CENTRE *for* DOCTORAL TRAINING *in*
CYBER SECURITY

CDT Technical Paper

07/16

Ensuring honest Behaviour in Cooperative Surveillance Systems

Dennis Jackson

Ensuring Honest Behaviour in Cooperative Surveillance Systems

Dennis Jackson

8th September, 2016

Abstract

With the development of cooperative surveillance systems such as ADS-B and AIS, great effort has been spent in order to ensure their specifications deliver safety properties. Unfortunately, less thought was put into how to secure these systems against hostile acts. As both systems have entered active use, recent research has explored attacks against these protocols and associated countermeasures. This paper presents a novel solution, designed to ensure the security properties of these surveillance systems without altering their existing behaviour.

Contents

1 Introduction **1**

2 Setting **2**
2.1 TDOA and TOA Calculations 3

3 Related Work **5**

4 Threat Model and Objectives **5**

5 System Description **6**
5.1 Protocol Flow 7
5.1.1 Initialisation 8
5.1.2 Discovery and Challenge 8
5.1.3 Reconciliation 10
5.1.4 Accidental Alerts 11

6 Evaluation **11**
6.1 Movement Model 12
6.2 Radio Model 12
6.3 Adversary Behaviour 12
6.4 Scenarios 13
6.5 Methodology 15
6.6 Scenario Results 15
6.7 Behaviour under noise 15
6.8 Summary 16

7 Extensions and Further Work **17**
7.1 Representation of D 17
7.1.1 Cubic Spline Construction 17
7.1.2 Cubic Spline Refinement 18
7.2 Representation of B 18
7.2.1 Prefix Trees 18
7.2.2 Bloom Filters 18

8 Conclusion **19**

1 Introduction

Historically, we have relied upon non-cooperative surveillance systems such as radar to manage separation and deconfliction in shared navigational space (be it air, land or sea). However, as these shared channels become busier and busier, there is increasing pressure to improve throughput by reducing separation zones between entities. Recently, cooperative surveillance systems have been introduced, which allow users of the space to report their position to their neighbours and hence manage their own safety.

Typically these systems make use of GNSS[1] such as GPS or Galileo and a common radio channel in the VHF range. In the aviation setting ADS-B was finalised (ICAO) in 1999 and has become mandatory for civil aviation in Europe since 2015, and will be in the USA from 2020 [22]. Similarly, nautical navigation (IMO) finalised AIS in 1998 and began to make it mandatory from 2002 [2]. Furthermore, research is being undertaken to adapting these systems to autonomous road vehicles [**?**].

Unfortunately, the development of these systems has focused on safety to the almost entire exclusion of security. Both ADS-B and AIS operate without any defences to mitigate malicious interference, impersonation or jamming. One offered reason for this is that at the time the specifications were drafted, the required equipment to transmit on the channels was reasonably expensive and consequently anyone with access to such resources would be able to employ alternative strategies to greater effect. Needless to say, the price of radio equipment has plummeted to over the last 20 years to the point where fake ADS-B messages can be transmitted to planes with little more than a laptop and a £200 antenna.

In addition to this, there are other limitations on the countermeasures that can be employed. Firstly, the remarkably low bandwidth and high congestion of the systems employed means that traditional encryption and signature schemes are unworkable. Additionally, such systems benefit from their open nature, hence they must remain open to users who are untrusted. Finally, planes and ships travel across many jurisdictions and consequently changes in protocol must be agreed by large multinational bureaucracies, yet changes in the security landscape rarely keep to this slow pace.

Considering all of these factors, we must accept that economic demands drive us towards these cooperative surveillance systems, due to technical limitations we cannot burden the systems with heavyweight cryptography, and we cannot rely on solutions which result in a change in protocol. This leads us to suggest the introduction of separate verification systems, which supplement but do not alter existing systems, and can be implemented without requiring every user of the old technology to upgrade.

The current state of the art is to use a combination of cooperative technologies (such as ADS-B) to provide safety, whilst non-cooperative technologies (such as Radar or WAM) provide security. Unfortunately non-cooperative systems are orders of magnitude more expensive than the cooperative technologies. Radar systems require careful placement and consume significant amounts of power, additionally they require constant maintenance. WAM systems require a large footprint to be effective and a reliable ground network to provide communication. Unlike radar, WAM systems are not secure against dedicated attackers and can be attacked by use of directional antenna broadcasting to known receiver positions. [3]

Our insight is to make use of the navigating entities themselves in order to verify the authenticity of transmitted data. By carefully selecting a number of trusted nodes, we can use the mobile entities to multilaterate untrusted transmitters and hence assure the correctness of transmitted information. This approach has the benefits of WAM in terms of accuracy and low-overhead, but is even more cost effective as it requires no additional infrastructure. Furthermore,

[1] Global Navigation Satellite System

it scales with the density of traffic and does not require participation from multiple countries.

In the following, we establish a threat model, describe a supplementary system which can be applied to any particular domain (land, sea or air) and finally validate its effectiveness with a computer simulation.

2 Setting

We first need a formal definition for cooperative surveillance systems that is sufficiently general to capture the systems we wish to study:

Definition 1. *A **cooperative surveillance system** consists of a set of surveilled entities in 3D space along with a set of surveilling entities. Note that these sets may not be disjoint and in some cases may even be equivalent. The surveilled entities emit claims regarding their positions in the form* $\{ID, POSITION\}$ *which are received by the surveilling entities subject to range and noise limitations. These position claims will be broadcast at a fixed rate (e.g. 2 claims per second) but not necessarily broadcast at a fixed interval. In some systems there may be velocity claims broadcast as well.*

Systems such as ADS-B and AIS easily fall within these schemes [15] [**?**]. The intended purpose of a CSS is to provide accurate and timely location information to surveilling entities. In particular both schemes allow for surveilling entities which do not make claims (e.g air traffic control towers, satellites) as well as entities which both surveil and make claims. In both such systems there has been little thought given to how to ensure their security properties (prevention of harm caused by a malicious adversary) as opposed to their safety properties (prevention of harm caused by accidental events).

Obviously one such solution is to phase out these systems and aim to replace them with schemes which offer both security and safety properties. Unfortunately due to the bureaucracy of supranational organisations such as the ICAO and IMO, the development, standardisation and implementation of these systems takes a considerable amount of time. As was noted earlier, ADS-B was finalised in 1999 but will not be fully deployed in the USA/Europe until 2020. Consequently, we must look to additional protocols and schemes that can ensure security properties of the cooperative surveillance system without modifying the underlying protocol. That is to say, both surveilling and surveilled entities must continue to function correctly even if they do not implement the additional protocol. Function correctly is taken to mean that they can provide the same information / receive the same information as an equivalent arrangement of entities not running the additional protocol.

We are primarily interested in the inhomogeneous nature of the entities that participate in these protocols. For example ADS-B surveils entities such as large passenger aircraft with hundreds of lives aboard, all the way down to cheap consumer drones (and a similar situation occurs on the seas). With the controlled nature of civil aviation, a great deal more control and security can be placed onto large passenger aircraft for relatively minimal cost. In contrast deploying ADS-B to drones is difficult enough[2] without adding any additional hardware. This asymmetry leads us to the proposal that our more significant entities can be entrusted with more responsibility (because the compromise of a large civil plane represents close to the worst case already) and then using additional protocols and perhaps equipment placed on these entities, to provide security guarantees regarding the overall system. Motivation aside, we will deal with the system and the additional protocol in in a general setting.

[2] the equipment required is only a GPS receiver and radio transmitter

2.1 TDOA and TOA Calculations

There are a number of characteristics that can be used to identify the source of a radio transmission such as AOA or signal strength estimation etc. Given that we do not want to install additional equipment onto our entities, we will look at Time Delay of Arrival (TDOA) and Time of Arrival (TOA) multilateration which requires only an antenna and a synchronised clock. TDOA Multilateration uses synchronised clocks on two or more receivers to calculate the time delay between when the signal reached each source. We can derive the equation below from first principles (using no more than $t = d/v$):

$$t_b - t_c = \frac{||p_a - p_b||}{c'} - \frac{||p_a - p_c||}{c'}$$

and then using the claimed position p_a and known locations p_b, p_c, the time of arrival t_b and t_c at p_b and p_c respectively and the speed of light in the atmosphere c', we see if equality holds (up to tolerance). If equality fails, then we know the position claim must be false. However, if the equality holds then it is still possible that the position claim is false, this is because for a fixed time delay, the valid solutions in p_a form a hyperboloid sheet. If you inspect the number of variables, it will quickly become apparent that three such equations would be needed to fix a particular point, which would require four measuring entities. Note that both entities must measure the same transmission (position claim) so in high noise environments, may be difficult to impossible to fully multilaterate a particular claim.

If we define a particular tolerance for the equality to be k and three distinct such equations hold then the bounding sphere around a position claim will be kc so for example 0.00001 leads to a bounding radius of about 3000m. There will be measurement errors in the position claim, the known position and the exact time delay (clocks can only be so accurate), however in practice they tend to be remarkably small. Using a GNSS such as GPS, distinct entities can be synchronised to within 10 nanoseconds of each other [7]. This means the multilateration could be in principal accurate to 30cm.

Another method is to measure the Time of Arrival of a particular signal. This can only be done if a trustworthy entity embeds the time a message was sent. The receiving entity can then measure the time it took for the message to arrive and hence the distance between the two entities. The equation in this case is simply:

$$t_a - t_t = \frac{||p_a - p_b||}{c'}$$

Where t_a is the time of arrival at known position p_b, t_t is the start time of transmission at claimed position p_a and c' is again the propagation speed of light in the atmosphere. A similar process for TDOA multilateration occurs but the valid set of claims for this position is in fact a sphere. Consequently we need three entities to receive a message in order to ensure it came from a particular point. Notice including the transmission time provides us with more information and hence we need fewer entities, however the transmitting entity must be honest, otherwise any position can be claimed.

These multilateration systems have been studied extensively and the basis for GNSS systems such as GPS, Galileo and so on. There are a number of subtleties, but in particular it is important to consider the Geometric Dilution of Precision (GDOP), which is a factor in error calculations. The measurement error given above is in fact the best case measurement error. Consider the hyperboloid / sphere not as a sheet, but as a 3D region of space due to the measurement uncertainties. Then certain configurations of receivers will lead to relatively large intersections between these regions and hence greater uncertainty in the actual position of the transmitting entity.

A fascinating result (derived in [8]) is that this geometric dilution of precision is in fact equivalent to the moment of mass[3] of receivers around the transmitting entity. This intuitively means that these systems are highly accurate when the receiving entities convex hull contains the source of the transmission, and most inaccurate when the source of the transmission is displaced from the receiving entities centre of mass on one or more axes[4].

A more complete treatment can be found in [9].

3 Related Work

ADS-B and AIS have long been suspected to be vulnerable to attacks from malicious actors. However in 2011, Sampigethaya and Poovendran published [17] which outlined these concerns in broad strokes (as part of a greater work on the new attack surface offered by digitised ATC systems) and proposed using "the group navigation property" in order to provide integrity for ADS-B.

The first paper to comprehensively document the vulnerability of ADS-B was published in 2013 by Martinovic et al [18] and documents a number of vulnerabilities which allowed for the manipulation of reported planes and the introduction of entirely fictional planes. A significant contribution of this work was that the attacks were concretely demonstrated using physical hardware, which made it much harder for regulatory bodies to ignore.

As a follow up, Martinovic et al published [21] in the same year which summarised the known attacks against ADS-B, as well as bringing together a number of proposed solutions. Significant effort was made into developing a taxonomy of these solutions, according to their benefits and drawbacks. Using group verification was discussed a solution, but the relative complexity of the additional protocols was seen as a significant drawback.

Work on ensuring the integrity of the ADS-B system began in earnest and [6] discusses one particular method using wide area multilateration to detect false position claims. Unfortunately such a system cannot protect individual aircraft against a spoofing attack using a directional antenna, as well as requiring significant investment on infrastructure to cover an area. In 2015, Martinovich et all published [20], which proposes an alternative solution using statistical methods to detect injected messages. Also in 2015, Schafer et all published [19] which exploits the mobility of the aircraft in order to derive more information, noting that movement along a particular vector is harder to fake than a position claim in a single instance.

[3] demonstrated the threat of attackers utilising multiple devices or directional antennae to fake TDOAs and furthermore showed how they could be defeated using additional information from the physical layer. However, they left open the question of whether a cleverer attacker could also mask these physical characteristics, or if in fact it is a fundamental asymmetry.

In general the vast body of work has focused on using static antennae operated by a single organisation to perform multilateration or similar techniques rooted in statistical inference to provide integrity over a wide area. As well as the relative cost and fragility of such systems, they only provide integrity for the operator of the equipment and can be fooled by attackers injecting messages with directional antennae. Furthermore such systems must be deployed on a region by region, country by country basis at considerable expense. In contrast the system we will explore will provide each participating entity with a measure of security and not rely upon adoption in particular geographic areas.

[3]Note that we are referencing the form of the derived matrix, in the calculations for accuracy, 'unit mass' is used.

[4]This is why ground based multilateration systems are accurate for longitude and latitude (as they often contain the aircraft on these axes) but inaccurate for altitude (where they do not)

4 Threat Model and Objectives

We will in general assume that certain entities are always honest and follow the defined protocol behaviour. This assumption would be remarkably strong in many settings, but in civil aviation for example, if a jet airliner carrying hundreds of people is behaving maliciously, much has already gone wrong and the situation cannot get much worse. Equally, we will assume that our attacker is limited in economy, i.e. they can deploy multiple devices with a clever coordinated strategy, some devices may be drones or other expensive pieces of equipment, however, the attacker cannot launch thousands of drones and conduct a Sybil attack. We are primarily concerned with low investment, high intelligence attackers conducting attacks which threaten the lives of hundreds of people. If the attack requires nation-state resources, we can be confident such nation states have cheaper, more reliable methods for disrupting civil aviation.

Assumption 1. *Upgraded entities (defined later) are always honest*

Assumption 2. *The adversary has full control over the network, however at least 4 upgraded entities will only receive legitimate GPS transmissions and will not have their message transmissions jammed.*

This assumption is intended to allow the adversary as much power as possible, without including the unsolvable case in which there is no correct information in the system. In practice this assumption is actually quite suitable as it is unlikely an economically restricted actor could dominate a number of radio channels over hundreds of miles without one or more military bodies becoming involved.

Goal 1. *If an entity makes a false position claim (actual location is at least a tolerance from the claimed position) which is covered by a successful protocol session, the session will ouput "ALERT".*

Goal 2. *If a successful protocol session only includes true position claims, the session will output "PASS".*

Here we set the threshold quite low at correctness. Notice that we omit a large number of attacks with these modest goals, including those against the protocol itself. We will also argue for properties such as aliveness even in the presence of misbehaviour and so on, but leave their proper discussion for future work.

5 System Description

As this is just a first outline, for now we will abstract out physical characteristics in favour of a general approach, of course for any particular implementation the amount of bandwidth and range available is important, but we again leave those important issues for future work. We begin by defining the attributes of the entities we consider:

Definition 2. ***Free*** *Entities may travel through 3D space up to a maximum velocity.* ***Confined*** *Entities are limited to a subset of 3D space, up to a maximum velocity.* ***Predictable*** *Entities have known public locations, they may be stationary (e.g. an air traffic control tower) or mobile (e.g. a satellite)*

Definition 3. ***Upgraded*** *Entities have a public/private key which is secure. Keys are signed by a central authority. We may specify the behaviour of Upgraded entities according to our system.* ***Legacy*** *entities have no special behaviour.*

As we discussed above, we will consider a number of entities in a volume of 3D space. We will define U to be a set of present upgraded entity IDs, and likewise L for legacy entities. We define the following functions:

$$Range : \{Channels\} \rightarrow \mathbb{R}$$

$$P : U \cup L \rightarrow \mathbb{R}^3$$

$$K : U \rightarrow Hashchain$$

$$T : \varnothing \rightarrow Time$$

$$S : U \rightarrow PrivateKey$$

I chosen to write $T()$ as a constant function to describe the current time when it is invoked. We will assume our upgraded devices have synchronised clocks, through GPS for example. K is a one way secure hash chain which mutates over time and signifies the authenticity of a message. For example it could be the result of the TESLA scheme [16]. In TESLA, the first message is signed by a digital signature and then subsequent messages can be authenticated with low overhead. The technical details of this will be left for future work, for now we stress that K must provide authenticity and integrity of a packet.

Our entities have access to a number of channels which we describe below:

Definition 4. *All entities have access to a channel which provides them with positioning and timing information at fixed regular intervals. Rather than consider implementation specific details, we will consider it simply as a channel, denoted ϕ, on which messages flow from an oracle to individual entities at a fixed rate. In normal operation, we will assume the navigational information is correct up to a certain margin of error.*

Definition 5. *All entities have access to a channel, denoted ψ, on which they broadcast position claims at a fixed rate in the form:*

$$\{ID, P(ID)\}$$

All entities also receive and process information on this channel. They may optionally transmit velocity claims in the same format as well.

The above matches our definition of a cooperative surveillance system.

Definition 6. *Upgraded Entities have access to a broadcast channel, denoted ω, over which they may transmit arbitrary messages.*

The motivation for this channel is the use of VDL Mode 4 [4] or similar vehicle to vehicle channel. We will make a general assumption on channel range that the maximum distance between any two upgraded entities $R = \max_{i,j \in U}(|P(i) - P(j)|)$ will be such that $R << Range(\omega)$ and $R < Range(\psi)$. There will be some number of upgraded entities $n = |U|$ and some number of legacy entities $k = |L|$. In general we expect $n < k$, but we will not enforce this as a requirement.

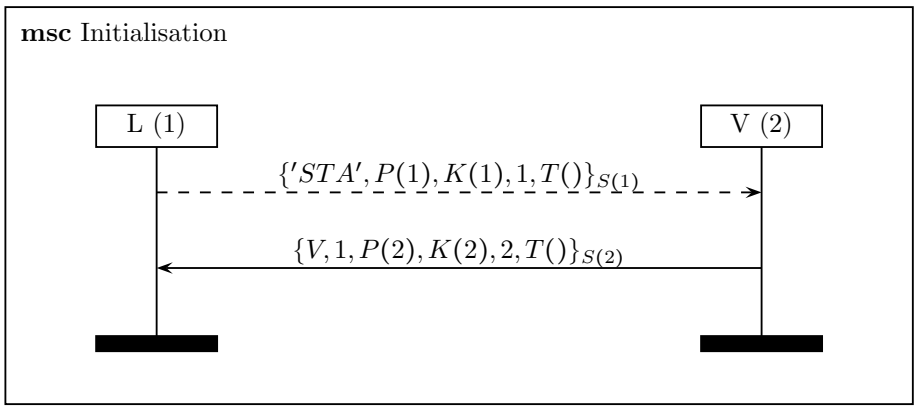
Upgraded entities will cache all transmissions on ψ received over the last time period, with an associated timestamp taken as the message was received. We will assume the local clocks are globally synchronised, either through their positioning channel or some other mechanism.

5.1 Protocol Flow

The protocol is split into three phases. The first and final phase coordinate the roles of the group members and verify that no member witnessed misbehaviour. In the intermediate phase, data is exchanged in order to verify consistent behaviour.

In the following diagrams, all messages are sent over the broadcast channel ω so in the absence of collision or malicious behaviour, we would expect all upgraded entities to receive all messages. However, such events may occur and consequently most channels have message delivery schemes (e.g. VDL Mode 4 [4]) which ensure (in the absence of malicious behaviour) that messages are delivered within a certain amount of time. Consequently, solid lines will indicate when we are transmitting a message with an explicit recipient and a 'safe' delivery guarantee. Dashed lines will indicate messages where delivery is ensured by another mechanism, for example indirectly by the later safe delivery of a message protected by a time out. We will detail this mechanism on a message by message case.

5.1.1 Initialisation



Where $V \in \{1, 0\}$ represents a yes vote or no vote respectively.

In this phase our upgraded entities decide whether or not they will participate in this run of the protocol. They may decide not to due to distance, poor connection or some other factor. We will require that an entity only agrees to participate if it believes it will be present to conclude the session. Otherwise jammed sessions and accidental loss of connection are hard to distinguish.

It should be noted that from the information provided at this stage, the upgraded entities have enough information to perform TOA calculations and check witnessed timestamps and public position claims for consistency. In the event of an inconsistency, the verifying entity should raise the alarm.

For the signatures, we can expect our entities to maintain a cache of ids against public keys. These could be stored long term and not expected to change often or alternatively downloaded over a side channel. There are many solutions to public key infrastructure and in this scenario we benefit from the centralised control already in place to operate the surveillance system.

If there are not at least 3 yes votes ($V = 1$) then the session does not proceed. We will expect votes in either direction after a certain amount of time, otherwise we must assume the adversary is jamming the channel. The exact details of how to implement this will need to be implementation specific due to the nature of the channel congestion.

We can ensure the first message is delivered to start the session by retrying transmission until we can be confident that the absence of a response implies jamming. The second message

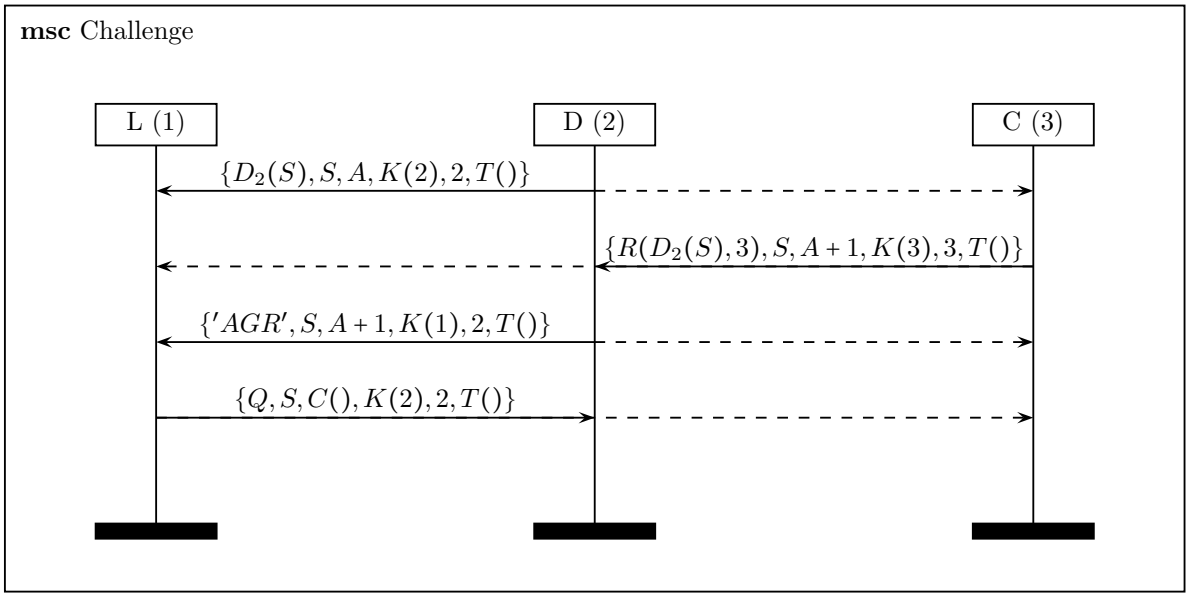
will be delivered 'safely' and we can ensure securely by imposing a time-out after the repeated transmission of the first message.

5.1.2 Discovery and Challenge

We now proceed to the discovery and challenge protocol. In this phase our upgraded entities exchange information in order to build a shared picture. Our approach is to take a particular legacy entity (proceeding by some fixed order) and then have the closest upgraded entity broadcast the witnessed position claims. Then, other upgraded entities which saw additional or differing information will broadcast it and a challenge occurs. Once a consistent view has been established which all upgraded entities agree on, then each upgraded entity can calculate the respective TDOA (detailed in section 2.1).

More precisely, we first use a lexicographic order on the witnessed ids. This allows to avoid congestion by having multiple planes race to be the first to transmit. There may be some confusion as only some planes may have witnessed an ID and hence broadcast out of order, this effect will be much reduced compared to proceeding randomly. Then we proceed by the closest entity, again their may be confusion here so we will use a counter to order claims. In the event of simultaneous transmission, the leader can ensure they come to consensus by taking a lexicographic order on the transmitting planes and ignoring the messages that did not come from the least element. Thus we maintain the advantages of avoiding congestion, but in the event of confusion we ensure there is a canonical flow.

The following diagram shows the exchange wherein entity 2 has announced a position claim for a given identifier and entity 3 has additional information which it wishes to report.



Where A represents an increasing counter on a per subject basis, $C()$ is an increasing counter recording the number of subjects recorded so far, S is the current subject ID, $D_I(S)$ is information describing the recent position claims and their timestamp witnessed from S by I. $R(D, I)$ is a function that refines data given the information from I. For example if D is simply a list/set of claims then R could be concatenation or set union. However it could be much more sophisticated

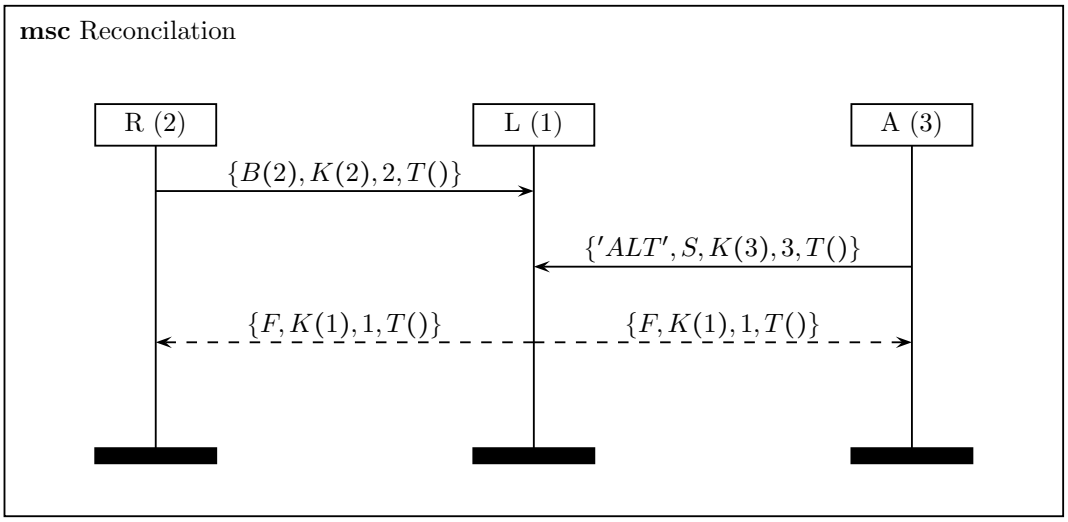
and later we will discuss efficient data structures for this information. Q is the hash from the final challenge or discovery message that the leader saw. Any entity which has a message matching this hash can verify that it has the most up to date position information.

There maybe zero or more challenges. These represent refinements on the claimed positions wherein the challenger has additional information. In general, we do not expect (m)any though as the closest upgraded entity should receive almost all claims. The $'AGR'$ message indicates that the discoverer's data points (which may not have been revealed fully) agree with the resulting refinement. In the event of a further challenger, we will need agreement from both the discoverer and the original challenger. The motivation for this message is to discover inconsistent claims in the event two witnesses saw different claims for the same instant.

For convenience, I will refer to the messages above as W, X, Y, Z. We can ensure W is received by 1 because our leader is expecting at least one discovery per plane. Equally, other planes are expecting to see this discovery and in the absence of it they will either interrogate the plane they believe closest or broadcast their own message. Consequently we can be sure W is in general received. In the event 1 does not receive W we know that the adversary has interfered. Equally any challenge must be safely delivered by the flow of X from 3 to 2 and Y from 2 to 1. Furthermore all planes are expecting to see the leader announce the finalised hash in Z and hence they can interrogate the leader for it. In the event this fails we know the adversary has intervened. Consequently we know that these discovery-challenge sessions will succeed (or else we can be sure of hostile interference).

5.1.3 Reconciliation

In this phase we close the session by establishing how much information was verified and whether there were any inconsistencies. We will expect the alerts and reports to come in lexicographic order of the upgraded entities in order to avoid congestion issues. Here we see R reporting the planes they witnessed, whereas A is transmitting an alert message to the leader (L).



Where B is the set of identities of witnessed planes and $S \in L \cup U$ and F is the result of the session.

The ID of an entity will be in B if either of the following (exclusive) conditions is met:

- The entity ID is upgraded and the reporting entity verified a TOA calculation against the claimed secure position. And the witnessed position claims are consistent with the claimed secure position.
- The entity ID is legacy and the reporting entity verified a TDOA calculation against the timestamped position claims.

$'ALT'$ messages will be sent in the event that an inconsistency has been witnessed. An inconsistency is when:

- Absence of any witnessed claims despite being in reception area
- Wrong TOA or TDOA from an upgraded or legacy entity (see 2.1)
- Mismatch between secure claim and normal claim
- Inconsistent position claims (between two normal claims).

The leader can then inspect these responses and verify whether enough entities verified enough claims to be sure of consistency. F will either be a positive integer indicating how many entities have been securely transmitted (indicating the leader has found nothing wrong) or negative (indicating failure). The other upgraded entities check if their own counter matches F and can decide the outcome from that.

We know that reports and alerts will be safely delivered, so the adversary must have intervened if they do not arrive (we are expecting one message per yes vote and the leader has a record of yes votes). Finally every voter is expecting a results message from the leader and hence can request a retransmission if they do not receive the result in time and if that fails, they know they can raise the alarm.

5.1.4 Accidental Alerts

We can also estimate the probability that a secure entity will not witness any position claims due to some kind of channel congestion. As long as we know the failure rate is remarkably low, we can assume missing information is indicative of malicious injection or other misbehaviour.

Let $P(Claim)$ be the probability that an individual position claim is received. And let m be the number of messages which are emitted in the witness period. Then

$$P(NoneSeen) = (1 - P(Claim))^m$$

Now when we know there are n upgraded entities within reception range and consequently, the chances that less than 4 of them witness any position claims is:

$$P(\text{n-3 are none}) = P(NoneSeen)^{n-3} = (1 - P(Claim))^{m \times (n-3)}$$

For example using ADS-B we need to two consecutive messages and typically 1/3 messages collide. Furthermore approximately 10 positions claims are made per 10 seconds in normal operationthe equipment only being a GPS receiver and radio transmitter. Finally we assume there are 6 upgraded planes nearby:

$$P(Claim) = (2/3)^2$$

$$P(NoneSeen) = (1 - P(Claim))^{10} \approx 0.3\%$$

$$P(3 \text{ are none}) = P(Noneseen)^3 = 0.0000027\%$$

Consequently, we can typically assume the absence of information will be due to malicious behaviour and decide to raise the alarm. However in a different scenario with less frequently transmitted claims or higher channel noise, we may well have to ignore such red flags.

6 Evaluation

Using Omnet++, a Discrete Event Simulator [14], a simulation was implemented in which a number of entities participate in a cooperative surveillance system, with the protocol described above being used on a subset of them. Due to the relatively short time scale of this project (9 weeks to research, implement and evaluate the protocol and simulation), a number of approximations have been made.

Firstly, the simulation only covers one run of the protocol in which the upgraded devices record the claims for a fixed amount of time, then launch into the session. Upon the conclusion of the protocol session, the simulation terminates. It is possible to control how many entities are present and their type (upgraded, legacy, spoofing). Furthermore channel noise is implemented only for the claim channel and location channel, as implementing the delivery retry / request / time out sub-protocol would require too much time. This noise is only implemented as a random dice roll (with configurable threshold) rather than being a proper simulation of radio transmissions and collisions and so on. Furthermore, due to this simplistic radio model, bandwidth is not modelled so it is not possible to measure how the system scales. This would of course be an important consideration to test, as a viable secure solution must also be functional under high load.

Additionally the simulation only allows for hard coded attacker behaviour in handwritten scenarios. In particular, it allows for the spoofing of the location channel for one or more upgraded devices, one or more legacy devices or the manipulation of the claims channel for any device. This of course only presents a small subset of the possible attacks we discussed before, but time was a limiting factor. Finally the challenge/refinement portion of the protocol has been simplified to the nearest plane announcing the timings it saw. This approximation is possible because of the lack of noise or jamming on the secure channel and because it represents a lower bound on the performance we can expect.

6.1 Movement Model

The simulation initialises each entity to a random location within a 200km x 200km x 200km cube and chooses a random velocity with speed at most 25 meters per second. Originally the movement model was designed to be relatively flexible and allow for a number of different behaviours e.g. random walk, stationary, linear path or loading data from an external file, but due to time limitations, only linear motion [5] on a random trajectory has been implemented. Furthermore whilst it would be most efficient to only update an entities position when it emits or receives a message, the current simulation updates all entities positions at fixed time steps.

6.2 Radio Model

Originally the intention had been to use the INET library [13] for the radio simulation, however due to time constrains this did not prove feasible. Using INET would have been desirable as it has

[5] $p = vt + c$ where v is the randomly chosen velocity, p is the position, t is the current time and c is the randomly chosen starting position.

a sophisticated model which takes into account background noise, message collision, shadowing and other real world effects. Instead a much more simplistic model has been created which simply models the probability that a given message will be successfully transmitted as a uniform random variable and the probability of success as a simulation parameter. This calculation is done for every receiver, so in general some but not all entities will receive a message. This is of course not realistic for the majority of settings, but our motivation is aircraft transmitting radio signals over relatively short distances at altitude and consequently multipath propagation, terrain masking and so on are not immediate concerns.

6.3 Adversary Behaviour

Capturing adversary behaviour in a simulation is always difficult. There are a number of complex and subtle behaviours that might conceivably be exploited in order to cause a malicious outcome. Due to the time constraints of this project, the simulation only captures the simplest of attacks in which the adversary either interferes with the location channel of an upgraded or legacy device (causing it to report an incorrect position). This also captures the adversary injecting messages from a non-existent legacy device. The spoofed position is chosen to be a random fixed offset of the true position, this is to ensure its track appears indistinguishable to normal behaviour in the simulation.

6.4 Scenarios

A total of 8 scenarios have been crafted, each with a different number of parameters.

Name	Scenario	Parameters
Standard	6 Upgraded entities and 15 legacy entities, randomly distributed over 200km x 200km x 200km cube with random linear paths. Each with completely accurate position information. All behave honestly. The session should pass.	Noise - the probability of any transmitted message being received at a particular antenna.
Noisy Standard	6 Upgraded entities and 15 legacy entities, randomly distributed over 200km x 200km x 200km cube with random linear paths. Each with fuzzed position information (each coordinate $+-5$m). All behave honestly. The session should pass.	Noise - the probability of any transmitted message being received at a particular antenna.
Standard Legacy Spoof	6 Upgraded entities and 15 legacy entities, randomly distributed over 200km x 200km x 200km cube with random linear paths. Each with completely accurate position information. One legacy device attempts to spoof its location, representing either an honest entity which the adversary has spoofed the GPS of or the adversary directly injecting a sequence of false claims. The session should result in an alert.	Noise - the probability of any transmitted message being received at a particular antenna.
Noisy Legacy Spoof	6 Upgraded entities and 15 legacy entities, randomly distributed over 200km x 200km x 200km cube with random linear paths. Each with fuzzed position information (each coordinate $+-5$m). One legacy device attempts to spoof its location, representing either an honest entity which the adversary has spoofed the GPS of or the adversary directly injecting a sequence of false claims. The session should result in an alert.	Noise - the probability of any transmitted message being received at a particular antenna.
Standard Upgraded Spoof	6 Upgraded entities and 15 legacy entities, randomly distributed over 200km x 200km x 200km cube with random linear paths. Each with completely accurate position information. One upgraded device attempts to spoof its location, representing an honest entity which the adversary has spoofed the GPS of. The session should result in an alert.	Noise - the probability of any transmitted message being received at a particular antenna.
Noisy Upgraded Spoof	6 Upgraded entities and 15 legacy entities, randomly distributed over 200km x 200km x 200km cube with random linear paths. Each with fuzzed position information (each coordinate $+-5$m). One upgraded device attempts to spoof its location, representing an honest entity which the adversary has spoofed the GPS of. The session should result in an alert.	Noise - the probability of any transmitted message being received at a particular antenna.
Not Enough Upgraded	2 Upgraded entities and 2 legacy entities, randomly distributed over 200km x 200km x 200km cube with random linear paths. Each with completely accurate position information. The session should not take place as there are not enough upgraded entities. There is no channel noise.	None
Free Parameters	Entities are randomly distributed over 200km x 200km x 200km cube with random linear paths. Each with completely accurate position information.	The number of Upgraded, Legacy, Upgraded Spoofing, Legacy Spoofing and channel noise.

6.5 Methodology

The simulation has been instrumented to record whether the session resulted in a success or failure and the total number of TDOA checks made and those that conclusively passed, versus those that conclusively failed or were uncertain. These results were taken from running the above files.

For each noise value, each scenario (except Not Enough Upgraded and Free Parameters) was ran three times with a different random seed each time. The simulation was ran on an i7-3820 processor with Omnet++ running in Express mode. Each run finished in < 5 seconds.

6.6 Scenario Results

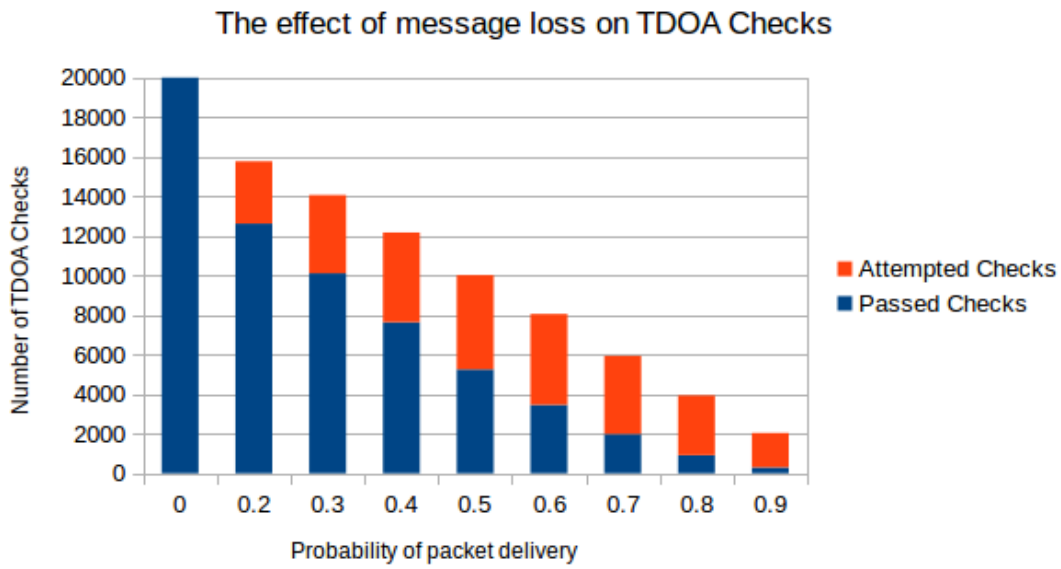
Scenario	Probability of a packet being lost (at receiver, independent)				
	No Loss	1/3 Loss	2/3 Loss	9/10 Loss	99/100 Loss
Normal	Pass	Pass	Pass	Pass	Alert
Normal with location error	Pass	Pass	Pass	Pass	Alert
Spoofing Legacy Device	Alert	Alert	Alert	Pass	Alert
Spoofing Legacy Device with location error	Alert	Alert	Alert	Alert	Alert
Spoofing Upgraded Device	Alert	Alert	Alert	Pass	Alert
Spoofing Upgraded Device with location error	Alert	Alert	Alert	Pass	Alert

Where the results differed upon the random seed, the majority result has been taken. Where the result differs from the expected behaviour, it has been highlighted in red. The deviations at 99/100 loss are somewhat expected as this artificially high loss leads to most entities having very incomplete information about their surroundings and consequently the alert is arguably correct and useful, even in the absence of actual malicious behaviour.

The other three deviations are more difficult to accept, despite the high noise level. They suggest that the high noise level allows for enough variation in signal reception to lead to missed alerts. Although with more instrumentation and more detail in the protocol, it would be possible to decipher what exactly lead to these erroneous results.

6.7 Behaviour under noise

The following graph was taken from running the standard simulation with 9 different noise levels and recording the number of TDOA checks that were attempted and succeeded. A TDOA check is attempted when an upgraded entity receives the timing information for a particular position claim. It passes if the equation given in section 2.1 holds to within tolerance, it fails if not and if the particular position claim was not received by the entity, it is considered only attempted.



		TDOA Checks			
		Passed	**Attempted**	**Total**	**Pass Ratio**
Probability of Packet Loss	0	19988	0	19988	100.00%
	0.2	12615	3145	15760	80.04%
	0.3	10105	3951	14056	71.89%
	0.4	7626	4538	12164	62.69%
	0.5	5251	4761	10012	52.45%
	0.6	3454	4590	8044	42.94%
	0.7	1989	3939	5928	33.55%
	0.8	918	3014	3932	23.35%
	0.9	303	1737	2040	14.85%

As expected we can see the falling number of attempted checks due to increasing noise levels meaning the referring upgraded entity transmits fewer timestamped position claims. Also notice that the proportion of passed checks against attempted checks falls. This is because the number of attempted checks is proportional to $1-T$ where T represents the noise threshold, but passed checks is proportional to $(1-T)^2$ so we would expect the pass proportion to be $(1-T)$. We can see this in the table. In the full implementation of the protocol, the challenge/refinement process would result in significantly better behaviour under high noise conditions, but it is reassuring to see such promising results without it.

6.8 Summary

Although the simulation is quite an abstraction from the real world and consequently these results are quite preliminary, they are certainly promising. Unfortunately they only confirm that the very simplest attacks can be defeated by this scheme, the real question is in whether more complex attacks will succeed, and whether the system will scale under load, which will need to

be investigated in future work.

The simulation source code, scenarios and results can be found at [5]

7 Extensions and Further Work

The immediate area of further work would be to improve the accuracy of the simulation in order to explore behaviour as the system scales, both with regard to false positives / negatives and the overall bandwidth consumed on the network. There are many ways in which the accuracy could be increased, but the primary areas are:

Firstly the movement model could be moved from random linear movement, to use actual recorded data from cooperative surveillance systems. One possible source would be the Open Sky Network [1] which would then allow for simulations with accurate density, claim information and so on to be modelled.

Secondly using a mature radio model such as the one offered by INET would allow for the modelling of congestion on the channel and the size of the information transferred would become relevant. This would require more work to refine the protocol's data structures as currently it does not go to any lengths to conserve bandwidth. Additionally the secure channel does not currently have any noise and by implementing the message delivery subsystem, it would be possible to check how well it held up under scale.

Thirdly, extending the scenarios to handle multiple concurrent sessions with different leaders and groups in different locations, as well allowing the same entity to participate in subsequent sessions with another or the same leader would allow for a steady state to be reached.

Fourthly attacker behaviour needs to be extended to cover more possible attacks such as the attacker selectively jamming planes or trying more sophisticated "boiling frog" spoofing attacks. There is an entire question of how to capture these scenarios as well, only using scenarios crafted by hands leads to potential attacks being missed due to lack of imagination, but equally the parameter space is huge so whether a more intelligent automatic method might be used remains unknown.

Another area of further work would be in refining the protocol description still further. More exotic data types could be used for transmission which could have a dramatic impact on the bandwidth required for the operation of the protocol. Additionally the protocol would benefit from having more nuance then resulting in a pass/fail, it could for example attempt to detect the exact nature of the inconsistency and potentially transmit accurate information to upgraded devices.

7.1 Representation of D

$D(\_)$ could be represented in a number of ways. All it represents is a number of position claims with associated timestamps (according to the synchronised clock). For example a set would be a perfectly reasonable representation and then $R(D, \_)$ would simply be set union. However, there is significantly more structure in the data which we can exploit, an important property for transmission over wireless channels. In particular, cubic spline interpolation [10] seems ideally suited to our needs.

7.1.1 Cubic Spline Construction

Firstly we take any received velocity claims v_i and then for any pair (p_i, p_{i+1}) of position claims we derive the average velocity between these points and introduce the additional implicit velocity

claim (at the midpoint of the timestamps of p_i, p_j). Then we can generate an interpolated spline for the velocity.

Using the velocity spline, we can construct the triple (position,velocity,time) for our witnessed position claims. The velocity information (as the derivative of position) will allow us to construct a more accurate position polynomial. Then we generate our interpolated spline using this triple according to a particular scheme and store it with the start and end points (which might be the same if we only had one position claim to work from). These schemes have the property that input position claims will always lie exactly on the resulting polynomial.

7.1.2 Cubic Spline Refinement

Refinement is the process of taking a cubic spline interpolation and a series of timestamped position claims and introducing their information to derive a new representation. For our polynomials, we can first define a constant maximum tolerance, we will not even issue a challenge / refinement if our claim lies within this tolerance of the existing polynomial. When trajectories are reasonably stable (as is the case for ADS-B and AIS), this will result in a considerable reduction of challenges, improving transmission efficiency.

When the additional points do exceed this tolerance, we will create the refinement with reference to a maximum velocity and then generate the spline polynomial subject tot that. This can be done by generating one or more spline polynomial using the new position claims and start/finish points from the existing polynomial. These start/finish points will the closest in time that have a resulting polynomial with maximum derivative less than the maximum velocity above. We may add multiple of these additional polynomials if our challenger saw additional data points that were not consecutive.

7.1.3 Summary

Using cubic splines could result in significantly reduced bandwidth requirements if position claims are transmitted more frequently than trajectory changes. Additionally, in the event of missed claims, it would reduce the need for retransmission as the trajectory would be interpolated to cover these gaps.

7.2 Representation of B

In the simplest case we could use a list or set of identifiers and then the leader can inspect this by simply testing for membership. However, if the identifiers require a significant number of bits to store (e.g. 24 for ICAO Aircraft addresses) this could be very inefficient. Two possible solutions are: prefix trees (compact tries) and bloom filters.

7.2.1 Prefix Trees

In the event the identifiers are highly structured with respect to geographic location or time, we can make use of this for data compression. Each witness can simply transmit a prefix / radix [12] tree representation and it can be easily checked for membership. The amount of compression depends on the structure possible. If the nearby identifiers could be represented by a set of size x and the total number of identifiers was of size y, then each identifier can be written as $log_2(x)$ bits and hence the prefix tree would be lower bounded by $x \times log_2(x) + log_2(y)$ bits in size. However this best case is unlikely to be realizable. With ADS-B, the identifiers correspond to Operator and Flight number tags, there would be a reasonable saving of approximately 3/8ths of the bits.

7.2.2 Bloom Filters

These allow for the probabilistic storage of set membership requires approximately $10 \times x$ bits with minimal false positive rate. (And zero false negative rate). This means that occasionally our leader would report entities were verified when in fact not enough reports had been witnessed, but we would never falsely conclude an entity was unverified. This would also open the protocol to an attack in which the attacker injects additional identities and magnify the bloom filter's error rate which would result in many false witnesses. However additional sanity checks could be introduced such as restricting the number of identities in any particular volume of space (and noting that identities with spoofed locations would result an alert in the discovery phase). [11]

8 Conclusion

This paper has discussed the general setting of cooperative surveillance systems and some of the ongoing research into practical attacks and associated hypothetical solutions. A novel protocol for ensuring honest behaviour in these systems has been described. Notably, it does not rely on all participating entities upgrading or joining the new scheme, it only requires a certain number to ensure the honest behaviour of the rest.

As well as describing the protocol and its intended security properties, we have discussed data structures which might provide scalability in the event of real world deployment. Finally we have build a simplistic simulation and verified the protocol works against simple attacks under a simple noise model. This work is only the very first step towards validating and developing the protocol as a whole. The next steps have been discussed, in particular the need for a more complete simulation with subtler attacks and a more sophisticated movement and radio model.

References

[1] Open sky network. https://opensky-network.org/.

[2] All About AIS. History of ais. http://www.allaboutais.com/index.php/en/aisbasics1/ais-history.

[3] Vincent Lenders Aanjhan Ranganathan Fabio Ricciato Daniel Moser, Patrick Leu and Srdjan Capkun. Investigation of multi-device location spoofing attacks on air traffic control and possible countermeasures. *Annual International Conference on Mobile Computing and Networking*, 22nd, 2016.

[4] EuroControl. *Assessment of VDL Mode 4 Frequency, Capacity and Performances.* 4th edition, 2010.

[5] Dennis Jackson. Source code and results. https://www.dennisjj.co.uk/static/source.zip.

[6] Jerry Johnson, Holger Neufeldt, and Jeff Beyer. Wide area multilateration and ads-b proves resilient in afghanistan. In *Integrated Communications, Navigation and Surveillance Conference (ICNS), 2012*, pages A6–1. IEEE, 2012.

[7] NIST Physics Laboratory. Common view gps time transfer. http://tf.nist.gov/time/commonviewgps.htm.

[8] Harry B Lee. Accuracy limitations of hyperbolic multilateration systems. *IEEE Transactions on Aerospace and Electronic Systems*, (1):16–29, 1975.

[9] Ivan Antonio Mantilla Gaviria. *New strategies to improve multilateration systems in the air traffic control*. PhD thesis, Editorial Universitat Politècnica de València, 2013.

[10] Wolfram MathWorld. Cubic spline. http://mathworld.wolfram.com/CubicSpline.html.

[11] NIST. Bloom filter. https://xlinux.nist.gov/dads/HTML/bloomFilter.html.

[12] NIST. Patricia tree. https://xlinux.nist.gov/dads//HTML/patriciatree.html.

[13] OMNeT++. Inet framework. https://inet.omnetpp.org/.

[14] OMNeT++. Omnet++ discrete event simulator. https://omnetpp.org/.

[15] INTERNATIONAL CIVIL AVIATION ORGANIZATION. *ADS-B IMPLEMENTATION AND OPERATIONS GUIDANCE DOCUMENT*. 8th edition, 2015.

[16] Adrian Perrig, Dawn Song, Ran Canetti, JD Tygar, and Bob Briscoe. Timed efficient stream loss-tolerant authentication (tesla): Multicast source authentication transform introduction. Technical report, 2005.

[17] Krishna Sampigethaya and Radha Poovendran. Security and privacy of future aircraft wireless communications with offboard systems. In *2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011)*, pages 1–6. IEEE, 2011.

[18] Matthias Schäfer, Vincent Lenders, and Ivan Martinovic. Experimental analysis of attacks on next generation air traffic communication. In *International Conference on Applied Cryptography and Network Security*, pages 253–271. Springer, 2013.

[19] Matthias Schäfer, Vincent Lenders, and Jens Schmitt. Secure track verification. In *2015 IEEE Symposium on Security and Privacy*, pages 199–213. IEEE, 2015.

[20] Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. Lightweight location verification in air traffic surveillance networks. In *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*, pages 49–60. ACM, 2015.

[21] Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. On the security of the automatic dependent surveillance-broadcast protocol. *IEEE Communications Surveys & Tutorials*, 17(2):1066–1087, 2015.

[22] Aviation Week. Global advance of ads-b. http://aviationweek.com/connected-aerospace/global-advance-ads-bl.