Advances in Science, Technology and Engineering Systems Journal Vol. 5, No. 6, 1657-1665 (2020)

www.astesj.com

Special Issue on Multidisciplinary Innovation in Engineering Science & Technology

ASTESJ

ISSN: 2415-6698

A Novel Way to Design ADS-B using UML and TLA+ with Security as a Focus

Pranay Bhardwaj*, Carla Purdy, Nawar Obeidat

College of Engineering and Applied Science, University of Cincinnati, Cincinnati, OH 45220, USA

ARTICLE INFO

Article history:
Received: 01 September, 2020
Accepted: 04 December, 2020
Online: 28 December, 2020

Keywords:
ADS-B
TLA+
Security

ABSTRACT

Automatic Dependent Surveillance-Broadcast (ADS-B) is the future of aviation. It is a vast system that provides situational awareness for the aviator and regulator at a very low cost and does so with the aid of multiple disparate systems working closely together and communicating with one another. ADS-B uses the Global Navigation Satellite System (GNSS/ GPS) to locate elements. Weather information and ground-based information is also transmitted wirelessly. The system is designed to be open, unencrypted, and accessible to actors throughout the world. However, this leaves it open to attacks. The use of GNSS and other wireless technologies also carries over their security vulnerabilities into ADS-B. Certain issues have arisen due to both component-system failures and malicious attacks. Most obvious solutions impinge on the openness and transparency of the system. Past research has indicated that security must be built into a system design itself and cannot be retrofitted. We want to showcase such a design process for ADS-B. Our pathway to do so is to first create Universal Modeling Language (UML) diagrams to showcase security and safety issues and responses. These UML diagrams will then help us to model state and sequence diagrams. These will then be used to create a TLA+ model of one selected security methodology. We then run the TLC model checking on it to find loopholes and plug gaps in our scheme. We managed to create such models and prove deadlock-free running using only software tools. Our eventual goal is to develop a comprehensive formal specification for ADS-B model-creation and checking.

1. Introduction

This paper is an extension of work originally presented at the 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), Dallas, TX, USA [1].

Under the 2020 ADS-B mandate [2, 3], the FAA has designated that all commercial traffic and most private traffic must make itself compatible with ADS-B by hardware and software upgrades by the date of January 1. 2020.

ADS-B will replace 'hard' sites like primary/secondary surveillance radar (PSR/SSR) stations with satellite-based GNSS. The onus to report position is now on the aircraft. Frequencies of 1090 MHz and 978 MHz will be used for the actual information transmission. Aircraft will also beam down velocity, altitude, and a host of other data. In exchange, weather data will be beamed up to them. Aircraft will share position information with each other as well, to prevent conflicts.

ADS-B is a lot more economical than the current system. In 2007 the cost to monitor 200 nautical miles of air space was estimated to be $10-14 million using PSR, $6 million using SSR, and $380,000 for ADS-B [4]. Cost savings occur due to not having to set up and maintain expensive PSR/SSR RADAR stations plus increased coverage of formerly 'RADAR-dark' areas, resulting in more safety dividends, among other advantages.

The cost factor makes the choice look easy, but the security and safety weaknesses of ADS-B present a conundrum. Security issues are noted in [5-7], and an example of a deliberate system shutdown is given in [8].

Any attempts to gatekeep ADS-B go against its ethos. Security thus becomes a challenge. Attempts are now being made to introduce security measures into an already deployed ADS-B. But in [9], the authors have postulated that security must be considered at the design phase itself and any attempts to fit it in later will result in underestimation of the system capacity required to run security.

The scope of ADS-B is not restricted to 1090/978 MHz. In [10], the authors talk about a future where the Internet Protocol is the communications backbone for this system, even for voice

* Corresponding Author: Pranay Bhardwaj, Email: bhardwpy@mail.uc.edu

communication for aircraft flying. The radio has not been the sole point of communication for an aircraft for a long time. Now component subsystems of an aircraft independently 'talk' to maintenance and route planning departments of the operator via Aircraft Communications Addressing and Reporting System [11]. IP is also slated to take this communication load.

Following [9], the goal of this research is to rethink the ADS-B system design from a security viewpoint. We want to show what models such a hypothetical design process could have made, and what tools they could have used. A similar methodology is adopted by the authors in [12]. Here we are not recommending specific solutions for the numerous safety and security problems, nor are we devising a way to 'fix' ADS-B as it exists. We want to model how the different subsystems that makeup ADS-B interact with each other when under attack.

Our methodology is to pick a particular attack, pick specific solutions to that attack, and then first model the attack and response using use case diagrams. These will be used to create state and sequence diagrams to give a step-by-step description of the security response process to that attack. Once the sequence of events is clear, we can develop a TLA+ model of our system and run it through the TLC model checker to see if there are any deadlocks or failure points.

The specific attack scenarios and response tools are not ours, nor do we want to create any. We just want to observe how these response tools would look if they were all designed into the system, rather than retrofitted and to see if their in-built design into the system will cause any problems with any other system.

To put it simply, we do not want a situation where, for example, a cyberattack is successfully stopped by the security system but with an unintended result that airplanes are left without GNSS coverage. This could happen if the inter-relationships of competing cybersecurity tools are not well- understood at design time. We want to showcase that our methodology can be followed to model all of ADS-B, and perhaps other large systems in this manner at a low cost to spot deadlocks and bugs early in the design process.

So why do we want to use UML?

In [13], the author has used UML, the Unified Modeling Language [14], to model the Controller Area Network Bus (CANBUS). It has also been used to create security parameters [15,16]. UML focuses on whole 'objects' and each component may be a whole system unto itself. Thus hardware, software, and cyber-physical systems can all be modeled in UML.

Looking from a very high level, we can visualize the ADS-B as similar to the CANBUS. Both are unencrypted, both have various 'nodes' in the system, both rely on messages, both use broadcast protocols and both have no 'central brain.' Finally, security was not a consideration in the original design of either.

In ADS-B, the 'nodes' are the aircraft, the GNSS is a 'sensor' and the bus itself could be the pathways of the 978/1080 MHz frequencies. This is quite an interesting comparison and can be made between ADS-B and many other systems of various sizes and scales.

If ADS-B analogies can be found with the CANBUS, then methodologies to secure the CANBUS can theoretically also be proposed for ADS-B. As in [13], we will begin by modeling selected security techniques in UML and create use case diagrams.

Aviation is a heavily regulated sector and the basic requirement in all these systems↓ operational and security↓ is that they all work well together. Thus, the early introduction of security into the design is necessary because a late addition may introduce unintended security flaws. That is exactly the model we suggest in this paper- a redesigned system that does not stray too much from the current design and does not require major changes in protocol or infrastructure.

Our inspiration to use TLA+ is to get a workable model of how a potential 'built-in security' ADS-B system will look and behave. If successful, we will also be able to spot any failure points or illegal states in our models.

We envision that our paper can be used as a guidebook to model any discrete distributed system and test it for security loopholes at a very low cost. This will lead to the introduction of formal methods in civil aviation cybersecurity.

Formal methods have been used for many years in modeling localized systems [17], and we believe large systems consisting of geographically discrete subsystems can also be modeled this way. ADS-B uses independent systems for position, weather, and collision avoidance↓and each one of these is important to ensuring security and safety.

Our end goal is to create an overall super-design of the ADS-B system incorporating all the different security measures we talked about. In the future, we will attempt to create state and sequence charts for all of ADS-B. We will seek to validate the TLA+ based model checking and formal methods for complete ADS-B design.

In section 2, we discuss some weaknesses of ADS-B security and some well-known attacks. We then do the same for its largest sub-system, GNSS in section 3. In section 4 we create UML use-case diagrams to show how some selected methods can protect us from a cyberattack, and we also look at many other methods specified in the literature and select some for the next stage. In Section 5, we create a state diagram and sequence diagram of how ADS-B could respond to an attack using tools described in section 4. This is not a general view↓ just a specific attack scenario↓ and it feeds into the next section. In section 6, we will finally convert the state diagram to an actual TLA+ text specification/model. We will run it through the TLC model checker in section 6.1 to look for deadlocks and failures. We will then suggest how to apply our method more broadly to deal with the safety and security issues we have identified for the ADS-B system in the concluding remarks and the future work described in section 7.

2. Major Weaknesses of ADS-B

Because of its accessible design, ADS-B is missing even minimal security features [5]. This makes attacking easy and defense complicated. Some overarching vulnerabilities are:

- Physical layer: The 1090 MHz channel used for commercial aircraft ADS-B is severely congested, with up to 50% message loss rate measured at 174 miles [5]. This will only get worse with time, and we are in a very early phase yet.
- Unencrypted: Because ADS-B is unencrypted, the positions of aircraft can be tracked by anyone with homemade equipment and a free open-source application like flightradar24.com (FR24). Eavesdropping is the first step for many attacks [5]. FR24 details a step-by-step guide to making an ADS-B 'snooper' and getting real-time aircraft identification, position, and performance data.
- Unauthenticated: ADS-B has no authentication on messages↓no signatures, no handshake protocol. Aircraft identifiers, like the 24-bit ICAO code, are publicly available in a platform like FR24. Thus, there is no way to identify the source of a message [3]. If the constraint of powerful transmission equipment is fulfilled, a well-funded actor can theoretically beam fake data about an existing contact, or a fake contact itself, to air traffic control. Even worse, it can remove data as well.

Some common attacks that can happen in the above environment are:

- Ghost contacts: Two works on cybersecurity [18,19] mention the possibility of fake, or 'ghost' targets in the system. An attacker could use the openness of the system to download authentic aircraft data and then broadcast it over the frequency with sufficient power. This would appear as a legitimate contact on the ATCS screen. Without confirmation of GPS position, it would be impossible to catch such an attack. An example that combined eavesdropping with ghost injection is described. The appearance of multiple ghost aircraft can cause system and sensory overload and can cause pilots and ATC to make dangerous decisions [6].
- Hybrid attacks: Hybrids of various attacks can be made. In [6], the authors talk about an attack that 'deletes' legitimate data from the system and injects false data in its place.
- Legacy systems: It is not uncommon to find aircraft flying today that are 20-30 years old or more. This is especially true in developing countries, cargo airlines, and various armed forces. Legacy aircraft systems were not designed with an interconnected super system like the ADS-B in mind. In the past, each aircraft was an independent entity. Legacy systems have always been a security challenge when hooked up to large networks. A determined adversary with ample support↓like a state-backed actor↓can overcome all compatibility and air gap barriers to infect an 'old' system with a 'new' bug, as was seen in [20].
- Physical access: Physical isolation or 'air- gapping' that ensured the protection of many systems before will cease to exist. In the past, the output of a PSR/SSR was inaccessible to the public. Now, the detection system, position system, and the results therein are all readable on a home computer in real-time. Physical isolation of infrastructure has given way to an open, unencrypted, unauthenticated setup.

Risks of GNSS Use in Aviation

- Survivability: As of today, four nations possess the capability to destroy satellites in orbit the USA, Russia, the PRC, and India [21]. These 4 nations have also all developed their own global or regional GNSS (GPS, GLONASS, BEIDOU, NAVIC respectively). All these nations have also militarized their GNSS systems↓to guide bombs, missiles, warplanes, naval vessels [22] and even to plan large-formation land maneuvers↓as was seen in the 1990 gulf war. Law enforcement and coast guarding also use GNSS. However, this also makes the GNSS a ripe target in the event of great power warfare. Knocking out GNSS satellites using anti-satellite (ASAT) missiles could lead to a mass blackout of ADS-B position reporting for all countries. This could have disastrous unintended consequences for civil aviation within seconds. In a future PSR/SSR-less world, this could mean crippling the global flight infrastructure. Although the probability of this is low, the authors find this to be one of the most disturbing eventualities.
- Deliberate shutdown: Like any other large system, GNSS may be on reduced performance from time to time, as noted, e.g., in [23]. Besides, deliberate shutdowns may be conducted for various purposes, like military training. The degraded signal quality sent an aircraft into high-speed oscillations in 2016 [24]. As the dependence of aircraft on GNSS grows, any shutdown or breakdown will become a safety issue for civilian aircraft. Dependence of autopilot systems on GNSS to navigate means aircraft could go into abnormal flight regimes without even alerting the crew if a sudden 'quiet death' of GNSS occurred locally during normal flight.
- Jamming and spoofing: GNSS is a satellite-based wireless system, and like every wireless system ever made, GNSS is susceptible to jamming of signals and spoofing↓incorrect location reporting. The Russian Federation has already demonstrated a GNSS spoofing/ jamming capability precise enough to make a single human difficult to locate [25]. It can be devastating for unsuspecting civilian aircraft in an ADS-B based system and could very well be the model of a terror attack.

3. Use Case Diagrams for ADS-B Security

For such a large and distributed system as ADS-B, it must be understood that no single solution exists. In [7], the authors tested the viability of many solutions and found none without a loophole. This is a glaring after-effect of not keeping security paramount in the design process. Of course, we cannot talk about a cybersecurity model without talking about the actual 'weapons' that will make it possible. We will now make UML diagrams of how the various cybersecurity tools we are familiar with could be deployed, in isolation or in combination, to ensure protection. Figure 1 provides the convention we use in the use case diagrams that follow.

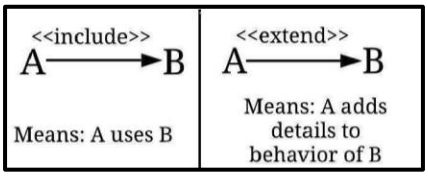


Figure 1: Convention for use case diagrams

In Figure 2, multilateration uses the time delay of signal arrival at multiple (>2) points to estimate the location of the transmitter [7]. Time Delay of Arrival (TDOA) of the signal is

calculated with the use of time stamps stamped on any message. Comparison of timestamps of transmission and reception gives a rough idea of the travel time, from which a rough distance can be calculated. If used at 3 receivers, a close X-Y-Z coordinate can be back-calculated. This is not very different from the SONAR equation.

If this back-calculated position is reasonably close to the reported GNSS coordinate of the transmitter, the allegiance of the message sender can be confirmed. Multilateration can be used to combat ghost aircraft, message modification, and man-in-the-middle attacks, and can provide a backup to GNSS breakdown in the absence of PSR.

Figure 3 is the most important here as it was the most complex one in our original paper. We have thus chosen to extend this diagram to the state diagram, sequence diagram, and later, to the TLA+ specification.

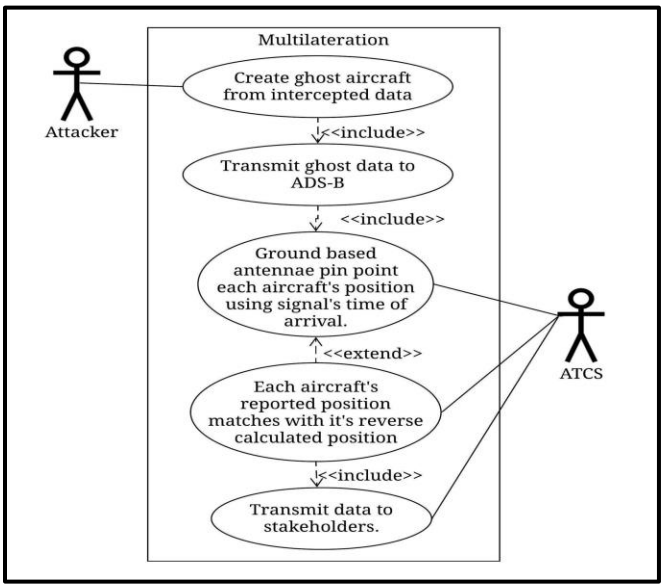


Figure 2: Multilateration and group verification can protect against ghost aircraft injections by matching the physical source of the signal to its reported one.

One of the weaknesses of using static ground posts as receivers for multilateration is that the system may be fooled by the attacker. If the location of these ground antennas is known, the attacker can tailor his transmission time stamps to present a 'false but accurate' picture to the multilateration-systems. A solution to this is to use dynamic receiver posts↓make the other, trusted aircraft in the airspace as multilateration, or triangulation points themselves!

Figure 3 shows multi-point multilateration. Group verification [7] is a kind of multilateration performed in the air. Aircraft already multilaterate each other using TDOA↓it is exactly how the Traffic Collision Avoidance System (TCAS) works. The equipment and algorithms are thus available on board already, and this is in line with our philosophy to cause minimal modification.

Each aircraft locates each member using TDOA of received ADS-B IN signals and estimates the others' positions. If the calculated position differs from the reported one, a 'suspect' airframe is identified, and group members move away from it. This can be successful only when 100% implementation of ADS-B is achieved. Location fixes from different sources are independent of GNSS. The fusion of this data can be an alternate method to locate aircraft. This will counter GNSS spoofing and GNSS jamming attacks.

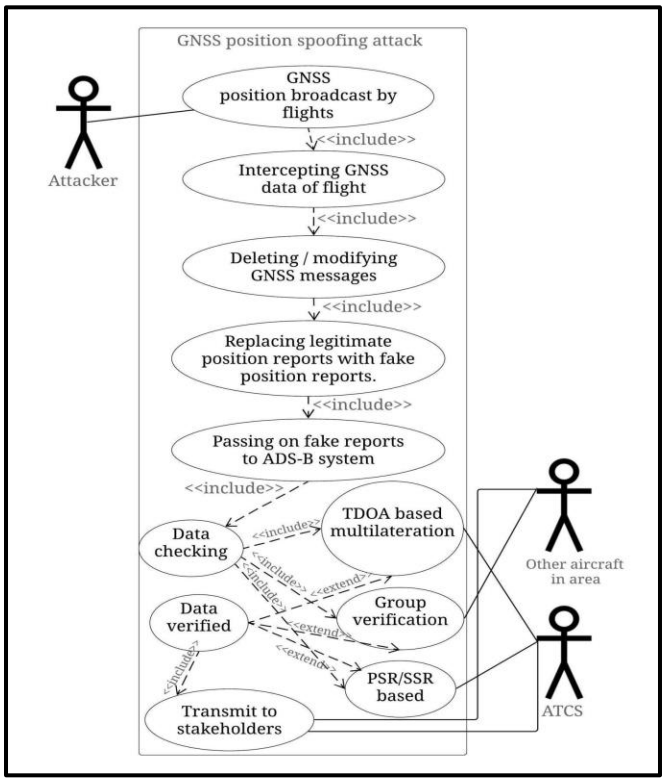


Figure 3: Multilateration may help with GNSS spoofing and unreliable position reports.

Other issues and solutions we will use in our overall model:

- Authentication of messages can be used to identify which messages are from what source. Authentication just confirms that the message is from who you think it to be. This directly allows us to trust or not trust it. Authentication of messages is the closest we come to a silver bullet solution for ADS-B as a single tool protects us from multiple issues↓Denial-of-Service (DOS) attacks and ghost aircraft attacks, just for starters. DOS attacks on airport or airline infrastructure are of concern as these are cheap to wage.
- Kalman filtering has many applications↓the ADS-B system can be 'taught' the typically correct values of airplane parameters [7]. According to their capabilities, scheduling, and operators, most aircraft will fly within a certain envelope of speed, heading, and locations usually. The Kalman filter can be rewarded whenever correct values are noticed, and not when unusual values are noticed. This will protect against ghost injection attacks and false data messages. The system will sense the departure from usual data trends and raise an alarm. If data trends change, it can be fed the new data trend and 'taught' any updates to the system this way. The only disadvantage is that the system is equally likely to be taught the 'wrong' way and thus become a 'bad child.'
- Hardware and software fingerprinting [26] have great potential to introduce economical authentication capability

without making major hardware or software changes. The authors in [26] provide a simple solution to 'fingerprint' the transmitters themselves. Every transmitter in the world is unique, just as every crystal is unique. Individual aircraft-borne ADS-B transmitters can be 'fingerprinted' either via unique hardware or through software properties. This can positively identify the transmitter and provide authentication.

- ADS-B as RADAR? A unique solution of using the ADS-B signal itself like a radar signal, by adding a random bi-phase modulation is proposed in [27]. This system can track other aircraft even with no GNSS. It can provide the much-needed reliability of radar at the low cost of ADS-B. This can counter issues of the survivability of GNSS. This agrees with [4], which talks of a fusion of GNSS and non- GNSS sources for locating aircraft. However, this will require changes to ADS-B transmitting recruitment and is thus low on our list of options.
- Weather RADAR for aircraft tracking: Weather RADAR is ubiquitous and big and small weather radar stations exist all over a developed country like the US. Since this RADAR can track small cloud formation, moisture, and bird/ locust flocks, can it be used to detect hard-skinned objects like airplane skin? Some research has been conducted in the use of weather RADAR to track aircraft and we believe that if it is properly developed, it can act as an emergency backup in the case of a major GNSS failure. For small general aviation (GA) aircraft at least, a method is suggested in [28]. This method can distinguish between small aircraft and large birds based on pressure waves created by aircraft propellers. In [29], the authors managed to detect aircraft with weather RADAR using some signal processing. More than 90% of aircraft in the USA are GA and most of these are propeller equipped [30]. Thus, a solution for GA aircraft as presented in [28] solves a huge chunk of aircraft tracking issues in case of GNSS blackout.
- Identification can be achieved by building a small ADS-B receiver like the ones FR24 uses, to receive aircraft data. Thus, in the event of a GNSS failure, weather RADAR can track the physical location of GA aircraft while an ADS-B receiver can identify them. One only must ignore the ADS-B location data in such a case. The use of a non-ADS-B source as a redundant backup is in the spirit of [7].
- TESLA protocol for encryption: In [31], the authors suggest a method that encrypts only the 24-bit ICAO aircraft identifier. It uses Timed Efficient Stream Loss-tolerant Authentication (TESLA) for authentication. It has performed well under testing without reducing security, performance, or affecting any major changes to ADS-B philosophy.

4. TLA+ conversion

TLA+ stands for 'Temporal Logic of Actions' [32]. It is a logical system or toolbox, developed by Leslie Lamport. which allows one to write formal specifications and to include scalable security protocols. The TLA+ specification can then serve as a formal 'model- checker' for any future realization of the ADS-B system and its components. It is not a classical programming 'language.' TLA+ is based on mathematics and uses set theory to model systems. Each system has states, and each state can be defined by a set of certain values. When these values change, so does the state. Any transition (step) is the change of one state to another or a change of one set of values to another set of values.

An entire system can be modeled as a superset of all possible state sets. We do not go too deep into the specifics of each state↓we are more concerned with the what rather than the how. Thus, abstraction and conciseness are a very important skill to master before modeling the system in TLA+. While economics and computer overhead would be major factors in practical systems design, we drop them when dealing with TLA+.

As a testimonial of its effectiveness, the application of TLA+ by Amazon revealed it to be excellent for bug detection [33,34]. Microsoft also used it to verify the design of a subsystem of the Alpha 21364 Microprocessor [17].

Our work here can be viewed as a guide on how to design large systems in the future↓by comparative analysis and extensive model-driven design.

ADS-B has many subsystems that make up the whole. GNSS is responsible for position reporting to aircraft. Aircraft then report their position to air traffic control.

Aircraft get weather and aerodrome information, and share position reposts with each other for collision avoidance. All this communication happens over the 978/1090 MHz spectrum. So not only are there several independent operations but also there are independent failure points. All these systems must be a part of any formal method examination of the ADS-B system. The specific solutions for responding to attack are not our own↓they are specified in section 4. What we are doing is modeling a security solution for the ADS-B system that will have safety and security included by design, and suggesting a procedure to design such a system and simulate it.

Several works on smart systems in TLA+ have been researched. In [35], the authors design a smart school system in TLA+. In [36], the authors model a sewerage system in UML and TLA+.

The authors in [35] start by making a UML specification, a sequence diagram, and a state diagram. These diagrams aid in making the sequence of events clear. The system's pathways, legal and illegal actions, and outcomes are modeled.

4.1. Why do we make sequence and state diagrams?

TLA+ defines every system as a sum of all the states it can exist in. A system can have multiple states, and there will be a defined 'step', or a sequence of events, to go from one state to another. Thus, we first develop a sequence diagram, which shows all the steps between states, and a state diagram which shows the states themselves. Only then can we draw a true TLA+ model.

Let us take the below example of applying methodologies of [35] and [36] to the ghost aircraft and GNSS spoofing issues as noted in Figure 3.

4.2. ADS-B response to Ghost aircraft and GNSS spoofing

In sections 2 and 4, we talked about security issues caused if an aircraft is either a fake contact or is reporting untruthful position information about itself to ADS-B, via GNSS spoofing. Since ADS-B envisions doing away with PSR/SSR, we had

examined some other backup systems that can physically locate an aircraft if GNSS reports are unreliable. The UML diagram for this was Figure 3. Let us now look at a sequence diagram for the situation, in Figure 4.

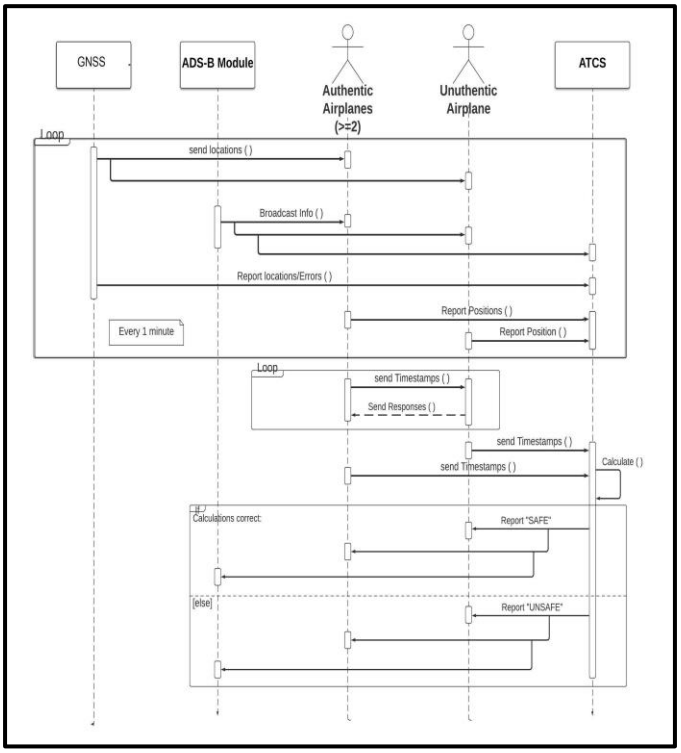


Figure 4: Sequence diagram: Reverse calculation of the aircraft's true position can be done using time-delay analysis of transmitted signals

GNSS reports aircraft position to airplanes, which includes both the 'good' and 'bad' airplane. All aircraft will then report their position to ATCS. A time delay analysis of the signals from all aircraft will be performed, and this will be used to get a location fix of the aircraft in physical space. Aircraft will also be able to perform multilateration and triangulation on each other in airspace↓within a '4-ship-cell' for example↓and get location fixes for each other's locations.

A comparison of the back-calculated position data with the GNSS reported position data will point out which contacts are reporting true data and which are not. Thus, aircraft can be labeled 'SAFE' or 'UNSAFE.' This information can then be provided to other aircraft in the area, and ATCS. Note here that 'ADS-B' stands for the system, but practically it may mean a supervisory body, like the FAA Technical Operations Services, or any other service which needs to be notified of security incidents. Let us now look at a state diagram of the same function in Figure 5.

This specifies an interval of 1 minute after which the whole cycle is repeated. This is arbitrary and will practically be a decision governed by radio traffic and economic factors. Since aircraft move fast and move in all 3 dimensions, this interval cannot be too long. Also, when we say 'ATCS', we mean a local or regional airspace manager. We do not envision a global central ATCS system in this paper.

Let us now use the 3 inputs, Figures 3,4, and 5 to start making the TLA+ formal specification for the above response.

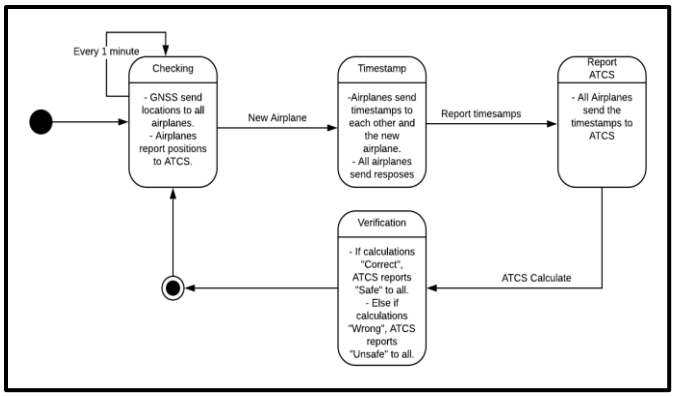
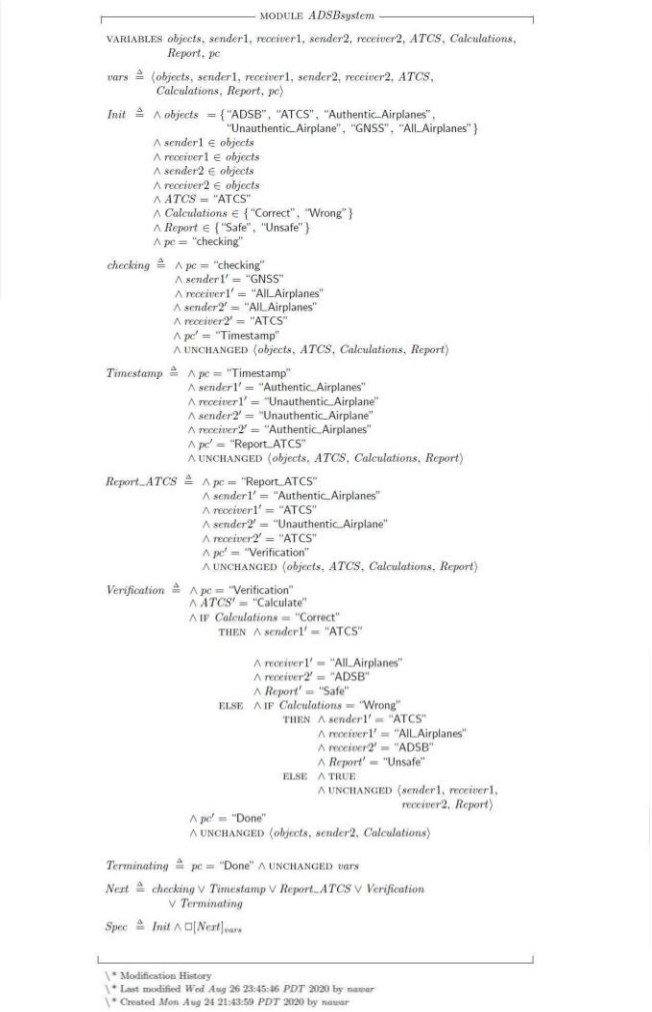


Figure 5: State Diagram of the system under Figure 4 conditions.

5. TLA+ Specification

We present now, the TLA+ specification for the attack response model sequenced in Figure 4 and stated in Figure 5. A block by block explanation follows.



MODULE *ADSBsystem*

VARIABLES *objects*, *sender1*, *receiver1*, *sender2*, *receiver2*, *ATCS*, *Calculations*, *Report*, *pc*

$vars \triangleq \langle objects, sender1, receiver1, sender2, receiver2, ATCS, Calculations, Report, pc \rangle$

$Init \triangleq \land objects = \{\text{"ADSB"}, \text{"ATCS"}, \text{"Authentic_Airplanes"}, \text{"Unauthentic_Airplane"}, \text{"GNSS"}, \text{"All_Airplanes"}\}$
$\land sender1 \in objects$
$\land receiver1 \in objects$
$\land sender2 \in objects$
$\land receiver2 \in objects$
$\land ATCS = \text{"ATCS"}$
$\land Calculations \in \{\text{"Correct"}, \text{"Wrong"}\}$
$\land Report \in \{\text{"Safe"}, \text{"Unsafe"}\}$
$\land pc = \text{"checking"}$

$checking \triangleq \land pc = \text{"checking"}$
$\land sender1' = \text{"GNSS"}$
$\land receiver1' = \text{"All_Airplanes"}$
$\land sender2' = \text{"All_Airplanes"}$
$\land receiver2' = \text{"ATCS"}$
$\land pc' = \text{"Timestamp"}$
$\land \text{UNCHANGED } \langle objects, ATCS, Calculations, Report \rangle$

$Timestamp \triangleq \land pc = \text{"Timestamp"}$
$\land sender1' = \text{"Authentic_Airplanes"}$
$\land receiver1' = \text{"Unauthentic_Airplane"}$
$\land sender2' = \text{"Unauthentic_Airplane"}$
$\land receiver2' = \text{"Authentic_Airplanes"}$
$\land pc' = \text{"Report_ATCS"}$
$\land \text{UNCHANGED } \langle objects, ATCS, Calculations, Report \rangle$

$Report\_ATCS \triangleq \land pc = \text{"Report_ATCS"}$
$\land sender1' = \text{"Authentic_Airplanes"}$
$\land receiver1' = \text{"ATCS"}$
$\land sender2' = \text{"Unauthentic_Airplane"}$
$\land receiver2' = \text{"ATCS"}$
$\land pc' = \text{"Verification"}$
$\land \text{UNCHANGED } \langle objects, ATCS, Calculations, Report \rangle$

$Verification \triangleq \land pc = \text{"Verification"}$
$\land ATCS' = \text{"Calculate"}$
$\land \text{IF } Calculations = \text{"Correct"}$
$\text{THEN } \land sender1' = \text{"ATCS"}$
$\land receiver1' = \text{"All_Airplanes"}$
$\land receiver2' = \text{"ADSB"}$
$\land Report' = \text{"Safe"}$
$\text{ELSE } \land \text{IF } Calculations = \text{"Wrong"}$
$\text{THEN } \land sender1' = \text{"ATCS"}$
$\land receiver1' = \text{"All_Airplanes"}$
$\land receiver2' = \text{"ADSB"}$
$\land Report' = \text{"Unsafe"}$
$\text{ELSE } \land \text{TRUE}$
$\land \text{UNCHANGED } \langle sender1, receiver1, receiver2, Report \rangle$
$\land pc' = \text{"Done"}$
$\land \text{UNCHANGED } \langle objects, sender2, Calculations \rangle$

$Terminating \triangleq pc = \text{"Done"} \land \text{UNCHANGED } vars$

$Next \triangleq checking \lor Timestamp \lor Report\_ATCS \lor Verification \lor Terminating$

$Spec \triangleq Init \land \Box[Next]_{vars}$

* Modification History
* Last modified *Wed Aug 26 23:45:46 PDT 2020* by *nawar*
* Created *Mon Aug 24 21:43:59 PDT 2020* by *nawar*

Figure 6: TLA+ model specification in literal form.

In the module, the keyword "VARIABLES" represents the variables in this module where "vars" is the tuple of variables:

<objects, sender1, receiver1, sender2, receiver2, ATCS, Calculations, Report, pc >

In every TLA+ module, there is an "Init" and a "Next" state. In the Init state, the variables get their initial values, while the next state represents all possible states that come next. In this module, the Next state represents:

Next == checking \/ Timestamp \/ Report_ATCS \/ Verification\/ Terminating

This includes 5 different states. The 'Checking' state occurs when the GNSS sends positioning data to all the airplanes, and then all the airplanes report their locations to ATCS.

In the group verification strategy noted in Figure 3, airplanes were using timestamps on exchanged signals to triangulate each other using time delay of transmission. The 'Timestamp' state represents this communication between airplanes. When a new airplane that possibly might be an attacker appears, it must communicate with other authentic airplanes (even if just for TCAS) and exchange timestamps.

In the 'Report_ATCS' state, all airplanes also report to ATCS with timestamps for routine operations↓and these can be used for position verification from a ground point of view.

The system must then enter the 'verification' state when ATCS does the time delay and radio directional calculations on these signals and compares them to the first GNSS location reports received from all airplanes. Based on these calculations the ATCS will identify which airplanes are reporting their position faithfully and which are not. It will then broadcast the result with a safe/unsafe report to everyone in the airspace.

The termination state allows infinite stuttering to prevent deadlock on the termination (when pc = "Done").

The statement:

Spec == Init /\ [][Next]_vars

represents the specification of the whole system, where Init means the starting state and []Next_vars means Next state must be true for the entire behavior with the option of keeping all variables unchanged.

5.1. TLC Result

The TLC model checker is a tool to validate the TLA+ module and check it for any errors. We run the TLC model checker for our TLA+ ADS-B module and it shows that the module is valid with no errors as well as no deadlock. Also, as appears in the figure below, the TLC model checker shows how many times each state was visited in the module, and what values changed to arrive at the next state.

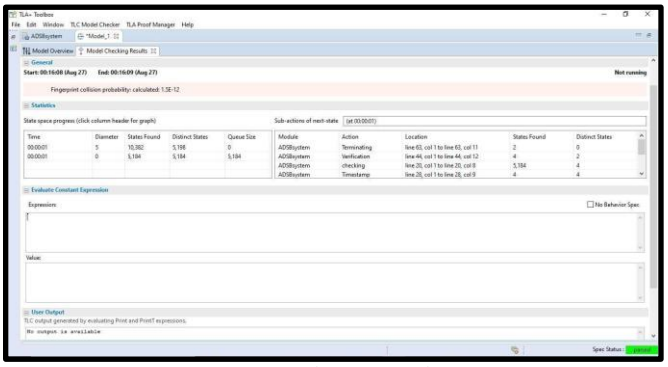


Figure 7: TLC Model check for validity of TLA+ spec in Figure 6.

6. Conclusion and Future Work

At this stage, our model for Figure 3 for verifying physical aircraft locations to combat ghost aircraft and incorrect position reporting issues shows no deadlock. We see that a security system using multilateration and time delay of arrival can identify which aircraft are ghosts and which are not, without getting into a deadlock for resources with the ADS-B system itself.

What this leads us to conclude is that this multi-level modeling of use case-> state diagram-> sequence diagram-> TLA+ specification-> TLC model check can be used to model a large and discrete system like ADS-B to see how it would function and what it would fail at. This is what TLC does it goes through iterations and finds failure points. It throws different numbers at the problem and sees what 'sticks to the wall,' what causes the system to break. We can conclude, at least on this level, that our procedure to model the ADS-B this way and the ability to test the model is valid.

It is always advisable to design security into a system from the beginning of the design process. We hope our method of using UML + state diagrams + sequence diagrams + TLA+ can not only provide the reader with a stable, practical ADS-B model that is secure and realistic but also act as a model of how to design security into other discrete systems in the future.

At the beginning of this paper, we compared the vast ADS-B to a small CANBUS. The roadmap we used here has already been used to formally model smart schools and sewerage systems. When a certain method is successfully used to design various systems of different sizes and scope successfully, it has the potential to develop into a standard. We hope other authors can expand on our work. Our diagrams can be used as an example of how to do this. They also provide a means to examine multiple solutions for strengths, weaknesses, and costs, before any detailed design or implementation takes place.

*6.1. Future work- **An Overall ADS-B** Model in TLA+*

The only weakness of our model is that it is not detailed enough. A true ADS-B system model with all subsystems and security systems and multiple attackers will be much larger in scope. While we model only one attack above, a real-world model will have all attacks and all security systems integrated and will run them in different ways and see when and where two security systems 'collide.'

And that is our future goal.

Following our methodology, the first step towards envisioning the entirety of ADS-B in TLA+ would start with a diagrammatic representation of what we want, as in Figure 8. We believe this will be an extensive task requiring more hands and time than we have right now. We can, however, think about what such a system would look like.

We make some basic assumptions.

No true communication can take place as long as the lines of communication remain unverified. We must find a way to secure the ADS-B physical layer without compromising on openness.

A lot of solutions impinge on trusting information systems like GNSS. Before we trust any data, we must know it came from the right source. Thus, a periodic 'wellness check' on vital infrastructures like transmission systems, radio frequencies, and GNSS should be the second step.

The third step will utilize the data gleaned from steps 1 and 2 to create a model of the system. We would get an idea of how safe or unsafe it is and would be able to deploy mitigation measures accordingly.

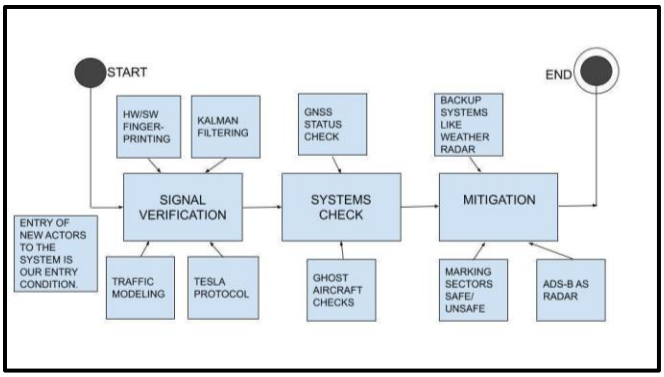


Figure 8: What an overall security solution for ADS-B might look like.

Once again, this would most probably happen on a local level. However, in the spirit of TLA+, we concern ourselves more with the 'what' than the 'how' and do not get too deep into details.

On a higher level, an overall ADS-B specification will consist of multiple figures like Figures 3,4,5, and 6, created for each security challenge, coming together and coalescing into one super-state diagram. This is why we see this paper as a stepping stone to achieving a formally designed ADS-B specification and showing the proper steps to carry out this process.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

All the use case diagrams in section 4 were made using chart-making tools at www.lucidchart.com and the authors would like to thank the website.
The authors would also like to thank Mr. Leslie Lamport for the TLA+ tool which is a fundamental part of this paper.
The authors would also like to thank the University of Cincinnati for its support.

7. References

[1] P. Bhardwaj, C. Purdy, "System design methodologies for safety and security of future wireless technologies in aviation," in 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), 235-238, 2019, doi: 10.1109/MWSCAS.2019.8885347.

[2] FEDERAL AVIATION ADMINISTRATION, May 2010, Federal regulation 14 CFR 91.225S, [Online], Available: http://www.ecfr.gov/cgi-bin/text-idx?node=14:2.0.1.3.10#_top.

[3] FEDERAL AVIATION ADMINISTRATION, May 2010, Federal regulation 14 CFR 91.227, [Online], available: http://www.ecfr.gov/cgi-bin/text-idx?node=14:2.0.1.3.10#se14.2.91_1225.

[4] International Civil Aviation Organization, Guidance material on comparison of surveillance technologies (Ed. 1), [Online], Available: http://www.icao.int/APAC/Documents/edocs/cns/gmst_technology.pdf.

[5] M. Strohmeier, V. Lenders, I. Martinovic, "On the security of the automatic dependent surveillance-broadcast protocol," IEEE Communications Surveys and Tutorials, 2015, doi:10.1109/COMST.2014.2365951.

[6] M. Schäfer, V. Lenders, I. Martinovic, "Experimental analysis of attacks on next generation air traffic communication," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013, doi:10.1007/978-3-642-38980-1_16.

[7] M. Riahi Manesh, 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, **19**, 16-31, 2017, doi:10.1016/j.ijcip.2017.10.002.

[8] FAA Safety Team, 2019, Team notice NOTC8274, [Online], Available: http://www.faasafety.gov/SPANS/noticeView.aspx?nid=8274.

[9] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, S. Ravi, "Security as a new dimension in embedded system design," in Proceedings - Design Automation Conference, 2004, doi:10.1145/996566.996771.

[10] K. Sampigethaya, R. Poovendran, S. Shetty, T. Davis, C. Royalty, "Future E-enabled aircraft communications and security: The next 20 years and beyond," Proceedings of the IEEE, 2011, doi:10.1109/JPROC.2011.2162209.

[11] ICAO International Communications Group, April 2006, Introduction to ACARS messaging services as implemented via Iridium satellite link, http://www.icao.int/safety/acp/inactive%20working%20groups%20library/acp-wg-m-iridium-7/ird-swg07-wp08%20-%20acars%20app%20note.pdf.

[12] C. W. Lin, A. G. Vincentelli, Security-aware design for cyber-physical systems, Book, Springer, 2017, doi: 10.1007/978-3-319-51328-7.

[13] G. Kalakota, Hierarchical partition based design approach for security of CAN bus based automobile embedded system, Electronic Thesis or Dissertation, University of Cincinnati, 2018.

[14] Object Management Group, 2005, Introduction to OMG's Unified Modeling Language (UML®), http://www.uml.org/what-is-uml.htm.

[15] J. Vidal, F. De Lamotte, G. Gogniat, P. Soulard, J.P. Diguet, "A co-design approach for embedded system modeling and code generation with UML and MARTE," in Proceedings -Design, Automation and Test in Europe, DATE, 2009, doi:10.1109/date.2009.5090662.

[16] J. Jürjens, P. Shabalin, "Tools for secure systems development with UML," in International Journal on Software Tools for Technology Transfer, 2007, doi:10.1007/s10009-007-0048-8.

[17] S. Tasiran, Y. Yu, B.Batson, "Using a formal specification and a model checker to monitor and direct simulation," Design Automation Conference, 2003, doi:10.1109/dac.2003.1219024.

[18] Darlene Storm, August 2012, "Curious hackers inject ghost airplanes into radar, track celebrities' flights," Computerworld, [Online], Available: http://www.computerworld.com/article/2472455/curious-hackers-inject-ghost-airplanes-into-radar--track-celebrities--flights.html.

[19] A. Costin and A. Francillon, July 2012, "Ghost in the air (traffic): on insecurity of ADS-B protocol and practical attacks on ADS-B devices," [Online], Available: http://www.researchgate.net/publication/267557712.

[20] S. Collins, S. McCombie, "Stuxnet: the emergence of a new cyber weapon and its implications," Journal of Policing, Intelligence and Counter Terrorism, 2012, doi:10.1080/18335330.2012.653198.

[21] Niall Firth, MIT Technology Review, June 2019, "How to fight a war in space (and get away with it),"http://www.technologyreview.com/2019/06/26/725/satellite-space-wars/.

[22] Defense Intelligence Agency, January 2019, "Challenges to Security in Space,"http://www.dia.mil/Portals/27/Documents/News/Military%20Power%20Publications/Space_Threat_V14_020119_sm.pdf.

[23] European GNSS Service Center, European Global Navigation Satellite Systems Agency, 23 November 2017, Notice advisory to Galileo users (NAGU) 2017045," [Online], Available: http://www.gsc-europa.eu/notice-advisory-to-galileo-users-nagu-2017045.

[24] Federal Aviation Administration, 2016, "EMB-300 Phenom yaw damper failure due to unreliable or unavailable GPS signal," [Online], available: http://www.faa.gov/documentLibrary/media/Notice/GENOT_7110_711_EMB-300.pdf.

[25] C4ADS innovation for peace, 2019, "Above us Only Stars- Exposing GPS Spoofing in Russia and Syria," [Online], Available: http://www.c4reports.org/aboveusonlystars.

[26] M. Strohmeier, I. Martinovic, "On passive data link layer fingerprinting of aircraft transponders," in CPS-SPC 2015 - Proceedings of the 1st ACM Workshop on Cyber-Physical Systems-Security and/or Privacy, co-located with CCS 2015, 2015, doi:10.1145/2808705.2808712.

[27] M.S. Huang, R.M. Narayanan, Y. Zhang, A. Feinberg, "Tracking of noncooperative airborne targets using ADS-B signal and radar sensing," International Journal of Aerospace Engineering, 2013, doi:10.1155/2013/521630.
[28] S. Bachmann, V. DeBrunner, D. Zrnic, "Detection of small aircraft with doppler weather radar," in 2007 IEEE/SP 14th Workshop on Statistical Signal Processing, 443-447, 2007, doi: 10.1109/SSP.2007.4301297.
[29] S. Rzewuski, K. Kulpa, A. Gromek, "Airborne targets detection using weather radar," in 2015 Signal Processing Symposium, SPSympo 2015, 2015, doi:10.1109/SPS.2015.7168305.
[30] Aircraft Owners and Pilots Association, 2019, "2019 State of general aviation,"http://download.aopa.org/hr/Report_on_General_Aviation_Trends.pdf.
[31] H. Yang, Q. Zhou, M. Yao, R. Lu, H. Li, X. Zhang, "A practical and compatible cryptographic solution to ADS-B security," IEEE Internet of Things Journal, 2019, doi:10.1109/JIOT.2018.2882633.
[32] Leslie Lamport, 4 September 2018, "A High-Level View of TLA+," [Online], Available: http://lamport.azurewebsites.net/tla/high-level-view.html.
[33] C. Newcombe, T. Rath, F. Zhang, B. Munteanu, M. Brooker, M. Deardeuff, "How amazon web services uses formal methods," Communications of the ACM, 2015, doi:10.1145/2699417.
[34] C. Newcombe, "Why Amazon chose TLA+," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, doi:10.1007/978-3-662-43652-3_3.
[35] N.H. Obeidat, C. Purdy, "Modeling a smart school building system using UML and TLA+," in Proceedings - 3rd International Conference on Information and Computer Technologies, ICICT 2020, 2020, doi:10.1109/ICICT50521.2020.00028.
[36] S. Latif, A. Rehman, N.A. Zafar, "Modeling of sewerage system linking UML, automata and TLA+," in 2018 International Conference on Computing, Electronic and Electrical Engineering, ICE Cube 2018, 2019, doi:10.1109/ICECUBE.2018.8610971.