Extending ADS-B for Mixed Urban Air Traffic

Thabet Kacem
Department of Computer Science
and Information Technology
School of Engineering and Applied Sciences
University of the District of Columbia
Washington, District of Columbia 20008, USA
Email: thabet.kacem@udc.edu

Alexandre Barreto, Paulo Costa, Duminda Wijesekera
Radio and Radar Engineering Laboratory
Volgenau School of Engineering
George Mason University
Fairfax, Virginia 22030-4444, USA
Email: adebarro@c4i.gmu.edu,[pcosta,dwijesek,]@gmu.edu

***Abstract*—Automatic Dependent Surveillance Broadcast (ADS-B) has become one of the most prominent protocols in the area of Air Traffic Control (ATC) due to its accuracy compared to traditional surveillance technologies, as well as for its suitability for deployment in areas where radar operations would be financially inviable or technically unfeasible (e.g., mountain ranges, at sea, etc.). In spite of its advantages, there have been considerable criticism from security and ATC experts on a serious vulnerability of the protocol. More specifically, its messages are exchanged in clear text over the air, which makes it an easy target for many attacks. In previous work, we proposed a framework to enforce the authenticity and integrity of ADS-B messages, for the 1090 Extended Squitter (ES) version of ADS-B, in ground-to-air and air-to-air scenarios by using security metadata based on keyed-hash message authentication code (HMAC) and a proper exchange of keys, which was presented as a secure extension to the ADS-B protocol named ADS-Bsec. However, due to the complexity of ATC operations within these two scenarios and the intricate nature of air space regulations, which can have different requirements from one country to the other, the implementation of such measures requires more in-depth considerations, especially in the presence of mixed air traffic in crowded urban areas, before ADS-Bsec could be consistently deployed. In this paper, we further explore these considerations by studying the impact on the performance, safety and security of Universal Access Transceiver (UAT) in a mixed airspace with and without ADS-Bsec. Consequently, we extend the previous work by exploring the complex and dynamic interactions between these components as a cyber-physical system handling ATC operations in different scenarios, and propose solutions to the issues we encountered during this process. Our ideas are illustrated using a simulated ATC case study and discussed though an analysis of its results.**

***Index Terms*—ADS-B; UAT, Cyber-physical systems; ATC**

I. Introduction

During recent years, Asynchronous Dependence and Surveillance Broadcast (ADS-B) has become one of the most promising protocols used for ATC surveillance operations [12] due to various reasons. First, it has a much better accuracy compared to traditional surveillance techniques, such as Primary Surveillance Radar (PSR) and Secondary Surveillance Radar (SSR), because it uses GPS to determine the aircraft position. Second, it offers a financially-viable option to deploy ADS-B stations covering areas that cannot accommodate radar stations, such as mountain ranges, sea, etc. In spite of these advantages, several critics of this technology [10] and [11] have warned against a serious flaw since ADS-B messages are sent in clear-text over the air without any security property being enforced, which makes it vulnerable to a plethora of attacks ranging from eavesdropping to message injection attacks.

In previous work [5] and [6], we proposed ADS-Bsec, a framework supporting secure ADS-B operations by providing ADS-B message authenticity and integrity by using metadata, based on keyed-Hash Message Authentication Code (HMAC), supported by a proper key exchange scheme that supports both air-to-ground and air-to-air operations. However, ever-growing complexities of airspace motivated the development of this paper. In particular, the issue of airspace organization for Unmanned Aircraft Systems (UAS) has been of great interest to the Air Traffic Management (ATM) community. The Unmanned Aircraft System Traffic Management (UTM) is an air traffic management ecosystem under development for autonomously controlled operations of Unmanned Aerial Systems (UAS) that aims to provide a set of ATM services in G airspace class [4] and a new kind of airspace compartmentalization for lower altitude operations (i.e., altitudes under 400 ft above ground level), providing operational safety and efficiency for UAS and manned aircraft operations [3].

There are several concerns with regards to UTM in comparison to ATM such as the segregation of airspace for UAS users while sharing it with other unmanned and manned aircrafts, helicopter operations and other aircrafts during landing and take-off procedures. Also, UTM requires higher technical capabilities and operational costs than ATM and knowing accurate positions of aircraft in the surrounding; especially for crowded urban areas. These differences motivate the need to investigate the suitability of ADS-B to provide the required level of safety, cyber-security and efficiency.

In this paper, we extend our previous work by studying the feasibility of using UAT version of ADS-B to support the complex nature of mixed airspace while guarantying the performance, safety and security. We support our ideas through a simulated case study, which is more realistic in its design than the ones in [1] and [2], and we discuss the results of our findings.

The paper is organized as follows. Section II presents the related work while section III describe how our ADS-Bsec

framework works. Section IV describes the design of our case study while section V discusses the results of the experiments we conducted.

II. Related Work

Guterres et al. [1] proposed to examine various operational scenarios and estimate the ADS-B performance and capacity in a mixed sUAS and NAS environment. Their focus was on combining radio signal transmission power with traffic density. That is, they started with 4 different variations of transmission power and combined with 4 different traffic intensity to generate 16 different scenarios. They calculated the probability of message decode for each scenario, and drew diverse conclusions about the overall system performance and claimed that balancing these two parameters would be sufficient to attain an acceptable demand on the UAT in areas of potentially high sUAS concentration while providing safety and utility to all aircraft. However no simulation was done here.

Matheou et al. [2] extended the work of Guterres et al. by validating the scenarios in a more realistic fashion (i.e. instead of just combining power and traffic intensity). They picked the first 12 of the 16 scenarios and implemented those in a simulator. The simulation itself added new components to the dual traffic intensity versus transmission power. For example, they had a mix of small UAS and aircrafts, with a percentage of pre-defined mode types. Their focus was also in assessing the communication performance, but their criteria was more than *probability of message decoding* used in [1]. They measured the probability of closing the communications link and of capturing a Message Start Opportunity (MSO) slots and completing the MAC layer process to fully send framed information data to the receiver, and provided a reasonable description of how their MATLAB simulation was setup.

However, both works did not consider the authentication of ADS-B messages, which is a significant issue in the UTM scenario. Different from the manned scenarios, UAS scenarios are likely to rely only on ADS-B devices to provide situational awareness and avoid collisions. In addition to ADS-B, the manned aircraft can leverage TCAS (Traffic Alert and Collision Avoidance System) [13] or other kinds of surveillance aircraft radar. As stated, the UTM airspace has more density than manned airspace, and the separation between aircraft is relatively small. Consequently, less forgiving of errors in positions or transmitter identities.

III. ADS-Bsec at a Glance

Our ADS-Bsec framework, as shown in Figure 1, provides the much needed integrity and authenticity to ADS-B messages which in turn supports secure air-to-air and air-to-ground communication between aircraft equipped with ADS-B IN on one hand and between aircraft and ground components on the other hand, respectively. At the sender side, we start by generating the coordinates of the trajectory of the flight in the Aircraft Prediction Module by using inputs from the flight path and from the Base of Aircraft Data (BADA). Then, the

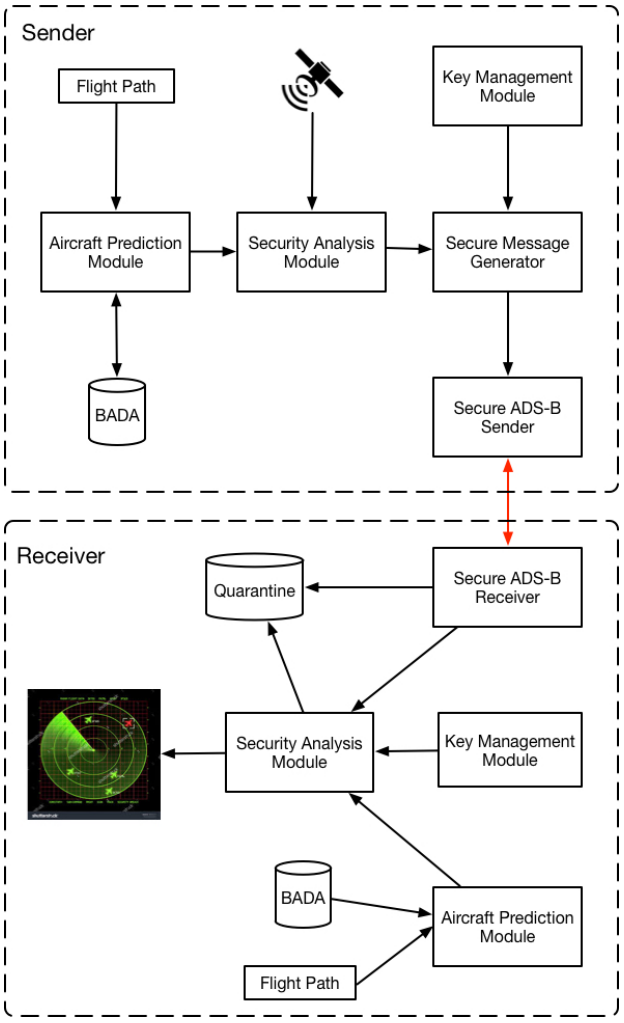

Fig. 1. Framework Overview

Security Analysis module continuously compares the predicted position along with the one obtained from GPS in order to detect any possible geo-locational data inconsistency or possible GPS spoofing attacks. But once the geo-locational data is sanitized, we pass it to the Secure Message Generator that generates the secure ADS-B message to be sent. As the secure ADS-B messages rely on the HMAC-based metadata, which is used instead of the Cyclic Redundancy Check (CRC) part of the ADS-B message, getting the adequate key is of utmost importance. In oder to do so, the Key Management Module continuously fetches the right key to deliver to the Secure Message Generator. The security metadata of each secure ADS-B message relies on two keys: one *individual* key used for ground-to-air communication with ground components and one *group* key used for air-to-air communication with neighboring aircraft.

Once the message is received it is passed for the Security Analysis Module that extracts the security metadata and validates it by first fetching the appropriate keys from the Key Management Module. Then, it computes the same metadata

based on the payload of the ADS-B message and compares it with the received one. During this process the Security Analysis Module also checks the dynamics of the decoded position. This check ensures that it falls within the expected range obtained from the Aircraft Prediction Module, constituting another check for keys compromises. If the position is valid it is displayed on the air traffic controller screen and related messages are stored in the quarantine for further analysis otherwise.

IV. The Case Study

A. Motivation

Previous version of ADS-Bsec was dedicated to 1090 ES version of ADS-B. Due to the complex nature of UTM, that version has a higher density than ATM while the separation between aerial vehicles are relatively small, this constituted a motivation for us to extend our framework to support UAT, which is widely used for sUAS, to provide the safety and security for mixed airspace while providing a good performance. Therefore, we designed a simulated ATC case study to describe the experiment design and the set of scenarios. The goal of this experiment is to show that our ADS-Bsec protocol can be extended to work in a UTM scenario by providing security and safety while preserving the performance of ADS-B.

B. Experiment Design

The design goals of this experiment are as follows:

- Evaluate the impact of encryption process used in ADS-Bsec inside the ADS-B performance index (air-to-air & air-to-ground update rate).
- Evaluate the impact the modification of ADS-BSec (i.e., switching the implementation from the one using 1090ES to the one using UAT) has in the ADS-B performance index.
- Understand the impact of more realistic UTM scenarios (manned and unmanned aircraft sharing some part of airspace) in the ADS-B performance index.

To achieve these goals, first, we defined some premises. To start, the scenario in [2] needs to be updated in order to became more realistic by using some UTM concepts, such as routes and non-random Kinetic movement. Second, the aircraft (manned and unmanned) needs to follow a simple collision avoidance algorithm to be able to evaluate if the encryption delay does not affect the safety condition of UAT. Third, the scenarios to be developed shall not consider vertical obstacles. In other words, we consider that there is no obstacle across the path used by the collision avoidance algorithm other than other aircraft. Fourth, the security is not evaluated, because previous work [5] addressed it.

C. Scenarios Description

To accomplish these goals, we developed three different scenarios that were designed taking into consideration the NASA airspace design in [3] as described in Figure 2

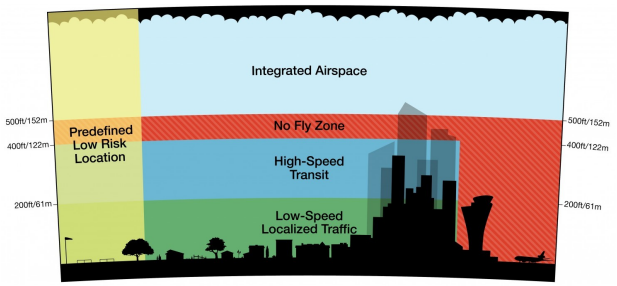

Fig. 2. NASA Airspace Design from [3]

The key features of the three scenarios are described as follows:

- Only UAS are considered.
- The UAS can have three different kind of applications: power line inspections, cargo delivery (such as Amazon) and surveillance.
- UAS are equipped with a collision avoidance sensors.
- The difference between the scenarios lies in the number of UASs that are used, which is 181, 323 and 1020, respectively for scenarios 1, 2 and 3.
- ADS-B works in mode In/Out.
- The scenarios are built over Sao-Paulo, Brazil.
- There are no air traffic control services for UAS.
- Scenarios includes zoning of urban town in small areas (for efficient and effective management of drone operations) and each zone will have retail areas designated to supply/delivering point in the city (like heliports) as described in Figure 3 from [7].
- Height of Take-off and Landing (ToLd: up to 60m (200ft) [7].

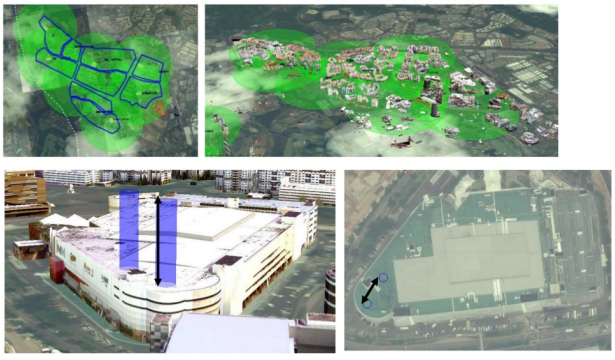

Fig. 3. NASA Airspace Design from [7]

D. Simulation

The Simulator, henceforth called SimATC, follows a Publish-Subscriber Message pattern-based architecture as shown in Figure 4.

First, in SimATC the *Channel* class acts as a broker and is responsible for relaying all the received updates from publishers to the corresponding subscribers. In addition, it has the

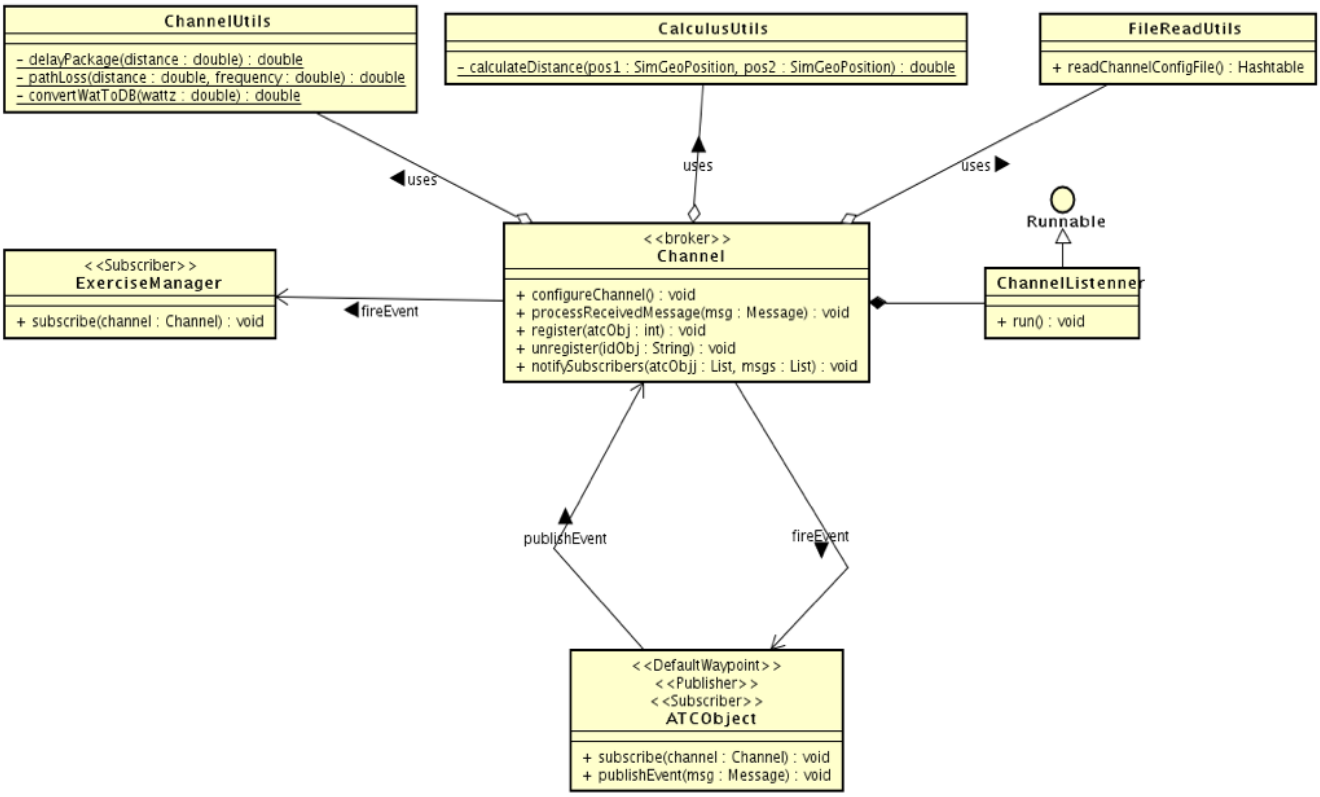

Fig. 4. Simulator Class Diagram

following features. (1) Processes position and order of the messages updates. (2) Determines the reachable subscribers based on the distance and/or the channel propagation capability before sending the update to them. (3) The Channel emulates the communication propagation delay, collisions and other channel access issues, padding the messages when applicable and by emulating any other required effect to increase the fidelity of the simulation.

To perform these tasks, the Channel class uses the *FileReadUtils* class that is responsible for parsing and extracting data from the configuration file, the *CalculusUtils* class responsible for measuring the distance between two geo-points defined in term of the triplet (longitude, latitude and altitude); and the *ChannelUtils* class that performs some miscellaneous measurements such as the delay, path loss as well as unit conversions.

The *ATCObject* entity is an abstract class whose derived concrete classes are *ATCController* and Aircraft as shown in Figure 5. The *ATCController* class extracts the position from the message sent from the aircraft, via the channel, in order to verify that the aircraft is following the planned flight plan and check for any potential conflict. If found, it sends an order message to the aircraft in question to resolve the conflict.

The aircraft is a more complex entity because it needs to perform the required calculations of the kinematics in order

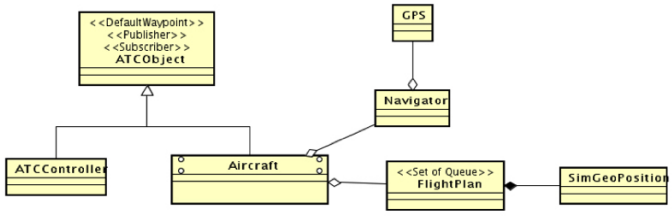

Fig. 5. ATCObject Model

to provide an accurate position. In addition, it periodically publishes its position, every second to other aircraft listening on the channel. More importantly, every aircraft that receives a location update from another aircraft needs to run the collision avoidance algorithm in order to avoid mid-air collision.

The *ExerciseManager* class is the main entity that defines the simulation clock and is the event handler of the simulation environment. Thus, all the messages that are sent through the channel are replicated to the *ExerciseManager* class and logged in a text file. In addition, the *ExcerciseManager* class shows the progress of the exercise to a user as shown in Figure 6. However in addition to *ExerciseManager* the channel also records various types of information, such as *timestamp* of reception or delivery of a message and t*ATCController* and

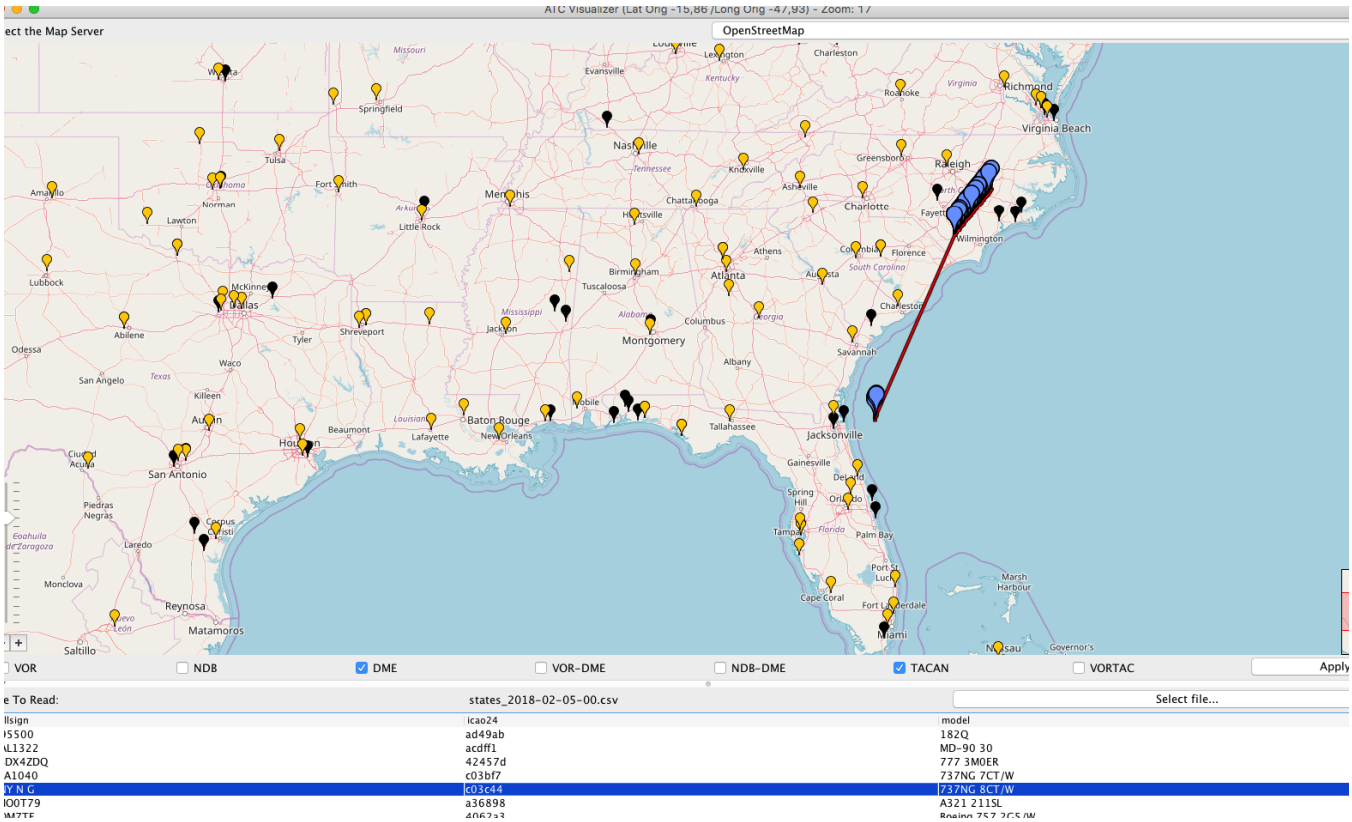

Fig. 6. Simulator Class Diagram

the *aircraft* also log the generated collision alerts.

Because of the paramount importance of safety and is one of the main goals of our ADS-Bsec framework, we developed a simple collision avoidance algorithm based on [9] as shown in Algorithm 1.

Algorithm 1: Collision Avoidance & Warning Generation Algorithm

```
1  Inputs: List of aircraft UAS_DB, double timestamp;
2  for i=0..UAS_DB.size() do
3      Aircraft id_i = AirDB.get(i);
4      for j=0..UAS_DB.size() do
5          if i!=j then
6              Aircraft id_j = AirDB.get(j);
7              Boolean canReach =
                 ChannelUtils.isReachable (id_i.getPosition(),
                 id_j.getPosition(), timstamp);
8              Boolean warnCollision =
                 ChannelUtils.checkCollision(id_i.getPosition(),
                 id_j.getPosition(), timestamp);
9              if warnCollision then
10                 Logger.logWarningCollision(timestamp,id_i,
                     id_j,canReach);
```

Line 1 of Algorithm 1 presents the input parameters, the list of aircraft as stored in the database and the timestamp of the collision warning. Lines 2-5 mark the start of the nested loop that iterates through the list of aircraft. Line 7 checks if each pair of aircraft at a given time can reach each other based on the horizontal separation defined at 50 meters and the vertical separation defined at 10 meters. Line 8 checks if a collision is likely based on the comparison of the computed horizontal and vertical distances with these thresholds. Lines 9-10 write the collision warning to the log in case its corresponding boolean variable evaluates to true.

In order to accurately evaluate the effect of using UAT instead of 1090ES in our ADS-Bsec framework, we modeled the end-to-end delay as the sum of the HMAC computation delay, the transmission delay and the HMAC verification delay. The HMAC computation delay is the time it takes the ADS-B sender to integrate the HMAC-based metadata in the ADS-B message. The transmission delay is the time it takes for an ADS-B message from source to destination. The HMAC verification delay is the time it takes the ADS-B receiver to verify the authenticity and integrity of the security metadata by comparing the received versus the computed HMAC digest.

V. Results

For each scenario, i.e. 181 UASs vs 323 UASs vs 1080 UASs, we run a Monte Carlo simulation of this end-to-end delay in function time as shown in Figure 7 aND the jitter in Figure 8.

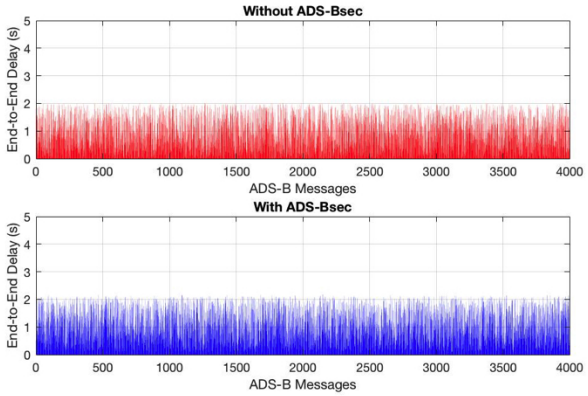

Fig. 7. End-to-End Delay Comparison

As shown in Figure 7, the added security has a minor impact on the end-to-end delay which supports our initial hypothesis that ADS-Bsec can be extended to support UAT variant of ADS-B as it was fully tested for 1090ES. Also, our findings indicate that the jitter values range between 4E-3 second and 2.1077 seconds with mean value of 0.6869 second. These results are encouraging as they validate the the soundness of our approach especially that the end-to-end delay is less than the 3 seconds threshold which is imposed for safe air-to-air updates. This would also contribute to the considerable reduction of collision warning as more reliable and precise position is provided by our ADS-Bsec framework.

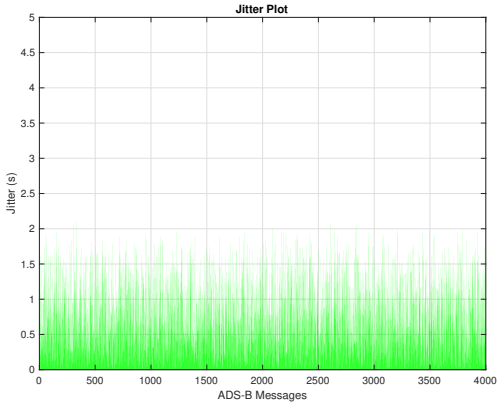

Fig. 8. Jitter Plot

VI. Conclusions

In this paper, we extended our previous work on ADS-Bsec validating its applicability in urban zones with emerging complex air traffic characteristics. As a solution, we developed a case study supported by a large scale simulation that is as close as it gets to real flight operated by UAS. Our findings support our initial claim that ADS-Bsec can be extended to support UAT for flights operated by UASs.

References

[1] M. Guterres, S. Jones, G. Orrell, and R. Strain. "ADS-B surveillance system performance with small UAS at low altitudes". In AIAA Information Systems-AIAA Infotech@ Aerospace, p. 1154. 2017.

[2] K.J. Matheou, R.D. Apaza, A.N. Downey, R.J. Kerczewski and J. Wang. "ADS-B Mixed SUAs and NASA System Capacity Analysis and DAA Performance". ICNS 2018, April 2018, Herndon, VA.

[3] P.H. Kopardekar. "Unmanned aerial system (UAS) traffic management (UTM): Enabling low-altitude airspace and UAS operations." (2014).

[4] ICAO's airspace classification scheme is defined in ICAO Annex 11: Air Traffic Services, Chapter 2, Section 2.6.

[5] Thabet Kacem, Duminda Wijesekera, Paulo Costa. "Key distribution scheme for aircraft equipped with secure ADS-B IN". The 20th IEEE Intelligent Transportation Systems (ITSC 2017), October 2017, Yokohama, Japan.

[6] Thabet Kacem, Duminda Wijesekera, Paulo Costa, Alexandre B Barreto. "Secure ADS-B framework ADS-Bsec". The 19th IEEE Intelligent Transportation Systems (ITSC 2016), November 2016, Rio di Janeiro, Brazil.

[7] Kin Huat Low. "A common framework with core boundaries for global harmonization. Framework for urban Traffic Management of Unmanned Aircraft System (uTM-UAS)". Presented in DRONE ENABLE ICAOs Unmanned Aircraft Systems (UAS) Industry Symposium (UAS2017), September 2017 @ ICAO HQ, Montreal, Canada.

[8] Haraldsdottir, Aslaug, et al. "ATM Operational Concepts and Technical Performance Requirements." New Concepts and Methods in Air Traffic Management. Springer, Berlin, Heidelberg, 2001. 63-74.

[9] 20-151B - Airworthiness Approval of Traffic Alert and Collision Avoidance Systems (TCAS II), Versions 7.0 & 7.1 and Associated Mode S Transponders, faa.gov, March 18, 2014, p. C1.

[10] Strohmeier, M., Lenders, V. and Martinovic, I., 2015. On the security of the automatic dependent surveillance-broadcast protocol. IEEE Communications Surveys & Tutorials, 17(2), pp.1066-1087.

[11] McCallie, D., Butts, J. and Mills, R., 2011. Security analysis of the ADS-B implementation in the next generation air transportation system. International Journal of Critical Infrastructure Protection, 4(2), pp.78-87.

[12] Federal Aviation Administration (FAA), Air Traffic Bulletin, Special Issue 2005-3, August 2005, Available at: www.faa.gov/air traffic/publications/bulletins/media/atb aug 05.pdf

[13] Kuchar, J.E. and Drumm, A.C., 2007. The traffic alert and collision avoidance system. Lincoln Laboratory Journal, 16(2), p.277.