive)
Attack	33187	32883	0.92% (False Negative)

In addition, the processing time noise ρ we added in the experiments goes up to 50 ns. Nevertheless, the standard deviation of the measured processing time error is only up to 2 ns, because the measurement is averaged over multiple challenges and multiple rounds. Hence, the measurement errors are far smaller than the noises. Our protocol can reach a satisfying accuracy in processing time measurement even in a very noisy environment.

Figure 9 shows the means (dots) and standard deviations (error bars) of the measured distance errors $\xi(\hat{d})$, i.e. the difference between the measured distance and the actual distance, as well as the means and standard deviations of the estimated distance errors. Similar to the results in Figure 8, only the noise factors determine the measured distance errors. When the processing time noise ρ is at the largest value 50 ns in the experiments (equivalent to 15 meters of location noise), the standard deviation of the measured distance error is only 1 meter, which is sufficiently accurate for UAV collision detection.

2) Distance-bounding with Fraudulent ADS-B: In attack cases, the prover claims to be at a location within 10 meters to the verifier in its ADS-B messages. Figure 10 and Figure 11 show the measured processing time errors and distance errors with fraudulent ADS-B broadcast.

As analyzed in Section IV-B, the processing time $\xi'(\hat{t_p})$ under attack is significantly larger than the actual processing time. Comparing Figure 10 and Figure 8, we observe that the fraudulent processing time error is always above the upper bound of the legitimate processing time error. The deviation of processing time is much larger for the attack data too. This is directly related to the difference between the real and claimed positions.

3) Attack Detection: Finally, we study how well the verifier can detect fraudulent and legitimate ADS-B messages. Because the worst standard deviation of $\hat{t_p}$ is about 4 ns as shown in Figure 9, we set $5\sigma^*_{\hat{t_p}} = 20ns$ in Eq(7). Table II shows the results of attack detection. The table includes the number of distance-bounding sessions, the number of sessions that are detected as fraudulent, and the corresponding false positive or negative rates.

The false positive rate is 0, i.e. no legitimate messages are flagged as suspicious. Meanwhile, just 0.92% of fraudulent messages were not detected by the protocol and thus the false negative rate is 0.92%. We further analyzed the false negative cases. They all appear when the prover is already within 20 meters distance to the verifier. Therefore, the prover is factually close to the verifier when the protocol cannot detect the false negative cases.

VI. Related Works

Security of the ADS-B proposal was well reviewed in [10] where a variety of vulnerability and security implementation were summarized. The attacks include ADS-B message deletion, modification, injection and jamming. The goals of the attacks mainly include creating ghost aircraft and disrupting reception of ADS-B.

To secure the ADS-B protocol, broadcast authentication [13], [14] was proposed to allow any receiving devices to verify the identity of the sending device. But these authentication methods do not verify the location information being transmitted.

As one of the location verification approaches, distance-bounding is suitable to determine the physical distance between two devices by measuring the time between responses. It is used in a variety of real world applications to defeat various distance fraud attacks. A variety of distance-bounding protocols [18]–[20] have developed to defeat certain types of attacks or meet different situational needs.

VII. Conclusion

The ADS-B protocol is mandated for use in manned aircraft and becoming increasingly popular in UAVs. The safety benefits introduced by this system are limited by the lack of security features to defend it. This paper proposes a new multi-point distance-bounding protocol to verify the location data in ADS-B messages. The protocol uses multiple points along with a prover's flight path and enables the verifier to measure the distance with the prover accurately in noisy settings. In our simulations, the protocol can detect 99.1% fraudulent ADS-B messages while having no mis-detection on legitimate ADS-B messages.

VIII. Acknowledgment

This research was partially funded by Texas State University REP program.

References

[1] FAA, "Equip ADS-B," http://www.faa.gov/nextgen/equipadsb.

[2] B. Stark, B. Stevenson, and Y. Chen, "ADS-B for small Unmanned Aerial Systems: Case study and regulatory practices," in *Proc. of International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013, pp. 152–159.

[3] D. B. Sesso, L. F. Vismari, and J. B. Camargo, "An approach to assess the safety of ADS-B based unmanned aerial systems," in *Proc. of International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 669–676.

[4] Y. Lin and S. Saripalli, "Sense and avoid for Unmanned Aerial Vehicles using ADS-BCostinF2012, year=2015, pages=6402-6407,," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*.

[5] C. Caron, "After Drone Hits Plane in Canada, New Fears About Air Safety," https://www.nytimes.com/2017/10/17/world/canada/canada-drone-plane.html, 2017.

[6] A. Pasztor, "Possible Drone Collision Renews Focus on Safety Systems," https://www.wsj.com/articles/possible-drone-collision-renews-focus-on-safety-systems-11544965203, 2018.

[7] uAvionix, "uAvionix Sense and Avoid for Drones and GA," https://uavionix.com/.

[8] Ardupilot, "ADS-B Receiver," http://ardupilot.org/copter/docs/common-ads-b-receiver.html.

[9] A. Costin and A. Francillon, "Ghost in the Air (Traffic) : On insecurity of ADS-B protocol and practical attacks on ADS-B devices," in *BlackHat USA*, 2012.

[10] M. Strohmeier, V. Lenders, and I. Martinovic, "On the Security of the Automatic Dependent Surveillance-Broadcast Protocol," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 1066–1087, 2015.

[11] M. R. Manesh and N. Kaabouch, "Analysis of vulnerabilities, attacks, countermeasures and overall risk of the Automatic Dependent Surveillance-Broadcast (ADS-B) system," *International Journal of Critical Infrastructure Protection*, vol. 19, 2017.

[12] M. R. Manesh, M. Mullins, K. Foerster, and N. Kaabouch, "A preliminary effort toward investigating the impacts of ADS-B message injection attack," in *Proc. of IEEE Aerospace Conference*, 2018, pp. 1–6.

[13] K. D. Wesson, T. E. Humphreys, and B. L. Evans, "Can Cryptography Secure Next Generation Air Traffic Surveillance?" http://users.ece.utexas.edu/ bevans/papers/2015/nextgen/, 2014.

[14] H. Yang, Q. Zhou, M. Yao, R. Lu, H. Li, and X. Zhang, "A Practical and Compatible Cryptographic Solution to ADS-B Security," *IEEE Internet of Things Journal*, 2019.

[15] Z. Languell, "Distance Bounding SITL Simulation," https://github.com/zlanguell/DistanceBounding-SITL-Simulation, 2019.

[16] SITL, "SITL Simulator (Software in the Loop)," http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html, 2014.

[17] A. Abu-Mahfouz and G. P. Hancke, "Distance Bounding: A Practical Security Solution for Real-Time Location Systems," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 16–27, 2013.

[18] J. Munilla and A. Peinado, "Distance Bounding Protocols for RFID Enhanced by Using Void-challenges and Analysis in Noisy Channels," *Wirel. Commun. Mob. Comput.*, vol. 8, no. 9, pp. 1227–1232, 2008.

[19] S. Lee, J. S. Kim, S. J. Hong, and J. Kim, "Distance Bounding with Delayed Responses," *IEEE Communications Letters*, vol. 16, no. 9, pp. 1478–1481, 2012.

[20] R. Entezari, H. Bahramgiri, and M. Tajamolian, "A Mafia and Distance Fraud High-resistance RFID Distance Bounding Protocol," in *Proc. of International Conference on Information Security and Cryptology*, 2014, pp. 67–72.

[21] A. Mitrokotsa, C. Dimitrakakis, P. Peris-Lopez, and J. C. Hernandez-Castro, "Reid et al.'s distance bounding protocol and mafia fraud attacks over noisy channels," *IEEE Communications Letters*, vol. 14, no. 2, pp. 121–123, 2010.

[22] G. P. Hancke, "Distance-bounding for RFID: Effectiveness of 'terrorist fraud' in the presence of bit errors," in *Proc. of IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*, 2012, pp. 91–96.

[23] C. Cremers, K. B. Rasmussen, B. Schmidt, and S. Capkun, "Distance Hijacking Attacks on Distance Bounding Protocols," in *Proc. of IEEE Symposium on Security and Privacy*, 2012, pp. 113–127.

[24] Y.-S. Kim and S.-H. Kim, "RFID distance bounding protocol using m-ary challenges," in *ICTC 2011*, 2011, pp. 782–783.

[25] D. H. Yum, J. S. Kim, S. J. Hong, and P. J. Lee, "Distance bounding protocol with adjustable false acceptance rate," *IEEE Communications Letters*, vol. 15, no. 4, pp. 434–436, 2011.

[26] M. Schafer, V. Lenders, and I. Martinovic, "Experimental Analysis of Attacks on Next Generation Air Traffic Communication," *Appl. Cryptography Netw. Security*, pp. 253–271, 2013.

[27] G. P. Hancke and M. G. Kuhn, "An RFID Distance Bounding Protocol," in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, 2005, pp. 67–73.

[28] FAA, "Fact Sheet - Small Unmanned Aircraft Regulations (Part 107)," https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=22615, 2018.

[29] Q. Gu, D. Michanowicz, and C. Jia, "Developing a Modular Unmanned Aerial Vehicle (UAV) Platform for Air Pollution Profiling," *Sensors*, vol. 18, p. 4363, 12 2018.

[30] K. B. Rasmussen and S. Čapkun, "Realization of RF Distance Bounding," in *Proc. of USENIX Conference on Security*, 2010.