ÉCOLE POLYTECHNIQUE DE MONTRÉAL
affiliée à l'Université de Montréal

Détection d'attaques informatiques sophistiquées contre les communications ADS-B en aviation

JEAN-YVES DE MICELI
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Décembre 2020

ÉCOLE POLYTECHNIQUE DE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Détection d'attaques informatiques sophistiquées contre les communications ADS-B en aviation

présenté par **Jean-Yves DE MICELI**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Giovanni BELTRAME, président
José FERNANDEZ, membre et directeur de recherche
Tarek OULD BACHIR, membre

DÉDICACE

À Adhémar, Aliénor, Camille, Charles, Marie-Jeanne, Romane,
pour votre soutien et votre précieuse amitié,
À mes anciens chefs, à ceux de la 1re Val de Seine et du Clan de la Brie,
pour m'avoir fait découvrir le sens du service et du don de soi,
À Mme Autran, M. Bouju, M. Fabresse, M. Rivière,
pour votre passion et la curiosité que vous m'avez transmis,
À tous ceux qui m'ont accompagné et fait grandir durant ces années,
Comment vous dire merci ?...

REMERCIEMENTS

Je tiens à remercier tous ceux qui m'ont soutenu durant ces années de maîtrise et qui ont permis à ce projet de recherche d'aboutir. Mes premières pensées sont pour le professeur José Fernandez, mon directeur de recherche, qui m'a fait découvrir ce monde incroyable de la sécurité informatique en aéronautique. Ses précieux conseils en tant que pilote, ses réunions de suivi et son soutien m'ont aidé à avancer dans la bonne direction. Un immense merci aussi à Nader Ammari et Christopher Neal pour ces nombreuses heures passionnantes d'échange et de travail sur des questions de modélisation et d'implémentation. Le laboratoire est vraiment constitué de perles, et il serait trop injuste d'oublier Militza Jean qui, tout en restant discrète, a veillé à ce que ma recherche se passe pour le mieux en prenant à bras-le-corps les problèmes administratifs pouvant surgir. Merci à Marielba Urdaneta pour sa patience et ses commentaires précieux lors de la relecture de ce mémoire. Merci à tous les membres du laboratoire pour vos échanges, pour ces moments de joie et de repos, ces discussions du quotidien qui sont banales mais tellement essentielles. Je tiens aussi à remercier Hans Obas pour son expertise apportée dans le domaine de l'aviation, son écoute et sa grande humanité.

Un travail de cette ampleur n'aurait jamais été possible sans une vie épanouie. Et là, mon cœur se tourne vers la Frat', lieu de vie incroyable, les jeunes de la paroisse St Jean-Baptiste et la troupe de la 38e, sans oublier bien sûr la Ferme, petit coin de paradis, et les jeunes des quatre coins du Québec rencontrés chez Charles et Sarah. Merci de m'avoir accepté dans votre «gang». Deux années à rire, à faire des randonnées, à former de belles amitiés et vivre de merveilleux moments.

Enfin, un immense merci pour tous ceux qui m'ont soutenu depuis la France. Par vos lettres, mails et messages, vous m'avez donné la force de continuer. Merci pour ma famille qui m'a vu discrètement traverser le monde pour terminer mes études et qui n'a pas arrêté de penser à moi. Je ne l'ai pas souvent montré, mais sachez que c'était réciproque.

RÉSUMÉ

Le contrôle aérien dispose de plusieurs types de radars pour détecter des avions et assurer la sécurité des passagers et équipages dans le ciel. Depuis le début des années 2000, un protocole de communication, l'*Automatic Dependent Surveillance-Broadcast* (ADS-B), a connu un véritable essor au point d'éventuellement supplanter les radars et de venir les remplacer dans certaines zones. Avec cette technologie, ce sont les avions eux-mêmes qui transmettent en continu leurs données de position. De plus, une constellation de satellites est déployée afin de recevoir ces messages et de permettre une détection des avions sur l'intégralité du globe terrestre.

Si cette technologie présente beaucoup d'intérêt par la qualité des informations transmises et son faible coût, elle possède des lacunes qui sont dangereuses du point de vue de la cybersécurité. En effet, les messages ADS-B sont envoyés en clair et ne possèdent pas de mécanismes d'authentification. Il est donc possible pour une personne ayant pris connaissance du protocole de pouvoir perturber le travail des contrôleurs aériens et impacter négativement le trafic aérien en injectant de faux paquets par une radio logicielle. Cela cause un problème de sécurité, car en cas d'attaque les contrôleurs ne pourraient pas mener correctement leur mission et les pilotes peuvent être amenés à effectuer des manœuvres dangereuses en suivant les indications du *Traffic Alert and Collision Avoidance System* (TCAS), un système de prévention de collision, qui utilise l'ADS-B.

Afin de corriger ces vulnérabilités, une grande partie des chercheurs se concentrent sur les différents moyens d'intégrer de la cryptographie dans le protocole ADS-B, ce qui nécessitera une mise à jour. Or, les processus de certification en aéronautique mettent généralement plusieurs années à aboutir. C'est pourquoi notre travail s'est penché sur le développement et l'évaluation d'un outil de détection d'attaques ADS-B.

L'idée directrice de notre recherche est l'utilisation d'ontologies pour développer un système expert utilisant des règles de détection basées sur des concepts abstraits de haut niveau. Le système expert nécessite d'abord un modèle ontologique qui représente l'environnement du contrôle aérien. C'est lui qui donne sens aux données enregistrées. Ensuite, pour établir les règles de détection, il a fallu identifier les principales menaces pesant sur le trafic aérien afin d'établir des scénarios d'attaques probables. À partir des menaces identifiées, nous avons créé quatre types de règles de détection, appelées logiques, qui sont : la logique d'identification, la logique des origines, la logique des radars et la logique de vol. Nous obtenons ainsi un système expert, ATC-Sense, composé d'une base de données ontologique et d'un raisonneur

qui applique les règles de détection pour établir la véracité des paquets ADS-B. Pour tester la performance de notre système expert pour la détection d'attaques sophistiquées contre les communications ADS-B, nous avons développé la plateforme ATC-Emu. Elle émule l'environnement du trafic aérien avec le vol des avions, les différents radars et le logiciel utilisé par les contrôleurs aériens. ATC-Sense vient s'y greffer et reçoit toutes les informations provenant des radars. Enfin, un module d'attaque a été ajouté pour simuler les attaques qui pourraient être perpétrées sur les communications ADS-B.

Des tests ont été effectués sur ces différentes logiques de détection qui donnent des résultats probants en termes de rapidité d'exécution et de taux de détection. De plus, une comparaison de notre solution a été faite avec une autre solution de détection qui propose d'utiliser l'apprentissage machine. Il s'en est révélé une plus grande efficacité et flexibilité de l'ontologie. Enfin, le fait que les régulateurs américains et européens s'intéressent à l'ontologie pour leurs futures mises à jour de leurs systèmes de gestion du trafic aérien amène un argument important pour la viabilité de notre solution.

Lors du développement de la plateforme ATC-Emu et des règles de détection, différents choix ont été posés afin de simplifier le problème et de parvenir à une preuve de concept. Il conviendra donc dans des travaux futurs de revenir sur ces choix pour réduire la frontière entre notre simulation et la réalité. De plus, des tests à grande échelle devraient être menés pour s'assurer que la qualité de la détection ne se dégrade pas dans des situations complexes avec un fort niveau de trafic.

ABSTRACT

Air traffic control has several types of radar to detect aircraft and ensure the safety of passengers and crews in the sky. Since the beginning of the 2000s, a communication protocol, *Automatic Dependent Surveillance-Broadcast* (ADS-B), has experienced a real boom, to the point where it may eventually supplant the radars and replace them in certain areas. With this technology, it is the aircraft themselves that continuously transmit their position data. In addition, a constellation of satellites is deployed to receive these messages and enable aircraft to be detected over the entire globe.

While this technology is of great interest because of the quality of the information transmitted and its low cost, it has shortcomings that are dangerous from a cybersecurity point of view. ADS-B messages are sent in clear text and have no authentication mechanisms. It is therefore possible for a person who is familiar with the protocol to be able to disrupt the work of air traffic controllers and adversely affect air traffic by injecting false packets via a software defined radio. This causes a safety problem, because in the event of an attack the controllers would not be able to carry out their mission properly and pilots may have to carry out dangerous manoeuvres following the indications of *Traffic Alert and Collision Avoidance System* (TCAS), a collision avoidance system which uses ADS-B.

In order to correct these vulnerabilities, many researchers are focusing on the various ways of integrating cryptography into the ADS-B protocol, which will require updating. However, certification processes in aeronautics generally take several years to complete. This is why our work has focused on the development and evaluation of a tool for detecting ADS-B attacks.

The guiding idea of our research is the use of ontologies to develop an expert system using detection rules based on high-level abstract concepts. The expert system first requires an ontological model that represents the air traffic control environment. It is this model that gives meaning to the recorded data. Then, in order to establish the detection rules, the main threats to air traffic had to be identified in order to establish probable attack scenarios. Based on the threats identified, we created four types of detection rules, called logics, which are: identification logic, origin logic, radar logic and flight logic. We thus obtain an expert system, ATC-Sense, composed of an ontological database and a reasoner who applies the detection rules to establish the veracity of ADS-B packets. To test the performance of our expert system for the detection of sophisticated attacks against ADS-B communications, we developed the ATC-Emu platform. It emulates the air traffic environment with the flight of aircraft, the various radars and the software used by air traffic controllers. ATC-Sense is

added to it and receives all the information from the radars. Finally, an attack module has been added to simulate attacks that could be carried out on ADS-B communications.

Tests have been carried out on these different detection logics, which give convincing results in terms of speed of execution and detection rate. In addition, a comparison of our solution has been made with another detection solution that proposes to use machine learning. This proved to be a more efficient and flexible ontology. Finally, the fact that American and European regulators are interested in the ontology for their future updates of their air traffic management systems provides an important argument for the viability of our solution.

During the development of the ATC-Sense platform and detection rules, various choices were made to simplify the problem and to arrive at a proof of concept. Future work will therefore have to revisit these choices in order to reduce the borderline between our simulation and reality. In addition, large-scale tests should be carried out to ensure that the quality of detection does not deteriorate in complex situations with a high level of traffic.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

AAE	Above Aerodrome Elevation
ACC	Area Control Centre
ADS-B	Automatic Dependent Surveillance-Broadcast
ADS-C	Automatic Dependent Surveillance-Contract
AGL	Above Ground Level
ALRS	Alerting Service
ANSP	Air Navigation Service Provider
ASL	Above Sea Level
ASM	Airspace Management
ASR	Airport Surveillance Radar
ATC	Air Traffic Control
ATCSS	Air Traffic Control System Simulator
ATFM	Air Traffic Flow Management
ATM	Air Traffic Management
ATOM	Abstractions Translation Ontology Method
ATS	Air Traffic Services
BDS	Comm-B Data Selector
CAATS	Canadian Automated Air Traffic System
CPDLC	Controller Pilot Data Link Communications
CPS	Cyber-Physical System
DIOSE	Détection d'intrusion avec l'ontologie par un système expert
DDS	Data Distribution Service
EFS	Electronic Flight Strip
FAA	Federal Aviation Administration
FDP	Flight Data Processor
FIR	Flight Information Region
FIS	Flight Information Service
FL	Flight Level
FSD	Flight Simulation Data
GADSS	Global Aeronautical Distress and Safety System
GNSS	Global Navigation Satellite System
ICAO	International Civil Aviation Organization
IDS	Intrusion Detection System

IFR	Instrument Flight Rules
ILS	Instrument Landing System
LSTM	Long Short-Term Memory
MAC	Message Authentification Code
METAR	Meteorological Aerodrome Report
MLAT	Multilateration
NIST	National Institute of Standards and Technology
OACI	Organisation de l'Aviation Civile Internationale
OMG	Object Management Group
OWL	Web Ontology Language
PAR	Precision Approach Radar
PIA	Privacy ICAO Address
PKI	Public Key Infrastructure
PSR	Primary Surveillance Radar
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RTCA	Radio Technical Commission for Aeronautics
SAR	Search and Rescue
SAT	Security in the Air using TESLA
SESAR	Single European Sky ATM Research
SDR	Software Defined Radio
SID	Standard Instrument Departure Route
SMR	Surface Movement Radar
SPARQL	SPARQL Protocol And RDF Query Language
SQL	Structured Query Language
SSR	Secondary Surveillance Radar
STAR	Standard Arrival Route
TCAS	Traffic Alert and Collision Avoidance System
TESLA	Timed Efficient Stream Loss-tolerant Authentication
TWR	Control Tower
URI	Universal Resource Identifier
URL	Universal Resource Locator
VATSIM	Virtual Air Traffic Simulation Network
VFR	Visual Flight Rules
WAM	Wide Area Multilateration
W3C	World Wide Web Consortium

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

Pouvoir détecter un avion dans le ciel et l'identifier est un des enjeux les plus importants pour le contrôle du trafic aérien, ou *Air Traffic Control* (ATC). Pour se faire, les services ATC utilisent différents systèmes. Il y a tout d'abord le radar primaire, *Primary Surveillance Radar* (PSR), qui est la plus ancienne technologie de détection d'aéronefs. Développée avant la Seconde Guerre mondiale, elle est constituée d'une antenne en rotation sur elle-même qui envoie des pulsations d'onde. Lorsqu'elles rencontrent un objet, une partie des ondes est réfléchie et renvoyée vers la source. En calculant ainsi le temps et la différence d'énergie entre l'émission et la réception, le système est capable de calculer la distance et l'azimuth de l'appareil par rapport au radar. Si cette méthode permet de détecter tout appareil volant, elle ne permet pas pour autant de les identifier. C'est pourquoi, un second type de radar, le *Secondary Surveillance Radar* (SSR), a été développé au début de la Seconde Guerre mondiale. Cette technologie est constituée d'une antenne qui tourne sur elle-même en émettant des interrogations à 1 030 MHz. Un dispositif électronique embarqué dans les aéronefs, le transpondeur, reçoit ces interrogations et répond à la fréquence 1 090 MHz en donnant le code identifiant à quatre chiffres de l'aéronef (*squawk code*) donné par les contrôleurs aériens. Cela permet de faire la distinction entre les différents appareils qui passent dans la zone de couverture du radar. Le SSR a connu une amélioration avec le mode S qui permet d'envoyer plus de données, dont l'identifiant 24-bit unique à chaque avion, ce qui facilite grandement l'identification des aéronefs par les contrôleurs aériens.

Ces deux technologies, bien que fiables, ont des limitations pour surveiller et contrôler le trafic aérien. D'abord, il y a des zones sur la planète où il est difficile (voire impossible) d'installer des radars (océans, montagnes, déserts, etc.). De plus, le coût de telles installations et de leur maintenance, s'élevant à plusieurs millions de dollars, représente un défi pour les gouvernements et limite leur déploiement. Finalement, la réception des radars est impactée négativement par des obstacles naturels (montagnes) entre les avions et les radars, ainsi que par les conditions météorologiques. C'est pourquoi dès 1992, la *Federal Aviation Administration* (FAA) s'intéresse à une nouvelle technologie, l'*Automatic Dependent Surveillance-Broadcast* (ADS-B), qui permet d'envoyer des informations depuis les avions. Comme pour le SSR, l'ADS-B utilise le transpondeur des avions, mais au lieu d'attendre une interrogation provenant d'une station au sol, c'est l'aéronef qui émet périodiquement des messages sur la fréquence 1 090 MHz ou 978 MHz. Ces informations sont diffusées par un équipement ADS-B

out[1] et reçues par les antennes au sol et par les appareils munis d'un équipement ADS-B *in* à portée (Figure 1.1). Ainsi, dans *Automatic Dependent Surveillance-Broadcast*, le terme *automatic* signifie que les informations sont envoyées par l'aéronef sans interrogation d'une source externe ; *dependent* traduit le fait que la surveillance ne se fait que si l'avion a activé son transpondeur et *broadcast* souligne cette diffusion. Avec cette nouvelle technologie, les informations transmises sont aussi plus précises et complètes. Il s'agit d'une amélioration du mode S du SSR. Parmi les informations envoyées se trouvent l'identifiant 24-bits *International Civil Aviation Organization* (ICAO) de l'appareil, la position calculée par satellite par le *Global Navigation Satellite System* (GNSS), l'altitude, la vitesse, le cap (direction) entre autres informations.

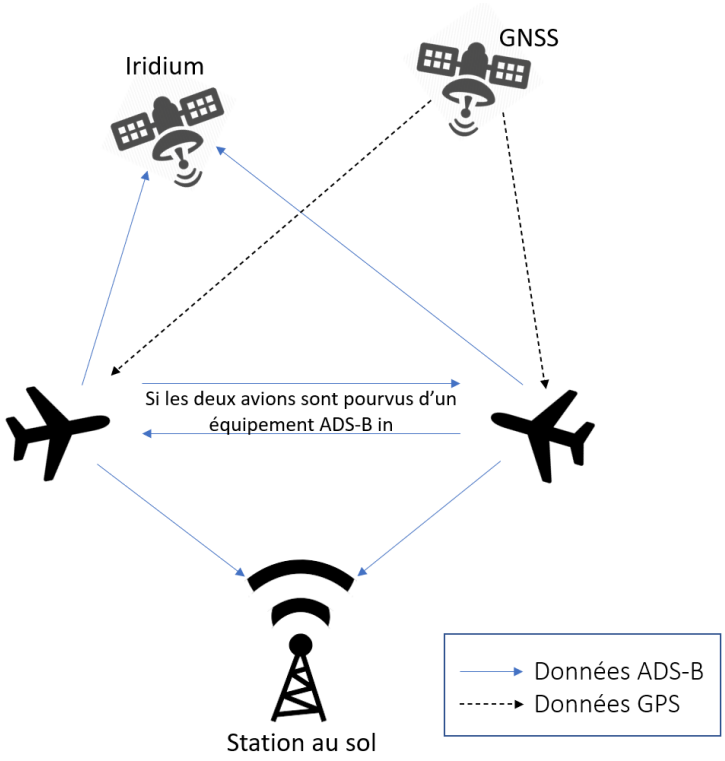


Figure 1.1 Les communications ADS-B

En 1998, la FAA a lancé une phase d'expérimentation de cette technologie, lors du programme Capstone [1], visant à améliorer la sureté aéronautique en Alaska. À la suite de ces résultats, la *Radio Technical Commission for Aeronautics* (RTCA) [2] a effectué un travail de normalisation aboutissant sur la norme *RTCA DO-242A - ADS-B Minimum Aviation System Performance*

1. Normalement, le transpondeur.
2. Organisme non lucratif ayant pour but de proposer des recommandations sur les différents systèmes utilisés en aéronautique.

Standard. Au 1er janvier 2020, la FAA a imposé, à quelques exceptions près, à tous les aéronefs circulant dans le secteur aérien américain d'être dotés d'un transpondeur émettant de l'ADS-B [2, 3]. Cette législation met en évidence l'engouement des différentes autorités et organisations mondiales pour adopter cette technologie utilisée depuis le début des années 2000.

En effet, la précision et le nombre d'informations envoyées par ADS-B permettent d'affiner la surveillance du trafic aérien. Enfin, le coût des infrastructures est dix fois moindre que celui des radars. Fort de ces avantages, cette technologie a d'abord été implantée dans de vastes régions inhabitées où le trafic aérien est quand même important. Ainsi, Nav Canada, la société qui s'occupe des services du contrôle aérien au Canada a déployé dès la fin des années 2000 une zone de couverture ADS-B autour de la Baie d'Hudson d'une superficie de 850 000 km^2 où passent annuellement plus de 35 000 avions. Les garanties offertes par ce protocole en matière de sureté et la transmission périodique de la position des avions ont permis entre autres de réduire la distance de séparation entre deux appareils de 80 NM à 5 NM [3], ce qui représente une réduction de la consommation de carburant de 18 millions de litres (soit 10 millions de dollars canadiens en 2009) [5]. Cet argument économique est de grand intérêt pour les compagnies aériennes qui cherchent à tout prix à réduire leur consommation [6].

La simplicité de l'ADS-B et les nombreux avantages offerts par ce protocole en ont fait un indispensable du développement des systèmes ATC dans le monde. C'est ainsi la pierre d'angle des projets NextGen de la FAA et *Single European Sky ATM Research* (SESAR) de l'organisme de contrôle aérien européen Eurocontrol afin de moderniser les structures de gestion de trafic aérien, ou *Air Traffic Management* (ATM) [7, 8]. L'un des points clés de ces projets est le développement de l'ADS-B satellitaire qui permettrait de s'affranchir d'installations physiques dans les coins reculés et offrirait une surveillance de ces espaces et des océans. Grâce à cela, la distance de séparation entre deux avions pourrait être réduite ce qui donnerait un gain de temps et carburant conséquent. La compagnie *Aireon*, fondée par les agences de services ATC du Canada, du Danemark, de l'Italie, de l'Irlande et du Royaume-Uni, se base là-dessus. Il utilise la constellation de satellites *Next* de la société *Iridium Communications* pour étendre la couverture ADS-B à la totalité de la surface terrestre [9]. L'enjeu de ce projet est de taille, car cela permettrait non seulement d'améliorer la gestion du contrôle aérien, mais aussi de nombreux pays, qui n'ont pas les moyens d'installer et entretenir des infrastructures radar, pourraient avoir accès à l'information diffusée par ce système pour assurer de meilleurs services ATC sur leur territoire. C'est aussi une avancée majeure pour les

3. Bien que depuis 2010 [4] l'Organisation de l'aviation civile internationale (OACI) (ou en anglais ICAO) préconise d'utiliser les unités du Système International, dans les faits ce sont les nœuds (kn), les milles marins (NM) et les pieds (ft) qui sont principalement employés en aviation.

opérations de recherche et sauvetage, *Seach and Rescue* (SAR), en cas d'écrasement ou de disparition d'un appareil. En effet, à moins que le transpondeur ne cesse d'émettre, à cause d'une panne ou d'une action délibérée du pilote, un avion transmettra sa position où qu'il soit sur le globe. Suite à la disparition du vol MH370 en 2014, l'OACI a lancé une série de recommandations pour un meilleur traçage des avions et améliorer l'alerte des SAR, menant à la mise en place du *Global Aeronautical Distress and Safety System* (GADSS) [10]. *Aireon* et *Flightaware*[4] ont alors lancé un partenariat pour proposer *GlobalCare*, une solution conforme au GADSS utilisant l'ADS-B [11, 12].

1.1 Problématique

L'ADS-B est devenu ainsi indispensable dans l'écosystème du contrôle du trafic aérien. Il a de nombreux avantages en termes de simplicité, de coût et de qualité des informations transmises. Néanmoins, cette simplicité se traduit par une grande vulnérabilité. En effet, si ce protocole possède quelques outils pour empêcher des erreurs dans la transmission (par exemple un code de redondance cyclique pour prévenir des interférences), il a deux grands défauts qui rendent son utilisation facilement exploitable par des tiers malintentionnés : un manque d'authentification et une absence de chiffrement.

Le "B" de *Broadcast* dans son acronyme rappelle le fait que c'est le transpondeur qui émet les messages sans avoir reçu auparavant d'interrogation de la part d'un radar. Ce ne sont donc pas les systèmes au sol qui initient la détection, mais l'avion lui-même. Il n'y a pas de connexions bilatérales, comme c'est le cas pour le protocole cousin *Automatic Dependent Surveillance-Contract* (ADS-C). De plus, tous les messages sont envoyés en clair, sans aucun chiffrement et sans l'authentification de la source émettrice. C'est notamment ce qui permet à des sites internet comme *flightaware.com* ou *flightradar24.com* d'afficher la position des avions en temps réel. Ils utilisent d'une part les données provenant de sources officielles et s'appuient d'autre part sur un réseau de *feeders*. Ce sont des particuliers disposant d'une radio logicielle, ou *Software Defined Radio* (SDR) et une antenne, qui reçoivent les messages ADS-B émis par les avions qui passent au-dessus de chez eux et envoient ces données à ces sites. Avec plus de 24 000 *feeders* dans 199 pays rien que pour le site de *flightaware.com*, c'est une grande partie du globe terrestre qui est couverte [13]. Puisque l'ADS-B contient l'identifiant 24-bit de chaque appareil, les déplacements de ces avions sont ainsi visibles par tout le monde. Dans des soucis de vie privée, la FAA a lancé à la fin des années 2010 sur le territoire américain le programme *Privacy ICAO Address* (PIA) afin de permettre un certain anonymat pour l'aviation générale et d'affaires [14]. Mais cela n'apporte aucune sécurité supplémentaire au

4. Société qui donne, à travers un site web et une application, accès au suivi des vols.

fonctionnement de l'ADS-B. Or, si l'on peut recevoir ces paquets avec un SDR, avec certains modèles il est également possible d'émettre des signaux spécifiques et donc de construire de faux messages ADS-B. Andrei Costin et Aurélien Francillon ont montré en 2012 lors de la conférence Black Hat qu'il est possible de créer de faux paquets ADS-B afin de perpétrer des attaques contre le trafic aérien [15]. Le protocole est connu, n'utilise ni chiffrement ni authentification, et la technologie pour l'utiliser est disponible au grand public.

Ainsi, le protocole est vulnérable à des attaques informatiques. Les SDR ont rendu possible l'injection de faux avions dans les systèmes ATC. Mathias Schäfer, Vincent Leders et Ivan Martinovic ont réalisé en 2013 une série d'attaques sur une simulation de trafic aérien et ont montré les différents effets que cela pouvait avoir [16]. Par exemple, ils montrent qu'en injectant un nombre élevé de faux avions, l'écran des contrôleurs aériens devient illisible, ce qui nuit fortement à leur travail. En ayant fait apparaître une centaine de faux avions dans le trafic aérien autour d'une même zone, il est impossible de traiter les vrais avions en temps réel. En inondant les récepteurs de faux paquets (*flooding attack*), ils créent une forme de déni de service. Dans ce cas, ce n'est pas le système qui est saturé, mais les utilisateurs. Cette paralysie du contrôle aérien pourrait avoir de lourdes conséquences en matière de sécurité aérienne et de coûts pour les compagnies aériennes dont les avions devront sûrement être déroutés.

D'autres groupes de recherches ont catégorisé les attaques et ont déterminé différents degrés de difficulté pour les réaliser [17–19]. Leurs taxonomies d'attaques regroupent le déni de service, l'usurpation d'identité (*spoofing attack*) ainsi que le brouillage des communications (*jamming attack*). Ces attaques viennent perturber le bon fonctionnement de l'ATC, mais elles sont peu sophistiquées et surtout sont visibles une fois déclenchées. En effet, si une centaine de faux avions apparaissent d'un coup, alors cela devient suspect. En plus, le brouillage des communications se révèle être facilement détectable. Cependant, les attaques peuvent être encore plus élaborées. Un acteur malveillant qui aurait des connaissances dans le domaine de la circulation aérienne pourrait alors créer des faux avions qui suivent une trajectoire logique. En ayant une cohérence entre les messages envoyés et en apportant une intelligence qui connaît le fonctionnement et les limites de l'environnement ATC, la confusion créée au niveau des contrôleurs peut avoir de plus grandes répercussions. Les attaques sur les communications ADS-B peuvent aussi avoir des effets directement sur les avions. En effet, les aéronefs sont équipés d'un système de prévention de collision, le *Traffic Alert and Collision Avoidance System* (TCAS) qui se base sur les informations transmises par les transpondeurs des appareils à proximité. Lorsqu'il y a un risque de collision, le TCAS émet une alerte et ordonne au pilote d'effectuer une manœuvre d'évitement. Ainsi, de faux messages ADS-B peuvent occasionner des manœuvres d'urgences, voire dangereuses.

Ce qui rend la situation encore plus inquiétante, ce sont les conséquences qui suivent l'engouement pour le déploiement de l'ADS-B. En effet, avec l'ADS-B satellitaire, les sociétés d'ATC songent maintenant à remplacer les stations au sol. C'est notamment le cas de Nav Canada qui prévoit de ne pas renouveler une partie de ses stations ADS-B et SSR au sol en 2028 [20]. La diversité des capteurs et des protocoles est une bonne pratique en sécurité informatique. Là où il y a un radar SSR et une antenne ADS-B qui couvrent la même zone, les contrôleurs aériens peuvent faire une comparaison entre les informations reçues. Un message reçu par l'un et non par l'autre deviendrait suspect. En uniformisant les moyens de surveillance à l'ADS-B seul, on prive ainsi la détection d'un moyen de comparaison. Cela laisse donc le champ libre à des acteurs malveillants pour lancer des attaques dans ces zones. Un autre risque avec l'abandon des stations ADS-B au sol est l'augmentation de la portée d'attaque, car un attaquant à Montréal pourrait théoriquement faire apparaître un avion à Vancouver par exemple. Avec un radar, il y aura un décalage logique et au niveau de la réception des données par l'antenne, on pourrait se rendre compte que les coordonnées sont contradictoires avec la portée de l'équipement.

1.2 Comment se protéger des attaques contre l'ADS-B ?

Lors de la découverte de vulnérabilités sur un protocole, différentes options sont possibles afin d'en améliorer la sécurité. La façon la plus simple est de faire une mise à jour, une nouvelle version du protocole en ajoutant des mesures de sécurité afin de supprimer la vulnérabilité. Dans certains cas, des raisons techniques (par exemple un nombre de bits restreint) ou d'implémentation font en sorte que le protocole doit être repensé entièrement. De nombreuses recherches avec ces points de vue ont été effectuées pour sécuriser l'ADS-B. Le chapitre 3 les passera en revue. Mais déjà plusieurs embûches à leur implémentation sont visibles. La première est la question de l'interopérabilité entre les appareils qui disposeront de la version mise à jour et ceux que ne l'ont pas. Et la plus grosse faiblesse reste le temps de certification qui est particulièrement long en aéronautique. Il suffit de relire l'histoire de l'ADS-B pour s'en convaincre : entre les premiers tests menés par la FAA en 1992 jusqu'à son obligation aux États-Unis en 2020, près de trente ans se sont écoulés. Ainsi donc, la modification d'un tel protocole, même si nécessaire, est un projet sur le long terme.

Dû aux enjeux mentionnés précédemment, une autre partie des recherches s'est orientée sur la détection de faux paquets ADS-B. Il s'agit non pas d'empêcher des acteurs malveillants d'utiliser à mauvais escient ce protocole, mais d'éviter que leurs actes aient des conséquences. Les deux principaux intérêts à cette vision c'est d'avoir à disposition un outil qui apporte une couche de sécurité pour les contrôleurs aériens avant que le protocole soit mis à jour

et surtout de continuer à avoir cette protection supplémentaire si le nouveau protocole fait face à des vulnérabilités. Ces deux branches de recherches sont donc complémentaires, l'une sécurisant le protocole et l'autre via la détection sécurisant l'utilisation du protocole.

Parmi ces différents travaux, se trouve une approche proposée par Louis-Philippe Morel ayant pour but la mise en place d'un module de détection sémantique [21]. Dans son mémoire de maîtrise, il propose d'utiliser les ontologies afin de créer un système expert pour détecter de faux paquets ADS-B. Il a ainsi développé une ontologie et quelques règles afin de discuter de la légitimité d'un paquet reçu. Un des intérêts principaux de cette recherche est que le but est le développement d'un outil d'aide à la décision pour les contrôleurs aériens dont les résultats sont compréhensibles par tous ceux travaillant dans le domaine ATC. En effet, en tant que système expert, les règles de détection s'appuient sur la connaissance qu'ont les experts du domaine des situations qui peuvent avoir lieu ou pas. Et donc si une situation suspicieuse est détectée, c'est parce qu'elle a enfreint une règle et on sait alors pourquoi. Ce travail, bien qu'intéressant, s'est arrêté à l'état de concepts avec une proposition d'ontologies, d'architecture et de règles.

1.3 Objectifs de recherche

L'objectif de cette recherche est de développer et valider une solution capable de détecter des attaques sophistiquées sur l'ADS-B. Bien que les vulnérabilités de l'ADS-B peuvent avoir des conséquences directes sur le pilotage à cause de son utilisation par les systèmes TCAS, nous ne le considérerons pas et limiterons nos travaux au contrôle au sol. Pour atteindre notre objectif, différentes questions de recherche ont été énoncées :

Q1 : Quelles sont les différentes cyberattaques plausibles sur les communications ADS-B ? Quels sont les impacts qu'elles peuvent avoir ?

Si différentes taxonomies d'attaques ADS-B ont déjà été réalisées [17–19], elles restent très générales en donnant des méthodes d'attaques plausibles. Mais pour évaluer une menace et son risque, il faut déterminer à la fois quels sont les acteurs potentiels et quels sont les scénarios et impacts possibles. Le but de cette question de recherche est donc de définir des exemples concrets de menaces pour pouvoir ensuite trouver les règles de détection nécessaires.

Q2 : Quelles sont l'efficacité et la flexibilité de l'utilisation d'ontologies pour détecter ces attaques en comparaison avec des solutions antérieures ?

Les ontologies sont choisies pour réaliser un système expert. Cela se traduit par :

1. La construction d'une ontologie représentant les concepts adéquats.
2. La mise en place de règles de détection basées sur des requêtes sur cette base de données ontologiques.

Il faut donc s'assurer que les règles sont bien appliquées et qu'elles ne souffrent pas d'erreurs de jugement. De plus, puisque différents scénarios d'attaques sont proposés, il faut vérifier que l'ontologie est suffisamment flexible pour toutes les couvrir.

Q3 : Quelle est la performance de la solution proposée et sa viabilité en temps réel ?

Cette proposition d'outil d'aide à la décision pour les contrôleurs aériens a pour vocation ultime d'être implémentée dans les systèmes ATC. Au-delà des critères d'efficacité et de flexibilité, il faut s'intéresser à des critères comme le temps d'exécution ou le calcul nécessaire pour chaque requête afin de vérifier si les modules développés peuvent être employés à grande échelle et avec une circulation aérienne intense, et ce en temps réel.

1.4 Plan du mémoire

Le chapitre 2 de ce mémoire est consacré aux concepts de base de l'aéronautique et de la sémantique qui sont utilisés tout au long de ce travail de recherche. Dans un premier temps, il décrit l'environnement du monde de l'ATM avec ses différents composants et systèmes utilisés. L'introduction a donné le contexte de la recherche, cette partie lui donne son cadre. Et puis ce chapitre définit plus précisément ce que sont les ontologies et ce que permet la technologie des bases de données ontologiques.

Le chapitre 3 est une revue de littérature qui recense les différentes avancés et les travaux menés dans la sécurisation de l'ADS-B. On y retrouve les deux branches introduites précédemment. D'un côté, une recherche qui s'intéresse à une amélioration du protocole afin de le rendre plus sécurisé et, de l'autre des travaux sur des outils afin de pouvoir détecter les attaques ADS-B. Enfin, une section est dédiée à l'utilisation d'ontologies dans le monde aérien et aussi en sécurité informatique.

Dans le chapitre 4 sont détaillées les différentes méthodologies employées lors de cette recherche. Dans un premier temps nous analysons les différentes attaques qui peuvent être menés sur les communications ADS-B afin de déterminer les menaces pesant sur le trafic aérien et leurs conséquences. Puis nous cherchons à établir un modèle ontologique qui nous permet de représenter à un haut niveau l'environnement du contrôle aérien et ainsi que ces attaques. Enfin, nous proposons quatre logiques de détection contre des attaques sophistiquées.

Ensuite, dans le chapitre 5 nous expliquons notre évaluation expérimentale. Pour pouvoir tester les attaques sophistiquées et les différentes règles de détection, nous développons une plateforme de simulation de contrôle aérien, ATC-Emu, ainsi que la solution de détection basée su l'ontologie, ATC-Sense. Nous préparons ensuite un cadre de test en reproduisant l'espace aérien canadien, avant de comparer le système expert basé sur des ontologies avec une autre méthode de détection utilisant l'apprentissage machine.

Dans le chapitre 6, nous présentons les résultats produits par cette recherche. Cela se traduit par l'évaluation du modèle ontologique, des logiques de détection de notre solution, ainsi que les résultats de la comparaison avec une méthode utilisant l'apprentissage machine pour détecter des faux paquets ADS-B.

Enfin, le chapitre 7 montre les limites de cette approche avec les choix qui ont été faits et les résultats obtenus. Il propose une discussion pour de travaux ultérieurs afin d'améliorer la plateforme et ses modules ainsi qu'une réflexion sur les étapes à suivre pour passer d'un prototype à un projet à plus grande échelle.

CHAPITRE 2 DÉFINITIONS ET CONCEPTS DE BASE

2.1 Le contrôle du trafic aérien

La description qui s'en suit de l'écosystème du trafic aérien est tirée des procédures et rapports de l'OACI [22] et d'Eurocontrol [23].

La gestion du trafic aérien (*Air Traffic Management* en anglais, ou ATM) regroupe toutes les activités visant à assurer la sécurité et l'efficacité des flux d'appareils dans l'espace aérien. Elle s'appuie sur trois différentes branches qui sont étroitement reliées.

***Air Traffic Services* (ATS)** Les services de la circulation aérienne sont offerts par des organisations nationales ou privées dans le but d'assurer la sécurité, l'efficacité et l'ordre des opérations des aéronefs dans le ciel et au sol. Ils sont au nombre de trois.

- Le contrôle du trafic aérien (*Air Traffic Control* ou ATC). Organe principal de l'ATM, il a pour mission d'assurer le suivi et la sécurité des aéronefs dans l'espace aérien tout en gardant une circulation fluide et efficace. Il veille notamment à éviter toute collision entre les appareils en prenant garde à ce qu'ils maintiennent une séparation horizontale et verticale suffisante. Les aéronefs sont soumis à ses décisions et doivent attendre une clairance (en anglais *clearance*) du contrôleur pour les phases importantes du vol comme le roulage, le décollage, le suivi d'une route ou l'approche.
- *Flight Information Service* (FIS). Le service d'information de vol envoie aux pilotes les informations et avis nécessaires pour la bonne exécution de leur vol en toute sécurité. Cela peut-être des renseignements comme une très mauvaise météo durant le vol ou à l'arrivée, la présence d'oiseaux aux abords des aérodromes ou des dangers plus inusuels comme une éruption volcanique ou la présence de nuages toxiques voire radioactifs.
- Le service d'alerte (*Alerting Service* ou ALRS). Il s'occupe d'informer, de faire le lien et apporter son aide et expertise aux autorités et organisations appropriées en cas d'opération de recherche et de sauvetage d'un appareil.

***Air Traffic Flow Management* (ATFM)** La gestion des flux de trafic aérien a pour objectif principal de réguler les mouvements des appareils afin d'éviter la saturation des aérodromes et des secteurs aériens. Elle s'appuie pour cela sur la capacité des centres de contrôle ATC, en nombre d'appareils par heure, ainsi que des plans de vol déposés. Les services ATFM doivent donc veiller à maximiser les flux aériens sans pour autant dépasser ces

capacités, afin que le contrôle aérien se fasse en toute sécurité. En cas de demande excessive, ils peuvent avoir des actions immédiates, comme faire patienter les appareils au sol, ou sur le long terme en définissant des créneaux horaires précis ou des routes de rechange pour la circulation aérienne.

***Airspace Management* (ASM)** Ce dernier service s'occupe de la gestion de l'espace aérien. Cela passe par la définition des routes, des zones de contrôle, des restrictions et des niveaux de vol. Il cherche des solutions pour concilier les intérêts de toutes les parties prenantes : gouvernements, militaires, compagnies aériennes et populations.

2.1.1 Infrastructures ATC

Pour répondre à ses missions de surveillance, le contrôle aérien peut s'appuyer sur un large panel de systèmes et technologies mis à sa disposition. Ils peuvent être classés en trois catégories : les systèmes basés sur les radars, les systèmes basés sur le satellite et les autres [24].

Le radar primaire de surveillance (PSR) est la première des technologies radar. Il utilise le principe d'écholocalisation pour détecter les aéronefs dans le ciel. Ce principe permet d'obtenir la position des aéronefs en calculant la distance entre le radar et la source, et à partir de l'azimut obtenu par l'orientation de l'antenne au moment de la réception de l'écho et. Par contre, il ne peut pas obtenir l'altitude ni identifier les avions. C'est pourquoi il est souvent associé avec un radar secondaire de surveillance (SSR). Le PSR est un moyen de détection indépendant, car il ne nécessite pas d'équipement embarqué dans les appareils pour déterminer la position des avions. C'est ce qui en fait sa grande force et c'est pourquoi il a aussi une mission de redondance et de secours. Il est utilisé en backup des systèmes de détection dépendants (SSR/ADS-B) dans les zones d'approche ou de phase critique de vol pour pallier un éventuel défaut de fonctionnement des transpondeurs. De plus, il est aussi utilisé pour détecter des phénomènes météorologiques comme les tempêtes, ainsi que les groupes d'oiseaux migrateurs et les autres objets volants pouvant affecter les vols. Enfin, c'est le moyen de détection le plus sûr pour le domaine militaire.

Le PSR désigne en fait une famille de radars fonctionnant sur le principe de réflexion d'onde. Le tableau 2.1 détaille leurs principales caractéristiques. On y trouve notamment les :

- *Airport Surveillance Radar* (ASR). Radar de surveillance d'approche. Couplé avec un radar SSR, il est déployé dans les zones de contrôle terminales. Il a une portée de 60 NM à 100 NM (de 111 à 185 km).
- *Precision Approach Radar* (PAR). Radar d'approche de précision. Ce radar a la particularité d'être lié à une piste d'atterrissage particulière. Installé dans l'axe de la piste

et avec une portée de 22 NM (41 km), il permet de suivre avec une grande précision la trajectoire d'un avion en phase d'approche. Il est principalement utilisé en cas de mauvaise visibilité. Si l'avion n'est pas aligné avec la piste ou que la trajectoire de descente n'est pas bonne, le contrôleur appelle le pilote afin de l'avertir et de l'aider à se remettre en bonne place. L'utilisation du PAR se fait principalement dans le domaine militaire. En effet, dans l'aviation civile, les approches par mauvaise visibilité se font surtout avec l'aide de la technologie *Instrument Landing System* (ILS) [1].

— *PSR en-route* : Radar longue portée (250 NM, soit 463 km) qui s'occupe du suivi des avions à l'extérieur des zones terminales.

— *Surface Movement Radar* (SMR) : Radar de détection des mouvements au sol. Il est utilisé dans les zones aéroportuaires en complément des observations visuelles. Sa portée est nettement moindre (moins de 10 km), mais suffisante pour son utilisation.

Tableau 2.1 Comparaison des radars PSR

	ASR	PAR	PSR en-route	SMR
Portée	60-100 NM	22 NM	250 NM	5 NM
Fréquence	2700-2900 GHz	9-9.2 GHz	1215-1370 MHz	9.1-9.5 GHz
Taux de renouvellement des données	5 s	1 s	5-10 s	1 s
Nombre de cibles par rotation	>1000	100	>1000	>300
Prix	3M $	/	6M $	500K $

Pour pallier les limitations des radars primaires quant à l'information qu'ils peuvent saisir, les systèmes ATC s'appuient sur les radars secondaires. Historiquement, le but premier lors de la Seconde Guerre mondiale était de pouvoir distinguer les avions alliés des autres en embarquant dans les avions un dispositif appelé *Identification Friend or Foe (IFF)*. Ce dispositif-ci répondait à des interrogations très précises des stations au sol. Ainsi un avion qui était détecté par les radars primaires, mais qui n'avait pas donné de réponse IFF pouvait être considéré comme un avion ennemi, car cela signifiait qu'il n'avait pas le dispositif embarqué. Cette technologie a par la suite évolué pour donner le radar secondaire utilisé actuellement. Le SSR est constitué de deux parties : 1) une antenne radar qui est en rotation sur elle-même et 2) un dispositif appelé *transpondeur* qui est embarqué dans les aéronefs. L'antenne radar

1. Méthode d'aide à l'approche pour des avions en vol aux instruments. Deux éléments placés en bordure de piste indiquent au pilote l'alignement par rapport à l'axe de la piste et à la pente nominale d'approche. Ce n'est donc pas un moyen de détection.

émet des interrogations à 1 030 MHz et le transpondeur répond en retour en envoyant des informations qui diffèrent selon le mode sélectionné à la fréquence 1 090 MHz. Lorsqu'il reçoit une réponse, le radar secondaire calcule la position (azimut et distance) de l'avion à partir du décalage entre l'interrogation et la réponse. Il existe différents modes de transpondeur pour le domaine civil et militaire. Dans l'aviation civile, on distingue :

- *Mode A*. Le mode *Alpha* est le plus ancien des modes. Le transpondeur envoie au radar son code *squawk*. Le radar peut ensuite associer un identifiant à la position qu'il vient de calculer. Le code *squawk* est un code en octal de quatre chiffres allant de 0000 à 7777. Il est donné soit par le contrôleur aérien, soit par la législation. Par exemple, le code 1200 est le code standard pour un vol à vue dans l'espace nord-américain quand aucun autre n'a été assigné par un contrôleur. De même, il existe certains codes que le pilote peut envoyer pour signaler une situation majeure : 7500 pour un détournement, 7600 pour une panne radio ou 7700 pour une urgence. Ce code est sélectionné par le pilote sur le transpondeur.
- *Mode C*. Le mode *Charlie*, ou A/C, est le mode SSR le plus utilisé. Il reprend les principes du mode A, mais ajoute l'altitude calculée par l'avion par incrément de 100 ft (30,48 m). C'est donc le premier moyen de détection qui permet aux contrôleurs aériens d'avoir une position 3D de l'avion : distance et azimut calculées par le radar et l'altitude donnée par l'avion. Cela permet une meilleure distinction des points radars lorsque ceux-ci se chevauchent ou sont proches sur l'écran.
- *Mode S* : Le mode *Sierra* est une évolution des transpondeurs qui offre beaucoup de possibilités. Le *squawk* code ne permet pas d'identifier totalement l'avion car il est donné par le contrôleur aérien pour une période de temps donné et il n'y a pas forcément de continuité entre deux secteurs aérien. Pour pallier ce problème, le mode S utilise l'identifiant unique de l'avion, l'adresse 24-bit assignée par l'OACI. Il ajoute aussi les informations du mode A/C avec une meilleure précision pour l'altitude (25 ft contre 100 ft). En plus du statut de l'avion (en vol ou au sol), il permet d'envoyer d'autres informations comme la vitesse par rapport au sol, le taux de montée ou le cap. Ces informations sont regroupées par registres de 56 bits appelés Comm-B Data Selector (BDS) [25].

La seconde catégorie des moyens de détection regroupe les systèmes dont la détermination de la position se fait par satellite. Ici, ce n'est pas l'antenne qui détermine la position de la source, mais c'est la source elle-même qui émet sa position calculée par un *Global Navigation Satellite System* (GNSS) qui est alors transmise par l'avion. Cette catégorie regroupe l'ADS-B et l'ADS-C.

L'ADS-B est un protocole dérivé du Mode S. Pour l'utiliser, il faut que l'avion soit pourvu

d'un transpondeur *Mode S extended squitter* ou d'un émetteur ADS-B *out*. Chaque seconde sont envoyées par broadcast différentes informations comme la position, l'altitude, l'identifiant OACI 24-bit, le *callsign*[2] et bien d'autres. Bref, on y retrouve toutes les informations du Mode S. Cette fréquence élevée de mise à jour des données ainsi que la précision et le nombre des informations envoyées en font un grand atout pour les contrôleurs aériens.

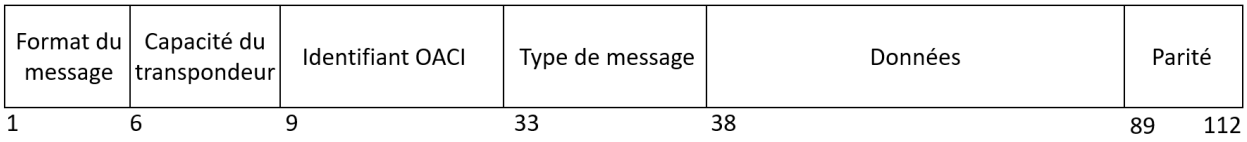


Figure 2.1 Format d'un message ADS-B de 112 bits

Un message ADS-B est codé sur 112 bits précédés d'un préambule envoyé sur $8\mu s$. La Figure 2.1 montre les différentes parties d'un message. Elles sont au nombre de six :

- *Format du message* - Codé sur 5 bits. Ce champ est commun à tous les messages émis par le transpondeur. Pour un message ADS-B, c'est la valeur 17 qui est renseignée.
- *Capacité du transpondeur* - Elle est codée sur 3 bits. Elle désigne la version du transpondeur.
- *Identifiant OACI* - Cet identifiant sur 24 bits est celui donné lors de la mise sur le marché de l'avion. Il est unique et attribué par l'OACI lors de l'immatriculation de l'appareil.
- *Type de message* - C'est le numéro du *Comm-B Data Selector* (BDS) codé sur 5 bits qui est contenu dans la partie suivante. Il permet donc de comprendre les informations qui sont envoyées en utilisant le bon algorithme de décodage.
- *Données* - Les données sont codées sur 51 bits. Elles proviennent des systèmes de navigation de l'avion et correspondent à différents registres. Le tableau 2.2 décrit le type de contenu présent dans ce champ à partir de la norme OACI 9871 [26].
- *Parité* - Afin de s'assurer qu'il n'y a pas eu d'interférences ou de données tronquées lors de l'envoi du message, la parité est calculée sur 24 bits à l'aide d'un code de redondance cyclique.

L'ADS-C est un protocole sous forme de contrat établi entre une unité de service du trafic aérien et l'avion. C'est le centre ATC qui initie le contrat et c'est seulement lui qui reçoit les données, contrairement à l'ADS-B où l'information est diffusée à tout le monde. Ce protocole est basé sur les communications satellites et permet un contrôle dans les zones désertiques

2. Indicatif radio de l'avion. Il correspond à son immatriculation ou au nom de la compagnie suivi de son numéro de vol, par exemple N1234A,C-GBAC, dans le premier cas, ou TS110 ou AC304 dans le second cas.

Tableau 2.2 Types de messages ADS-B

Type de message	Contenu
0	Pas d'information
1 - 4	Identification
5 - 8	Position à la surface
9 - 18	Position de l'avion (altitude barométrique)
19	Vitesse
20 - 22	Position de l'avion (altitude calculée par le GNSS)
23 - 30	Réservé
31	Statut de l'avion

et océaniques. L'avion transmet des informations comme sa position, son identifiant ou encore des données météorologiques. Dans le cas de contrat périodique, un rapport ADS-C est émis en fonction de différents évènements comme l'arrivée à un *waypoint*[3], un changement de taux de montée ou une déviation de trajectoire [27]. Ce n'est pas un outil de détection ou de contrôle dans son usage, mais plutôt d'information. Les données sont envoyées ponctuellement. La durée entre deux messages ADS-C fluctue selon les pays. Elle est en moyenne de 14 minutes dans plusieurs pays [28], mais peut varier de 10 à 27 minutes aux États-Unis selon l'endroit où se situe l'avion [29]. Ce taux de rafraichissement des données est très loin des taux de mise à jour des autres moyens décrits ci-avant (entre 1 et 10 secondes).

Enfin, un dernier moyen de détection des avions est la *multilatération*. Il en existe deux types, un près des aéroports appelé *Multilateration* (MLAT) et l'autre sur les zones en-route appelé *Wide Area Multilateration* (WAM). Il s'agit d'un maillage d'antennes recevant les signaux envoyés par les transpondeurs lors des réponses pour le SSR (mode A/C ou S) ou de manière spontanée comme l'ADS-B. La localisation des avions se fait en calculant la différence de temps d'arrivée des messages pour chaque antenne. En effet, les balises étant à différents endroits, lorsqu'un avion émet un signal, les antennes le recevront avec un léger décalage selon leur distance à la source. Cette technologie est intéressante parce qu'elle permet d'utiliser des technologies déjà en place (SSR, ADS-B) et apporte dans la détection une couche supplémentaire de sécurité, car le transpondeur est détecté là où il émet.

Ces technologies sont celles utilisées par les systèmes ATC civils, les principales étant quand même les radars primaires, secondaires ainsi que l'ADS-B. L'annexe A fait une comparaison de chacune des technologies décrites précédemment.

3. Repère cartographique. Une succession de waypoints forme la route qu'un avion doit suivre.

2.1.2 Un espace réglementé

Selon l'article premier de la convention de Chicago [4], «[...] chaque État a la souveraineté complète et exclusive sur l'espace aérien au-dessus de son territoire» [30]. Ce sont aux gouvernements de décider de l'utilisation de leur espace aérien. Néanmoins la plupart des états suivent un schéma similaire de classification et d'organisation. D'une façon générale, ces espaces sont divisés en différentes classes. Elles définissent les règles de circulation, les services offerts par les organismes de contrôle aérien ainsi que les communications et équipements nécessaires.

Au Canada [5], le ministère des transports appelé Transports Canada définit sept classes d'espace aérien [31]. La Figure 2.2 représente leur agencement. Les classes de A à E désignent l'espace aérien contrôlé. La plus restreinte est la A où seuls les appareils volants aux instruments sont autorisés. À l'inverse, la plus permissive est la E où par exemple il n'est pas obligatoire pour les appareils volants à vue d'avoir une radio. Ensuite vient la classe F qui délimite des zones à usages spéciales où le trafic est réglementé. Ce sont par exemple des zones militaires. Et pour terminer, la classe G correspond à l'espace aérien non contrôlé.

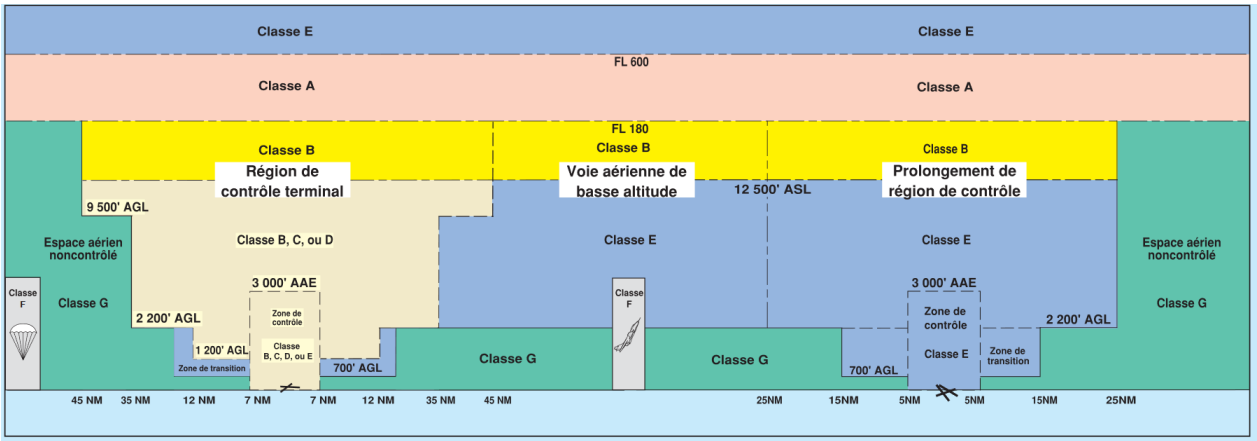


Figure 2.2 Les classes de l'espace aérien canadien [31]

Pour faciliter la tâche des contrôleurs, l'espace aérien est découpé en différentes zones appelées secteurs aériens. Chaque secteur est représenté sur la carte par une forme géométrique, une altitude maximale et minimale, et correspond typiquement à une seule classe aérienne. Il existe différentes façons de mesurer l'altitude comme on peut le voir sur la Figure 2.2. Elle peut être exprimée par rapport au niveau de la mer (*Above Sea Level*, ASL), au sol (*Above*

4. Convention en 1944 qui a créé l'OACI pour s'occuper de la coordination et de la réglementation du trafic aérien international.

5. Sauf mentions contraires, les définitions qui suivent sont expliquées dans le cas du Canada.

Ground Level, AGL), ou encore d'un aérodrome (*Above Aerodrome Elevation*, AAE). Enfin, des niveaux de vol, ou *Flight Level* (FL), sont définis par rapport à l'isobare 1013,35 hPa. Un niveau de vol sont une altitude barométrique qui est calculée en centaine de pieds. Ainsi, lorsque le pilote ou le transpondeur donne des informations sur l'altitude, il est important de connaître le type d'altitude qu'ils utilisent. La forme géométrique des secteurs aériens est souvent cylindrique au niveau des aéroports importants et ressemble à des parallélépipèdes entre ces aéroports formant ainsi des couloirs aériens. Ce maillage est mis à jour périodiquement et est disponible sous forme de publications électroniques et chartes [32]. Ces secteurs sont pris en charge par des contrôleurs aériens et ont des limitations de capacité.

À cause de la grande superficie du Canada et d'un trafic aérien important, notamment au niveau des aéroports internationaux et de l'océan Atlantique, le territoire est décomposé en sept régions : Vancouver, Edmonton, Winnipeg, Toronto, Montréal, Moncton et Gander [33]. Chaque région d'information de vol, ou *Flight Information Region* (FIR), possède un centre de contrôle régional, ou *Area Control Centre* (ACC). C'est ici que s'opèrent la surveillance et le contrôle d'un appareil *en-route* (phase du vol entre la fin de la montée et le début de l'approche) ainsi que lors de la phase terminale dans certains aéroports. Pour les aéroports les plus importants, il s'agit de la tour de contrôle ou *Control Tower* (TWR) qui s'occupe de la phase d'approche, du décollage et des opérations au sol [34].

Il existe aussi différentes catégories d'aviation qui ont chacune leurs propres réglementations. Outre la partie militaire, Transports Canada définit trois formes d'aviation civile [35] :

- *L'aviation commerciale* : Plus communément appelée le transport aérien, de personne ou de fret.
- *L'aviation à voilure tournante* : Ou hélicoptères. Ce sont tous les appareils qui utilisent des rotors verticaux.
- *L'aviation générale* : Toutes les autres activités qui ne rentrent pas dans la première ni la deuxième catégorie. On y retrouve ainsi les avions privés, de loisir, d'affaires, ainsi que ceux utilisés pour le sport aérien, l'agriculture, etc.

En plus de ces trois types d'aviation, il existe différentes catégories de vol, dont deux principales qui sont le vol aux instruments, ou *Instrument Flight Rules* (IFR), et le vol à vue, ou *Visual Flight Rules* (VFR). Dans le premier cas, la prévention des collisions et le suivi de la trajectoire se font par l'utilisation des instruments de bord, tandis que dans le second cas ils se font par la vue. Ces deux catégories ne sont pas sujettes à la même réglementation concernant l'équipement à avoir à bord, le suivi par les services ATC, l'autorisation de voler en fonction des conditions météorologiques, l'accès à certaines classes aériennes, etc. L'aviation commerciale se fait principalement en IFR tandis que l'aviation générale se fait plus souvent en VFR.

Avant de pouvoir entamer son vol, un pilote doit, sauf exception, avoir préalablement déposé son plan de vol auprès des autorités compétentes. Ce dernier comporte notamment l'identifiant de l'appareil ou le numéro de vol, le type de vol, le type d'appareil, les lieux de départ et d'arrivée avec une estimation des horaires et de la durée de vol, l'altitude et la vitesse de croisière, la route prévue, entre autres informations. Toutes ces informations importantes servent ensuite aux contrôleurs aériens dans leurs opérations de surveillance et de gestion du trafic. Il peut arriver qu'en cours de route le plan de vol soit modifié en fonction des conditions météorologiques, du trafic aérien, ou d'éléments imprévus. Cela se fait toujours avec l'accord du contrôleur aérien et du pilote. Enfin, pour assurer le suivi d'un même appareil au sein d'un secteur aérien, le contrôleur dispose d'une bande de papier ou électronique, le *flight strip*, qu'il remplit avec différents renseignements issus originellement du plan de vol : heure d'arrivée et de sortie du secteur, autorisations données, heures de passage à différents points, remarques importantes, et d'autres. Aujourd'hui, plusieurs *Air Navigation Service Provider* (ANSP) [6] ont remplacé ces *flight strips* par des versions électroniques sur des systèmes informatiques ; on parle maintenant de *Electronic Flight Strip* (EFS).

2.1.3 Éviter les collisions en vol

Bien que la sécurité des aéronefs soit principalement assurée par les contrôleurs aériens, il existe de nombreux espaces aériens qui ne sont pas contrôlés, où il n'y a donc personne au sol pour prévenir les aéronefs de possibles collisions. De plus, il peut arriver des cas où les contrôleurs ne se rendent pas compte d'une situation pouvant aboutir à une collision, à cause d'un fort achalandage, d'une distraction ou d'autres problèmes sur le trafic aérien. C'est pourquoi l'OACI demande à ce que les appareils de plus de 5 700kg, ou ceux transportant plus de 19 passagers, soient munis d'un *Traffic Alert and Collision Avoidance System* (TCAS) [36].

Le TCAS est un équipement embarqué à bord des aéronefs qui permet d'afficher les aéronefs proches de l'avion, de détecter d'éventuelles collisions, de générer des alarmes et, à partir du TCAS II, des consignes pour effectuer des manœuvres d'évitement. Il fonctionne à l'aide des transpondeurs des autres appareils. Comme dans le cas de la détection SSR, le TCAS interroge les autres transpondeurs à portée toutes les secondes à la fréquence de 1 030 MHz qui répondent à la fréquence de 1 090 MHz. À partir de ces informations et de diverses paramètres de l'avion (vitesse, cap, taux de montée, et d'autres), il affiche les appareils à proximité en donnant un indicateur de prévention de collision en quatre niveaux (trafic non conflictuel, trafic à proximité, *traffic advisory*, *resolution advisory*). Les deux derniers niveaux génèrent des alertes. Par exemple, lorsqu'un appareil se trouve à proximité mais

6. Fournisseur de service de la navigation aérienne. Nav Canada est un ANSP pour le Canada.

ne présente pas de danger immédiat et entre dans la zone de *traffic advisory*, il est indiqué par un cercle orange sur le TCAS et une alerte sonore est émise : « *Traffic, Traffic*». En cas de collision imminente, l'indicateur devient un carré rouge et une alerte sonore de type *resolution advisory*, qui contient un ordre de manœuvre qui doit être immédiatement exécutée par le pilote, est émise. Si les deux avions qui risquent d'entrer en collision sont munis d'un TCAS II, alors ceux-ci se coordonnent sur les manœuvres. Par exemple, un appareil recevra la consigne « *Climb, Climb*» (monter à 1 500 - 2 000 ft/min) et l'autre la consigne « *Descend, Descend*» (descendre à 1 500 - 2 000 ft/min). Dès qu'un conflit est terminé, c'est l'alerte « *Clear of conflict*» qui est prononcée.

Le TCAS permet donc de prévenir de possibles collisions entre appareils, et en cas de danger, d'indiquer des manœuvres à suivre pour se mettre en sécurité. Les ordres du TCAS sont prioritaires sur ceux donnés par les contrôleurs aériens suite aux recommandations du bureau fédéral allemand d'enquête sur les accidents aéronautiques (*Bundestelle für Flugunfalluntersuchung*) depuis l'accident d'Überlingen (Allemagne) le 1er juillet 2002 qui a coûté la vie à 71 personnes [37] [7].

2.2 Le réseau Data Distribution Service (DDS), lieu d'échange de données

Les systèmes ATM font face à une grande diversité des sources et types de données dont font partie les plans de vol, les paquets PSR, SSR et ADS-B. Pour faire le lien entre ces différentes sources et utilisateurs, Nav Canada s'appuie sur une norme d'échange de données proposé par le consortium *Object Management Group* (OMG) [8]. Il s'agit du *Data Distribution Service* (DDS). C'est un réseau d'échange de données basé sur le principe de publication/souscription (ou *publisher/subscriber*). Il n'y a pas de relations directes entre les sources de données et les receveurs. Les éditeurs (*publisher*) associent leurs données à des catégories (*topics*) et les publient sur un réseau d'échange de données. De leur côté, les abonnés (*subscriber*) souscrivent à des catégories précises. Dès qu'un message apparaît avec la bonne catégorie, alors ils le lisent.

Le DDS est une norme centrée sur les données. Elles sont regroupées dans un ou plusieurs domaines. Les domaines étant cloisonnés, il n'y a pas d'échanges entre eux. À l'intérieur de ces domaines, les données sont liées à des *topics*. Chaque *topic* a sa propre structure de données. Cela permet à ce que les données soient utilisées de manière uniforme. La communication

7. Lors de cet accident, le contrôleur aérien a demandé à l'un des pilotes de monter alors que son TCAS lui ordonnait de descendre. En suivant les consignes du contrôleur, le pilote a provoqué la collision avec le second appareil, ce qui ne serait pas arrivé s'il avait suivi son TCAS.

8. Organisme à but non lucratif ayant pour but de proposer des normes afin d'aider à l'adoption du modèle objet.

entre applications se fait en deux étapes : la publication et la souscription. Tout d'abord, une source va publier des données dans un ou plusieurs *topics*. Celles-ci doivent être conformes à la structure demandée. Ensuite, toutes les applications qui se seront abonnées à ces *topics* recevront l'information via leur souscripteur [38].

La flexibilité du DDS, son système centré sur les données et le fait qu'il s'agisse d'un moyen d'échange en temps réel ont fait qu'il a été adopté par de grands acteurs du trafic aérien. Nav Canada a ainsi décidé d'utiliser la solution RTI Connext DDS lors de la modernisation de son infrastructure ATM [39]. En Europe, c'est Coflight, le système de traitement des données de vol, ou Flight Data Processor (FDP), qui a choisi d'utiliser OpenSlice DDS pour son fonctionnement et répondre aux objectifs de ciel unique européen définis par le projet SESAR [40, 41].

2.3 L'ontologie, une formalisation des connaissances

En philosophie, l'ontologie est une théorie sur l'être, sur ce qui existe. Le mot ontologie est dérivé du grec *ontos* qui signifie «ce qui est, étant» et de *logos* qui veut dire «traité». Cette discipline qui s'intéresse à la nature de l'être, a inspiré les chercheurs en intelligence artificielle et en web sémantique [9]. En informatique, l'ontologie est une formalisation des connaissances, une spécification des relations entre des entités [42]. D'une façon plus générale, Gruber décrit l'ontologie comme «*a statement of a logical theory*» [43].

Le développement des ontologies est lié à celui du web sémantique. C'est pourquoi les différentes couches et technologies utilisées proviennent de standards définis par le *World Wide Web Consortium* (W3C) [10].

2.3.1 Savoir décrire les ressources

Les ontologies sont basées sur des ressources identifiées par un *Universal Resource Identifier* (URI). Ces identifiants sont uniques pour chaque chose et ressemblent aux *Universal Resource Locator* (URL), sauf qu'ils ne nécessitent pas d'être accessibles sur le web. Toute entité possède un URI unique (par exemple `http://www.example.com/Airport#CYUL`). Pour pouvoir faire le lien entre ces URI, on s'appuie sur un modèle d'échanges de données, le Resource Description Framework (RDF).

Le *Resource Description Framework* (RDF) forme des graphes de connaissance. Il repose sur

9. Appelé aussi web de données, c'est un ensemble de technologies qui permettent aux machines de comprendre la signification des ressources du web et de les utiliser.

10. Organisme à but non lucratif fondé par Tim Berners-Lee dont le but est de normaliser les technologies du web.

un ensemble d'énoncés (*statements*) qui sont des triplets (*sujet,prédicat,objet*). Le prédicat est une propriété qui va caractériser le sujet. Hormis l'objet qui peut être un URI ou un littéral, le sujet et le prédicat doivent être des URI. La Figure 2.3 montre un exemple de graphe RDF. Il pourrait représenter l'information suivante : *Air Canada est une compagnie aérienne dont le siège social est situé à Montréal. Son site web est aircanada.com.* Cela fait quatre triplets dont l'URI `http:www.example.com/Airlines#AirCanada` est le sujet.

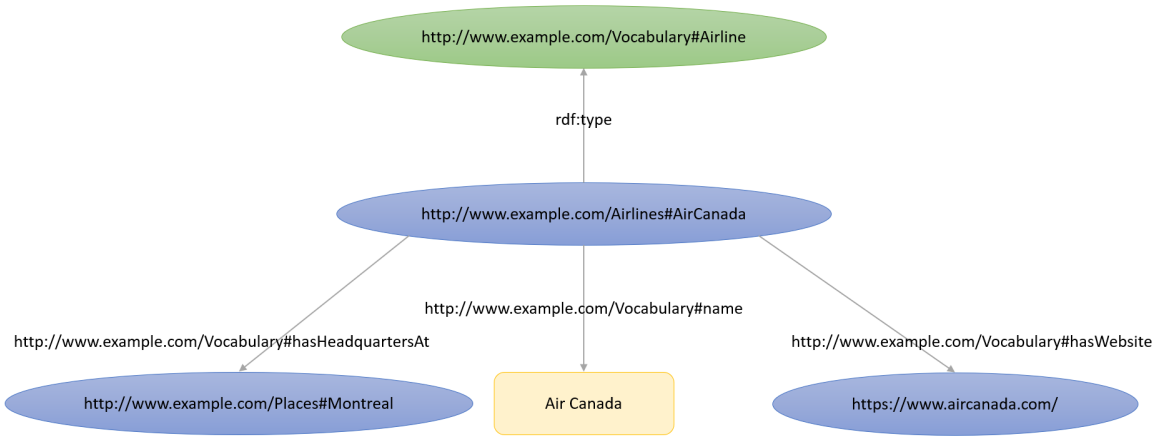


Figure 2.3 Un exemple de graphe RDF

Pour formaliser ces graphes, il existe différentes syntaxes RDF. Les plus utilisées sont RDF/XML, Notation3, N-Triples et Turtle. La syntaxe RDF/XML est lourde pour un utilisateur humain, mais rejoint l'objectif du web sémantique d'avoir un partage et une utilisation des ressources par des applications. Plus compréhensible pour l'humain et allégée, la notation Turtle est souvent choisie pour la création des ontologies. Elle fonctionne avec des préfixes qui permettent de raccourcir les URI et d'éviter ainsi de fastidieuses répétitions. De plus, le prédicat *a* permet de remplacer `rdf:type` qui identifie le type d'une entité. Enfin, lorsque plusieurs triplets ont le même sujet, à la place de devoir le répéter à chaque fois, Turtle permet d'écrire un premier triplet puis avec le symbole «;» on peut ensuite ajouter le prédicat et l'objet d'un autre triplet. La Figure 2.4 traduit en Turtle le graphe présenté ci-avant. Enfin, le W3C propose le *Resource Description Framework Schema* (RDFS) pour décrire les relations entre les ressources. Il introduit la notion d'héritage avec les sous-classes, et apporte des précisions pour les propriétés avec les notions de *domain* et *range*.

Le dernier outil pour représenter une ontologie est l'utilisation d'une logique descriptive, car elle permet d'exprimer des relations plus complexes que l'héritage et l'association entre les différents concepts. Le consortium W3C a émis la recommandation d'utiliser le *Web Ontology Language* (OWL) qui est une extension du RDFS. Il permet de mieux caractériser

```
@prefix :       <http://www.example.com/Airlines#>.
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix voc:    <http://www.example.com/Vocabulary#>.
@prefix places: <http://www.example.com/Places#>.


:AirCanada a voc:Airline;
    voc:name "Air Canada";
    voc:hasHeadQuartersAt places:Montreal;
    voc:hasWebsite <https://www.aircanada.com/>.
```

Figure 2.4 Un exemple de notation Turtle

les relations en introduisant différentes notions de logique. On y retrouve ainsi la conjonction et la disjonction (un objet ne peut pas appartenir à deux classes disjointes), les notions de transitivité et d'inverse et les quantificateurs universel (∀) et existentiel (∃).

OWL se base sur l'*hypothèse du monde ouvert* qui désigne le fait que ne pas connaître une information n'implique pas que celle-ci soit fausse. Cela permet donc une grande souplesse pour la création d'ontologies, mais amène quelques limites comme le fait de devoir préciser les classes qui sont disjointes. En effet, si dans une ontologie on explique que Milou est un chat et qu'il est un chien et que les classes ne sont pas disjointes, ces énoncés sont cohérents d'un point de vue ontologique alors que ce n'est pas le cas dans la réalité.

Dans la dynamique d'universalité et de partage des données du web sémantique, différents vocabulaires ont été créés. Ce sont des ontologies utilisées pour exprimer des données en RDF. On retrouve par exemple *FOAF* qui porte sur la description des personnes et de leurs liens [44], *Dublin Core* qui permet de décrire des documents [45] ou encore *SKOS* pour représenter des classifications, thésaurus et taxonomies [46]. Ces différents vocabulaires sont utilisés par plusieurs centaines de vocabulaires et ontologies [47], ce qui permet d'avoir une uniformité des notions de base et évite à chaque fois de les redéfinir.

2.3.2 Raisonner sur les données

Une des grandes forces de l'ontologie réside dans la notion d'inférence. Il s'agit de pouvoir déterminer une proposition à partir de relations entre des prédicats. Par exemple, si dans une ontologie on a `:packetX a voc:PSRPacket.` (le paquet X est un paquet PSR) et que la relation `ex:PSRPacket rdfs:subClassOf voc:RadarPacket.` (un paquet PSR est un paquet radar) a été définie, alors par inférence on peut en déduire que `:packetX a voc:RadarPacket` (le paquet X est un paquet radar). Cela fonctionne aussi avec les propriétés et sous-propriétés. L'inférence permet donc de découvrir de nouvelles relations qui ne sont pas explicitement indiquées dans l'ontologie.

Pour interagir avec les données, le W3C propose d'utiliser le langage *SPARQL Protocol and RDF Query Language* (SPARQL) [48]. Il est basé sur la syntaxe Turtle et permet d'interroger, de modifier, d'ajouter ou supprimer des données RDF. Il est très semblable au *Structured Query Language* (SQL) au niveau des mots clés et de la construction des requêtes. Par contre, puisque les données RDF sont sous forme de graphes, il n'y a pas de notions de table avec des clés primaires comme pour les bases de données relationnelles. La Figure 2.5 donne un exemple de base de requête SPARQL. Elle se base sur l'exemple présenté dans la Figure 2.3. Il s'agit d'une requête qui cherche les différents sièges sociaux de l'entité `:AirCanada`. Il n'y aura qu'un seul élément de retourné, la ressource `<http://www.example.com/Places#Montreal>`.

```
1  PREFIX : <http://www.example.com/Airlines#>
2  PREFIX voc: <http://www.example.com/Vocabulary#>
3  SELECT ?o WHERE {
4          :AirCanada voc:hasHeadquartersAt ?o
5  }
```

Figure 2.5 Un exemple de requête SPARQL

De plus, comme le RDF est basé sur la notion de triplet, SPARQL permet d'avoir comme variable chaque élément, soit le sujet, le prédicat et l'objet. Ainsi, une requête qui porte sur un seul triplet aura différents sens selon les variables qui auront été choisies. C'est ce que montre le Tableau 2.3 en indiquant tous les résultats qui peuvent être obtenus à partir d'un seul triplet dans la requête. Pour une meilleure illustration, on fait l'hypothèse d'un graphe complet et que la propriété `voc:hasHeadquartersAt` a pour domaine une organisation et que `:AirCanada` soit par inférence une organisation.

2.4 Discussion

Dans ce chapitre, nous avons présenté les principales notions du contrôle aérien, du réseau DDS, et fait une présentation de l'ontologie. Ces éléments sont importants pour la compréhension du reste du mémoire. En effet, nous allons modéliser un environnement ATC et définir des règles de détection sur des situations faisant intervenir ces concepts. De plus, la description succincte de l'ontologie permet de comprendre la méthodologie utilisée pour la modélisation ainsi que le fonctionnement de la détection.

Après avoir présenté ces définitions et concepts de base, nous allons effectuer une revue de littérature sur les différentes mesures proposées par la recherche pour se protéger des attaques contre l'ADS-B.

Tableau 2.3 Sens des requêtes SPARQL selon le choix des variables d'un triplet

Variables	Triplet	Résultats de la requête
?s	?s voc:hasHeadquartersAt places:Montreal.	Toutes les organisations qui ont un siège social à Montréal
?p	:AirCanada ?p places:Montreal.	Toutes les relations entre Air Canada et Montréal (dans ce sens)
?o	:AirCanada voc:hasHeadquartersAt ?o	Tous les sièges sociaux d'Air Canada
?s, ?p	?s ?p places:Montreal	Toutes les entités et relations qui ont rapport avec Montréal
?s, ?o	?s voc:hasHeadquartersAt ?o	Toutes les organisations avec leur(s) siège(s) social(aux)
?p, ?o	:AirCanada ?p ?o	Toutes les relations d'Air Canada
?s, ?p, ?o	?s ?p ?o	Tous le triplets du graphe

CHAPITRE 3 TRAVAUX ANTÉRIEURS SUR LA SÉCURITÉ DE L'ADS-B

Depuis la découverte des vulnérabilités des communications ADS-B, les chercheurs se sont penchés sur des façons de rendre plus sécuritaire le contrôle aérien. Deux grands volets de recherche se distinguent. L'une est concentrée sur la refonte du protocole afin d'y apporter du chiffrement, tandis que l'autre s'intéresse à des outils pour déterminer la légitimité des paquets.

Dans ce chapitre, nous présentons dans un premier temps des solutions cryptographiques pour sécuriser l'ADS-B. Ensuite, nous nous intéressons à différentes méthodes de détection basées sur des caractéristiques physiques ou logiques. Pour finir, nous présentons un aperçu de l'utilisation d'ontologies en aéronautique et en sécurité informatique.

3.1 L'ajout de cryptographie

La sécurité informatique est basée sur trois piliers : la confidentialité, l'intégrité et la disponibilité. La confidentialité traduit le fait qu'une information ne doit être accessible qu'aux personnes qui en sont autorisées. L'intégrité d'une information se vérifie par l'absence d'altération entre l'émission et la réception. Elle va de paire avec l'authentification, qui consiste à garantir l'identité de celui qui émet ou reçoit un message. Enfin, la disponibilité caractérise le fait que les informations soient accessibles et utilisables par les utilisateurs autorisés. Le chiffrement consiste à chiffrer des données afin de les rendre compréhensibles uniquement par des personnes autorisées. Ces derniers doivent connaitre la clé qui permet de passer des données modifiées aux données originales. C'est la clé de déchiffrement. Un chiffrement répond aux objectifs de confidentialité. L'ajout d'un Message Authentification Code (MAC), permet de vérifier l'intégrité d'un message, tandis que des outils comme les signatures numériques apportent la notion d'authentification.

Chiffrer les communications ADS-B est donc un bon moyen d'apporter de la confidentialité pour les communications ADS-B. De plus, l'introduction de MAC et de signatures numériques permet de sécuriser ce protocole en réduisant les vulnérabilités de ce protocole. On distingue deux types de chiffrement : symétrique et asymétrique. Dans le premier cas, une clé unique connue que par l'émetteur et le récepteur est utilisée pour chiffrer et déchiffrer le message. Tandis que dans le second, le message est chiffré en utilisant une clé qui n'est connue que par l'émetteur (clé privée) et le récepteur utilise une clé diffusée par l'émetteur (clé publique) pour le déchiffrer. Bien que plusieurs travaux se soient lancés dans une méthode de chiffrement

symétrique pour l'ADS-B, le chiffrement asymétrique permet une meilleure gestion des clés [49]. Dans cette section, nous présentons une sélection de travaux portant sur ces deux types de cryptographie.

3.1.1 Le chiffrement FFX

L'algorithme FFX a été développé en 2010 par le *National Institute of Standards and Technology* (NIST) [1] [50]. Il a la particularité d'être un chiffrement à préservation de forme. Cela veut dire que le format des données en sortie correspond à celui des données en entrée. Il est très pratique dans le cadre d'un système codifié comme l'aéronautique. Une variante, FFX-A2, est spécialement conçue pour les chaînes binaires de 8 à 128 bits. Ce qui est le cas des messages envoyés par un transpondeur. C'est pourquoi C. Finke *et al.* l'ont modifié pour l'appliquer au cas de l'ADS-B [51]. Les simulations faites montrent que l'entropie des messages est augmentée, dépassant 7,96 bit par octet dans le cas d'un message transmettant la position, l'altitude et l'identifiant OACI. Si cette augmentation d'entropie est suffisante pour les chercheurs, certaines questions restent en suspens. Notamment sur la transmission de la clé. Pour pouvoir déchiffrer le message, le receveur doit connaître la clé utilisée par l'émetteur. Or, si cette clé tombe entre de mauvaises mains, alors cet acteur malveillant peut à son tour chiffrer les messages et créer de faux avions. C'est le principal défaut des méthodes de chiffrement symétrique. Elles nécessitent la mise en place d'un système de gestion de clé précis et sécurisé.

3.1.2 L'utilisation de courtes signatures

De leur côté, Wu *et al.* choisissent d'utiliser de courtes signatures sans certificats qui sont apposés à la fin d'un message pour permettre l'authentification [52]. À chaque nouveau vol, l'aéroport transmet une clé privée partielle à l'avion qui calcule sa propre clé privée et retourne la clé publique associée. Dorénavant, durant le vol, il appose au message ADS-B une signature calculée à partir de la clé privée. Les stations au sol ont accès à la liste de toutes les clés publiques et lorsqu'elles reçoivent un message, elles essaient de le déchiffrer. Si elles peuvent le faire, le message est validé. La signature apposée en fin de message dans leur méthode a une longueur de 1024 bits, soit plus de neuf fois la taille d'un message ADS-B. C'est beaucoup trop pour pouvoir être implémenté dans la réalité.

1. Organisme de normalisation aux États-Unis

3.1.3 La méthode SAT

En 2017, Berthier *et al.* proposent la solution *Security in the Air using TESLA* (SAT) pour authentifier la source des messages ADS-B [53, 54]. Elle est basée sur un schéma asymétrique et utilise le protocole *Timed Efficient Stream Loss-tolerant Authentication* (TESLA) développé par Perrig *et al.* [55]. C'est un mécanisme de rétroauthentification basé sur le temps. L'émetteur a au préalable déterminé une période découpée en intervalles de temps sur laquelle il enverra ses messages. Pour chaque intervalle, il obtient une clé K_i à l'aide d'une fonction à sens unique [2] F en calculant $K_i = F(K_{i+1})$. Pour chaque intervalle, une seconde fonction F' calcule la clé $K'_i = F'(K_i)$ qui sera utilisée pour chiffrer les messages ADS-B et produire leur code d'authentification de message (*Message Authentification Code* (MAC)) correspondant. Lors du premier envoi d'un message ADS-B dans l'intervalle T_i, l'émetteur calcule son MAC avec K'_i et l'envoie sans la clé. De même pour tous les messages de cet intervalle. Dans T_{i+1}, l'émetteur utilise K'_{i+1} pour calculer les MAC des messages envoyés dans cet intervalle-ci et transmet en plus K_i. Le récepteur peut alors utiliser les fonctions à sens unique pour authentifier les messages précédents. C'est donc une forme de rétro-authentification : pendant un laps de temps, l'origine d'un message ne peut être confirmée. Enfin, cette méthode nécessite que les transpondeurs et les systèmes au sol soient synchronisés pour avoir la même référence pour les intervalles de temps. Berthier *et al.* suggèrent donc d'utiliser l'heure donnée par les satellites de navigation (par exemple GPS).

Les paramètres choisis pour implémenter SAT sont conformes avec la règlementation sur l'ADS-B. Des tests ont été effectués entre deux radios logicielles. Le premier SDR servait d'émetteur et le second de récepteur. À la vue des résultats, il semblerait que les bits supplémentaires n'entraînent pas d'erreurs lors de l'envoi de message [54]. L'intérêt de la solution SAT est aussi qu'elle est rétrocompatible avec des transpondeurs qui ne l'utilisent pas. C'est un atout important, car cela permet de commencer à l'utiliser sans perturbation du trafic aérien. Des tests plus poussés sur des transpondeurs sont nécessaires pour confirmer ou infirmer cette observation. Pour terminer, une petite analyse de facteurs humains a été menée auprès de pilotes pour connaître la meilleure façon d'émettre une alarme.

Néanmoins, SAT nécessite auparavant d'avoir défini une *Public Key Infrastructure* (PKI) pour certifier les clés publiques des avions. La solution SAT propose une infrastructure hiérarchique dont l'autorité de certification principale est l'OACI et les secondaires sont les organismes de réglementation aérienne des différents pays (FAA, Transports Canada, etc.). La mise en place d'une telle infrastructure doit se faire en concertation avec les différentes parties prenantes et en créant un projet à l'échelle internationale. Cela demande beaucoup de temps, c'est

2. Fonction dont la connaissance de $F(x)$ ne permet pas de déterminer x.

pourquoi Berthier *et al.* n'envisagent pas un déploiement possible de SAT avant au moins 2030 [54].

3.2 Des méthodes de détection

De façon complémentaire à vouloir modifier le protocole ADS-B, des chercheurs se sont penchés sur des solutions permettant de déterminer la légitimité des paquets envoyés. Ces méthodes sont d'autant plus intéressantes qu'en cas de mise à jour de l'ADS-B, les procédés utilisés seront encore valables et la détection pourra continuer à se faire, palliant ainsi de possibles nouvelles vulnérabilités ou des menaces résiduelles qui ne disparaîtront pas avec l'ajout de telles modifications au protocole, telles que les menaces internes. Cette section recense des travaux qui ont été faits pour détecter des attaques ADS-B en s'appuyant sur des principes physiques, logiques ou sur de l'intelligence artificielle.

3.2.1 La multilatération et l'effet Doppler

La multilatération (MLAT, ou la *Wide Area Multilateration* (WAM) pour les zones en-route) est une technique qui commence à être implémentée de plus en plus dans les systèmes ATC [24]. Elle permet d'éviter d'avoir une couverture partielle des radars secondaires à cause de la rotation de leur antenne, d'avoir un système de détection redondant et une grande précision de la position des avions grâce à un maillage complet d'antennes unidirectionnelles. Avec la compatibilité prévue pour des transpondeurs émettant de l'ADS-B, la multilatération devient un précieux atout pour la détection de faux paquets. En effet, la MLAT détermine la source d'un paquet en calculant la différence de temps d'arrivée à chaque antenne. C'est sur ce principe que Schafer *et al.* établissent un moyen de détection de faux paquets ADS-B [56]. Un message dont la position ne correspond pas à celle calculée par le maillage MLAT devient suspect. En théorie, cela réduit le champ d'action des acteurs malveillants qui doivent se placer là où ils situent leur faux avion. Cependant dans les faits, la multilatération présente de grandes faiblesses dans la partie en-route. Une analyse de cas [57] faite sur un réseau de huit antennes qui ont reçu 50 millions de paquets ADS-B d'avions en vol de croisière (soit 12 000 m), a montré que seulement 5% des paquets ont été reçus par quatre antennes différentes, ce qui est nécessaire pour faire de la multilatération. La MLAT nécessite un maillage important d'antennes ; avec au moins quatre receveurs pour sécuriser une zone. Cela a un coût important en termes d'installation et de maintenance. De plus, des endroits comme les déserts et les océans ne peuvent pas prétendre à une couverture MLAT. Enfin, il est nécessaire qu'il y ait une synchronisation générale, car si deux antennes n'ont pas le même temps de référence, cela engendre des erreurs de calcul et de possibles faux positifs.

Pour remédier au problème d'absence de couverture des endroits désertiques et océaniques, une solution pour faire un travail similaire à la multilatération a été proposée par Sampigethaya et Poovendran [58]. Cette fois-ci, ce sont les avions eux-mêmes qui s'occupent de la détection. Un groupe d'avions qui sont à portée les uns des autres et qui volent à une vitesse similaire forme une cellule de base de détection. Un leader est désigné et sert de repère de référence. Dès qu'un message ADS-B est reçu par un avion, ce dernier envoie au leader l'heure d'arrivée du message. En utilisant la méthode des temps d'arrivée, comme pour la MLAT, le leader est capable de déterminer la provenance du message. Cependant, cela nécessite qu'il y ait toujours des groupes d'avions qui soient à portée pour assurer une surveillance complète sur l'espace aérien. De plus, cela pose des problèmes pour désigner le leader de groupe. Cette solution exige une communication abondante entre les appareils et donc une surcharge du réseau de communication qui sera utilisé.

Une façon de pallier les problèmes de la MLAT en gardant cette idée de détection multisite est d'utiliser l'effet Doppler [59]. Ce n'est pas la différence de temps d'arrivée qui est calculée, mais l'écart de fréquence entre celle envoyée et celle reçue. En effet, un déplacement radial de la source crée un léger décalage de fréquence au niveau du récepteur. Et même si un attaquant fixe veut moduler sa fréquence pour tromper une antenne, il ne pourra pas tromper les différentes antennes d'un maillage, car elles ne devraient pas recevoir le même décalage de fréquence. Même si cette solution a l'avantage de ne pas avoir à synchroniser les antennes entre elles, elle reste quand même dépendante d'un maillage de capteurs, ce qui peut difficilement se faire sur les zones désertiques ou océaniques. Une variante, étudiée par Ghose et Lazos [60], s'intéresse à l'étalement du spectre du signal reçu. Elle vérifie la cohérence de la position et de la vitesse du message ADS-B en s'appuyant sur une analyse de la vitesse radiale entre l'émetteur et le récepteur. L'idée principale est que le déplacement d'un avion par rapport à une antenne devrait se ressentir sur la fréquence des messages reçus, avec une légère variation en fonction de la vitesse et de la position de l'émetteur. Leurs résultats montrent que les attaques réussissent dans moins de 1% des cas et que ce taux diminue en augmentant le nombre de messages pris en compte. Leur solution a entièrement été simulée. Il reste donc à utiliser de vrais capteurs et paquets pour pouvoir vérifier que les interférences naturelles ne dégradent pas sensiblement la qualité de la détection.

3.2.2 L'intelligence artificielle

Certains auteurs se sont intéressés à l'utilisation de l'intelligence artificielle pour détecter de faux messages ADS-B. Ainsi, Habler et Shabtai ont développé une solution de détection basée sur l'algorithme d'apprentissage machine *Long Short-Term Memory* (LSTM) [61]. C'est un

réseau de neurones qui essaie d'établir un motif dans une série de données. Dans la famille des algorithmes LSTM, il y a l'encodeur-décodeur. Il fait le traitement dit *de séquence à séquence*, c'est-à-dire une séquence d'un domaine est traduite en celle d'un autre domaine. Par exemple la traduction d'une langue en une autre. L'encodeur-décodeur fonctionne de la manière suivante. Premièrement, l'encodeur s'occupe d'extraire le motif qui caractérise une séquence de données (par exemple «It's a plane !»). Le vecteur obtenu est transmis au décodeur avec une consigne (par exemple «C'est un avion !») et cherche à trouver le motif entre son vecteur d'entrée et la consigne. L'idée de ces deux chercheurs est d'entraîner un algorithme LSTM afin de reconstruire des séquences de vol. Ils partent du principe qu'entre deux aéroports ce sont presque tout le temps les mêmes routes qui sont utilisées. En entraînant leur algorithme sur des vols légitimes, ils montrent que lorsque des vols subissent des modifications de paquets, alors le LSTM ne reconstruit pas bien la séquence de données. En calculant ces anomalies, ils parviennent à détecter des attaques. Cette méthode ne permet donc pas la détection d'attaques par rejeu. De plus, les attaques perpétrées dans leurs tests sont peu subtiles (ils modifient les paquets d'un vol en Europe par ceux d'un vol en Thaïlande) et donc on ne connaît pas la tolérance du LSTM face à de petites variations. Enfin, pour établir un diagnostic, il est nécessaire d'avoir toutes les positions de l'avion depuis son départ. Cela veut dire que les messages ADS-B doivent être partagés entre les différents services de contrôle aérien à travers le monde. Nous allons étudier en détail cette solution et la comparer à la nôtre dans le chapitre 6.

Cette idée de trouver des incohérences par le LSTM a abouti à une variante qui s'intéresse à des zones géographiques [62]. Cette fois-ci, le récepteur construit une image 2D à partir des informations des avions qu'il reçoit pendant un temps Δt. Ces avions sont représentés par des flèches autour du récepteur. La latitude et la longitude donnent la position de la flèche, le cap est indiqué par sa direction, l'altitude est traduite par sa taille et la vitesse par sa longueur. La partie encodeur du LSTM essaie de trouver un motif dans une série d'images et le décodeur essaie de reconstruire les images d'origine. Le gros avantage de cette méthode par rapport à la précédente version du LSTM c'est qu'elle utilise des données locales. La détection s'appuie uniquement sur les informations que son capteur reçoit. Les résultats des expériences donnent de forts taux de fausses alarmes (jusqu'à 10% pour des attaques injectant de faux avions pour $\Delta t = 2s$). C'est beaucoup trop pour être utilisé par des contrôleurs aériens, car cela va encombrer leur travail par des alertes inutiles sur près d'un dixième des vols.

Les algorithmes d'encodeur-décodeur comme le LSTM ne forment qu'une des approches de l'intelligence artificielle utilisées pour faire de la détection d'anomalies. Chandola *et al.* ont élaboré une étude détaillant les différentes branches de la détection d'anomalies [63]. Elle est complétée par une autre étude, proposée par Chalapathy et Chawla, qui s'intéresse aux algo-

rithmes d'apprentissage en profondeur [64]. Les techniques de détection d'anomalies peuvent être distinguées en trois catégories :

- *Détection supervisée.* À partir d'un jeu de données correctement étiqueté (c'est-à-dire dont on sait si une donnée est légitime ou si elle présente une anomalie), les algorithme utilisant l'apprentissage supervisé créent un modèle prédictif pour classer toute nouvelle donnée.
- *Détection semi-supervisée.* C'est à cette catégorie qu'appartient le LSTM et tous les algorithmes avec le principe d'encodeur-décodeur. Dans l'apprentissage semi-supervisé, on dispose d'un jeu de données légitimes et on créé et entraîne un modèle afin de pouvoir prédire ces données. L'idée principale est que lorsqu'une donnée présentera une anomalie, l'algorithme prédictif obtiendra des valeurs aberrantes qui peuvent être quantifiées. Ainsi, dans le cas du LSTM, l'injection d'une anomalie dans une série de messages ADS-B se traduit par une mauvaise reconstruction de la trame originale.
- *Détection non supervisée.* Dans cette dernière catégorie, les données ne sont pas étiquetées. Les algorithmes essaient de déterminer des caractéristiques et des structures sur le jeu de données afin de pouvoir observer des valeurs aberrantes.

Ainsi, il existe différentes applications de l'intelligence artificielle pour détecter des anomalies. Bien qu'elles aient été testées dans différents domaines comme la fraude, l'intrusion sur des réseaux, ou encore les attaques sur des objets connectés, elles sont encore peu employées pour la détection de faux paquets ADS-B.

3.2.3 Un système de détection par contraintes

Tout récemment, une étude s'est servie du travail effectué par Morel [21], pour développer un système de détection d'intrusion, ou *Intrusion Detection System* (IDS). Dans son mémoire de maîtrise [65], Babar présente une méthode de détection basée sur des contraintes. En partant de requêtes SPARQL sur une ontologie contenant des concepts de haut niveau sur le contrôle du trafic aérien, elle développe des contraintes bas niveau qui sont appliquées dès la réception d'un paquet émis par un radar. Dans un système ontologique, la détection par requête SPARQL se fait *a posteriori*. Cela veut dire que les données sont enregistrées dans la base de données que les règles de détection vont interroger. Dans l'IDS par contraintes, lorsqu'un paquet arrive sur le réseau d'échanges de données du service ATC, il est analysé par les contraintes. Si le test est réussi, il passe, sinon une alerte est générée. C'est une solution qui s'applique bien à de simples requêtes. Mais pour des cas plus complexes qui nécessitent d'avoir recours à des données enregistrées (correspondance du plan de vol, cohérence physique entre une succession de paquets, et d'autres), les contraintes doivent être capables d'aller chercher ces données et les interroger.

3.3 L'ontologie au service de la sécurité informatique et de l'aviation

Une partie des projets de modernisation des systèmes de gestion du trafic aérien en Europe (SESAR) et aux États-Unis (NextGen) s'intéresse à la gestion et au partage des données entre les différentes infrastructures ATM. L'utilisation d'ontologies a été étudiée dans divers travaux et s'est révélée une approche prometteuse. Le projet BEST du consortium SESAR a ainsi abouti sur la création d'une ontologie représentant les systèmes ATM [66]. Cette ontologie est complète et donne ainsi du sens à tous les échanges de données entre les différents systèmes et centres.

De leur côté, Insaurralde et Blash proposent une ontologie pour les systèmes NextGen [67]. Elle est centrée sur l'aide à la décision pour les services ATM. Cette ontologie est relativement simple puisqu'elle ne contient que cinq classes (*Aircraft*, *Airport*, *Airspace*, *Route* et *Weather*) ayant chacune entre quatre et sept axiomes. Ils l'ont testé sur trois scénarios. Le premier permet de connaître l'impact de la météo sur un vol en cours et propose aux contrôleurs des routes alternatives. Le second vérifie si les pistes d'atterrissage sont accessibles pour savoir si un avion peut atterrir ou doit être dérouté. Et le troisième vérifie que la séparation réglementaire entre deux avions est bien respectée. Ces scénarios ont été testés sur des cas très simples avec peu de données (quatre aéroports, quatre routes et six avions). Bien que la preuve de concept soit concluante pour leurs scénarios, cette ontologie n'est pas suffisante pour y faire de la détection. En effet, il y a trop peu de concepts. De plus, les règles étudiées ont été testées directement sur Protégé, c'est-à-dire sur une ontologie fixe.

Plus complète, l'ontologie proposée par Keller s'approche de celle du projet SESAR [68]. Elle a pour vocation d'être utilisée pour la gestion du trafic aérien. Elle modélise la gestion d'un vol sous différentes formes : équipement, préparation et exécution du vol. Les infrastructures aéroportuaires sont extrêmement détaillées, incluant aussi bien les pistes et les différents revêtements de surface que les marquages au sol ou encore les portes d'embarquement. Ce qui intéresse les services ATM ce sont les caractéristiques des différents vols prévus : type d'appareil, moyens de communication possibles, équipage et route prévue, et bien d'autres. Ce sont ces informations qui les aident à planifier l'occupation de l'espace aérien. Cette ontologie, bien que détaillée pour l'ATM, n'est pas destinée aux services ATC. Ainsi, différentes notions importantes pour le contrôle aérien, comme les radars ou les types de messages reçus, ne sont pas présentes. C'est pourquoi elle n'est pas utilisable en tant que telle pour faire de la détection d'attaques sophistiquées contre les communications ADS-B.

Plusieurs travaux aboutissent à la création d'ontologies pour représenter les concepts clés de la sécurité informatique. Une étude nommée *Ontologies for Security Requirements : A Lite-*

rature Survey and Classification [69] recense les principales publications du domaine afin de les évaluer. Il existe différents types d'ontologies : des générales et d'autres plus spécifiques. Les ontologies générales décrivent les concepts de la sécurité informatique dans les moindres détails, sans pour autant les appliquer à une situation particulière. Par exemple, dans *An Ontology of Information Security* [70], la reconnaissance vocale est représentée comme moyen de contremesure et est obtenue à partir de la hiérarchie de classe suivante : *Countermeasure > Login System > Biometric Authentification > Behavioural Biometric Recognition > Voice Recognition*. Il s'agit ainsi d'une taxonomie de la sécurité informatique. Les ontologies plus spécifiques modélisent des concepts de sécurité informatique appliqués à un réseau ou des sites web. Par exemple, Undercoffer *et al.* proposent une ontologie pour les IDS [71]. Elle permet de détecter des attaques de déni de service ou par débordement de tampon en créant différentes règles. L'ontologie et les règles de détection sont une transposition des systèmes de détection d'intrusion dans un modèle sémantique. Cette recherche montre que l'on peut utiliser l'ontologie à des fins de détection en tant que système expert. D'autres chercheurs se sont penchés sur l'utilisation d'ontologies en tant que système de détection d'intrusion. Par exemple, Ducharme propose un système expert, appelé *Détection d'intrusion avec l'ontologie par un système expert* (DIOSE), basé sur les ontologies [72]. Cette solution comporte une base de données ontologique qui contient les évènements enregistrés par le système expert et une machine à état qui en reconstituant les évènements décrit de potentiels scénarios d'attaques. Ainsi,en faisant la corrélation entre les événements enregistrés, leur contexte et des informations sur les vulnérabilités, il permet ainsi d'améliorer la détection d'attaques informatiques. Il construit son modèle ontologique à partir de la méthode *Abstractions Translation Ontology Method* (ATOM) proposée par Malenfant-Corriveau qui aboutit à un système expert en partant de requêtes définies en langage naturel et des informations brutes [73]. Enfin, à partir de l'idée d'utiliser les informations provenant de différents capteurs afin d'améliorer la détection, Sadighian *et al.* proposent une solution de détection basée sur la fusion d'alertes provenant d'IDS en s'appuyant sur une base de données ontologique qui contient les informations venant de différents capteurs et une liste des vulnérabilités [74]. Cette solution a pour vocation d'être déployée dans de grandes infrastructures informatiques qui possèdent différents types d'IDS relevant différents types d'attaques informatiques. C'est un élément centralisateur. Toutes ces ontologies ne prennent pas en compte l'environnement du contrôle aérien, ou un environnement similaire, et ne peuvent donc pas être utilisées directement pour la détection de faux paquets ADS-B.

Une ontologie alliant à la fois aviation et cybersécurité a été proposée par Massaci *et al.* en 2011 [75]. Elle a été développée pour répondre à des attaques de type *GPS Spoofing*. On retrouve bien les concepts principaux d'un environnement de contrôle aérien : avion, station

au sol, opérateur ATC, radars primaires et secondaires, satellites. Mais une fois encore elles ne décrivent qu'une partie des concepts de la circulation aérienne. De plus, en choisissant de se spécialiser sur un type d'attaque particulier, les auteurs limitent la portée de leur ontologie qui sera donc à retravailler pour inclure de nouvelles menaces. L'idéal est donc d'avoir une ontologie générale à partir de laquelle on peut détecter différents types d'attaques. Enfin, comme plusieurs des travaux précédents, cet article propose une architecture sans apporter de méthode de détection et de tests. Il est difficile de juger de la fiabilité et de la qualité de l'ontologie dans un système de défense.

3.4 Discussion

Dans ce chapitre nous avons passé en revu différents travaux portant sur la sécurisation et la détection de l'ADS-B. Apporter de la cryptographie et de l'authentification dans les messages ADS-B nécessite de modifier le protocole, ce qui prend plusieurs années pour être certifié et déployé à l'échelle internationale. Pour apporter une solution à moyen-terme, des travaux se sont penchés sur la détection de faux messages ADS-B. Les méthodes utilisant la multilatération exigent la mise en place de nombreuses antennes pour créer un maillage de récepteurs, ce qui est une forte contrainte. L'intelligence artificielle semble prometteuse, mais les tests sont effectués sur des cas simples, avec des attaques peu subtiles et donnent des taux élevés de faux positifs. Une solution se basant sur des contraintes a été proposée, mais elle est limitée pour des attaques sophistiquées qui nécessitent de faire appel à des données enregistrées. Enfin, nous avons remarqué que l'ontologie est déjà utilisée pour modéliser des systèmes liés à l'aviation, mais qu'elle est peu utilisée pour faire de la détection. C'est donc une voie à explorer, d'autant plus qu'il a été montré que l'ontologie peut servir de système expert pour des systèmes comme les IDS.

Le chapitre suivant sera consacré à la méthodologie employée durant notre recherche. Il décrira les différentes attaques contre les communications ADS-B que nous avons relevé, ainsi que les processus utilisés pour l'élaboration du modèle ontologique et des règles de détection.

CHAPITRE 4 DÉTECTION DES ATTAQUES SOPHISTIQUÉES CONTRE LES COMMUNICATIONS ADS-B

À travers ce chapitre, nous nous intéresserons aux différentes attaques sophistiquées possibles contre les communications ADS-B. Ensuite, nous détaillerons l'obtention du modèle ontologique représentant l'écosystème du contrôle aérien qui est utilisé pour modéliser ces attaques et représenter l'environnement sur lequel elles influent. Enfin, nous présenterons le moyen choisi pour permettre la détection de ces attaques.

4.1 Des attaques sophistiquées contre les communications ADS-B

Les vulnérabilités sur les communications ADS-B étant connues, il est nécessaire dans un premier temps de déterminer les menaces et impacts que peut avoir une attaque ADS-B sur le trafic aérien. Pour cela, on définit une menace comme étant le couple (*Acteur,Scénario*) et un scénario comme une suite d'actions pouvant avoir des conséquences sur le trafic aérien.

Les acteurs pouvant être nombreux et inconnus, il faut alors les catégoriser. L'approche choisie est centrée sur ce qu'ils cherchent à obtenir en exploitant les vulnérabilités de l'ADS-B. En effet, la cause et le moyen d'action utilisé influent sur la façon dont l'attaque est menée. On distingue ainsi quatre types d'acteurs :

Amateur : Une personne qui a découvert une vulnérabilité et qui veut l'exploiter par curiosité. Ses connaissances du domaine sont assez limitées. Il ne cherche pas à causer du tort au trafic aérien, mais plutôt à satisfaire son ego. Un exemple serait une personne qui, comme défi personnel ou pour impressionner des amis, voudrait faire apparaître des faux avions sur *flightaware*.

Professionnel : Désigne quelqu'un qui souhaite attaquer les services ATC pour des raisons pécuniaires. Ses attaques visent à perturber et surtout forcer à arrêter le travail des contrôleurs afin d'obtenir une rançon pour un retour à la normale. On peut y retrouver des organisations cybercriminelles qui sont connues pour extorquer de l'argent via le *fishing* ou le *ransomware*.

Activiste : Cette catégorie désigne une personne ou un groupe qui souhaite causer des préjudices financiers à l'aviation. On peut penser à des groupes écologistes extrémistes. En effet, perturber le trafic aérien, voire fermer des espaces aériens implique des coûts supplémentaires aux compagnies aériennes. Dans le cas où une attaque forcerait un avion à une escale technique, cela représente une perte d'au moins 30 000 $ pour un

avion long-courrier comme l'A321LR [76].

Terroriste : Le terroriste se distingue des autres acteurs par une volonté de faire passer son message politique ou idéologique par des actions violentes voire létales. Avec des attaques ADS-B il pourrait ainsi provoquer la perte d'un appareil en perturbant lourdement le trafic aérien de telle sorte à ce que des avions tombent en panne de kérosène. De plus, si les missions de contrôle ne peuvent pas être assurées correctement dans les zones fortement achalandées, les risques de brusques manœuvres ou collisions deviennent plus élevés. En induisant le TCAS d'un avion en erreur en créant des situations de collision, l'attaquant force le pilote à effectuer des manœuvres d'évitement et déroute ainsi l'appareil. Ce cas est d'autant plus grave que les ordres du TCAS sont prioritaires sur les consignes des contrôleurs aériens.

4.1.1 Types d'attaques

Après avoir identifié les acteurs potentiellement malveillants, il faut déterminer les attaques qu'ils peuvent entreprendre sur l'environnement ATC via l'ADS-B. Pour commencer, il y a quatre principales façons de profiter des faiblesses du protocole ADS-B : l'injection de données, la modification de paquets, le brouillage des communications et la suppression de paquets. La modification et la suppression de paquets sont des attaques extrêmement difficiles à réaliser. Elles nécessitent d'avoir accès au transpondeur de l'avion et d'avoir modifié son signal de sortie ou bien détruire le signal au niveau des antennes de réception via des interférences destructives et d'en créer un nouveau.

Ainsi, le principal danger est l'injection de faux paquets. Il se décline de différentes manières.

Modification de trajectoire : En plaçant un faux avion proche d'un véritable appareil, ou en l'ajoutant à des points névralgiques des routes aériennes, les contrôleurs aériens peuvent procéder à modifier la trajectoire d'un vrai avion afin d'éviter une quelconque collision.

Duplication : La duplication consiste à reproduire une ou plusieurs fois les paquets d'un avion en vol en y modifiant quelques données. Cela fait apparaître un même avion à différentes positions. Cette attaque risque surtout de gêner le contrôleur aérien dans son travail en lui demandant une attention particulière pour identifier le vrai point radar (en demandant confirmation par radio) et le suivre ensuite.

Anticipation de vol : C'est un cas particulier de la duplication. Cette attaque demande de connaître l'état du trafic aérien et les vols qui sont prévus lors d'une journée. Il s'agit d'anticiper le passage d'un avion d'une zone sans couverture ADS-B à une zone

couverte. Sur les écrans radars, les contrôleurs observent alors un avion qui entre dans leur espace aérien sans autorisation. C'est une autre source de confusion.

Déni de service : Cette attaque vise à paralyser le trafic aérien. Il y a plusieurs degrés d'intensité et de subtilité. Le premier consiste à ajouter suffisamment de faux avions pour faire croire qu'un secteur aérien a atteint sa capacité maximale et provoquer sa fermeture temporaire. Le niveau maximal peut-être atteint avec une attaque coordonnée visant plusieurs secteurs aériens voire ceux d'un pays entier en rendant inutilisables les écrans des contrôleurs aériens. Si la sécurité des passagers et membres d'équipage n'est pas garantie, alors les secteurs aériens sont fermés et les avions cloués au sol, ce qui a un énorme coût économique pour les compagnies aériennes. Par exemple, lors des attentats du 11 septembre 2001, l'espace aérien américain a été entièrement fermé. Cela a forcé les autorités canadiennes à lancer l'*opération ruban jaune* visant accueillir en urgence 224 vols intercontinentaux, les autres ayant été déroutés vers leurs aéroports d'origine [77]. Un scénario similaire causé cette fois par des attaques ADS-B aurait un impact financier énorme.

La principale conséquence des injections de faux paquets ADS-B est donc la confusion des contrôleurs aériens. Cette confusion peut amener à des situations dangereuses, car elle risque de leur faire oublier des informations importantes. Le *Handbook of Aviation Human Factors* [78] désigne la distraction à cause d'informations similaires comme la principale cause d'oubli d'information importante. En se concentrant sur la situation inhabituelle liée à l'attaque, les contrôleurs perdraient momentanément de vue les autres situations, ce qui mènerait à des pertes de séparation voire des possibles collisions. De plus, en ciblant particulièrement un appareil et en lui injectant des faux paquets à proximité, les attaquants peuvent faire réagir son TCAS afin de le dérouter et de créer dans des situations dangereuses, par exemple en provoquant une série de manœuvres qui amènent l'appareil vers le sol. Cependant, la conséquence la plus dangereuse est le déni de service, car non seulement les avions ne pourraient plus se fier aux contrôleurs aériens pour voler en toute sécurité, mais en plus le TCAS de nombreux avions pourraient réagir, créant ainsi des manœuvres d'évitement désordonnées.

4.1.2 Un espace aérien sous tension

Le trafic aérien est inégalement réparti dans l'espace aérien. Les zones à forte concentration d'avions ainsi que les points d'entrée de l'espace contrôlé seraient des cibles privilégiées pour les attaquants. Trois zones se démarquent du lot : l'arrivée océanique, les zones de transition entre contrôleurs aériens et les zones terminales près des grands aéroports.

Le Canada est bordé par deux océans, le Pacifique et l'Atlantique. L'océan Atlantique contient

un trafic aérien important. Il est divisé en sept FIR dont les principales sont Gander Oceanic (Canada) et Shanwick Oceanic (Irlande et Écosse). C'est par là que passent la majorité des liaisons entre l'Europe et l'Amérique du Nord. Comme le trafic aérien est élevé, différentes routes, dont les routes atlantiques nord, sont définies afin d'assurer la sécurité du trajet des avions. En effet, il n'y a pas de contrôle radar sur l'océan. La position des avions est relevée environ toutes les 15 minutes via l'ADS-C. Il est donc possible d'usurper l'identité d'un avion avant que celui-ci ne soit à portée des radars. Toutefois, l'arrivée de l'ADS-B satellitaire permettra une détection dans les zones océaniques, ce qui réduira la possibilité de faire des attaques par anticipation.

Une autre frontière sensible est la transition entre deux contrôleurs aériens, et plus généralement à l'entrée d'un secteur aérien contrôlé. Si l'acteur malveillant crée un faux avion qui n'a pas reçu l'autorisation de pénétrer dans le secteur aérien, le contrôleur va essayer de l'appeler en continu, quitte à devoir passer la main à l'armée de l'air si cette intrusion pose des problèmes de sécurité. Les attaques par anticipation ou duplication peuvent aussi être perpétrées afin de gêner le travail du contrôleur.

Enfin, parmi les endroits les plus sensibles se trouvent les zones terminales. Dans les aéroports à fort achalandage, des procédures ont été mises en place pour réglementer les arrivées et les départs, ce qui permet de faire une gestion plus efficace et sûre des flux entrants et sortants. Parmi celles-ci, on trouve les *Standard Instrument Departure* (SID), qui déterminent la démarche à suivre entre le décollage et la partie en-route d'un vol, et les *Standard Arrival Route* (STAR) qui permettent de passer de la phase en-route à un point d'approche. Chaque aéroport peut avoir plusieurs STAR/SID en fonction de l'origine (ou la destination) de l'avion et de la piste choisie. Ces procédures sont indiquées lors de l'émission d'une autorisation de vol aux instruments et peuvent être modifiées par le contrôleur aérien en fonction de l'état du trafic.

Ces procédures sont constituées d'une liste de points de navigation, ou *waypoints*, avec des indications sur l'altitude maximale et minimale, ainsi que la vitesse maximale et minimale à avoir. En période de pointe, près d'une dizaine d'appareils peuvent se trouver sur la même STAR. La violation d'une des contraintes peut avoir un grand impact sur la circulation aérienne comme la mise en circuit d'attente pour des appareils, voire un détournement de la circulation qui peut se faire de façon brusque.

4.2 Établir le modèle ontologique

Pour détecter les attaques contre les communications ADS-B, nous avons choisi d'utiliser un système expert basé sur des ontologies. Il faut dans un premier temps modéliser l'écosystème du contrôle aérien.

Dans son mémoire, Louis-Philippe Morel [21] propose une première ontologie à utiliser pour la sécurité du contrôle aérien qui est relativement basique. Elle est composée autour de trois grands concepts : la position, le plan de vol et l'appareil. Elle est incomplète parce qu'elle écarte des concepts importants (comme les infrastructures) et certaines entités ne sont pas entièrement détaillées (il n'y a pas de distinction entre les différents modes SSR, par exemple).

Une seconde ontologie, plus complète, a donc été développée. Pour cela, nous avons utilisé un processus qui est un dérivé de la méthode ATOM proposée par Malenfant-Corriveau [73]. Cette méthode choisit de partir d'un haut niveau avec des questions en langue naturelle que l'on souhaite poser à l'ontologie. En traduisant ces questions en requêtes SPARQL et en déterminant les règles de traduction entre les informations brutes et le requêtes, cela permet de déterminer les concepts et relations qui sont utilisés. Ainsi, ce processus permet d'enrichir l'ontologie à chaque question. Cette méthode permet d'utiliser des ontologies sur mesure dans des systèmes qui peuvent être complexes à modéliser. Néanmoins, la qualité de l'ontologie résultante dépend uniquement des questions posées, et il peut y avoir des omissions de concepts importants nécessaires pour les règles subséquentes. Chaque nouvelle règle peut donc déboucher sur une mise à jour de l'ontologie.

La figure 4.1 montre le processus en quatre étapes qui a été suivi et permet l'élaboration du modèle ontologique. Celui-ci se déroule en présence de spécialistes du domaine du contrôle aérien et de l'ontologie qui apportent leur expertise à chaque étape.

4.2.1 Définir le cadre à modéliser

Le plus grand problème avec un environnement aussi complexe que celui du contrôle aérien c'est de savoir ce qui doit être modélisé ou pas. Vouloir tout modéliser représente dans notre contexte une perte de temps et d'efficacité, car il y a sûrement de nombreux concepts qui seront superflus. Pour faire une bonne sélection, il faut d'abord déterminer l'utilisateur du modèle. Pour les contrôleurs aériens, la perception du trafic aérien est importante dans son aspect global, tandis que pour les pilotes c'est important d'un point de vue local via le TCAS. Bien que la problématique de l'insécurité de l'ADS-B puisse avoir un impact important sur le pilotage à cause de son utilisation par les systèmes TCAS, nous limitons nos travaux aux systèmes ATC.

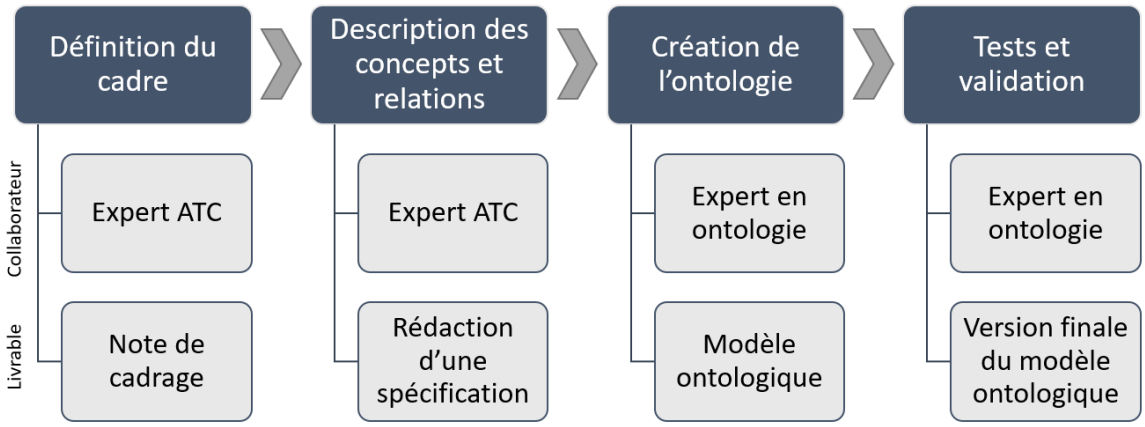


Figure 4.1 Processus de réalisation du modèle ontologique

Ainsi, le modèle ontologique est tourné du point de vue du contrôleur. Or, il en existe différents types, comme le précise le chapitre 2. Les sources de données et d'informations pour les contrôleurs sont aussi différentes. Cela a des répercussions sur la modélisation et la génération des attaques. Nous allons alors nous concentrer sur le contrôle des aéronefs lors des différentes phases de vol, depuis le décollage jusqu'à l'atterrissage en omettant le contrôle au sol.

4.2.2 Les concepts et relations de l'environnement ATC

Cette seconde étape consiste à formaliser et définir les concepts clés du contrôle aérien. Elle est réalisée avec l'aide d'experts. Chaque entité doit être détaillée avec une description générale et ses propriétés. Comme l'ontologie est relationnelle, les liens qu'ont les ressources entre elles sont indiqués (associations). Ce travail doit aboutir à une spécification exhaustive à partir de laquelle l'ontologie sera créée. Pour aider à l'établissement d'un tel document, les entités sont obtenues du point de vue du contrôleur en fonction de trois critères : ce qu'il sait (l'environnement), ce qu'il observe (les données) et ce qui lui permet d'observer (les sources).

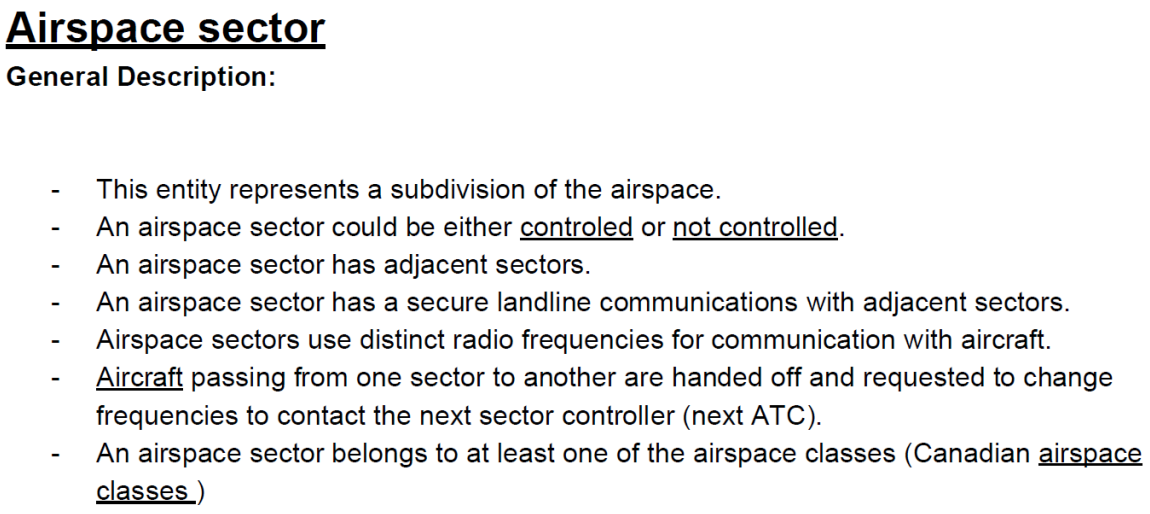
L'environnement : Dans ce terme générique, on a regroupé tout ce qui est extérieur aux infrastructures ATC. L'espace aérien est ainsi détaillé avec les différentes notions apportées par la législation (secteur aérien, classe, niveau de vol), de même que tout ce qui est lié à la notion de vol. Il y a aussi les utilisateurs (aéronefs), les règles de vol et tout ce qui touche au trajet (aéroport, route, *waypoint*).

Les données : Ce sont toutes les informations que le contrôleur reçoit et qui lui permettent de faire son travail. Les plus utilisées sont notamment le plan de vol et les paquets PSR/SSR/ADS-B, mais on y retrouve aussi les *flight strips* qui commencent à être informatisées. Par contre, c'est plus compliqué au niveau des communications

avec les pilotes. En effet, celles-ci se font essentiellement par voix et sont donc difficilement exploitables. Dans certains cas néanmoins les communications se font via le service *Controller Pilot Data Link Communications* (CPDLC) qui permet d'échanger des données comme les autorisations de vol et d'autres informations non urgentes. Il faut aussi spécifier le format de chaque donnée et les définir, même si ce sont des concepts simples comme l'altitude ou la longitude.

Les sources : Il s'agit des infrastructures physiques et des équipements. Cette catégorie décrit plutôt les caractéristiques techniques (comme la vitesse de rotation pour les radars ou la période d'envoi des messages sur le transpondeur).

La rédaction de la spécification se fait en plusieurs temps. Tout d'abord, les entités obtenues grâce aux catégories précédentes sont détaillées. Ensuite, on vérifie que les éléments décrits dans les propriétés sont bien définis. Dans le cas contraire, et si c'est pertinent, alors on les rajoute. La pertinence est définie par le point de vue du contrôleur aérien. Par exemple, l'entité avion ne sera pas modélisé de la même manière qu'un manufacturier le ferait. La figure 4.2 est un extrait de la spécification qui porte sur la définition d'un secteur aérien. Au sein de la description générale, il y a une brève définition ainsi que quelques relations. La phrase «*An airspace sector has adjacent sectors*» implique la création d'une propriété symétrique (si A est adjacent à B, alors B est adjacent à A) et antiréflexive (un secteur aérien ne peut pas être adjacent à lui-même) dont le domaine et l'image sont des secteurs aériens.



Airspace sector

General Description:

- This entity represents a subdivision of the airspace.
- An airspace sector could be either controled or not controlled.
- An airspace sector has adjacent sectors.
- An airspace sector has a secure landline communications with adjacent sectors.
- Airspace sectors use distinct radio frequencies for communication with aircraft.
- Aircraft passing from one sector to another are handed off and requested to change frequencies to contact the next sector controller (next ATC).
- An airspace sector belongs to at least one of the airspace classes (Canadian airspace classes)

Figure 4.2 Extrait de la spécification

4.2.3 D'une spécification à une ontologie

La spécification donne une représentation haut niveau en langage courant du contrôle aérien. Le fait d'avoir énuméré les concepts et donné les propriétés et relations sous forme de liste permet de passer aisément au niveau ontologique. L'ontologie est créée à partir du logiciel *Protégé* [79]. Il permet de faire abstraction des différentes syntaxes lors du processus de création. *Protégé* considère trois types d'entités : les classes, les propriétés d'objets et les propriétés de données.

- Les classes représentent les différents concepts et la spécification. En incluant RDFS et OWL, *Protégé* permet de les hiérarchiser, d'ajouter des restrictions avec les propriétés d'objets et de données, de les annoter (comme l'ajout d'étiquettes ou de commentaires) ou de déclarer deux classes comme disjointes ou équivalentes.
- Les propriétés d'objets sont les prédicats des triplets *sujet/prédicat/objet* dont le domaine et l'image (*range*) sont des classes. Comme d'un point de vue ontologique ce sont des ressources comme les classes, on retrouve les mêmes possibilités que celles décrites dans le point ci-dessus. Il y a aussi des notions propres aux propriétés comme le domaine et l'image et différentes caractéristiques permises par OWL (réflexivité, transitivité, symétrie, fonctionnalité, ou leur inverse).
- Les propriétés de données sont les prédicats des triplets dont l'image est un littéral [1]. Contrairement aux propriétés d'objets, elles n'ont pas d'image ou de caractéristiques autres que la fonctionnalité.

En prenant point par point les informations décrites dans la spécification, on peut alors commencer à décrire l'ontologie. Le spécialiste vérifie la cohérence de la traduction, notamment au niveau des caractéristiques des propriétés. Pour reprendre l'exemple du secteur aérien, la relation «*An airspace sector has adjacent sectors*» est traduite par une propriété d'objets `hasAdjacentSector` dont le domaine et l'image sont la classe `AirspaceSector` et qui possède les caractéristiques `Symmetric` et `Irreflexive`.

4.2.4 Tests et validation

Avant de pouvoir être utilisée, l'ontologie doit être testée et validée. Cela se fait notamment par la vérification de la consistance des concepts. Il y a deux façons d'obtenir une inconsistance. La première est quand un concept est sous-concept de deux entités disjointes. La seconde est quand il n'y a pas d'individus de ce concept. Pour déterminer la consistance, il faut peupler l'ontologie avec des données de façon à simuler une utilisation normale. Ensuite, différents outils appelés *raisonneurs* permettent de vérifier la consistance. Les raisonneurs les

1. Chaîne alphanumérique ou valeur typée

plus connus sont FaCT++, HermiT et Pellet. Si une inconsistance est relevée ou que l'on n'est pas capable d'insérer des données parce qu'il manque un concept, alors on recommence les étapes à partir de la création de la spécification. L'autre avantage des raisonneurs et de pouvoir élargir la base de connaissance en découvrant de nouvelles relations par inférence.

Avant d'être validé définitivement, le modèle ontologique que nous avons développé a été comparé aux deux grandes ontologies de gestion du trafic aérien déjà existantes : celle du projet BEST [80] et celle de la NASA [81]. Ces ontologies sont plus poussées et détaillées, car elles concernent le monde ATM en général (ATS, ATFM et ASM) alors que le contrôle aérien n'en est qu'une petite partie. Néanmoins, la comparaison est utile pour vérifier que le modèle qui a été développé s'inclue bien dans ceux proposés par ces organismes. En effet, si tel est le cas, alors les règles qui auront été développées sous le modèle utilisé dans ce mémoire pourront l'être aussi par des ontologies «officielles» sans grands ajustements.

4.3 Les règles du système expert

Après avoir identifié les principales cyberattaques pouvant avoir lieu sur le trafic aérien via le protocole ADS-B et établi le modèle ontologique, le processus de création de règles de détection peut commencer. La démarche se fait de manière itérative en commençant par repérer les incohérences flagrantes jusqu'à finir par les plus subtiles. Dans cette section, nous présentons la méthodologie suivie pour obtenir différentes logiques qui contiennent les règles de détection implémentées dans notre solution, et nous détaillons ces logiques.

4.3.1 Les règles de détection

Le modèle de détection choisi est un système expert. Il établit des règles pour répondre à des scénarios précis. La qualité du système de détection réside dans la capacité à savoir décrire toutes les attaques possibles et comment les prévenir. C'est pourquoi cette étape nécessite à nouveau d'être fait avec un expert en contrôle aérien. Son expertise s'est traduite dans un processus en six étapes qui permet la génération d'attaques et des règles de détection associées. Ce processus est illustré dans la figure 4.3.

1. La première étape est la définition d'une attaque. Cela revient à déterminer une action qui peut être entreprise par un tiers malintentionné. L'action doit être générale dans son implémentation, mais précise dans ce qu'elle fait intervenir. Par exemple, créer un avion fixe en bout de piste de l'aéroport de Montréal ou de Québec est d'un point de vue logique la même attaque. Par contre, créer un avion fixe en bout de piste de l'aéroport de Montréal et en créer un en phase d'approche de ce même aéroport sont

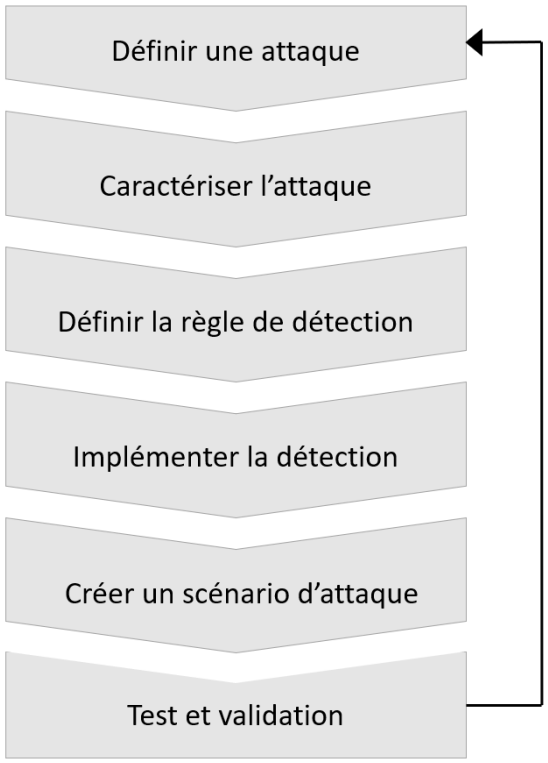


Figure 4.3 Processus de détermination des attaques ADS-B et de leur détection

deux attaques différentes, car ce ne sont pas forcément les mêmes types de capteurs et de données qui interviennent et les conséquences ne sont pas les mêmes. On commence avec des attaques bas niveaux et au fil des cycles du processus, les attaques se perfectionnent et deviennent sophistiquées. Cela permet de traiter des attaquants avec différents niveaux de connaissance du monde du trafic aérien.

2. Ensuite, il faut caractériser l'attaque. Cela signifie déterminer ce qui la rend unique. Cette caractérisation doit aboutir sur une description du processus précis de l'attaque (comment elle est lancée, comment elle est vue par les radars, comment elle se termine, quelle est sa logique, etc.) et des différences avec une situation normale. L'expert aide à déterminer quels sont les points de suspicion qui permettent de catégoriser un paquet (ou un enchaînement de paquets) comme une attaque. Pour reprendre l'exemple du faux avion fixe, le fait qu'un avion soit dans les airs sans bouger est impossible physiquement [2] et donc si plusieurs paquets ADS-B successifs indiquent un avion dans le ciel à la même position, alors cela devient suspicieux.
3. À partir des différences relevées, il faut alors établir les règles de détection. C'est un

2. Ce qui n'est pas le cas avec les hélicoptères qui peuvent faire du vol stationnaire.

schéma de conditions qui, lorsqu'elles sont réunies, détermine la légitimité ou non d'un paquet ADS-B. Ces règles sont écrites en langage courant et recensent les différentes sources de données qui doivent être utilisées. Elles sont ajoutées dans le rapport technique aux attaques qu'elles préviennent, ce qui permet un niveau d'explicabilité élevé du détecteur, en comparaison à des approches d'apprentissage machine par exemple.

4. Une fois les règles définies, elles doivent être implémentées. Cela veut dire traduire les règles en requête SPARQL et créer les différents scripts qui iront interroger la base de données ontologique et retourner les résultats. La création de ces scripts est aussi normalisée afin de garder une uniformité dans le module de détection et que la rigueur présente dans le rapport technique soit aussi présente au niveau du code.
5. Lorsque les attaques et règles de détection sont bien définies, on peut alors les tester. Un trafic aérien est simulé et des faux paquets ADS-B, correspondants aux attaques, sont injectés. On vérifie ainsi que les règles détectent bien les situations pour lesquelles elles ont été créées. Mais auparavant, il faut qu'un scénario ait été établi. Il est constitué en deux parties : un *scenario file* qui contient des avions qui représentent le trafic aérien «normal», pour faire la simulation, et une procédure ou un fichier d'attaque qui reproduit les injections de faux paquets ADS-B que cet attaquant aurait introduites.
6. Enfin, vient une phase de test et de validation. Le *scenario file* est chargé dans l'environnement de la simulation. Les attaques sont produites à partir de la procédure ou du fichier d'attaque. Ensuite, les règles de détection sont lancées. La validation des tests et leur évaluation concernent la capacité des règles à détecter leurs attaques et les critères demandés par les deux dernières questions de recherche. Une fois la validation effectuée, alors un nouveau cycle est lancé en partant sur une attaque plus poussée ou en se posant la question : «Quelles attaques pourraient ne pas être détectées par les règles que l'on vient de définir ?». Cela permet de définir des attaques de plus en plus sophistiquées.

Ce processus aboutit à des scénarios d'attaques et aux règles de détection qui sont associées. Plus elles sont nombreuses et sophistiquées, plus la détection est précise et nécessite une grande connaissance de l'écosystème ATC de la part de l'attaquant. La meilleure façon pour y arriver est de répéter le processus en boucle, comme le décrit l'étape 6, en cherchant à outrepasser les règles déjà existantes. Pour déterminer quelles sont les règles de détection les plus sophistiquées, on se base sur le critère de la complétude. C'est quand les seules attaques qui peuvent outrepasser la règle nécessitent d'attaquer aussi des systèmes que l'on considère comme sécurisés, ou des attaques qui ne rentrent pas dans le cadre de notre étude. Ce sont par exemple attaques qui sont faites à partir d'une intrusion sur le réseau des services ATC. Nos règles de détection ne s'appliquent qu'aux attaques menées sur les communications ADS-B.

Il est à noter que des règles dites *complètes* ne s'appliquent pas forcément dans tous les cas de figure. Par exemple, une règle qui dit que dans une zone couverte à la fois par un radar PSR et une antenne ADS-B un paquet ADS-B est considéré comme suspect s'il n'existe pas d'échos équivalents sur le PSR est complète. En effet, le radar PSR étant indépendant de tout équipement embarqué à bord, le seul moyen pour l'attaquant de mettre à défaut la détection est de faire voler un appareil au même endroit (dans ce cas ce n'est plus vraiment qu'une attaque informatique) ou de mettre hors service le radar ou d'injecter de faux paquets PSR sur le réseau ATM. On se trouve bien face à une règle complète puisque le PSR fait partie des systèmes sécurisés. Pour autant, cette règle ne peut pas être appliquée dans les zones sans couverture de radar primaire.

4.3.2 Logique d'identification

L'attaquant de plus bas niveau est considéré comme quelqu'un ayant pris connaissance de la vulnérabilité d'ADS-B et qui génère des paquets sans grande connaissance du domaine. Avant de s'intéresser à la cohérence spatiale des messages, il faut vérifier que les données envoyées soient possibles. Ce sont des vérifications très basiques.

Parmi les différents champs d'un paquet ADS-B se trouve l'identifiant 24 bit donné par l'OACI. Tous les aéronefs sont ainsi enregistrés dans leurs pays auprès des organismes gouvernementaux agréés. La logique d'identification se base donc sur le fait qu'un paquet ADS-B ne peut venir que d'un appareil déjà enregistré. C'est une simple requête SPARQL qui va vérifier si l'identifiant 24 bit contenu dans le paquet ADS-B existe bel et bien. Dans le cas contraire, l'aéronef qui a émis le message est déclaré comme suspect.

La logique d'identification exécute la requête montrée par la figure 4.4. Cette requête retourne la position (latitude, longitude, altitude) et le nom de tous les paquets ADS-B dont l'identifiant OACI de 24 bits n'est pas enregistré auprès des régulateurs du trafic aérien.

Cette simple règle est facilement contournée par des attaquants ayant effectué plus de recherche, d'autant plus que les bases de données d'identifiants OACI sont publiques et disponibles sur Internet. De plus, elle traduit le fait que toutes les attaques qu'elle laisse passer sont des usurpations d'identité. Ainsi, en plus de gêner le trafic aérien et le travail des contrôleurs, ces attaques peuvent porter préjudice aux propriétaires des aéronefs dont l'identité a été utilisée.

D'un point de vue technique, cette logique nécessite que la base de données ontologique soit synchronisée avec celles des immatriculations des aéronefs afin qu'un nouvel appareil ou un changement d'immatriculation ne soit pas perçu comme un faux positif. Dans le cas des

```
PREFIX atc-adsb: <http://[...]/graph/atc/ontologies/dds-topics/adsb-broadcast#>
PREFIX atc-core: <http://[...]/graph/atc/ontologies/atc-core#>
select ?report ?icaoID ?lat ?alt ?long ?call where {
    {?report a  atc-adsb:ADSBFlightPosition;
            atc-adsb:hasLatitude ?lat;
            atc-adsb:hasCallsign ?call;
            atc-adsb:hasLongitude ?long;
            atc-adsb:hasAltitude ?alt;
            atc-adsb:hasTargetID ?icaoID.}
    minus{
        select ?report ?ad where {
            ?report a  atc-adsb:ADSBFlightPosition;
            atc-adsb:hasTargetID ?icaoID.
            bind(str(?icaoID) as ?24bitID)
            service <http://[...]/repositories/ATC> {
                ?aircraft a atc-core:Aircraft;
                atc-core:hasICAOAddress ?24bitID.}
        }
    }
}
```

Figure 4.4 Requête SPARQL pour la logique d'identification

États-Unis, avec le programme Privacy ICAO Address, les aéronefs américains peuvent sous certaines conditions demander à obtenir un identifiant temporaire qui ne sera pas enregistré à l'aviation civile [14]. C'est principalement dans un intérêt de préservation de la vie privée afin que les sites de suivi des vols ne puissent pas faire le lien entre le vol d'un appareil et son propriétaire. Néanmoins, ce cas particulier n'est pas une exception à la règle puisque pour des raisons de contrôle, les identifiants sont disponibles pour les services ATC.

4.3.3 Logique des origines

Dans les attaques les plus simples à mettre en place, il y a la création de faux avions dans des endroits où le trafic est chargé comme les zones d'approche ou les routes aériennes. Or, l'apparition soudaine de signaux radars à ces endroits est suspecte.

On définit une trace ADS-B comme étant la succession des rapports ADS-B émis par une même source. Le premier paquet, qui correspond à l'origine doit pouvoir être justifié. Il y a trois raisons pour qu'une nouvelle trace soit créée :

— *Le décollage* : L'apparition d'une nouvelle trace se fait dès le démarrage, lorsque le pilote allume son transpondeur. Ainsi le premier paquet est situé au niveau d'un aéroport ou d'un aérodrome.
— *L'entrée en couverture radar* : Le premier paquet d'une trace peut se faire lorsqu'un appareil entre dans la zone de couverture d'une antenne ADS-B. Les paquets transmis par le transpondeur sont alors reçus par l'antenne et enregistrés.
— *Le passage à un espace contrôlé* : Dans certains cas, des avions peuvent être amenés à passer d'un secteur aérien non contrôlé ou dont l'utilisation du transpondeur n'est pas requise, à un secteur aérien où le transpondeur doit être allumé. La première position enregistrée de l'appareil se situe donc à la frontière entre ces zones.

Dans le cadre de ces travaux, nous émettons l'hypothèse que l'origine d'une trace doit donc se trouver dans ces zones. Dans le cas contraire, on peut considérer le message associé comme étant suspect. Pour la détection, on ajoute une marge dans le plan vertical et sur l'altitude à ces différents lieux (par exemple 3 NM au niveau d'un aéroport pour la position et 1,000ft pour l'altitude, ce qui correspond à la zone de contrôle). Si un avion a une trace dont l'origine ne figure pas dans ces zones, alors il est suspect.

Cette logique permet de détecter l'apparition de faux avions sur une grande partie de l'espace aérien. En plus de restreindre fortement le champ d'action des attaquants. Cela les force à faire des attaques plus poussées et donc d'avoir une meilleure connaissance en aéronautique. En effet, pour outrepasser cette règle, les faux messages ADS-B doivent contenir des rapports de position qui commencent dans des zones très précises : les aérodromes et les limites de portée radar et d'espace aérien contrôlé. De plus, avec l'ADS-B satellitaire dont le but est de couvrir entièrement la surface terrestre, ce troisième type de zone aura tendance à disparaître. Ainsi, si une personne malintentionnée veut causer de la gêne à un endroit précis, il devra commencer son attaque en amont.

4.3.4 Logique des radars

Certaines portions de l'espace aérien sont couvertes par différents types de radars, dont les radars primaires. C'est le cas au niveau des grands aéroports. Puisque les PSR sont des radars dont la détection est purement physique et ne dépend pas de la présence d'équipements embarqués à bord, alors en cas d'apparition d'un message ADS-B sur une zone couverte par un PSR, il devrait aussi y avoir un écho radar. En cas d'absence de cet écho, le message ADS-B est considéré comme suspect.

Pour s'assurer de la cohérence entre les relevés ADS-B et radars, il faut au préalable pouvoir faire la correspondance entre ces deux moyens de détection. Or ce n'est pas une tâche aisée.

Tout d'abord, ils ne sont pas synchronisés. Dans un cas, c'est l'avion qui émet périodiquement ses messages, et dans l'autre l'aéronef est détecté lorsque l'antenne est pointée dans sa direction. Ainsi, ce ne sont pas forcément les mêmes positions qui sont relevées. De plus, le radar primaire ne permet pas d'identifier les appareils qu'il détecte. Il faut alors faire du traitement de données pour associer un écho PSR à un avion. Cette tâche de correspondance entre plusieurs moyens de détection est effectuée par des systèmes de suivi qui permettent de clarifier les écrans radars [24].

Les radars secondaires utilisent des informations transmises par le transpondeur. Cette dépendance à un dispositif externe les rend donc vulnérables à de possibles cyberattaques. Ainsi, la comparaison entre un écho SSR et un écho ADS-B ne garantit pas la véracité du paquet ADS-B. Néanmoins, l'absence de correspondance SSR est considérée comme suspecte.

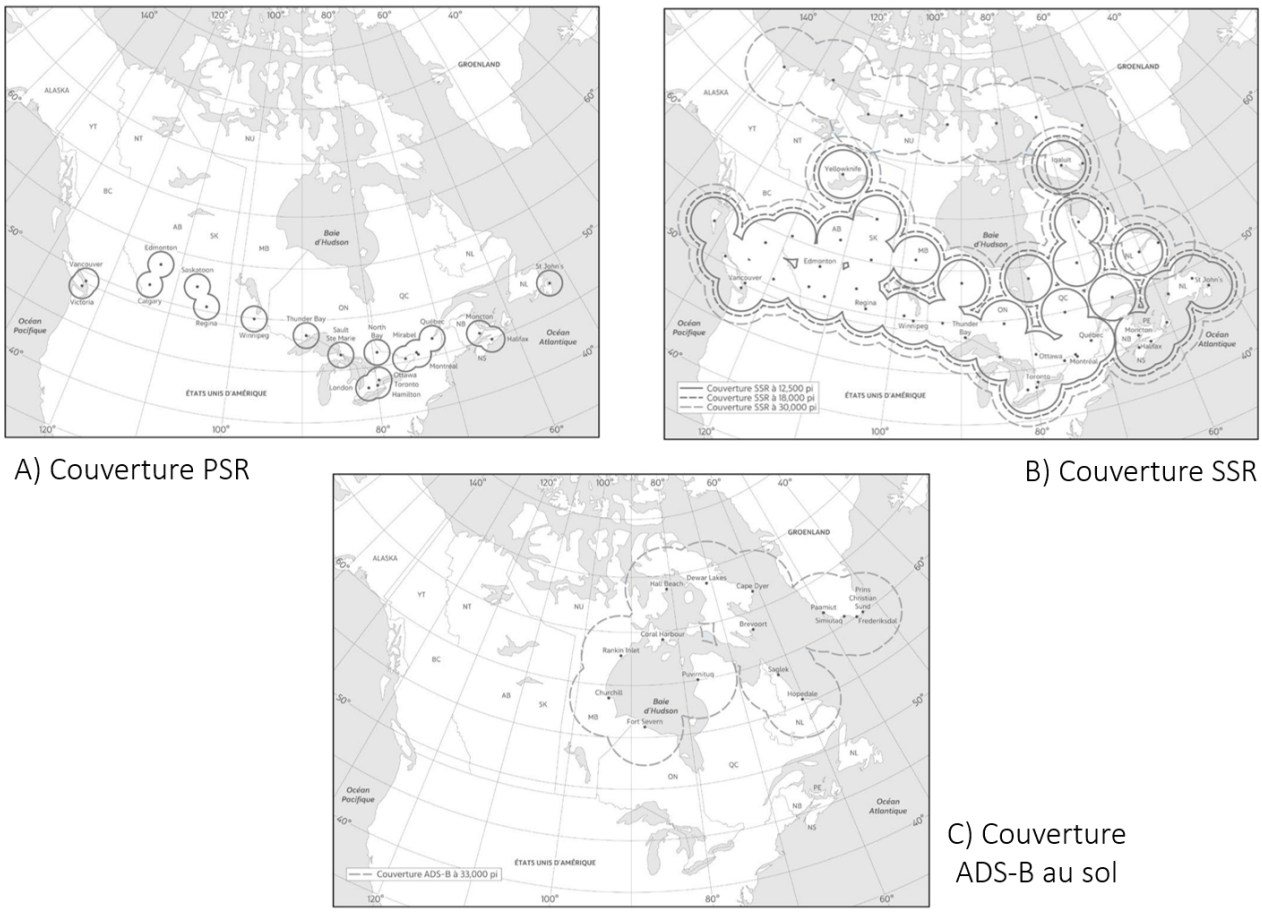


Figure 4.5 Couverture Radar et ADS-B en mode diffusion sol au Canada [82]

Bien qu'actuellement l'ADS-B en mode réception au sol semble être uniquement déployée au niveau de la Baie d'Hudson et de la mer du Labrador, comme le montre la figure 4.5C, nous avons décidé de tester la règle sur la FIR de Montréal en ajoutant des récepteurs ADS-B, qui

n'est couverte que par des radars PSR et SSR (figure 4.5 A et B). Cette prise de liberté est justifiée par le fait que cette zone sera effectivement couverte par de l'ADS-B satellitaire à partir du 25 février 2021 pour les secteurs de classe A (et 2022 pour les classes B) [83]. Enfin, l'algorithme pour la correspondance des traces a été créé pour cette recherche à des fins de test. Le but de cette règle n'est pas, en effet, de créer un algorithme de correspondance, mais de pouvoir vérifier l'efficacité des requêtes à partir des données qu'elle reçoit. Les algorithmes utilisés par les systèmes ATC sont beaucoup plus performants et présentent moins de lacunes, c'est-à-dire de difficultés d'identification voire de mauvaises identifications dans des ciels chargés.

4.3.5 Logique de vol

Avec les règles précédemment énoncées, le champ d'action des attaquants pour les *spoofing attacks* se réduit à des attaques avec usurpation d'identité dont l'origine est la limite de couverture des antennes ADS-B, la limite des secteurs aériens qui sont contrôlés ou bien des aérodromes qui ne sont pas sous couverture radar. De plus, il faut que la trajectoire des faux avions ne croise pas une zone de couverture PSR ou SSR. Avant de s'intéresser à la cohérence des paquets d'un point de vue physique (viabilité de la trajectoire, prise en compte de la météo, etc.), il reste différents moyens de détecter de faux avions, dont les restrictions apportées par la réglementation aérienne sur les secteurs aériens et les règles de vol.

```
1  PREFIX atc-adsb: <http://[...]/graph/atc/ontologies/dds-topics/adsb-broadcast#>
2  PREFIX atc-core: <http://[...]/graph/atc/ontologies/atc-core#>
3  select ?callsign ?report ?lat ?long ?alt ?time where {
4          {?report a  atc-adsb:ADSBFlightPosition;
5                  atc-adsb:hasCallsign ?callsign;
6                  atc-adsb:hasLatitude ?lat;
7                  atc-adsb:hasLongitude ?long;
8                  atc-adsb:hasAltitude ?alt;
9                  atc-adsb:hasTimeStamp ?time.}
10         MINUS {?flightPlan atc-core:hasCallsign ?callsign}
11
12 }
```

Figure 4.6 Requête SPARQL pour la logique de vol avec vérification du plan de vol

Une de ces réglementations est l'enregistrement d'un plan de vol. Il est obligatoire pour tout vol IFR ainsi que pour les vols à vue sous certaines conditions (dont les vols internationaux).

L'absence de plan de vol dans ces cas peut être considérée comme suspect. Il est à noter que le plan de vol est obligatoire dans les zones les plus achalandées et celles où les conséquences d'une attaque sont les plus importantes, par exemple les zones contrôlées autour des aéroports achalandés et l'espace aérien de bas niveau (espace de classe B et C). Les plans de vol sont déposés auprès des autorités compétentes et sont disponibles pour les services ATS. En exécutant la règle de détection montrée par la figure 4.6, on vérifie que les paquets ADS-B reçus proviennent des avions qui ont un plan de vol associé. Cette règle simple doit être étoffée pour être utilisable dans une situation réelle, notamment en vérifiant que la position de l'appareil ou son type correspondent à un vol IFR, ou VFR contrôlé [3]. Enfin, des vérifications sur l'heure de départ prévue, l'origine et la route entreprise par l'avion permettent de détecter des usurpations d'identités qui ont lieu dans une autre partie de l'espace aérien.

4.4 Discussion

Dans ce chapitre, nous avons détaillé différentes attaques informatiques pouvant survenir contre les communications ADS-B. Nous avons établit une liste d'acteurs potentiels et de scénarios dans des lieux sous tension, où les conséquences seraient importantes sur la sécurité aérienne. Ensuite, pour modéliser ces attaques et représenter l'environnement du contrôle aérien afin de créer un système expert de détection, nous avons expliqué la méthodologie qui a été suivie pour établir le modèle ontologique. Enfin, la détection se fait à partir de règles qui permettent de vérifier d'un point de vue logique la crédibilité des messages ADS-B reçus par les services ATC. Nous avons ainsi obtenu quatre règles logiques qui détectent des attaques de différents types.

La prochaine étape est d'obtenir un environnement de test afin de pouvoir implémenter le modèle ontologique et de vérifier que les règles de détection soient efficaces. Ainsi, dans le chapitre suivant nous présenterons ATC-Sense, notre solution de détection basée sur l'ontologie. Puis, afin de pouvoir la tester, nous avons créé une plateforme d'émulation d'environnement ATC, ATC-Emu dotée d'un module d'attaque. Enfin, nous décrirons le cadre de test de notre solution et nous la comparerons à une méthode utilisant l'apprentissage machine.

3. Au Canada, le vol VFR contrôlé permet aux avions en vol à vue de voler dans l'espace aérien contrôlé de bas niveau entre 12 500 et 18 000 ft.

CHAPITRE 5 ÉVALUATION EXPÉRIMENTALE

Ce chapitre présente ATC-Sense, la solution développée pour détecter des attaques sophistiquées contre les communications ADS-B, ainsi qu'ATC-Emu, une plateforme qui émule un système ATC. Pour tester notre système expert de détection, différentes attaques ont été étudiées afin de bien cibler les menaces pesant sur le trafic aérien. Enfin, nous proposons d'effectuer une comparaison avec une approche différente pour juger de la qualité, de la flexibilité et de l'efficacité de l'ontologie dans la détection de faux messages ADS-B.

5.1 ATC-Sense : une solution de détection d'attaques contre les communications ADS-B

Notre solution de détection d'attaques contre les communications ADS-B est un système expert basé sur les ontologies. Il est composé de deux principaux éléments : une base de données ontologique qui contient le modèle ontologique et les évènements qui seront enregistrés, ainsi qu'un raisonneur qui interroge la base de données. Ces éléments forment une plateforme, ATC-Sense, qui peut être greffée à des environnements ATC.

5.1.1 La base de données ontologique

La base de données ontologique contient la modélisation obtenue dans le chapitre précédent. Elle est accompagnée de scripts qui permettent la traduction et le peuplement des données brutes venant de l'environnement auquel ATC-Sense est greffé. Le produit utilisé dans notre implémentation est *GraphDB Free*, la première des versions développées par Ontotext [84]. Il a la particularité d'être conforme aux standards définis par le W3C sur RDF et SPARQL. Ses performances sont suffisantes pour développer la preuve de concept au niveau d'un centre de contrôle. De plus, il suffit de mettre à niveau le produit vers la version *Standard Edition* ou *Enterprise* lors d'un déploiement à plus grande échelle. La seule véritable limite, et qui n'est pas des moindres, c'est que cette version est limitée à deux requêtes SPARQL en parallèle. Cela signifie que si l'on réserve une requête pour le peuplement des données reçues en temps réel par les radars, il ne reste plus qu'une requête pour la détection. Il faut donc que leur temps d'exécution soit très faible et que les différentes règles soient séquencées pour éviter une congestion des requêtes.

La base de données doit être reliée là où transitent les informations dédiées aux contrôleurs. Un *parser* fait l'analyse et la conversion des données en RDF. De plus, la base de données doit

être aussi reliée à l'environnement de préparation de la simulation pour charger des données qui ne circulent pas en temps réel, mais qui sont utiles à la représentation du monde ATC. Ce sont, par exemple, les différentes instances des secteurs aériens, des aéroports ou encore des types d'avions. Il y a d'autres *parsers* qui s'en chargent. Pour uniformiser les liaisons entre les *parsers* et la base de données et éviter des erreurs en cas de mise à jour de celle-ci, un script python appelé *SparqlModule* est utilisé (voir figure 5.1). Il sert de constructeur de requêtes. Ainsi, lorsqu'une donnée doit être enregistrée dans GraphDB, elle est d'abord analysée et traduite en RDF par un *parser*. Ce dernier fait appel à des fonctions du *SparqlModule* pour générer des requêtes SPARQL et interagir avec la base de données.

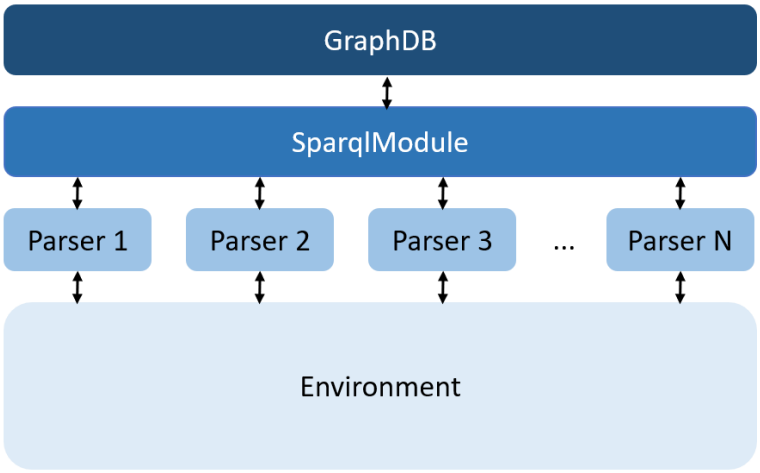


Figure 5.1 Étapes du peuplement de la base de données ontologique

Il existe deux types de données. Celles dites «statiques», qui ne varient pas durant la simulation, comme le modèle ontologique, les secteurs aériens, etc. et celles «dynamiques» qui correspondent au trafic aérien. Ces données sont stockées dans deux répertoires différents. Cela évite de donner l'accès en écriture au cœur de l'ontologie depuis l'environnement extérieur. De plus, sur une implémentation à grande échelle, le répertoire statique est global et permet à n'importe quel centre de contrôle d'avoir accès à ces données générales et au modèle ontologique, tandis que le répertoire dynamique est propre à chaque centre et contient les données qui lui sont nécessaires. Enfin, dans le cas des simulations, les données enregistrées peuvent être effacées en vidant le répertoire dynamique pour faire place à une nouvelle simulation sans avoir à recharger la partie statique à chaque fois.

5.1.2 Le module de détection

La détection se fait par un raisonneur. Il contient les différentes règles logiques que nous avons obtenues au chapitre précédent. Ce sont des requêtes SPARQL qui viennent interroger la base de données ontologique afin de retourner d'éventuels paquets suspects. Or, SPARQL ne permet pas de faire correctement de calcul dans l'espace en trois dimensions [85]. Cela force à devoir faire la partie calcul à l'aide de scripts Python. C'est notre sixième contrainte. L'ontologie n'est donc pas utilisée avec tout son potentiel est le résultat d'une détection est un mélange entre des informations retournées par les requêtes SPARQL et les calculs effectués par les scripts. Le module de détection est un ensemble de ces scripts qui regroupent les différentes règles qui auront été définies. Ils interrogent la base de données et retournent leurs résultats sous forme visuelle ou par affichage dans un terminal de commande. Afin de garder une uniformité dans le code, les scripts sont normalisés.

5.2 ATC-Emu : émuler un environnement ATC

Après avoir développé la solution de détection d'attaques ADS-B, il faut lui créer un environnement de test. Nous destinons notre solution aux contrôleurs aériens. Ainsi, nous avons développé une plateforme, ATC-Emu, qui émule une infrastructure ATC.

Le contrôle aérien fait partie des systèmes cyberphysiques, ou *Cyberphysical System* (CPS). Ce sont des systèmes informatiques qui sont en relation avec le monde (des éléments physiques) via un réseau de capteurs et d'actionneurs. Dans le cadre des systèmes ATC, les capteurs sont les différentes antennes et radars qui permettent de rendre compte de la situation de l'espace aérien. Pour les actionneurs, le processus est plus compliqué que dans des CPS typiques, parce que ce sont les contrôleurs via les informations transmises par leurs écrans qui agissent sur le trafic aérien en donnant par radio les autorisations de vol et autres consignes, et ultimement ce sont les pilotes qui agissent sur les systèmes physiques (les avions) qui changent l'«état» du système. Vouloir émuler un système ATC implique donc de recréer à la fois ses composantes matérielles et logicielles, l'environnement dans lequel elles évoluent ainsi que les humains qui les font fonctionner.

5.2.1 Travaux antérieurs du laboratoire SecSI

Le laboratoire de Sécurité des Systèmes d'Information de l'École Polytechnique de Montréal (Canada) s'est intéressé à l'émulation des systèmes ATC. Notre recherche est basée sur un de leurs travaux [21]. Il s'agit d'une proposition de plateforme d'émulation de système ATC appelée *Air Traffic Control System Simulator* (ATCSS). Elle se base sur Euroscope, un client

ATC virtuel utilisé dans le réseau de simulation virtuel de trafic aérien *Virtual Air Traffic Simulation Network* (VATSIM). Ce réseau virtuel simule le trafic aérien mondial avec d'un côté des pilotes utilisant des logiciels de simulation comme *XPlane* ou *Microsoft Flight Simulator* et de l'autre les contrôleurs ATC [86]. Euroscope reproduit ainsi les outils utilisés par les contrôleurs aériens. ATCSS propose une architecture afin de lier ce terminal radar à l'ontologie et aux outils de détection. Elle est décrite dans la figure 5.2. La partie orange (couleur saumon) représente l'environnement du contrôle aérien avec le simulateur de logiciel ATC composé d'Euroscope, du serveur *Flight Simulation Data* (FSD) et de scénarios contenant des avions, ainsi que des pilotes virtuels pouvant se connecter sur le réseau avec des logiciels de simulation de vol. Les éléments en bleu composent le réseau DDS, système d'échange de données où circulent les informations transmises par les radars. Enfin, les composants violets font la traduction entre les réseaux FSD et DDS. Cette infrastructure avec deux réseaux séparés (le réseau de simulation et le réseau de contrôle) suit le modèle de co-émulation de CPS proposé par Lemay *et al.* [87] pour les expériences en cybersécurité.

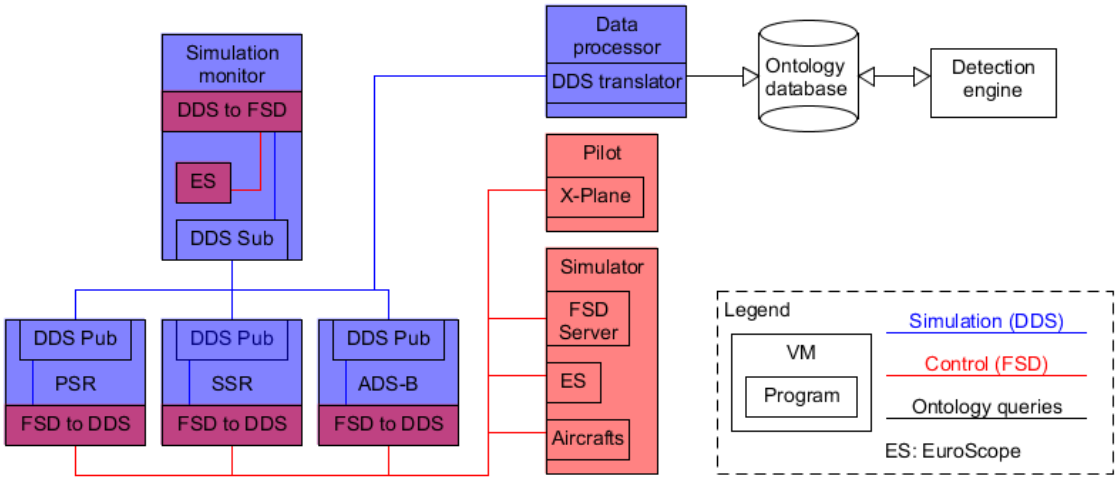


Figure 5.2 ATCSS, une proposition de simulation de système ATC [21]

Avant de pouvoir implémenter cette infrastructure, il faut d'abord l'analyser afin de vérifier son réalisme et sa complétude et la modifier en conséquence. La complétude traduit le fait de retrouver tous les composants d'un système ATC ainsi que les modules qui permettent de faire la détection et de simuler des attaques. Le réalisme concerne d'une part la gestion du trafic aérien et d'autre part la transmission des informations et plus précisément la façon dont elles sont reçues et exploitées par les systèmes ATC.

5.2.2 Reproduire les infrastructures radars

Si Euroscope constitue une reproduction raisonnable des logiciels de contrôle aérien, il fait abstraction de toute la partie physique (radars et antenne) d'une infrastructure ATC. Cela se traduit par un protocole simplifié pour l'échange de données avec le serveur FSD. Ainsi, à la place d'avoir des paquets reçus par les PSR, SSR ou les antennes ADS-B, ce sont des paquets FSD qui font les mises à jour de l'information des aéronefs. De plus, Euroscope a une vue globale sur tout le trafic aérien. C'est-à-dire que chaque paquet FSD qu'il reçoit est traité et affiché à l'écran. Il n'y a donc pas de notion de couverture radar limitée à proprement parler. Ainsi, dans certains cas, le contrôleur peut voir des informations qu'il ne peut normalement pas connaître. C'est pourquoi il est important de bien reproduire les infrastructures radars, afin que les données utilisées pour la détection soient bien celles reçues par les capteurs dans la réalité.

Dans l'architecture ATCSS, trois *publishers* sont proposés pour traduire les paquets FSD en paquets PSR, SSR, ADS-B et les faire transiter sur le réseau DDS. Cependant, le problème est mal posé. Si *in fine* ce sont bien des paquets PSR, SSR, ADS-B que l'on obtiendra, il faut que ce soient ceux de vraies sources. Le *publisher* n'est donc pas seulement un outil de traduction. C'est avant tout un composant émulant une infrastructure radar. Ainsi, il ne devrait pas y avoir uniquement trois *publishers*, mais il devrait y en avoir autant que de radars et d'antennes présents sur la zone à simuler. D'un point de vue technique, cette émulation pose deux défis :

1. Le serveur FSD n'autorise que trois connexions dont une est déjà prise par Euroscope. Il n'est donc pas possible que les *publishers* puissent tous s'y connecter directement. Pour y remédier, nous avons dupliqué le trafic FSD sur un autre réseau, appelé FSD'. Un script, nommé *Multiplexer*, est connecté en tant que client FSD et a pour but d'absorber le trafic et de le recopier sur FSD' où les *publishers* viendront s'y connecter.
2. La principale difficulté en émulant un radar est d'arriver à définir correctement sa zone de couverture. Bien que les constructeurs précisent des valeurs théoriques de portée, la couverture est sujette à plusieurs variables, dont le relief, l'élévation du radar et son inclinaison. Il y a ainsi plusieurs zones «aveugles», où un radar ne détecte rien. Cela contient entre autres le «cône de silence» qui correspond à l'espace au-dessus du radar qui n'est pas sondé. Il y a aussi les zones d'espace qui peuvent être masquées par un relief plus élevé faisant obstacle aux radars. Pour simplifier ces différents cas et avoir une première implémentation des radars, nous faisons le choix de considérer leur zone de couverture comme un cylindre. C'est une autre contrainte de cette recherche et elle permet d'être sûr que tous les avions qui devraient être détectés par un radar le sont

vraiment. Mais en contrepartie, certaines zones où des aéronefs seraient passés hors des écrans radars, sont rendues détectables.

Les données qui circulent sur le clone du serveur FSD (FSD') proviennent de tout l'espace aérien. Pour émuler une antenne ou un radar, il faut lui spécifier son type (PSR/SSR/ADS-B), sa position et sa portée. Chaque *publisher* capte les données qui sont envoyées par le *Multiplexer*. Puisque nous avons fait le choix d'une zone de couverture cylindrique, il lui suffit de vérifier si la position (latitude, longitude) et l'altitude du message reçu se trouvent dans le cylindre dont la base est centré sur l'emplacement du radar et le rayon est la portée du radar. De plus, nous simplifions le fonctionnement du radar en ne prenant pas en compte le cône de silence. Dans les cas des radars dont l'antenne est en rotation, la vitesse de rotation doit être spécifiée et le script la simule. Ainsi, lorsqu'une mise à jour de la position d'un avion est faite sur Euroscope, un paquet correspondant est envoyé sur le serveur FSD et est copié par le *Multiplexer* sur un second réseau. Les différents scripts simulant les radars vérifient s'ils peuvent recevoir ce paquet (l'avion est à portée et se trouve dans l'axe de détection du radar). Dans ce cas, le paquet FSD est traduit en paquet PSR, SSR ou ADS-B (selon le type de radar) puis est envoyé sur le réseau DDS.

La conversion des données est l'avant-dernière étape pour émuler un radar, juste avant leur envoi. Il existe différents types de paquets FSD : informations de connexion, mise à jour des clients, et bien d'autres. Ceux qui nous intéressent sont les paquets de mise à jour de position des avions. Ils contiennent les sept champs suivants : flag, callsign, squawk, latitude, longitude et heading [88]. Les valeurs et le format des informations de chacun d'eux sont détaillés dans le tableau 5.1.

Tableau 5.1 Format des paquets FSD de mise-à-jour de position

Champ	Valeur
Flag	"N" pour le mode *normal*. Ce sont tous les avions en vol. Et "S" pour *stand by*. Ce sont les avions qui sont prévus dans un scénario qui ne sont pas encore en vol.
Callsign	Identifiant de l'appareil qui peut être son immatriculation ou une combinaison entre l'identifiant de l'opérateur et le numéro d'immatriculation ou de vol
Squawk	Identifiant squawk à quatre chiffre en octal
Lat	Latitude en format décimal
Long	Longitude en format décimal
Alt	Altitude en pieds
Heading	Cap (direction). Converti dans un format propre à Euroscope

Comme Euroscope est un logiciel destiné à des contrôleurs aériens (pour le réseau virtuel

VATSIM), les simulations ne sont pas entièrement automatisables. En effet, les autorisations doivent être données par le contrôleur, en particulier celles pour l'atterrissage. Les contrôleurs en zone terminale ont ainsi jusqu'à quatre pistes qu'ils peuvent gérer. Et ce n'est qu'une fois l'autorisation d'atterrir donnée et l'atterrissage effectué que les avions disparaissent du serveur FSD. C'est comme s'ils éteignaient leur transpondeur. Si cette situation est assez conforme à la réalité, elle pose des problèmes dans le cas du contrôle en-route. En effet, ces zones de contrôle peuvent contenir plusieurs zones terminales. Or, comme il n'est pas possible de simuler leurs contrôleurs sur un unique client Euroscope, dans les simulations, les avions ne peuvent pas atterrir à ces aéroports. Pour rendre les simulations réalistes, il existe un moyen de forcer les autorisations et faire «arrêter» les aéronefs à une altitude et une position précise, comme s'ils venaient d'atterrir. Mais dans ce cas, les avions ne disparaissent pas du serveur FSD. Cela signifie que les aéronefs sont sans cesse repérés par les radars à leur dernière position. Pour éviter cette situation, les scripts radars sont dotés d'une mémoire de situation. C'est-à-dire une liste contenant chaque appareil qu'ils ont scanné et leur dernière position recensée. Si les scripts reçoivent du *Multiplexer* un paquet dont les valeurs sont les mêmes que celles qui sont en mémoire, alors ils ne le traitent pas.

Pour émuler un radar ou une antenne, nous avons implémenté l'architecture montrée à la figure 5.3. Tout d'abord, on introduit les capacités techniques et géographiques des radars et antennes. Lorsqu'un paquet arrive, on vérifie si l'avion qui émet le paquet est détectable. Puis, grâce à la mémoire de situation, on vérifie que le paquet reçu n'ait pas déjà été envoyé. Une fois ces tests passés, on met à jour la mémoire de situation et le paquet est converti en message DDS.

5.2.3 Le réseau DDS

Le réseau DDS est l'endroit où circulent les différentes informations reçues par les senseurs qui sont utilisés par les logiciels de contrôle aérien. C'est d'ailleurs à cet endroit que devrait se trouver logiquement Euroscope. Mais par son implémentation et son fonctionnement, il est situé à part au niveau du réseau FSD. C'est pour cela que Morel propose dans son architecture ATCSS de connecter au réseau DDS une machine virtuelle roulant un autre terminal Euroscope et un traducteur DDS/FSD (machine *Simulation Monitor* dans la figure 5.2). Le but de cette machine est d'afficher toutes les données issues des radars et antennes. Cela ajoute une redondance (puisqu'il existe déjà un terminal Euroscope sur lequel se déroule la simulation), mais alourdit la plateforme de simulation. En plus, ce second terminal est passif (il ne reçoit que des informations) et n'étant pas connecté à la simulation, il ne sert pas à contrôler les avions qui sont dans la simulation. Cela perd donc son utilité et c'est pourquoi

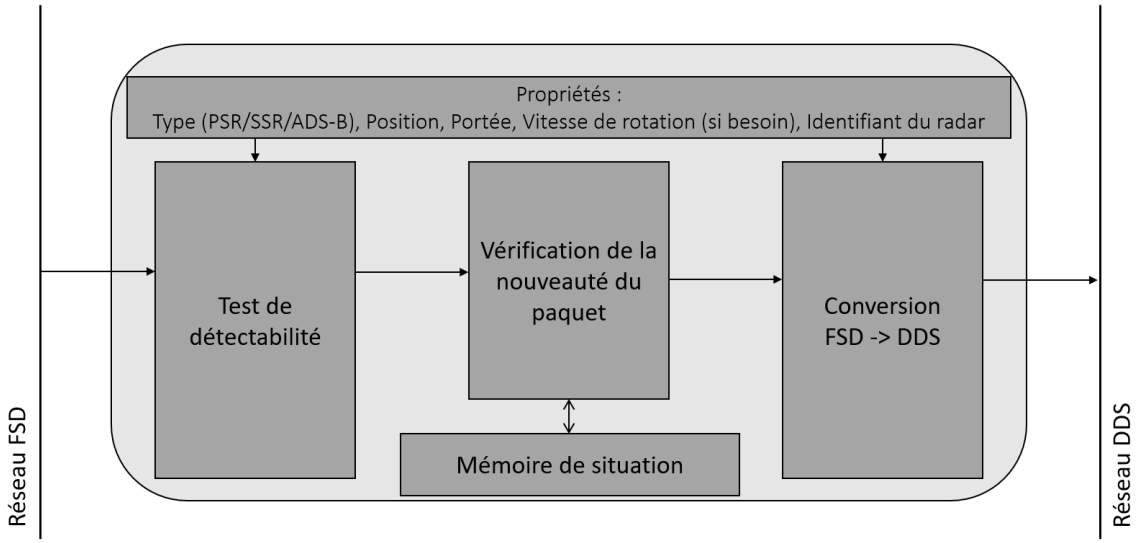


Figure 5.3 Émulation d'une infrastructure radar

cette proposition n'est pas retenue. Nous gardons uniquement un terminal Euroscope auquel sera greffé le réseau DDS.

5.2.4 Simuler des attaques

Une fois que l'environnement est correctement défini, il reste à introduire l'élément perturbateur qui est l'attaquant ainsi que le défenseur. Ici, il s'agit d'une logique de détection sans blocage. C'est-à-dire que le contrôleur observe l'environnement et ses perturbations, mais ne peut pas les empêcher. Sa seule action possible est de signaler ces perturbations. Tandis que l'attaquant peut interagir directement en envoyant de faux paquets. Il y a ainsi une asymétrie des actions et des façons de procéder.

Pour exécuter des attaques sur le système ATC, il faut avoir doté la plateforme de simulation d'un module qui injecte de faux paquets ADS-B. En situation réelle, une personne malveillante voulant perpétrer des attaques ADS-B enverra de faux paquets via un émetteur SDR. Ces paquets seront récupérés par des antennes ou des satellites et transférés sur le réseau DDS. Or, dans la plateforme de simulation ATC, ces communications physiques sont représentées par les paquets FSD. Cela amène à faire un choix. Soit on considère que les attaques doivent être aussi simulées avec un format ADS-B et dans ce cas on les injecte dans le réseau DDS, soit on les crée sur le réseau FSD et les *publishers* ADS-B s'occuperont de faire la traduction.

— Injecter les attaques ADS-B directement sur le réseau DDS pose un problème de

logique et un d'utilisation. En effet, le fait d'injecter des données sur ce réseau revient à dire que l'attaquant s'est connecté directement au réseau de partage des données ATC, ce qui n'est pas le cas. De plus, comme ce sont les données du réseau FSD qui mettent à jour la simulation sur Euroscope, les attaques perpétrées sur le réseau DDS ne sont pas visibles sur l'écran du contrôleur et donc qu'on ne peut pas voir les possibles conséquences qu'elles peuvent avoir (comme les conflits de résolution).

— Si les attaques se font sur le réseau FSD, alors elles sont bien visibles sur l'écran d'Euroscope. Cependant comme tout le trafic FSD est copié par le *Multiplexer* dans le réseau FSD' et est traduit lorsqu'il est détectable par chacun des *publishers* qui représentent une antenne ou un radar, alors ces attaques sont susceptibles d'être traduites par tous les types de radars, dont les PSR et SSR. Or cela ne correspond pas à la réalité. Il faut dans ce cas que les *publishers* puissent déterminer si ces paquets les concernent ou pas.

Le fait que les attaques puissent être visibles sur l'écran de contrôle et que le réseau DDS reste inviolé implique que ce soit le second choix qui soit fait. L'injection des faux paquets ADS-B se fait ainsi au niveau du réseau FSD. Il faut alors faire en sorte à ce que les paquets ne soient pas retransmis par les radars PSR et SSR. Pour cela, les scripts d'attaques vont insérer un caractère supplémentaire, le «_», en première position du champ du callsign (voir tableau 5.1 pour rappel du format des paquets FSD). Ce caractère spécial n'est pas utilisé par les callsigns. Au niveau des radars, on ajoute un test avant celui de détectabilité. Pour les PSR et SSR, si un paquet a un callsign qui commence par ce symbole, alors ils ne traitent pas l'information. Et pour les antennes ADS-B, elles suppriment ce caractère pour retrouver le vrai callsign spécifié par l'attaquant. Cette contrainte implique donc de devoir modifier un peu les *publishers* mais est justifiée dans le sens où les attaques sont bien présentes sur la couche physique (représentée par le réseau FSD) et bien représentées sur le réseau DDS (seuls les paquets ADS-B sont traduits, comme dans la réalité). Ainsi, pour visualiser les attaques sur le logiciel ATC, nous mettons une nouvelle contrainte à notre plateforme en injectant les attaques au niveau du réseau FSD.

Le module d'attaque est donc composé d'une série de scripts qui permettent de créer des paquets ADS-B à partir d'un fichier, ou manuellement, et de les injecter sur le réseau FSD.

5.2.5 Architecture de la plateforme

Toutes ces modifications ont abouti à ATC-Emu, une plateforme qui émule un système ATC. Elle est donc composée de deux grandes parties : l'environnement ATC qui s'occupe de simuler la circulation aérienne, les logiciels utilsés par les contrôleurs aériens et les infrastructures

radars ; et le module d'attaque qui joue le rôle d'élément perturbateur en reproduisant les différents acteurs malveillants ainsi que les attaques qu'ils peuvent perpétrer par l'injection de faux paquets ADS-B. Ces parties constituent deux des trois principaux éléments d'une étude de cybersécurité : l'environnement et l'attaquant. Le troisième élément, le défenseur, est présent lorsque l'on lie ATC-Sense à ATC-Emu.

Pour fonctionner correctement, la plateforme ATC-Emu a besoin d'utiliser de nombreuses données pour se faire une représentation correcte de l'environnement qui est émulé. Parmi ces données on y trouve :

- *Les secteurs aériens.* Ce sont des fichiers qui représentent la séparation de l'espace aérien. Euroscope utilise un *sector file* pour afficher sur l'écran du contrôleur les séparations entre chaque secteur.
- *Les classes aériennes.* Elles sont utilisées par l'ontologie afin de pouvoir déterminer les restrictions liées au secteur où se trouve un avion.
- *La liste des waypoints et routes.* Ces données cartographient l'espace aérien et sont utilisées par le pilote pour définir leur chemin. On les retrouve notamment dans les plans de vol et sur les procédures de départ et d'arrivée. La liste des *waypoints* et des routes est intégrée à Euroscope pour afficher la trajectoire qui suivra un vol en fonction de son plan de vol.
- *Les aéroports.* Les aéroports principaux sont détaillés par une carte sur Euroscope afin de permettre de faire le contrôle terminal. Pour les autres, seule la position est enregistrée. Au niveau ontologique, ils sont utilisés pour vérifier que l'origine et la destination d'un vol sont bien un aérodrome.
- *La liste des radars.* Elle contient les positions des différents radars et antennes d'un secteur aérien. Elle est utilisée pour créer les *publishers* du réseau DDS et simuler ainsi les données qui sont reçues par ces différentes sources.
- *Les procédures.* Différentes procédures régissent les arrivées et départs au niveau des aéroports. Il y a les *Standard Arrival Route* (STAR) pour arriver à la phase d'approche, et les *Standard Instrument Departure* (SID) pour les départs. Elles sont utilisées pour la détection. Enfin, d'autres procédures, qui ne sont pas encore implémentées dans la plateforme, comme les approches aux instruments, pourraient être employées pour aider à la détection lors des phases d'atterrissage.
- *Les types d'avion.* La liste des avions avec leurs performances permet à Euroscope de prédire correctement la physique des trajectoires des différents vols. Pour la partie ontologie, une telle liste serait utilisée par les requêtes de détection pour vérifier la cohérence physique des messages ADS-B. Cependant, cette dernière fonctionnalité n'est pas encore implémentée.

— *Les plans de vol.* Ces informations sont transmises normalement aux différents régulateurs par les pilotes.
— *Les données météo.* Elles sont utilisées pour la simulation des vols, c'est-à-dire par Euroscope. Ce sont les *Meteorological Aerodrome Report* (METAR), des rapports météorologiques émis par les aéroports. Ils aident les contrôleurs aériens à choisir les pistes en service en fonction du vent (une piste a deux directions possibles). Le décollage et l'atterrissage doivent idéalement se faire face au vent. Les METAR ont donc une influence sur la trajectoire des avions lors des arrivées et des départs. Ils sont indiqués dans les fichiers de scénarios.
— *Les scénarios.* Ce sont des fichiers qui représentent la situation à simuler. Il existe deux types de fichiers de scénario. Le premier, appelé *scenario file* par Euroscope, contient les fréquences des contrôleurs, la localisation des pistes d'atterrissage, les données METAR et le point de départ des différents avions avec leurs routes prévues. C'est à partir de ces informations que sont simulées les trajectoires. Le second type de scénario permet de rejouer une situation qui s'est passée. Un script Python injecte sur le réseau FSD les données de différents vols enregistrées dans des fichiers *CSV*. Dans ce cas, le contrôleur est passif et n'a plus la main sur les avions.

5.3 Environnement expérimental

En liant ATC-Sense et ATC-Emu, nous obtenons un environnement de co-émulation qui représente un écosystème ATC avec ses attaquants et ses défenseurs. Notre solution étant à l'état de preuve de concept, il n'y a pas de lien de ATC-Sense vers ATC-Emu et donc les actions du défenseur se limitent à de la détection pure, c'est-à-dire à l'affichage des paquets suspects. Néanmoins, à terme ses actions peuvent être plus complètes en agissant directement sur l'environnement (affichage d'alertes, non-affichage des paquets incriminés, et d'autres).

Pour pouvoir répondre à notre question de recherche qui s'intéresse à la performance de notre solution et à sa viabilité en temps réel, il faut la tester dans des simulations de trafic aérien.

5.3.1 Automatisation des tests

Afin de faciliter la préparation, l'exécution et la reproduction des simulations, et de lier correctement ATC-Emu et ATC-Sense, les tests ont été en grande partie automatisés. La figure 5.4 détaille les différentes étapes qui précèdent le lancement d'une simulation.

L'automatisation peut être décomposée en quatre grandes parties ayant des fonctionnalités différentes : la partie commande, les ressources, la plateforme ATC-Emu et la détection

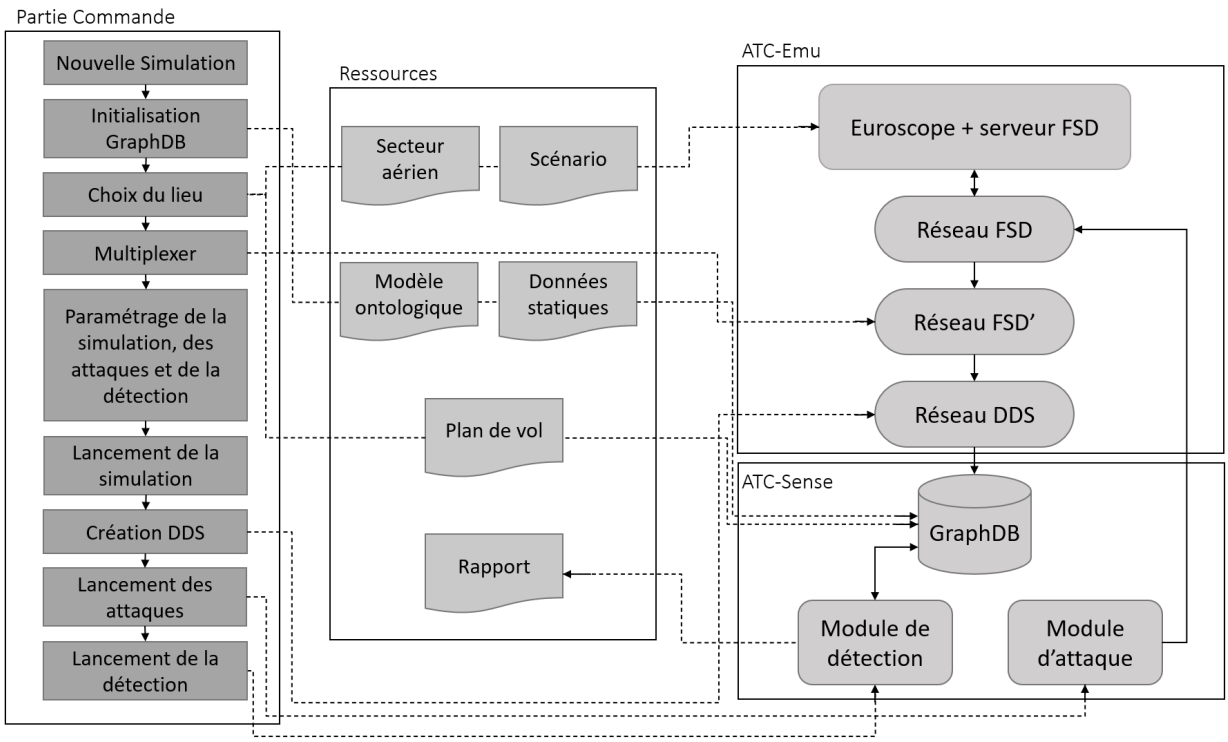


Figure 5.4 Processus de lancement des tests sur ATC-Sense et ATC-Emu

avec ATC-Sense. La partie commande est constituée d'un script principal qui exécute successivement différents scripts qui créent, paramètrent et contrôlent les différents modules de simulation. Le lancement d'une simulation se déroule comme suit :

1. Le premier script initialise la base de données ontologique. Il s'occupe de lancer GraphDB et de vider le répertoire contenant les données dynamiques. S'il s'agit de la première utilisation de la plateforme, ou que les données statiques doivent être modifiées, alors une option permet de vider le répertoire des données statiques et de le peupler à nouveau.
2. L'utilisateur est amené à choisir sur quel secteur aérien il souhaite travailler. Euroscope et le serveur FSD sont lancés avec les fichiers de configuration qui correspondent à ce choix. De plus, un scénario, qui est associé au secteur aérien choisi, est chargé dans Euroscope et les informations des plans de vol sont enregistrées dans GraphDB. Il est possible pour l'utilisateur de sélectionner à la place son propre scénario.
3. Le *Multiplexer* créé le réseau FSD'. Il commence à faire les copies des messages du réseau FSD vers FSD'.
4. L'utilisateur renseigne les paramètres de la future simulation. Il définit la durée de simulation, le nombre d'attaques, leur type, le temps et leur durée, ainsi que les règles

de détection à appliquer et leur fréquence d'exécution. Le module d'attaque permet d'automatiser la création des attaques, mais l'utilisateur peut choisir d'utiliser son propre script d'attaque.

5. L'utilisateur décide de donner le départ de la simulation.
6. Le réseau DDS est créé. Les données reçues par les antennes et radars sont enregistrées dans la base de données ontologique.
7. Le module d'attaque est lancé avec les paramètres choisis au préalable. Les attaques sont injectées sur le serveur FSD.
8. Le module de détection est créé avec les paramètres choisis par l'utilisateur. Il interroge GraphDB à la fréquence demandée et produit les rapports de détection aux formats *PDF* et *CSV*.

Ainsi, les tests sont automatisés. Ils se font par une succession de scripts qui exécutent les différents logiciels, chargent les données et scénarios selon les choix de l'utilisateur et lancent les scripts d'attaque et de détection. À ce jour, les paquets suspicieux sont affichés dans une console de terminal et enregistrés dans des rapports de détection aux formats *PDF* et *CSV*.

Avec ses différents paramètres et la possibilité pour l'utilisateur de sélectionner ses propres scénarios de simulation et d'attaques, les expériences menées peuvent être vraiment flexibles et permettent d'étudier de multiples situations différentes.

5.3.2 Une implémentation au Canada

La solution ATC-Sense a été testée sur l'exemple de l'espace aérien canadien. Les différents éléments de la plateforme ATC-Emu sont construits dans l'optique de se rapprocher au plus près possible du fonctionnement ATC au Canada.

Au niveau du logiciel ATC, la division canadienne de VATSIM, VATCAN, a créé le projet CANscope [89]. Il permet de configurer Euroscope afin de ressembler au *Canadian Automated Air Traffic System* (CAATS), le système ATM développé et utilisé par Nav Canada. On y retrouve les sept FIR gérées par Nav Canada (Vancouver, Edmonton, Winnipeg, Toronto, Montréal, Moncton et Gander). Chacune de ces régions est entièrement détaillée : secteurs aériens, aéroports, routes, balises, *waypoints*, géographie, procédures d'arrivées (STAR) et de départ (SID), etc. Même les alertes sonores et visuelles sont configurées pour ressembler à celles utilisées par Nav Canada.

Le CAATS utilise la technologie RTI Connext DDS. Nous utilisons donc cette même technologie. Ce sont plus exactement les connecteurs pour Python qui permettent de faire la liaison entre les scripts de traduction et le réseau DDS. Il est évident que la solution développée

pour les systèmes ATC canadiens est bien plus complexe et complète, car elle fait intervenir divers intermédiaires, mais la technologie est la même.

Enfin, les scénarios étudiés reprennent des vols qui sont en provenance ou à destination des aéroports canadiens. Ce rapprochement avec la réalité permet de mieux comprendre les enjeux liés à une perturbation du trafic aérien et observer les différentes actions qui peuvent être prises par le contrôleur ainsi que leurs répercussions.

5.3.3 Cas d'étude

La configuration utilisée pour déployer ATC-Sense et ATC-Emu est la suivante : une machine virtuelle Ubuntu 16.04 avec 8 Go de mémoire vive, 4 cœurs alloués à 2,9 GHz et 48 MB de mémoire vidéo pour l'affichage.

La troisième question de recherche s'intéresse à la performance de la solution proposée. Il s'agit tout d'abord de vérifier que la détection se fait bien et d'observer le temps de calcul qui a été nécessaire pour avoir les résultats des requêtes.

Pour y parvenir, nous avons créé un scénario comportant une simulation d'une heure sur la FIR de Montréal dans laquelle vingt avions évoluent sur un secteur aérien. La détection commence au bout de cinq minutes après le début de la simulation. Ce temps t_{init} d'initialisation permet aux avions légitimes de se mettre en place. Cinq faux avions sont ensuite insérés entre la dixième et la vingtième minute de la simulation. Chaque faux avion est présent pendant sept minutes. Durant cette heure, on calcule pour chaque règle le temps qu'il a fallu pour interroger la base de données, faire le traitement des données et retourner un résultat. La figure 5.5 montre le cadre de cet expérience.

Les ressources allouées à la machine virtuelle se sont révélées limitées. Euroscope consomme beaucoup de mémoire et de temps processeur, d'autant plus que ce logiciel fonctionne sous Windows et qu'il est donc nécessaire de rajouter une interface de compatibilité pour l'utiliser dans notre environnement Linux. Cette trop grande utilisation des ressources a des impacts sur les différents scripts en causant, par moments, des ralentissements. Si les expériences décrites précédemment permettent de juger de la viabilité de notre solution de détection sur un environnement ATC, les ralentissements causés par les performances physiques de la machine employée pour faire les tests apportent un biais sur les performances de notre solution en terme de temps de calcul. C'est pourquoi nous proposons une seconde série d'expériences qui s'appuient sur l'utilisation d'un système à évènements discrets. Au lieu d'utiliser la plateforme ATC-Emu pour reproduire le trafic aérien, nous utilisons les données obtenues lors de la première série de tests. Ces données sont triées chronologiquement et

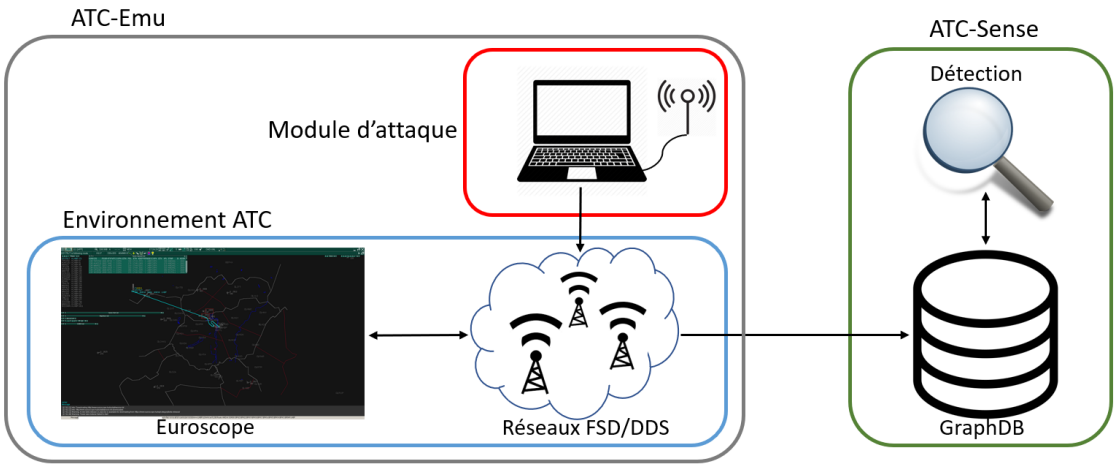


Figure 5.5 Architecture des tests menés avec ATC-Sense et ATC-Emu

enregistrées dans des fichiers. Lors de la phase de test, un script va les envoyer vers ATC-Sense avec les mêmes délais que lorsqu'ils ont été émis. Cela reproduit exactement la même situation que lors de la première phase de tests, mais sans utiliser toute la partie logicielle et réseau d'ATC-Emu. Cette fois-ci c'est de la co-simulation. La nouvelle architecture de test est schématisé dans la figure 5.6.

5.4 Comparaison avec des méthodes antérieures

Pour étayer l'évaluation de la solution de détection proposée, il faut pouvoir la comparer à d'autres solutions de détection d'attaques ADS-B. Les solutions les plus proches de celle que nous avons développée sont basées sur les ontologies et l'intelligence artificielle. Le mémoire de Babar [65] présente un système de détection d'intrusion basé sur la création de contraintes de bas niveau qui utilise en partie l'ontologie développée par Morel. Pour leur part, Shabtai et Habler [61] utilisent l'apprentissage machine. Nous allons donc nous pencher sur la solution de détection qui utilise l'intelligence artificielle pour faire la comparaison avec la nôtre.

5.4.1 Une méthode limitée

Dans leurs travaux, Shabtai et Habler proposent d'utiliser un réseau de neurones récurrents, le LSTM (*Long Short-Term Memory*), afin de pouvoir détecter des attaques ADS-B. Ils utilisent le modèle d'*encoder-decoder* pour essayer de comprendre la logique qu'il y a entre une série de messages ADS-B. Ils découpent ainsi l'ensemble des messages d'un vol en séries de 15 rapports consécutifs. L'encodeur essaie de déterminer des motifs logiques à ces séries, et le décodeur

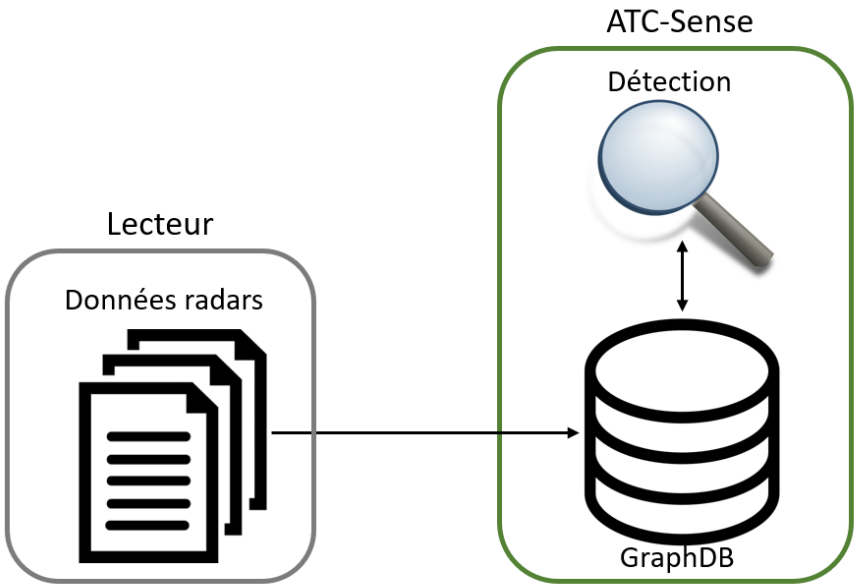


Figure 5.6 Architecture des tests menés directement sur ATC-Sense

tente, à partir des motifs obtenus, de retrouver les messages originaux. Leur idée est que s'il y a une attaque ADS-B, le décalage entre les vrais paquets et ceux injectés par l'attaquant sera amplifié par le LSTM. Pour déterminer cette amplification, ils calculent la ressemblance entre les paquets réels et ceux reconstruits par le décodeur en utilisant la similarité cosinus [1]. Ils calculent ensuite un score d'anomalie à l'aide de l'équation 5.1, où $W[i, L]$ est la série de messages en entrée de l'encodeur, $L = 15$ est la taille de cette série, $x^{(j)}$ est le j-ième message ADS-B et $\hat{x}^{(j)}$ correspond à sa reconstruction faite par le décodeur. Pour terminer, lors de la phase de test, ils déterminent une valeur de seuil, le *threshold*, correspondant à 95% des scores d'anomalie, à partir duquel un message est classé comme suspicieux. Un message dont le score d'anomalie est supérieur à ce seuil est déclaré comme suspicieux. Et Shabtai et Habler indiquent qu'une série de 15 messages suspicieux est considérée comme une attaque.

$$Anomaly(W[i, L]) = \sum_{j=i}^{i+L}(1 - CosSimilarity(x^{(j)}, \hat{x}^{(j)})) \tag{5.1}$$

Ces travaux proposent d'une certaine façon une méthode pour analyser la logique de vol et la cohérence entre les différents paquets. Si cette piste est intéressante, les choix qu'ils ont faits présentent quelques limites et lacunes.

1. Fonction permettant de calculer la similarité de deux vecteurs. Elle calcule le cosinus de l'angle entre ces deux vecteurs.

— Leur modèle d'attaque ne prend en compte que des cas où les attaquants font du *jamming*, c'est-à-dire ils brouillent les signaux d'un avion en vol pour y injecter de faux messages. C'est une attaque extrêmement compliquée, voire quasiment impossible dans le cas de l'ADS-B satellitaire, car l'attaquant doit supprimer les paquets venant d'un aéronef puis d'injecter les siens. Ainsi ce modèle ne prend pas en compte toutes les attaques faisant apparaître de faux avions, et qui sont plus simples à réaliser.
— Pour les besoins de leur algorithme, ils doivent connaître l'aéroport d'origine et de destination des vols et avoir une base de données de vols légitimes existants entre ces deux lieux. Cela nécessite d'entraîner l'algorithme sur toutes les liaisons aériennes possibles, ce qui est gourmand en termes de temps de calcul et de nombre de vols à traiter dans le jeu de données. De plus, cette méthode ne fonctionne pas dans le cas où l'origine ou la destination d'un aéronef sont inconnues, comme pour les vols qui n'ont pas de plan de vol.
— Leur cas d'étude ne semble s'intéresser qu'aux vols commerciaux. En effet, leur dataset est composé uniquement de vols assurés par des compagnies aériennes et ils considèrent qu'entre deux aéroports ce sont normalement les mêmes routes qui sont utilisées. Ceci est vrai dans le cas des vols commerciaux et pour la plupart des vols d'aviation générale. Or dans le cas de l'aviation générale, le choix des routes empruntées est à la discrétion du pilote. De plus, certains vols de l'aviation générale partent et reviennent au même aéroport. Enfin, même dans le cas des vols commerciaux ce ne sont pas toujours les mêmes routes qui sont utilisées entre deux aéroports. En effet, l'état du trafic aérien, de la météo ou des évènements imprévus peuvent amener un avion à changer de trajectoire et de route. En outre, l'arrivée des avions dans des aéroports d'importance est soumise à l'encombrement de l'espace aérien.
— La méthode pour déterminer la véracité d'un paquet présente un biais. En effet, elle est basée sur le calcul d'un seuil (*threshold*). Puisque le seuil est défini lors de la phase de test, la qualité de la détection ne dépend pas du modèle, mais des attaques testées. Comme résultat de cette méthode, des fichiers tests avec une attaque trop importante (gros changement de valeurs des paquets ADS-B) et de longue durée donneront un seuil élevé qui laissera passer des attaques plus subtiles. Tandis que des fichiers de test avec une attaque subtile (petits changements de valeurs) ou très brève risquent d'obtenir un seuil assez bas qui pourrait engendrer beaucoup de faux positifs.

5.4.2 Cas d'étude : l'arrivée à Montréal

Afin de pouvoir comparer notre solution à la leur, il faut trouver un scénario qui soit crédible et dans lequel l'attaque puisse potentiellement être détectée par les deux solutions. Les limites

de leur méthode énoncées précédemment restreignent le champ d'études aux attaques par modification de paquets.

L'article détaille trois types d'attaques qu'ils ont testés. La première, appelée *RND* (pour Random noise) est la modification des données (latitude, longitude, altitude, cap et vitesse) des paquets originaux de l'avion par un réel entre 0 et 2. La seconde, appelée *ROUTE*, consiste à remplacer des paquets par un segment d'un autre vol (en l'occurrence, un vol entre la Thaïlande et l'Ouzbékistan). Enfin, la dernière fait varier l'altitude en l'augmentant (SHIFT UP) ou en la diminuant (SHIFT DOWN) de 400 pieds à chaque message. C'est ce dernier scénario qui nous intéresse. En effet, il existe des zones où l'altitude et la vitesse sont réglementées et où un non-respect des consignes données peut avoir de grandes conséquences sur le trafic aérien et la sécurité des aéronefs. Dans leur évaluation expérimentale, Shabtai et Habler injectent leurs attaques pendant la phase de croisière. Or, dans la réalité, ces attaques seraient rapidement détectées, puisque l'avion est censé garder une altitude relativement semblables. C'est pourquoi nous avons choisi d'appliquer leur algorithme dans un scénario plus sophistiqué avec des attaques ayant lieu dans des zones sensibles.

Les procédures d'approche sont un des endroits sensibles qui ont été évoqués dans la section 4.1. Les avions qui sont dans une STAR sont soumis à des contraintes de vitesse et d'altitude entre différents *waypoints*. Le non-respect de ces critères peut avoir de fortes conséquences sur le trafic aérien. Il peut arriver, mais il doit être signalé au contrôleur qui est alors conscient de la situation. Ainsi, lors du vol TSC485 du 16 mai 2018, pendant la phase d'approche vers l'aéroport de Montréal-Trudeau, le pilote de l'appareil a prévenu le contrôleur aérien qu'il ne pourrait pas respecter l'altitude minimale requise. Deux minutes plus tard, il s'est produit une perte d'espacement avec un Cessna arrivant lui aussi à Montréal par une autre direction. Au point le plus proche, ils étaient à moins de 500 ft en résolution verticale et 1,7 NM en résolution horizontale (contre 1000 ft et 3 NM selon la législation) [90]. Bien que ce ne soit qu'un incident et que la principale cause retenue soit un manque de contrôleurs et une distraction du superviseur, ce cas de figure permet de comprendre qu'une attaque sur l'altitude est particulièrement dangereuse, surtout en période de fort achalandage. Le cas d'étude pour comparer les deux méthodes porte ainsi sur une modification de l'altitude d'un avion lors d'une procédure STAR. Pour préparer le scénario, nous nous appuyons sur quatre éléments clés :

— *Type de vol* : L'efficacité du LSTM en tant que moyen de détection repose sur la similarité entre les vols légitimes utilisés pour l'entraînement du réseau de neurones. La partie en-route est la phase de vol la plus stable : l'altitude, la vitesse et le cap varient peu. Ce n'est pas le cas lors de l'ascension ou de la descente, ce qui explique les différences qu'il peut y avoir entre un paquet et sa reconstruction par le LSTM

dans ces deux phases. Cela se traduit par une variation plus ou moins importante du score d'anomalie qui caractérise la différence entre un message et sa reconstruction. Le type de vol a donc une influence sur la granularité de la détection.

— *Données* : Comme dans l'implémentation de Shabtai et Habler les relevés ADS-B proviennent du site *flightradar24*. Elles ont donc le même format. Chaque jeu de données contient 60 vols. Ce sont des liaisons qui ont lieu au Canada puisque l'implémentation de la plateforme ATC-Sense représente le système ATC de ce pays. Ils ont pour destination l'aéroport de Montréal Trudeau. Deux liaisons seront testées : un vol court-courrier (Toronto à Montréal) et un vol long-courrier (Vancouver à Montréal).

— *Attaques* : Tout d'abord, afin d'approcher les résultats des chercheurs, l'attaque *SHIFT UP* est implémentée telle que décrite dans l'article. Néanmoins, dans leurs travaux ils exécutent l'attaque sur une série de 70 paquets ADS-B entre les messages 180 et 250 envoyés par un avion. Or, ces paquets correspondent à une partie de la phase de croisière et non à une STAR. Pour que l'attaque corresponde à notre scénario, elle n'est pas lancée sur les paquets 180 à 250 comme dans leur étude, mais au niveau de la STAR. Selon leur plan d'expérimentation, à chaque message l'altitude est incrémentée de 400 pieds, comme si l'avion décollait de nouveau, ce qui n'est pas très subtil. Un second type d'attaque (SHIFT subtile) est alors testé. Cette fois-ci les variations d'altitude sont plus petites afin de dépasser les limites indiquées par la procédure tout en gardant une trajectoire crédible. Il s'agit d'une augmentation de 10 pieds à chaque paquet, donc à chaque 6 secondes (temps de mise-à-jour des paquet en phase d'approche sur *flightradar24*).

L'algorithme LSTM a été construit à partir des informations fournies dans l'article de Shabtai et Habler. Il reprend donc les travaux effectués par les auteurs, mais ne contient pas forcément les mêmes hyperparamètres et moyens d'optimisation d'optimisation. Enfin, nous utilisons la méthode d'évaluation basée sur la valeur de seuil et que nous avons décrite dans la section précédente.

Puisque les STAR sont des procédures, il est possible d'une part de les modéliser et d'autre part d'y appliquer des règles dessus. Dans notre solution, la modélisation a déjà été faite lors du processus de création de l'ontologie. La détection se fait en différentes étapes. Pour commencer, il faut savoir si l'avion qui a émis un paquet ADS-B a un plan de vol et une STAR associés. Ensuite, il faut vérifier que l'avion se trouve bien dans la zone de la STAR, et si c'est le cas, à quel niveau il en est exactement. Lorsqu'il passe certains *waypoints*, des restrictions sur l'altitude et la vitesse lui sont imposées. La règle de détection va déterminer le caractère suspect d'un paquet ADS-B en vérifiant la concordance entre les données du paquet et les restrictions.

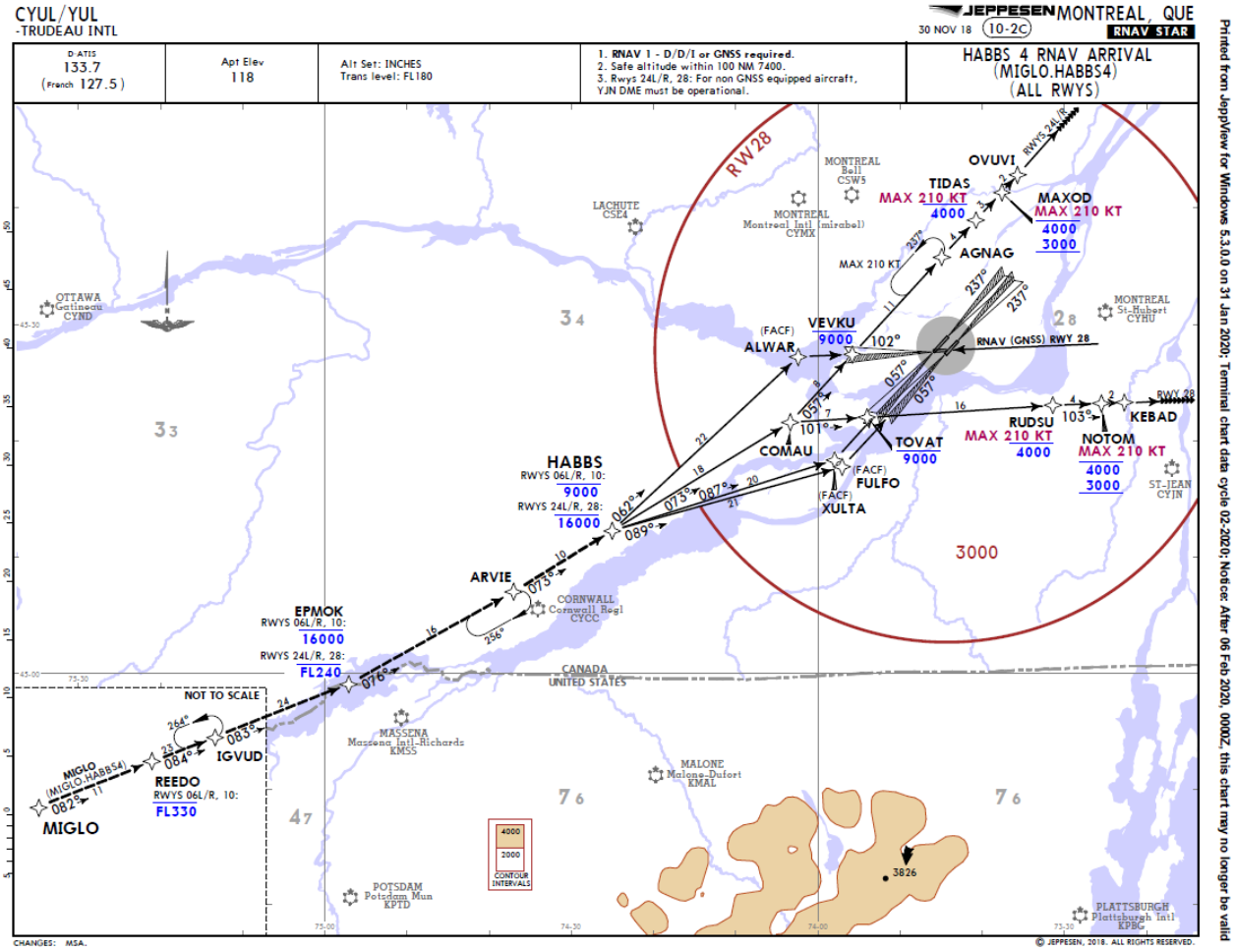


Figure 5.7 Une procédure STAR pour Montréal-Trudeau : HABBS4 [91]

Les vols de Toronto à Montréal utilisent principalement la procédure d'arrivée HABBS4 qui est détaillée dans la figure 5.7[2]. Il y a différentes routes à prendre selon la piste d'atterrissage assignée. Pour illustrer le fonctionnement d'une STAR et comment sont générées les attaques, nous allons prendre l'exemple d'un vol AC416 atterrissant sur la piste 24L. Il est soumis à différentes restrictions d'altitudes. Tout d'abord, le *waypoint* EPMOK limite l'altitude au niveau de vol FL240. Puis, à partir du *waypoint* HABBS, l'altitude est limitée à 16 000 ft. Et ainsi de suite (VEVKU à 9 000 ft, TIDAS à 4 000 ft). La dernière restriction du *waypoint* MAXOD impose une altitude maximale de 4 000 ft et une altitude minimale de 3 000 ft jusqu'à la fin de la STAR. La figure 5.8 montre l'exemple d'un vol légitime AC416 entre Toronto et Montréal, atterrissant sur la piste 24L, ainsi que les différentes attaques (SHIFT UP et SHIFT subtile) qui peuvent être perpétrées. On remarque que l'attaque SHIFT UP viole les contraintes de la STAR dès le *waypoint* HABBS, mais qu'il faut attendre le passage

2. À ne pas utiliser à des fins de navigation aérienne.

de VEVKU pour être réellement au-dessus des limites imposées ; tandis que pour l'attaque subtile, il y a une première violation des contraintes au passage de VEVKU, mais elles sont encore plus visibles après TIDAS.

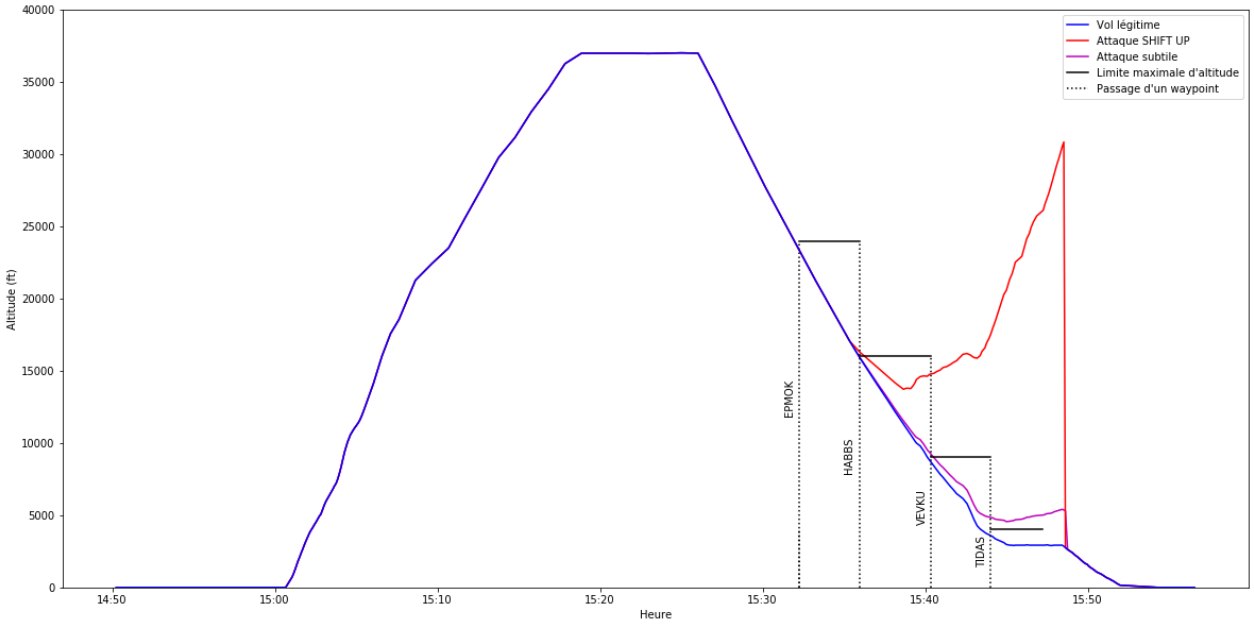


Figure 5.8 Exemple d'attaques lors d'une STAR pendant un vol AC416

Pour la comparaison entre les deux solutions, nous utilisons l'arrivée HABBS4 pour les vols en provenance de Toronto, et l'arrivée LFLER5 pour les vols venant de Vancouver.

5.5 Discussion

Dans ce chapitre nous avons présenté ATC-Sense, notre solution de détection d'attaques ADS-B basée sur l'ontologie, qui s'appuie sur le modèle ontologique et les règles développés au chapitre 4. Ensuie, nous avons détaillé la création d'ATC-Emu, une plateforme émulant un environnement ATC. Puis, nous avons décrit le cadre expérimental pour tester notre solution de détection.

Nous avons enfin proposé une comparaison de notre solution à une solution antérieure qui se base sur l'apprentissage machine. Après avoir analysé la méthode proposée et trouvé ses différentes limites, nous avons détaillé une série d'expériences sur des vols intérieurs pour pouvoir comparer notre solution avec l'apprentissage machine.

Les résultats obtenus à partir de la plateforme ATC-Sense et de la comparaison avec le LSTM sont détaillés dans le prochain chapitre.

CHAPITRE 6 RÉSULTATS ET COMPARAISON

Dans ce chapitre nous présentons les résultats de nos travaux de recherche. Tout d'abord nous discutons de la pertinence du modèle ontologique obtenu, ensuite nous évaluons les différentes logiques de détection avant de détailler les résultats obtenus lors de la comparaison avec le LSTM.

6.1 Pertinence du modèle ontologique

Notre objectif de recherche consistait à développer et valider une solution pour détecter des attaques sophistiquées contre les communications ADS-B. Nous avons ainsi créé un système expert basé sur les ontologies. En l'élaborant, nous avons développé avec l'aide de spécialistes en aviation et en ontologies un modèle qui décrit une partie de l'environnement ATC. Or, nous avons vu que des organisations comme la NASA [81] ou l'organisme européen SESAR (via le projet BEST [80]) ont développé de leur côté des ontologies sur les systèmes de gestion du trafic aérien (ATM), dont les services ATC en font parti. Notre ontologie est bien plus petite que les leurs, comme le montre le tableau 6.1. Cela permet de dire que la détection d'attaques ADS-B se fait en utilisant un nombre restreint de concepts.

Tableau 6.1 Comparaison de différentes ontologies ATM

	ATC-Sense	BEST	NASA
Classes	63	1133	165
Propriétés d'objet	47	1335	126
Propriétés de données	39	531	244

En observant de plus près les concepts clés de notre modèle ontologique, nous nous rendons compte qu'ils sont inclus dans au moins une de ces ontologies (quelques exemples sont donnés dans le tableau 6.2), ce qui vient conforter la méthode utilisée lors du développement de ce modèle. Ainsi, nous pouvons considérer que l'ontologie que nous utilisons est pertinente pour la détection d'attaques ADS-B et qu'elle représente assez bien les principaux concepts utilisés dans les systèmes ATC. En conséquence, nous pouvons dans un premier temps conclure que le processus de création du modèle ontologique, basé sur la méthode ATOM, est fiable, stable (dans le sens où il donne des ontologies qui sont directement utilisables) et va à l'essentiel. De plus, comme c'est un processus général et que ce sont les experts en contrôle aérien qui

ont insufflé à l'ontologie les connaissances, il est raisonnable de penser que ce processus peut être décliné pour créer des systèmes experts dans d'autres domaines grâce à d'autres experts.

Tableau 6.2 Correspondance de noms de différentes concepts ontologiques

	ATC-Sense	BEST	NASA
Adresse OACI	hasICAOAddress	ICAOAircraftAddress	modeSCode
Position	Position	AircraftState-Position	Location
Secteur aérien	AirspaceSector	SectorConfiguration	Sector
Site ADS-B	ADSBGroundSite	ADSBGroundStation	*Nil*
Waypoint	Waypoint	ReferenceLocation	LatLonFix

En détaillant les différentes attaques possibles contre les communications ADS-B, et en créant les règles logiques pour les détecter, nous nous sommes rendus compte que pour déterminer le caractère suspicieux d'un paquet ADS-B, il faut mettre en évidence une incohérence physique, mais surtout logique (due à la réglementation). C'est en recoupant les informations provenant de différentes sources (par exemple les paquets ADS-B, la base de données des identifiants OACI et celle des plans de vol) que l'on découvre ces incohérences. Les règles logiques que nous avons développées montrent ainsi la force de l'ontologie qui permet de raisonner sur des concepts de haut niveau.

6.2 Évaluation des requêtes

Les critères traditionnels de sécurité informatique (vrai positif, faux positif, vrai négatif, faux négatif) ne peuvent pas correctement s'appliquer pour les règles à cause d'un biais. En effet, il n'existe pas de jeux de données contenant des attaques ADS-B. Ce sont des attaques qui ont été créées pour cette expérience, pour vérifier le comportement des règles de détection, c'est pourquoi nous n'utiliserons pas cette métrique. Nous nous intéressons surtout au temps de détection pour déterminer la viabilité, ou non, d'une telle solution dans une véritable infrastructure ATC. On définit le temps de détection comme étant le temps mis par une requête pour vérifier si des paquets ADS-B sont légitimes ou non.

La première série de tests, dont les résultats sont affichés dans la figure 6.1, montre que notre solution de détection ATC-Sense est viable dans un environnement ATC. Comme nous l'avons indiqué dans le chapitre précédent, les ressources physiques de la machine sur laquelle les tests ont été faits étant limitées, et la plateforme d'émulation ATC-Emu étant gourmande en ressources, nous avons pu observer des légers ralentissement du système, ce qui s'est traduit

par une certaine latence dans l'exécution des scripts de détection. Les temps d'exécution des requêtes pour les différentes règles logiques sont de l'ordre de 1 à 3 secondes. Hormis pour la logique des radars, on observe que ce temps d'exécution reste assez stable au cours du temps.

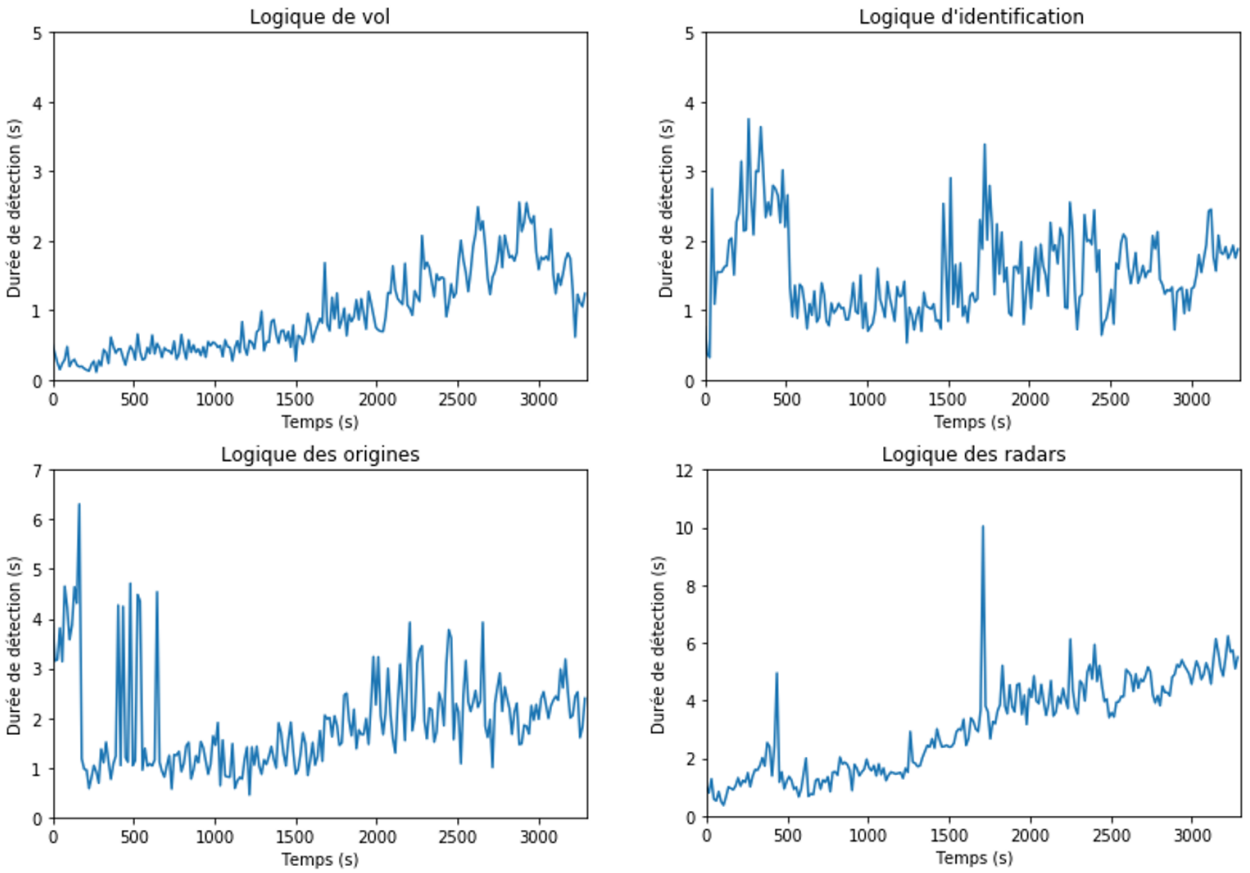


Figure 6.1 Évolution de la durée de détection des différentes logiques au cours du temps lors d'une simulation avec ATC-Emu

La seconde série de tests à été fait dans un contexte de co-simulation. Cette fois-ci, les performances d'ATC-Sense n'ont pas été impactées par la forte consommation de ressource d'ATC-Emu. Ce qui nous permet de comparer les résultats, c'est qu'ils ont été obtenus en interrogeant exactement les mêmes données. Ainsi, cette fois-ci, la durée d'exécution de requêtes est négligeable (hormis pour la logique des radars), comme le montre la figure 6.2. Elle est de l'ordre du centième de seconde.

Dans les deux séries de tests, on remarque que le temps d'exécution d'une requête reste stable au cours du temps, hormis pour la logique des radars. Or, plus la simulation avance, plus la base de données dynamique augmente au cours du temps à cause des enregistrements

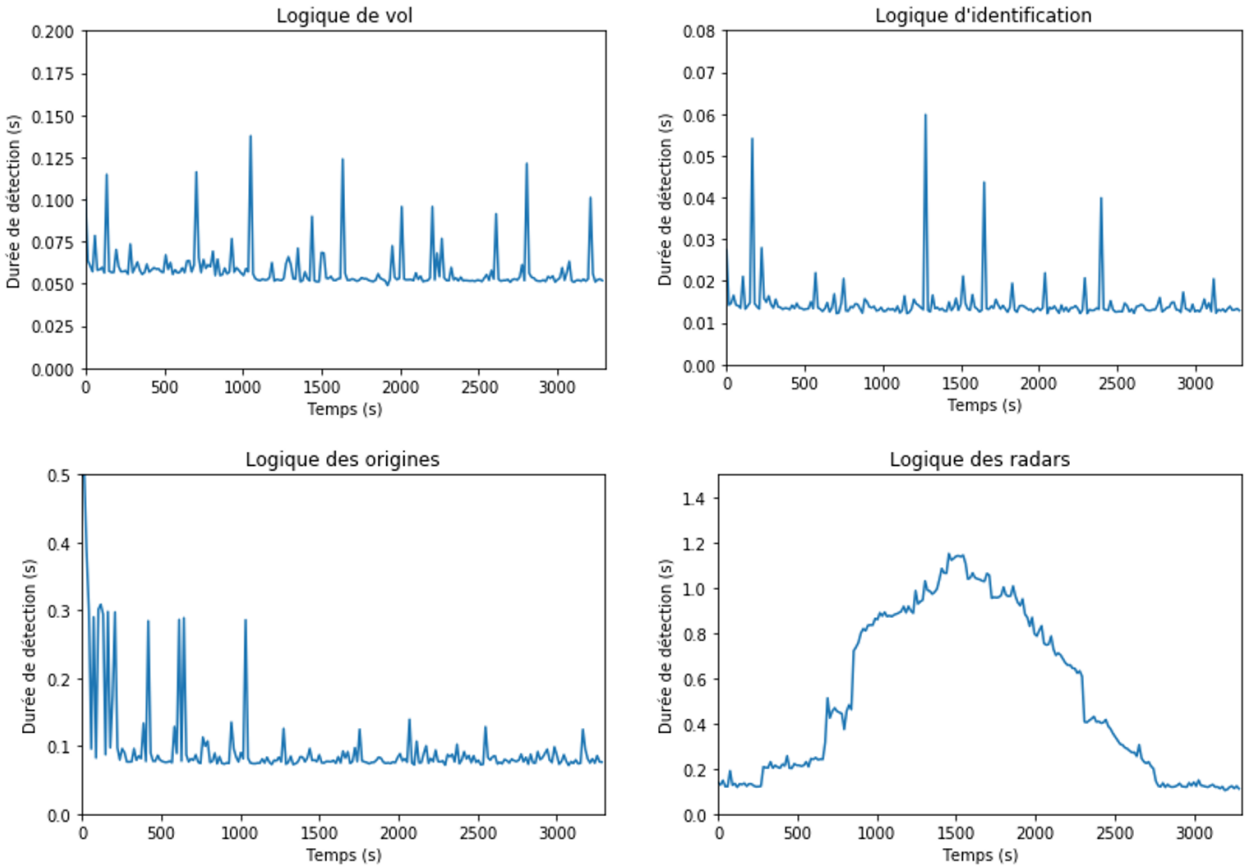


Figure 6.2 Évolution de la durée de détection des différentes logiques au cours du temps lors d'une simulation sans ATC-Emu

des données pour arriver à près de 100 000 triplets au bout d'une heure. Puisque le temps d'exécution reste stable, cela montre que la taille de l'ontologie a une faible influence sur le temps de détection. Le temps moyen de détection est de l'ordre de la seconde avec ATC-Emu et est inférieur à la seconde sans la plateforme d'émulation. Dans les deux cas, il s'agit de valeurs acceptables. En effet, c'est inférieur à la période de mise à jour des PSR et SSR (qui est entre 5 et 10 secondes). De plus, dans ses travaux [54], Berthier a demandé à des pilotes la durée qu'ils jugeaient acceptable entre l'apparition d'un écho radar et sa confirmation en tant qu'avion réel ou attaque. C'est dans le cas d'une attaque ADS-B en vue de créer un détournement de la trajectoire d'un vrai avion. Ils considèrent que 30 secondes sont tolérables et suffisantes pour engager des manœuvres d'évitement si les paquets ADS-B reçus proviennent bien d'un vrai avion. Bien que cette enquête n'ait concerné que des pilotes, on peut penser que ce chiffre doit être assez similaire pour les contrôleurs aériens. Ainsi, avec des requêtes de l'ordre de la seconde, le système expert peut être considéré comme très réactif.

Néanmoins, il faut tempérer ces résultats par la taille des tests. Ce sont des successions de

simulations d'une heure et qui ne comportent que vingt avions légitimes et cinq faux avions. Pour pouvoir garantir la qualité de ces résultats, il faudrait faire les tests en simulant un ciel complètement chargé, comme le cas des secteurs aériens les plus en tension. Enfin, ces tests ont montré que dans tous les cas, les attaques sont bien détectées et qu'il n'y a pas de faux positifs. La seule exception est la logique des radars qui affiche un taux de faux positifs de 15%. L'erreur ne provient pas de la règle de détection, mais de l'algorithme de correspondance des échos radars qui a été développé pour l'occasion et qui est simpliste.

6.3 Comparaison avec le LSTM

Les résultats de cette expérience sont montrés dans le tableau 6.3. Ils sont à prendre avec des précautions. En effet, le fait que l'algorithme utilisé ne soit pas le LSTM que les chercheurs ont entraîné, mais une version basée sur leur article, implique que les résultats que nous avons obtenus sont différents de ceux qui pourraient être obtenus avec le leur. Ainsi, il se peut que le travail de *reverse-engineering* effectué ne soit pas optimisée et présente des lacunes. De plus, nous avons effectué la comparaison avec un scénario qui est plus sophistiqué, mais qui rend le LSTM plus instables. Cela amène ainsi un biais dans notre comparaison. Cependant, la comparaison effectuée met tout de même en évidence les forces et faiblesses des deux solutions proposées. Le taux de détection de 100% de la solution ontologique s'explique par la particularité du scénario. En effet, nous avons choisi un scénario dont les attaques ont une incidence directe sur le trafic aérien. Dans les procédures d'arrivée, une façon de provoquer une confusion au niveau des contrôleurs aériens pouvant aboutir sur la mise en circuit d'attente des appareils en phase d'approche, est de faire dépasser les altitudes requises au passage de certains *waypoint*. C'est donc pour ça que la règle ontologique fonctionne bien. Si ces attaques ne faisaient pas dépasser les altitudes maximales demandées, alors l'ontologie ne les détecterait pas, mais les conséquences seront minimes ou inexistantes sur la circulation aérienne.

Comme nous l'avions mentionné dans les limites de l'implémentation de Shabtai et Habler, le seuil de détection dépend fortement de la «force» de l'attaque, et la qualité de détection va de pair. Le mauvais score du réseau de neurones s'explique par plusieurs raisons. La première est que l'attaque a eu lieu en phase de descente. Cette phase, avec le décollage, est la partie la plus instable du vol. En effet, c'est à ce moment qu'il y a le plus de variations de paramètres (altitude, cap, vitesse, position). Cette instabilité se traduit par un score d'anomalie élevée lors des phases de tests, contre des résultats presque nuls pour la phase de croisière. On retrouve cette instabilité dans la différence entre le vol court-courrier et le vol long-courrier. Dans le premier cas, la partie croisière du vol ne dure qu'environ 16% du temps de vol, contre

79% pour le vol long-courrier. Le seuil d'anomalie est donc plus élevé dans le premier cas. La seconde raison du mauvais score du LSTM est un problème lié aux données. Pour ressembler à l'implémentation de Shabtai et Habler, elles sont extraites du site *flightradar24*. Or, leur jeu de données est actualisé toutes les 6 à 30 secondes, loin des 1 à 2 secondes de la réalité. Ce taux d'actualisation est suffisant pour la partie en-route, mais apporte plus d'instabilité pour les phases de montée et de descente. De plus, le temps de détection de notre règle ontologique est de l'ordre de la seconde, comme pour les précédentes règles créées. Tandis que pour le LSTM, Shabtai et Habler ont choisi un temps de détection correspondant à la création de 15 séries de messages malicieux. Or, comme les messages ADS-B de *flightradar24* sont actualisés toutes les 6 à 30 secondes, cela donne un temps de détection de 1 minute 30 à près de 8 minutes.

Tableau 6.3 Taux de détection du système expert selon la logique de détection et par type d'attaque lancée dans la phase de descente d'un avion suivant la procédure STAR

Attaque	CYYZ-CYUL (LSTM)	CYYZ-CYUL (Ontologie)	CYVR-CYUL (LSTM)	CYVR-CYUL (Ontologie)
SHIFT UP	23,33%	100%	46,66%	100%
SHIFT subtile	3,33%	100%	5%	100%

La grande force du LSTM réside dans le fait qu'il ne dépend d'aucune donnée extérieure, si ce n'est l'origine et la destination de l'appareil ainsi qu'un modèle entraîné sur des vols passés. Ce qui n'est pas le cas pour la règle de détection qui a dû vérifier le plan de vol et chercher les informations liées à la STAR associée. Cette indépendance est encore plus importante dans le cas où les paquets modifiés concernent aussi la position. Cela augmenterait encore plus le score d'anomalie et l'attaque serait plus vite détectée, tandis que la règle de détection des STAR serait inutilisable. Par contre, comme les autres règles sont appliquées de leur côté, cette attaque serait détectée par la logique des radars (la majorité des STAR concernent des aéroports d'importance qui ont des radars d'approche) ou par une logique de trajectoire. L'autre intérêt du LSTM est son unicité. C'est le même algorithme qui est appliqué, quelle que soit la situation. Alors que pour le système expert, ce sont toutes les règles qui doivent être appliquées pour juger de la véracité d'un paquet ADS-B, ce qui en fait un système plus complexe et gourmand en termes de ressources.

Pour conclure, le taux de faux positifs est important à prendre en compte. Le but des outils de détection de faux messages ADS-B est d'aider les contrôleurs aériens dans leur travail en évitant qu'une attaque ADS-B suscite une incompréhension voire les amène à poser de mauvaises décisions. Or, ces attaques sont encore loin de faire partie de leur quotidien. Avec

un taux de faux positif de 4%[1] obtenus pour le LSTM [61] et une circulation quotidienne moyenne d'environ 45 000 avions dans les espaces contrôlés par la FAA aux États-Unis [92], ce sont donc 1800 fausses alertes auxquelles les contrôleurs devront faire face chaque jour. Et il est peu probable qu'ils subissent autant d'attaques, ce qui remet en question l'utilisation du LSTM comme moyen de détection.

6.4 Discussion

À travers ce chapitre, nous avons présenté les différents résultats de nos travaux. Ainsi, quatre logiques de détection contre différents types d'attaques ont été évaluées. Tout d'abord, la création du modèle ontologique et des règles du système expert ont montré que l'utilisation d'ontologies est appropriée pour détecter des attaques ADS-B, notamment grâce aux corrélations qui peuvent être faites avec la sémantique. Ensuite, lors de différents tests, il résulte que le système expert ontologique est performant dans le sens où la durée de détection est de l'ordre d'une seconde et que toutes les règles détectent correctement les attaques contre lesquelles elles sont créées. Il est à noter que l'exception de la logique des radars avec un taux de faux positifs de 15% n'est pas due à la détection en tant que telle, mais à l'algorithme de correspondance des points PSR,SSR et ADS-B. Cet algorithme a été créé pour des tests et est largement en deçà de ceux qui se trouvent dans le marché des systèmes ATC.

Nous avons enfin comparé notre solution vis-à-vis d'une solution antérieure se basant sur l'apprentissage machine. Les tests effectués nous ont permis de démontrer la performance de notre ontologie entre terme de rapidité de détection et de flexibilité. Néanmoins, nous avons remarqué que les règles de détection nécessitent d'avoir accès à diverses sources de données et que l'environnement soit correctement modélisé, contrairement à l'algorithme d'intelligence artificielle qui se base uniquement sur les messages ADS-B.

Le dernier chapitre de ce mémoire sera consacré à une synthèse des travaux effectués, une description des limites et contraintes rencontrées avant de proposer diverses pistes pour des travaux futurs.

1. Nous avons bien obtenu un taux de 15% pour la logique des radars, mais cette exception est due à un algorithme de correspondance des données PSR, SSR et ADS-B non performant. Les autres logiques n'ont pas catégorisé de faux positifs.

CHAPITRE 7 CONCLUSION

À travers cette recherche, nous avons proposé une solution de détection d'attaques informatiques sophistiquées contre les communications ADS-B. Trois questions de recherche ont aidé à la réalisation de notre objectif :

Q1 : Quelles sont les différentes cyberattaques plausibles sur les communications ADS-B ? Quels sont les impacts qu'elles peuvent avoir ?

Q2 : Quelles sont l'efficacité et la flexibilité de l'ontologie pour détecter ces attaques en comparaison avec des solutions antérieures ?

Q3 : Quelle est la performance de la solution proposée et sa viabilité en temps réel ?

Pour conclure ce mémoire, les travaux entrepris sont synthétisés et analysés afin d'en déterminer les limites et de proposer des améliorations pour ouvrir la voie à d'autres travaux de recherche.

7.1 Synthèse des travaux

La solution de détection d'attaques ADS-B, ATC-Sense, s'appuie sur un modèle ontologique représentant l'environnement ATC. Il est intégré au sein d'une plateforme émulant une infrastructure ATC. Cette plateforme, nommée ATC-Emu, est constituée d'un logiciel de contrôle du trafic aérien, Euroscope, et d'un module qui simule les différents radars, le réseau d'échange de données (DDS) et leur enregistrement dans la base de données ontologique. ATC-Emu représente le système de contrôle aérien utilisé au Canada. Un module permet de simuler les différentes attaques injectées sur le réseau DDS et un autre s'occupe de la détection. La détection s'appuie sur des règles ontologiques et forme un système expert.

La première question de recherche nous a amenés à considérer quatre types d'acteurs potentiellement malveillants (amateurs, professionnels, activistes et terroristes), de déterminer trois zones sensibles (l'arrivée océanique, la transition entre deux secteurs aériens et les STAR/SID au niveau des aéroports) ainsi que les impacts que peuvent avoir les attaques allant d'une simple gêne à la fermeture de l'espace aérien. Cela a abouti à la création de scénarios d'attaques et de règles de détection. Ces dernières forment différentes logiques : identification, origine, radars et logique de vol. Chaque règle restreint le champ d'action des attaquants.

Pour répondre à la seconde question de recherche, nous avons mené une comparaison avec une solution [61] proposant d'utiliser l'apprentissage machine pour détecter de faux paquets ADS-B. Nous avons reproduit l'algorithme LSTM décrit par les créateurs de cette solution et avons choisi de faire la comparaison au niveau des STAR, une des zones à haut risque identifiée. Pour la solution ontologique, il a fallu implémenter une nouvelle règle de détection qui prenne en compte les particularités de la STAR. De la comparaison il résulte que notre solution se montre plus efficace en termes de taux de détection et de temps de détection, et flexible. En effet, le LSTM ne détecte pas des variations subtiles de l'altitude en phase d'approche, ce qui n'est pas le cas du raisonnement ontologique puisqu'il est basé sur des conditions.

Enfin, pour la dernière question de recherche, nous observons un temps d'exécution des requêtes qui dépend peu de la taille du graphe et du nombre de données enregistrées. Il est de l'ordre de la seconde, voire du centième de seconde pour certaines requêtes. Ces tests sont encourageants et comme l'enregistrement des données radars et ADS-B se fait de façon locale au niveau de chaque centre de contrôle, cela restreint la quantité de données à utiliser et donc le temps de requête. La viabilité de la détection de faux avions est aussi justifiée par l'intérêt que portent les régulateurs à l'ontologie et leur envie de déployer des outils basés sur un modèle ontologique pour les systèmes ATM. Le système expert se trouve alors totalement intégré dans cette vision ontologique.

7.2 Limitations de la solution proposée

Le choix d'utiliser un système expert pour détecter des faux messages ADS-B implique que la qualité de la détection dépende de la capacité à bien modéliser l'environnement ATC et à créer des règles qui couvrent toutes les situations. Ce mémoire propose ainsi cinq règles. Elles permettent d'éviter une bonne partie des attaques sophistiquées, mais laissent encore passer d'autres types d'attaques, notamment le *jamming*. De plus, ces règles représentent des situations idéalisées. Ainsi, nous avons défini les radars comme ayant une zone de couverture cylindrique, ce qui n'est pas le cas dans la réalité. La logique de détection des origines et celle des radars se basent donc sur une représentation simplifiée de l'espace aérien. D'une façon générale, il y a donc une abstraction des contraintes physiques que l'on trouve dans la réalité et qui perturbent la transmission des données : interférences, conditions météo et problème de couverture à cause du terrain, panne de transpondeur ou allumage tardif pour les pilotes, et bien d'autres. Ce sont des situations exceptionnelles qui deviennent des faux positifs si la détection ne les prend pas en compte. Les règles présentées dans ce mémoire donnent un aperçu du fonctionnement d'un système expert ontologique pour l'aviation, mais doivent être

approfondies pour être déployées sur les logiciels ATC.

D'un point de vue technique, les règles de détection n'utilisent pas pleinement le potentiel d'une base de données ontologique. En effet, le fait que le calcul 3D ne soit pas possible avec les requêtes SPARQL oblige à devoir passer par des scripts Python pour effectuer ces calculs. Il n'y a donc pas toujours de traduction directe entre la règle haut niveau que l'on décrit et son équivalence en requête SPARQL. Cela fait intervenir un script tiers et occasionne donc une étape supplémentaire.

La solution proposée ne prend pas en compte les échanges vocaux entre les contrôleurs et les pilotes. C'est une source d'information en moins. Si, par exemple, les contrôleurs voient un écho ADS-B sur leur radar et ne sont pas capables de joindre l'appareil, il peut s'agir d'un problème de communication, d'un oubli de changement de fréquence, d'un geste volontaire du pilote ou d'un faux avion. Pour continuer avec les informations manquantes, les *flight strips*, bien qu'ils soient modélisés, ne sont pas encore utilisées dans les solutions de détection. Or ils recensent les décisions prises par le contrôleur aérien, dont des modifications de trajectoire, des mises sur circuit d'attente, etc.

7.3 Améliorations futures

La présente recherche a mis en évidence la capacité des ontologies à former un système expert fiable et performant pour détecter des attaques ADS-B. Avec l'engouement des régulateurs européens et américains pour l'utilisation des ontologies dans les systèmes de gestion du trafic aérien, c'est un domaine qui a un grand potentiel. Cela nécessite donc de chercher des pistes d'améliorations et axes de recherche futurs pour passer d'une simple preuve de concept à un projet plus abouti.

Tout d'abord, les différents scénarios durent de quelques minutes à quelques heures. Cela n'est pas suffisant pour tester la viabilité de cette solution à grande échelle. Il faudrait avoir des simulations sur plusieurs journées. Mais pour continuer à interagir sur le trafic aérien, cela nécessite la présence de contrôleurs tout au long de la simulation. Une autre façon de pallier ce problème serait de faire les simulations en utilisant directement les données du trafic en cours. Par contre, on perdrait alors tout l'aspect de contrôle et d'interaction avec les pilotes.

La gestion de la base de données ontologique devrait être également analysée et revue. En augmentant la durée des scénarios, cela permet d'observer les conséquences que peuvent avoir les enregistrements des données dans la base de données ontologique sur la qualité et la rapidité de détection. Pour remédier à de potentielles lenteurs, une idée serait de créer un

système avec trois composants : mémoire courte durée, mémoire longue durée et archivage. Cela amènerait de la fluidité dans les requêtes.

Enfin, il serait intéressant de compléter l'approche de cybersécurité par des études comportementales sur les contrôleurs aériens. À travers différentes expériences, ont pourrait s'intéresser aux comportements qu'ils peuvent avoir avec des variables comme la connaissance de la possibilité de créer de faux paquets ADS-B et la présence ou non d'un outil de détection d'attaques ADS-B. On obtiendrait non seulement des mesures sur la réelle incidence que ces attaques ont sur le contrôle du trafic aérien, mais aussi sur l'aide ou la gêne que le dispositif apporte. Cela permettra alors de créer une approche de cyberrésilience en préparant le comportement à adopter par les contrôleurs aériens en cas d'attaque.

RÉFÉRENCES

[1] Federal Aviation Administration - Alaskan Region, “Capstone test and evaluation master plan for ADS-B radar-like services,” 30 jan. 2000. [En ligne]. Disponible : https://www.faa.gov/nextgen/programs/adsb/Archival/media/TEMPFIN.PDF

[2] Federal Aviation Administration, “14 CFR 91.225,” 2019. [En ligne]. Disponible : https://ecfr.federalregister.gov/current/title-14/chapter-I/subchapter-F/part-91/subpart-C/section-91.225

[3] ——, “14 CFR 91.227,” 2010. [En ligne]. Disponible : https://ecfr.federalregister.gov/current/title-14/chapter-I/subchapter-F/part-91/subpart-C/section-91.227

[4] International Civil Aviation Organization, “Annex 5 - Units of measurement to be used in the air and ground services,” 2010.

[5] W. Goerge, “NAV CANADA Implements ADS-B,” dans *Integrated Communications, Navigation and Surveillance (ICNS) Conference*, Arlington, VA, 13-15 mai 2009, p. 1–9. [En ligne]. Disponible : https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5172868

[6] International Civil Aviation Organization, *Operational Opportunities*, 2010. [En ligne]. Disponible : https://www.icao.int/environmental-protection/Documents/EnvironmentReport-2010/ICAO_EnvReport10-Ch3_en.pdf

[7] Federal Aviation Administration, “What is NextGen ?” 2020. [En ligne]. Disponible : https://www.faa.gov/nextgen/what_is_nextgen/

[8] SESAR Joint Undertaking, “ADS-B and other means of surveillance implementation status,” 15 mai 2018. [En ligne]. Disponible : https://ec.europa.eu/transport/sites/transport/files/20180515-sesar-ads-b-report.pdf

[9] Aireon. (2020) About aireon. Dernier accès le 25 août 2020. [En ligne]. Disponible : https://aireon.com/company/

[10] International Civil Aviation Organization. (2019) Global tracking initiatives. [En ligne]. Disponible : https://www.icao.int/safety/globaltracking/Pages/Homepage.aspx

[11] Aireon. (2017) Aireon and flightaware partner to launch globalbeacon airline solution for ICAO airline flight tracking compliance. [En ligne]. Disponible : https://aireon.com/2016/09/21/aireon-flightaware-globalbeacon/

[12] GlobalBeacon. (2020) Are you ready for GADSS ? [En ligne]. Disponible : https://globalbeacon.aero/

[13] FlightAware. (2020) Flightaware ADS-B statistics. Dernier accès le 25 août 2020. [En ligne]. Disponible : https://flightaware.com/adsb/stats/

[14] Federal Aviation Administration. (2020) ADS-B privacy. Dernier accès le 25 août 2020. [En ligne]. Disponible : https://www.faa.gov/nextgen/equipadsb/privacy/

[15] A. Costin et A. Francillon, "Ghost in the air (traffic) : on insecurity of ADS-B protocol and practical attacks on ADS-B devices," dans *Black Hat USA*, Las Vegas, NV, USA, 2012. [En ligne]. Disponible : http://www.eurecom.fr/fr/publication/3788/download/rs-publi-3788.pdf

[16] M. Schäfer, V. Lenders et I. Martinovic, "Experimental analysis of attacks on next generation air traffic communication," dans *Integrated Conference on Applied Cryptography and Network Security*, Springer, édit., Berlin, Heidelberg, 13-15 mai 2013, p. 253–271. [En ligne]. Disponible : https://link.springer.com/chapter/10.1007/978-3-642-38980-1_16

[17] M. Strohmeier, "Security in next generation air traffic communication networks," thèse de doctorat, Kellogg College, University of Oxford, Oxford, UK, 2016. [En ligne]. Disponible : https://ora.ox.ac.uk/objects/uuid:c5c61de4-ffef-479e-9f49-de38c2a8e9ec

[18] D. McCallie, J. Butts et R. Mills, "Security analysis of the ADS-B implementation in the next generation air transportation system," *International Journal of Critical Infrastructure Protection*, vol. 4, p. 78–87, 8 2011. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S1874548211000229

[19] M. Riahi Manesh et N. Kaabouch, "Analysis of vulnerabilities, attacks, countermeasures and overall risk of the automatic dependent surveillance-broadcast (ADS-B) system," *International Journal of Critical Infrastructure Protection*, vol. 19, p. 16–31, 12 2017. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S1874548217300446

[20] Nav Canada, "Aeronautical study - Canadian ADS-B out performance requirements mandate," août 2018. [En ligne]. Disponible : https://www.navcanada.ca/EN/products-and-services/Study%20Recommendations/Study%20Final%20Report%20-%20ADS-B-EN.pdf

[21] L.-P. Morel, "Using ontologies to detect anomalies in the sky," mémoire de maîtrise, Dép. de génie informatique et génie logiciel, École Polytechnique de Montréal, Montréal, QC, 2017. [En ligne]. Disponible : https://publications.polymtl.ca/2818/

[22] International Civil Aviation Organization, "Procedures for air navigation services - air traffic management," 2016, doc 4444.

[23] Commission des Communautés Européennes, “La gestion du trafic aérien,” Bruxelles, Belgique, 6 mars 1996. [En ligne]. Disponible : https://eur-lex.europa.eu/legal-content/FR/TXT/PDF/?uri=CELEX:51996DC0057&from=SL

[24] Thales Air Systems, “Air traffic management - a guide to global surveillance,” 2014. [En ligne]. Disponible : https://www.icao.int/NACC/Documents/Meetings/2014/ADSBIMP/ADSBIMPP12.pdf

[25] International Civil Aviation Organization, “Manual on mode S specific services,” 2004, Doc 9688 AN/952.

[26] ——, “Technical provisions for mode S services and extended squitter,” 2017, doc 9871 AN/460.

[27] A. Mattos, “ADS-C technical aspects and implementation status,” dans *ICAO Seminar on the Implementation of Aeronautical Surveillance and Automation Systems in the SAM Region*, San Carlos de Bariloche, Argentine, 6-8 déc. 2010. [En ligne]. Disponible : https://www.icao.int/SAM/Documents/2010/SURAUTOSEM/15%20SITA_ADS%20C%20Technical%20Aspects%20and%20Implementation%20Status.pdf

[28] Inmarsat et A. Australia, “Australian enhanced flight tracking evaluation,” 2015. [En ligne]. Disponible : https://www.icao.int/safety/globaltracking/Documents/ICAO%20Global%20Flight%20Tracking%20-%20Regional%20evaluation%20using%20ADS-C.pdf

[29] Federal Aviation Administration, “Operational policy ADS-C distance-based separation - AIP part. 2 enr 7.5,” 2020. [En ligne]. Disponible : https://www.faa.gov/air_traffic/publications/atpubs/aip_html/part2_enr_section_7.5.html

[30] International Civil Aviation Organization, “Convention relative à l’aviation civile internationale,” 2006. [En ligne]. Disponible : https://www.icao.int/publications/Documents/7300_cons.pdf

[31] Transports Canada, “Canada’s airspace - information on airspace classification and structure,” 2018. [En ligne]. Disponible : https://www2.tc.gc.ca/publications/bil/tp6010/pdf/hr/tp6010b.pdf

[32] Nav Canada, “Manuel des espaces aériens désignés,” 2020. [En ligne]. Disponible : https://www.navcanada.ca/EN/products-and-services/Documents/DAH_Current_FR.pdf

[33] ——, “Nav canada regions,” 2018. [En ligne]. Disponible : https://www.navcanada.ca/EN/careers/techops/workenvironment/Documents/Technical%20Operations%20Regions_EN.pdf

[34] ——. (2020) Air traffic control. Dernier accès le 25 août 2020. [En ligne]. Disponible : https://www.navcanada.ca/EN/about-us/Pages/what-we-do-atc.aspx

[35] Transports Canada, “Aviation civile et industrie aéronautique,” dans *Un ciel à partager - Guide de l’industrie de l’aviation à l’intention des gestionnaires de la faune*, 2004, p. 80–81, TP 13549F. [En ligne]. Disponible : https://tc.canada.ca/sites/default/files/migrated/tp13549f.pdf

[36] International Civil Aviation Organization, “Annex 10 volume IV- Surveillance and collision avoidance systems,” 2014.

[37] Bundesstelle für Flugunfalluntersuchung, “Investigation report of a midair collision near Überlingen,” mai 2004. [En ligne]. Disponible : https://www.bfu-web.de/EN/Publications/Investigation%20Report/2002/Report_02_AX001-1-2_Ueberlingen_Report.pdf

[38] DDS Foundation, “What is DDS ?” dernier accès le 25 août 2020. [En ligne]. Disponible : https://www.dds-foundation.org/what-is-dds-3/

[39] Nav Canada, “Nav canada améliore la plateforme de technologies ATM avec RTI Connext,” 2013, dernier accès le 25 août 2020. [En ligne]. Disponible : https://www.navcanada.ca/FR/media/pages/news-releases-2013-nr25.aspx

[40] Adlink, “Coflight consotium selects vortex openslice DDS middleware for next generation european flight data processor,” dernier accès le 25 août 2020. [En ligne]. Disponible : https://www.adlinktech.com/en/Coflight

[41] A. Corsaro, “Coflight eFDP,” 2006. [En ligne]. Disponible : https://www.dds-foundation.org/sites/default/files/dds_07-07-04.pdf

[42] T. Berners-Lee et J. Hendler, “Publishing on the semantic web,” *Nature*, vol. 410, p. 1023–4, 05 2001.

[43] T. R. Gruber, “Toward principles for the design of ontologies used for knowledge sharing,” *International Journal of Human-Computer Studies*, vol. 43, 08 1994.

[44] Birckley, Dan and Miller, Libby. (2014) FOAF vocabulary specification 0.99. Dernier accès le 25 août 2020. [En ligne]. Disponible : http://xmlns.com/foaf/spec/

[45] Dublin Core Metadata Initiative. (2020) Dernier accès le 25 août 2020. [En ligne]. Disponible : http://dublincore.org/

[46] W3C. (2012) SKOS simple knowledge organizations systems - home page. Dernier accès le 25 août 2020. [En ligne]. Disponible : https://www.w3.org/2004/02/skos/

[47] Linked Open Vocabularies, “Linked open vocabularies,” 2020, dernier accès le 25 août 2020. [En ligne]. Disponible : https://lov.linkeddata.es/dataset/lov/

[48] Prud’hommeaux, Éric and Seaborne, Andy, “SPARQL query language for RDF,” 2008, dernier accès le 25 août 2020. [En ligne]. Disponible : https://www.w3.org/TR/rdf-sparql-query/

[49] K. D. Wesson, T. E. Humphreys et B. L. Evans, “Can cryptography secure next generation air traffic surveillance ?” *IEEE Security and Privacy Magazine*, 2014.

[50] M. Bellare, P. Rogaway et T. Spies, “The ffx mode of operation for format-preserving encryption,” *NIST submission*, vol. 20, p. 19, 2010.

[51] C. Finke, J. Butts, R. Mills et M. Grimaila, “Enhancing the security of aircraft surveillance in the next generation air traffic control system,” *International Journal of Critical Infrastructure Protection*, vol. 6, n°. 1, p. 3–11, 2013.

[52] Z. Wu, A. Guo, M. Yue et L. Liu, “An ADS-B message authentication method based on certificateless short signature,” *IEEE Transactions on Aerospace and Electronic Systems*, 2019.

[53] P. Berthier, “SAT : Sécurisation de l’ADS-B grâce à TESLA,” mémoire de maîtrise, Dép. de génie informatique et génie logiciel, École Polytechnique de Montréal, Montréal, QC, 2017. [En ligne]. Disponible : https://publications.polymtl.ca/2540/

[54] P. Berthier, J. M. Fernandez et J.-M. Robert, “SAT : Security in the air using Tesla,” dans *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*. IEEE, 2017, p. 1–10.

[55] A. Perrig, R. Canetti, J. D. Tygar et D. Song, “The tesla broadcast authentication protocol,” *Rsa Cryptobytes*, vol. 5, n°. 2, p. 2–13, 2002.

[56] M. Schäfer, V. Lenders et J. Schmitt, “Secure track verification,” dans *2015 IEEE Symposium on Security and Privacy*. IEEE, 2015, p. 199–213.

[57] M. Strohmeier, V. Lenders et I. Martinovic, “Lightweight location verification in air traffic surveillance networks,” dans *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*, 2015, p. 49–60.

[58] K. Sampigethaya et R. Poovendran, “Visualization & assessment of ADS-B security for green ATM,” dans *29th Digital Avionics Systems Conference*. IEEE, 2010, p. 3–A.

[59] M. Schäfer, P. Leu, V. Lenders et J. Schmitt, “Secure motion verification using the doppler effect,” dans *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2016, p. 135–145.

[60] N. Ghose et L. Lazos, “Verifying ADS-B navigation information through doppler shift measurements,” dans *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*. IEEE, 2015, p. 4A2–1.

[61] E. Habler et A. Shabtai, “Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages,” *Computers & Security*, vol. 78, p. 155–173, 2018.

[62] S. Akerman, E. Habler et A. Shabtai, “VizADS-B : Analyzing sequences of ADS-B images using explainable convolutional LSTM encoder-decoder to detect cyber attacks,” *arXiv preprint arXiv :1906.07921*, 2019.

[63] V. Chandola, A. Banerjee et V. Kumar, “Anomaly detection : A survey,” *ACM computing surveys (CSUR)*, vol. 41, nº. 3, p. 1–58, 2009.

[64] R. Chalapathy et S. Chawla, “Deep learning for anomaly detection : A survey,” *arXiv preprint arXiv :1901.03407*, 2019.

[65] A. Babar, “An approach to represent and transform application specific constraints for an intrusion detection system,” mémoire de maîtrise, Queen’s University, Kingston, ON, 2020. [En ligne]. Disponible : https://qspace.library.queensu.ca/handle/1974/27685

[66] BEST Consortium, “Project summary,” 2016. [En ligne]. Disponible : https://project-best.eu/downloads/The_BEST_project_summary.pdf

[67] C. C. Insaurralde et E. Blasch, “Ontological knowledge representation for avionics decision-making support,” dans *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. IEEE, 2016, p. 1–8.

[68] R. M. Keller, “Ontologies for aviation data management,” dans *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. IEEE, 2016, p. 1–9.

[69] A. Souag, C. Salinesi et I. Comyn-Wattiau, “Ontologies for security requirements : A literature survey and classification,” dans *International conference on advanced information systems engineering*. Springer, 2012, p. 61–69.

[70] A. Herzog, N. Shahmehri et C. Duma, “An ontology of information security,” *International Journal of Information Security and Privacy (IJISP)*, vol. 1, nº. 4, p. 1–23, 2007.

[71] J. Undercoffer, A. Joshi et J. Pinkston, “Modeling computer attacks : An ontology for intrusion detection,” dans *International Workshop on Recent Advances in Intrusion Detection (RAID)*. Springer, 2003, p. 113–135.

[72] E. Ducharme, “Détection d’intrusion à l’aide d’un système expert basé sur l’ontologie,” mémoire de maîtrise, Dép. de génie informatique et génie logiciel, École Polytechnique de Montréal, Montréal, QC, 2017. [En ligne]. Disponible : https://publications.polymtl.ca/2923/

[73] S. Malenfant-Corriveau, “Proposition d’une méthode de développement d’ontologie pour un système expert en sécurité,” mémoire de maîtrise, Dép. de génie informatique et génie logiciel, École Polytechnique de Montréal, Montréal, QC, 2017. [En ligne]. Disponible : https://publications.polymtl.ca/2922/

[74] A. Sadighian, S. T. Zargar, J. M. Fernandez et A. Lemay, “Semantic-based context-aware alert fusion for distributed intrusion detection systems,” dans *2013 International Conference on Risks and Security of Internet and Systems (CRiSIS)*. IEEE, 2013, p. 1–6.

[75] F. Massacci, J. Mylopoulos, F. Paci, T. T. Tun et Y. Yu, “An extended ontology for security requirements,” dans *International Conference on Advanced Information Systems Engineering*. Springer, 2011, p. 622–636.

[76] H. Obas, E. M. Niang, I. Ouchen, S. Bahbah, G. Chauvy et O. Dia, “Exploitation de l’A321LR chez Air Transat,” École des Sciences de la Gestion de l’UQÀM et Université Paris Dauphine-PSL, Rapport technique, mai 2020.

[77] Transports Canada, “11-09-2001 four days in september,” 2002. [En ligne]. Disponible : https://tc.canada.ca/sites/default/files/migrated/sept_11_2001_four_days_in_september.pdf

[78] J. A. Wise, V. D. Hopkin et D. J. Garland, “Air-traffic controller memory,” dans *Handbook of aviation human factors*. CRC Press, 2016, ch. 21, p. 21–1 à 21–29.

[79] Protégé. A free, open-source ontology editor and framework for building intelligent systems. [En ligne]. Disponible : https://protege.stanford.edu/

[80] BEST Consortium. (2020) Publications. Dernier accès le 25 août 2020. [En ligne]. Disponible : https://project-best.eu/publications.html

[81] R. M. Keller, “The NASA air traffic management ontology,” 2018, dernier accès le 25 août 2020. [En ligne]. Disponible : https://data.nasa.gov/ontologies/atmontoCore/

[82] Nav Canada, “AIP canada (OACI) partie 2 - en-route (ENR),” 2020. [En ligne]. Disponible : https://www.navcanada.ca/FR/products-and-services/Documents/AIP/Current/part_2_enr/2enr_fre.pdf

[83] ——, “The future of air traffic surveillance is here,” 2020. [En ligne]. Disponible : https://www.navcanada.ca/EN/products-and-services/surveillance/Documents/ADS-B_The%20Future_%20ENG_FINAL.pdf

[84] Ontotext. (2020) GraphDB the best RDF database for knowledges graphs. Dernier accès le 25 août 2020. [En ligne]. Disponible : https://www.ontotext.com/products/graphdb/

[85] R. M. Keller, “Building a knowledge graph for the air traffic management community,” dans *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, p. 700–704. [En ligne]. Disponible : https://dl.acm.org/doi/abs/10.1145/3308560.3317706

[86] VATSIM. What is VATSIM ? Dernier accès le 25 août 2020. [En ligne]. Disponible : https://www.vatsim.net/about

[87] A. Lemay, J. Fernandez et S. Knight, “An isolated virtual cluster for scada network security research,” dans *1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013) 1*, 2013, p. 88–96.

[88] G. Csernák, “Euroscope user’s manual v3.2.1.20,” 2019. [En ligne]. Disponible : https://www.euroscope.hu/documents/EuroScopeUsersManual.3.2.0.20.pdf

[89] CANscope, “Canscope CAATS modification for euroscope,” 2020. [En ligne]. Disponible : https://www.vatcan.ca/canscope/index.html

[90] Bureau de la sécurité des transports du Canada, “Rapport d’enquête sur la sécurité du transport aérien A18Q0069,” 10 juil. 2019. [En ligne]. Disponible : https://www.tsb.gc.ca/fra/rapports-reports/aviation/2018/a18q0069/a18q0069.pdf

[91] Jeppesen, “HABBS4 RNAV arrival,” 2020. [En ligne]. Disponible : https://ww2.jeppesen.com/

[92] Federal Aviation Administration. Air traffic by the numbers. Dernier accès le 25 août 2020. [En ligne]. Disponible : https://www.faa.gov/air_traffic/by_the_numbers/

[93] J. Bodart, “Mode S surveillance principle,” dans *Surveillance/MICA Workshop*, Le Caire, Égypte, 26-28 fév. 2019. [En ligne]. Disponible : https://www.icao.int/MID/Documents/2019/MICA/MICA-MID%20-%20WP%2002%20-%20Mode%20S%20Surveillance%20Principle.pdf

ANNEXE A COMPARAISON DES MOYENS DE DÉTECTION D'AVIONS

	PSR	SSR	ADS-B	ADS-C	MLAT
Requiert un équipement embarqué	Non	Oui	Oui	Oui	Oui
Identification	Non	Oui	Oui	Oui	Oui
Données reçues pour un aéronef sans transpondeur	Position et vitesse calculées	Nil	Nil	Nil	Nil
Données reçues par un transpondeur mode A/C	Position et vitesse calculées	Position et vitesse calculées, altitude barométrique, squawk	Nil	Nil	Position et vitesse calculées, altitude barométrique, squawk
Données reçues par un transpondeur mode S downlink ou un transmetteur ADS-B out	Position et vitesse calculées	Mode A/C + BDS [1] 1,0 ; BDS 1,7 ; BDS 2,0 (call sign) ; BDS 4,0 (information saisies sur l'auto-pilote) ; BDS 5,0 ; BDS 6,0 (cap et vitesse)	Si compatibilité ADS-B : Identification, position à la surface, position de l'avion (altitude barométrique ou GNSS), vitesse, statut de l'avion	Si compatibilité ADS-C : Position, altitude, numéro de vol, paramètres d'urgence, waypoints passés ou estimés, données additionnelles	Mode A/C + BDS 1,0 ; BDS 1,7 ; BDS 2,0 (call sign) ; BDS 4,0 (information saisies sur l'auto-pilote) ; BDS 5,0 ; BDS 6,0 (cap et vitesse)
Taux de rafraichissement	5-10s	5-10s	1-4s	14mn en moyenne	1-10s

Les informations proviennent du résumé du fonctionnement ATM fait par Thalès [24] et d'une présentation sur le Mode S [93].

1. *Comm-B Data Selector* (BDS), registre d'informations.