THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Institut National des Sciences Appliquées de Toulouse (INSA Toulouse)

Discipline ou spécialité :

Systèmes Informatiques Critiques / Systèmes Embarqués

Présentée et soutenue par :
Mohamed Slim BEN MAHMOUD
le : vendredi 17 février 2012

Titre :

Addressing Security Challenges in Emerging Data-based Aeronautical Communications

---

JURY

Ernesto Daminani, Full Professor - Université de Milan - Italie
Haitham Cruickshank, Senior Lecturer - Université de Surrey - UK
Pascal Urien, Professeur des universités - Télécom ParisTech, France
Thierry Gayraud, Professeur des universités - Université Paul Sabatier de Toulouse, France
Alain Pirovano, Enseignant-chercheur - ENAC, France
Nicolas Larrieu, Enseignant-chercheur - ENAC, France

Ecole doctorale :
Systèmes (EDSYS)

Unité de recherche :
Laboratoire d'optimisation des architectures des réseaux de télécommunications LEOPART

Directeur(s) de Thèse :
Alain Pirovano, enseignant-chercheur à l'ENAC
Nicolas Larrieu, enseignant-chercheur à l'ENAC

Rapporteurs :
Ernesto Daminani, Full Professor - Université de Milan - Italie
Haitham Cruickshank, Senior Lecturer - Université de Surrey - UK

Résumé

Ce travail de thèse s'intéresse à la sécurité des futures communications aéronautiques de donnée. Le travail est divisé en trois grandes parties. La première contribution est une architecture de sécurité adaptative pour les communications aéronautiques intégrant un segment sol-bord par satellite. Un module de gestion de la sécurité a été conçu, développé, puis validé lors de la phase finale d'intégration du projet FAST (Fibre-like Aircraft Satellite Communications). La deuxième contribution est une méthodologie quantitative d'estimation du risque lié à la sécurité réseau. L'originalité de notre approche est d'être basée sur la notion de propagation du risque au sein des différents noeuds du réseau. Comme cas d'étude, un réseau de communication aéroportuaire utilisant le protocole AeroMACS a été étudié dans le cadre du projet SESAR (Single European Sky ATM Research). La troisième contribution est une infrastructure à clés publiques (PKI) qui permet d'optimiser les échanges de signalisation (échanges de clés, certificats, vérification des signatures) entre l'avion et l'autorité de certification au sol. Le modèle de PKI proposé est un modèle hiérarchique utilisant la certification croisée entre les autorités de certification mères.

Mots clés: Sécurité, Réseau, Communications Aéronautiques, Risque, Satellite, PKI

Abstract

This research work deals with the information and network security in the aeronautical communication domain. Three fundamental research axes are explored. First, a quantitative network security risk assessment methodology is proposed. Our approach is based on the risk propagation within the network nodes. As study cases, the algorithm has been validated in the scope of the European industrial project entitled SESAR (Single European Sky ATM Research) and the Aerospace Valley FAST (Fibre-like Aircraft Satellite Communications). Particularly, experimental results relative to the case study devoted to the FAST project shown that the global network risk in the non secured system architecture is relatively high, meaning the system needs more consideration from a security point of view. To cope with this issue, an adaptive security management framework for a satellite-based aeronautical communication architecture has been proposed as a second contribution. A security manager module has been designed, implemented, then tested in the scope of the FAST project. Finally, as the security primitives used in the adaptive security management framework need to be efficiently exchanged, the last contribution consists in a scalable PKI adapted for the upcoming network-enabled aircraft. The idea is to minimize the air-ground additional overhead induced by the security procedures (keys, digital certificates, revocation/verification procedures). The PKI model we propose is a cross-certified multi-rooted hierarchical model.

Keywords: Security, Network, Datalink Communications, Risk, Satellite, PKI

Acknowledgments

After almost three entire years of research, a few hundred pages of thesis, paper, report, and deliverable writing (by the way, I ask forgiveness to mother nature after removing half of the amazonian forest in paper printing), oral presentations at conferences, seminars, and work package meetings, I can finally add the most enjoyable part of the dissertation where I can thank the many people who gave me their assistance to complete this thesis.

First and foremost, I would like to express my gratitude to my thesis supervisors, Doctors Nicolas Larrieu and Alain Pirovano for the opportunity they gave me to carry out my research in the LEOPART research laboratory. Their suggestions, creativity, keen intuitive insights, and professionalism have been a great value for me to complete this thesis. From a non-professional point of view, I own them a lot for the warm greetings and hospitality they showed me when I first came in France, and for their support to resolve all the administrative issues I faced during my journey.

I would like also to express my gratitude to Professors Ernesto Damiani and Haitham S. Cruickshank for reviewing this manuscript and for making the trip to attend the thesis defense. My thanks go also to the other members of the committee: Professors Pascal Urien, Thierry Gayraud, and Doctor Pascal Andrei.

During my journey in France, I had the chance to collaborate with many colleagues of mine from the LEOPART research team and industrial projects. I had the opportunity to work in a very friendly and inspiring atmosphere and I learned a lot from all of them. Fabien, Frédéric, Antoine, Ons, Mathieu: I will never forget the quantity of coffee we drunk together during these years,

My thanks are also addressed to all the students I had the opportunity to supervise at the ENAC (Amira, Hamza, Michel, Hamdi, Antoine, Nicolas, Mathieu, Alice). You all contributed in one or the other way, through technical discussions, sharing experiences or simply by your company and your friendship.

Finally, I am deeply indebted to my friends (Ines, Aymen, Meriem, Salma, Asma, Pauline, Sébastien, Walid, Yessine) and members of my family for their support specially my parents for trusting me and my abilities: in spite of the long distance that separated us, I felt their faith in my heart every single day. This thesis is also dedicated to my sisters Azza and Sirine, and my little nephew Fares: I hope this work will motivate you forthcoming years to succeed in your studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Motivation and Problem Statement

Aeronautical communications are about to shift the paradigm of digital data in near future. Thanks to IT (Information Technology) progress made in last decades, aviation industry stakeholders are enhancing and expanding their networks to overcome problems related to voice radio communications, but also to modernize the ATM (Air Traffic Management) environment. In 1978, a datalink system have already been introduced, namely ACARS (Aircraft Communication Addressing and Reporting System) [ARINC 2006], to essentially sustain radio voice systems, which were already running at their maximum capacity.

In civil aviation, a datalink is a two-way communication between an aircraft and a ground station, such as an air traffic tower control or an airline company, used to exchange digital information. In 1983, ICAO (International Civil Aviation Organization)[1] established the FANS (Future Air Navigation System) special committee, charged with developing operational concepts for the future ATM, including datalink communications [Smith *et al.* 2001]. In 2001, EUROCONTROL (European Organization for the Safety of Air Navigation)[2] launched the "LINK 2000+" program which is a set of implementation rules for datalink systems and technologies in the European airspace [EUROCONTROL 2010a]. Despite these early concerns for datalink communications, aeronautical voice radio communication is still a hot topic today, mainly because of an increasing growth in the worldwide air traffic.

[1] http://www.icao.int/

[2] http://www.eurocontrol.int/

1.1.1 Growth in Air Traffic

When a voice radio communication technology is used, all pilots in the same sector and communicating with an air traffic controller are tuned to the same frequency. This can be challenging considering the expected air traffic growth. Statistical data on air traffic increase are very important in identifiying trends within the air transportation industry. In 2009, EUROCONTROL reported in its annual NOP (Network Operations Report) an increase of 14.77% of the average daily traffic between 2004 and 2008, just for the European airspace [EUROCONTROL 2010c]. The same organization reported also in a 2010 long term forecast study an expected average air traffic growth of 1.6% to 3.9% per year between 2009 and 2030 [EUROCONTROL 2010b]. There will be 16.9 million IFR (Instrumental Flight Rules) movements in 2030, 1.8 times more than in 2009. Figure 1.1 depicts the forecast average annual growth in the European Airspace on behalf of EUROCONTROL long-term forecast:

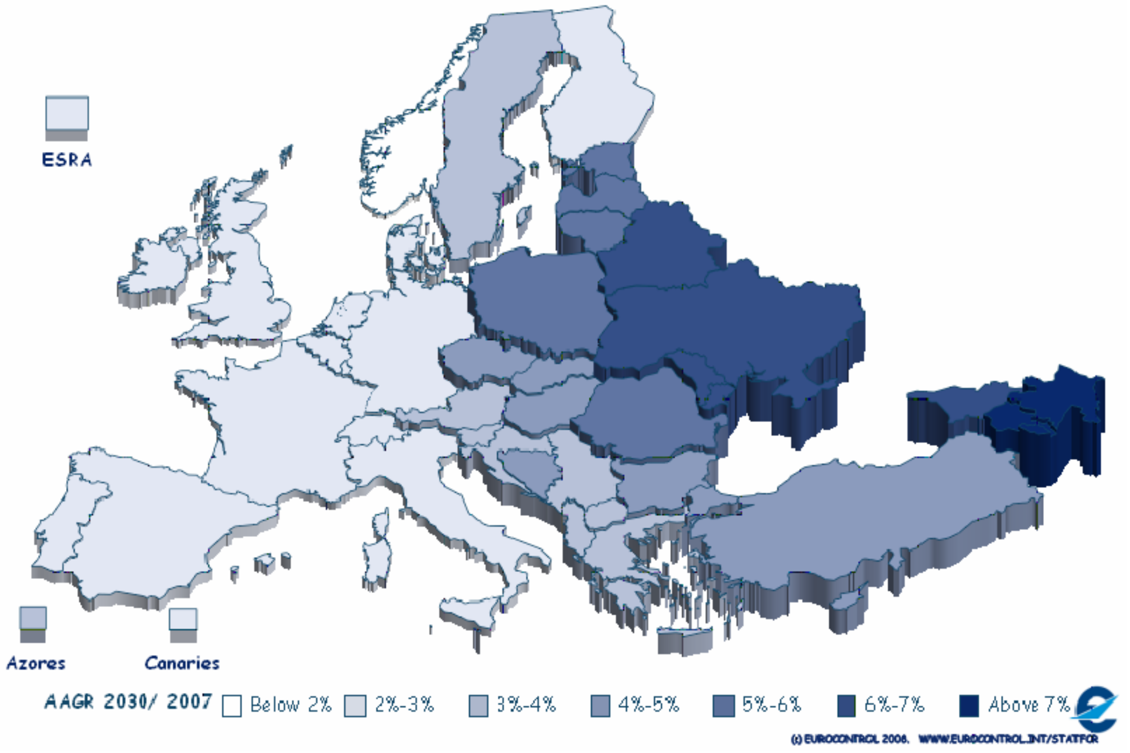


Figure 1.1: Forecast Average Annual Air Traffic Growth [EUROCONTROL 2010b]

This continuous positive growth is due to many factors such as an increased aircraft manufacturers market, more competitive low-cost airlines, an increased passenger demand and the greater need for companies to provide a better service to their customers. If the tendency continues to increase, it would probably result in a worldwide frequency saturation.

1.1.2 Radio Voice Frequency Congestion

The civil aviation airband spectrum, which uses radio frequencies between 108 and 137 MHz (Mega Hertz), is already congested. A frequency saturation may likely delay the communication between the pilot and the controller and make them unreachable for a certain period of time. ICAO has also shown its concern and expressed the need for additional spectrum for future aeronautical air-ground communications systems [ICAO 2006a].

As consequences could be catastrophic in such circumstances, many solutions have been provided in the past in order to address this frequency congestion issue such as optimizing frequency re-use, using more spectrum (since 1979, it has been increased from 136 to 137 MHz), or even splitting the radio spectrum into narrower bandwidths (50 kHz to 25 kHz channels). In 1994, it has been decided to reduce the channel spacing from 25 to 8.33 KHz (Kilo Hertz) under the ongoing the EUROCONTROL "8.33 KHz program" initiative [ERAA 2006].

International aeronautical organizations also proposed to divide saturated air traffic sectors into smaller sectors, but these solutions seem to imply many shortcomings. First, the number of available voice communication channels is still limited: there may not be enough frequencies when reaching a saturation point in a high density airspace in few years. Also, sector division requires the transfer of a given flight between two successive sectors. When the aircraft has to cross through a sector limit, the air traffic controllers assigned to each sector have to coordinate, then the pilot has to switch to the second channel frequency, test it, and turn-off the previous channel frequency. In many cases, it would no more be feasible to divide down a sector.

Hence, it is clear that air traffic sector division is not an efficient long-term solution for radio voice frequency saturation. Currently, the voice still remains the primary mean for air-ground communications, using HF(High Frequency) or VHF(Very High Frequency) bands technologies, but the industry is expected to progressively leave analog voice at the expense of digital data communications.

1.1.3 Datalink

Datalink systems seem to be an efficient conflict resolution strategy to:

- Increase the effective capacity of communication channels;
- Reduce communication delay;
- Increase voice channel availability;
- Enhance working environment for both pilots and air traffic controllers;
- Avoid misunderstanding and misinterpretation due to poor voice quality;
- Avoid signal corruption due to frequency saturation.

Many aircraft are already supporting data-based communications such as the CPDLC (Controller to Pilot Data Link Communication) [Evan Darby 1998] system, which is an ATN (Aeronautical Telecommunication Network) [ICAO 2002] datalink application that allows text-based message exchange between airline/air traffic ground facilities and the aircraft. Promising statistics showed that analog voice usage for operational services decreased linearly in aircraft supporting data-based ATS (Air Traffic Service) and AOC (Airline Operational Communication) applications: for 100% CPDLC aircraft equipped, the radio voice usage decreased by 84% [Shingledecker *et al.* 2005]. This kind of result points out the general trend for aeronautical communications: less voice and more data in the future.

1.1.4 Future Data-based Aeronautical Communications

Future air communication services and their supporting systems will be promptly based on datalink technologies to provide air-ground safety communications. Beside these operational services, airline companies are willing to enhance their offer to attract more passengers by providing new IFC (In Flight Connectivity) services. In fact, classic IFE (In Flight Entertainment) services, such as watching movies or listening to music stored on-board the aircraft are no longer satisfactory for passengers: modern customers recently showed their interest in a continuous Broadband Internet access on-board the aircraft to share multimedia contents, send or receive e-mails, or connect to their favorite social network. Also, additional airline applications may be supported in a long term perspective such as remote medical supervision or video surveillance systems to cope with a sick passenger or to monitor accidental/malicious events on-board.

With the introduction of new IFC/IFE passenger services and the forecast air traffic growth for future ATM environment, the aviation industry has so far shown a great interest in the use of satellite systems as supplementary means of air-ground communications due to their high capabilities. Firstly, high bandwidth SATCOM (Satellite Communication) technologies allow airlines to improve their IFE solutions and provide new IFC services such as Internet on-board. Secondly, SATCOM may play a great role for safety-related services as a complement to existing datalink systems such as VHF or VDL (Very High Frequency DataLink) mode 2 [Murawski *et al.* 2004]. Current aeronautical communications are based on several segregated systems where each system is working independently over a dedicated link (HF, VHF, VDL mode 2, etc). Ideally, communications between the pilot and the air traffic controllers should always be available upon demand and the air-ground channel availability should exceed 0.9995 in most cases, as mentioned in [EUROCONTROL 2002].

Therefore, using an additional SATCOM access network should provide an attractive and cost-effective solution for both operational (*i.e.* ATS and AOC) and APC (Airline Passenger Communications) datalink services, especially that high capacity *Ka*-band SATCOM allows to aggregate all these aeronautical services and provides satisfying global performances by improving ATM RCPs (Required Communication Performance) [ICAO 2005]. The previous PhD thesis defended in our laboratory illustrated the feasibility of such a system [Radzik *et al.* 2007], the main idea was to focus on RCPs and QoS (Quality of Service) constraints to design an on-board satellite terminal that allows the aggregation of operational and passenger services on the same SATCOM link.

Based on these new datalink system specifications, new ATN applications shall be designed using existing and mature COTS (Commercial Off The Shelf) products such as IP (Internet Protocol). There could be many reasons to use COTS technologies for future aeronautical communications such as:

- Reduction of development, implementation, and maintenance costs;
- Maturity of some product specifications such as TCP (Transmission Communication Protocol) or IP;
- Inter-operable software implementations;
- Seamless interoperability with existing terrestrial networks.

Thereby, some COTS technologies may fit for aeronautical requirements, particularly for a seamless inter-operation of datalink systems with existing and future ground networks. Although, ICAO has defined an ATN/IPS protocol stack to support IPS (Internet Protocol Suite) in next generation ATN applications [ICAO 2008]. Nevertheless, in order to successfully deploy all these new networks and applications, some emerging security problems should be carefully considered.

1.1.5 Emerging Security Challenges

The impact of these meaningful changes is a substantial increase of security risk in a critical context where human lives are involved and air-ground connectivity breakdowns cannot be tolerated. In fact, security and safety are strongly tied in the aeronautical area. These two terms are used differently depending on the considered context, therefore it is essential to primarily clarify their meanings in this thesis:

◇ **Safety** here is assimilated to the set of procedures, methodologies, and techniques deployed to mitigate risks related to the regularity and operation of the flight. For instance, we talk about safety-related or operational services for communications between the pilot and the air traffic controller;

◇ **Security** corresponds to the approaches and methods used to mitigate risks resulting from a malicious intent like an unauthorized intrusion on avionic systems. As an external attack may induce some consequences on the regularity of the flight, security risks definitely imply safety risks in the aeronautical context. Besides, in its general meaning, security includes both physical security and information systems/network security. This thesis focuses on information systems and network security as shown below.

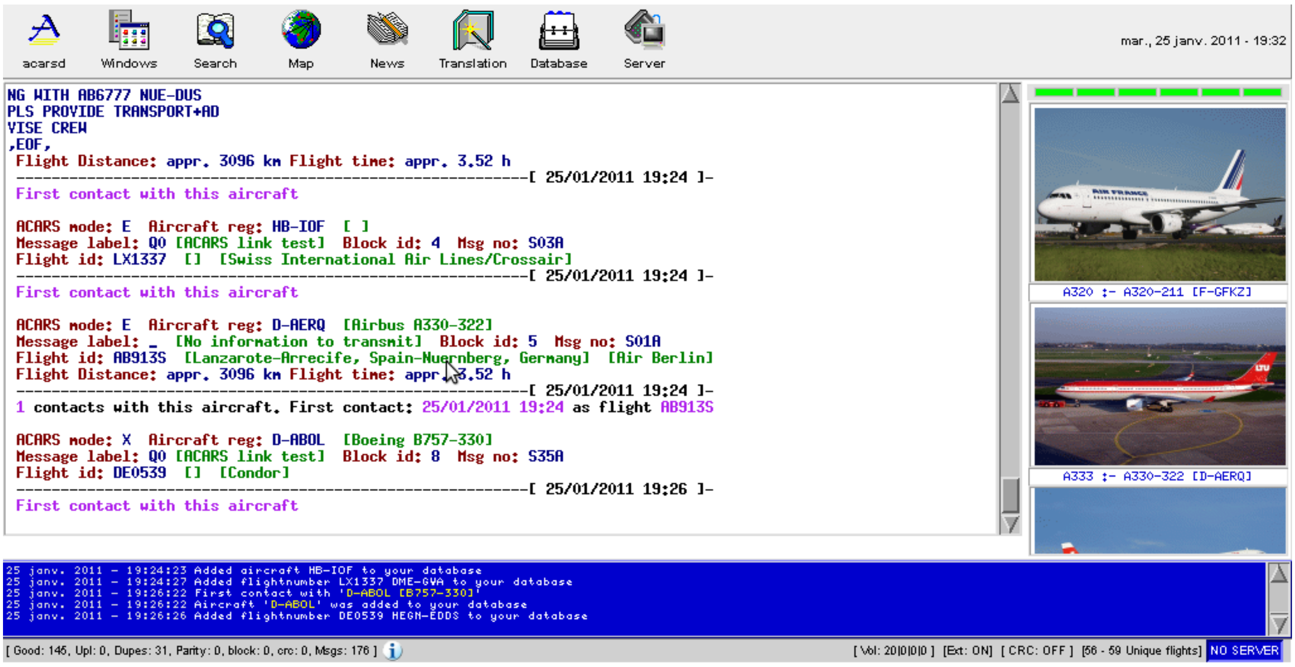
The civil aviation industry has always shown its interest in safety issues. With the emerging new datalink technologies and security consequences they may bring, security should be considered as seriously as safety in upcoming years. The openness and heterogeneity of datalink access technologies expose the ATN to malicious attacks, which emphasizes the security concern. To mitigate these risks and provide suitable solutions, some considerations should be taken into account when security requirements are being identified. For example, scaling issues in ATM environments, mainly induced by a growing air traffic load and passenger flight demand, underline the need for an optimized key management. The IATA (International Air Transport Association) reported in the 53^{th} edition of WATS (World Air Transport Statistics) a passenger growth of +22.1% between 1999 and 2008 [IATA 2009]. As the number of aircraft/passengers/avionic systems using cryptography grows, it is clear that the amount of keys and digital certificates increases.

Usually, PKIs (Public Key Infrastructure) [Douligeris & Serpanos 2007] are used to cope with scaling issues in wired and wireless networks and distribute the cryptographic credentials to end entities. However, such a solution should take into considerations the characteristics of the future ATN (large network, limited bandwidth, heterogeneity of services, etc), and consequently must induce a minimal overhead in the network. The opportunity to take advantage of safety-related service aggregation with airline and passenger services will necessitate appropriate security countermeasures to protect the operational services from new threats. Service scheduling techniques shall be used to give priority to operational ATS and AOC services.

Despite the increasing interest of many researchers and aeronautical actors in developing security solutions for current datalink technologies, these problems remain unsolved. For instance, a secure framework for ACARS called AMS (ACARS Message Security) [Roy 2001] has been developed by Honeywell[3] and standardized by ARINC to protect datalink messages exchanged between aircraft and ground systems. Indeed, ACARS messages were previously exposed to eavesdropping attacks: on www.acarsd.org website, it is possible to download a free decoder (available for WINDOWS and LINUX operating systems) and then listen to ACARS-based communications in real-time using a simple sound-card.

Figure 1.2 shows a screenshot of the acarsd GUI (Graphical User Interface). We can see that many information could be easily acquired, such as daily airline statistics, flight level, flight destination, message content, ICAO aircraft code, etc. Even if the AMS protocol has been well-designed (key management, life-cycle management of cryptographic keys, etc), the system has never been implemented for economic reasons. Also, the system is already out-of-date as long as ACARS will be superseded by ATN/IPS applications for ATS and AOC services over the next 20 years [SITA 2004].

[3] http://honeywell.com/Pages/Home.aspx

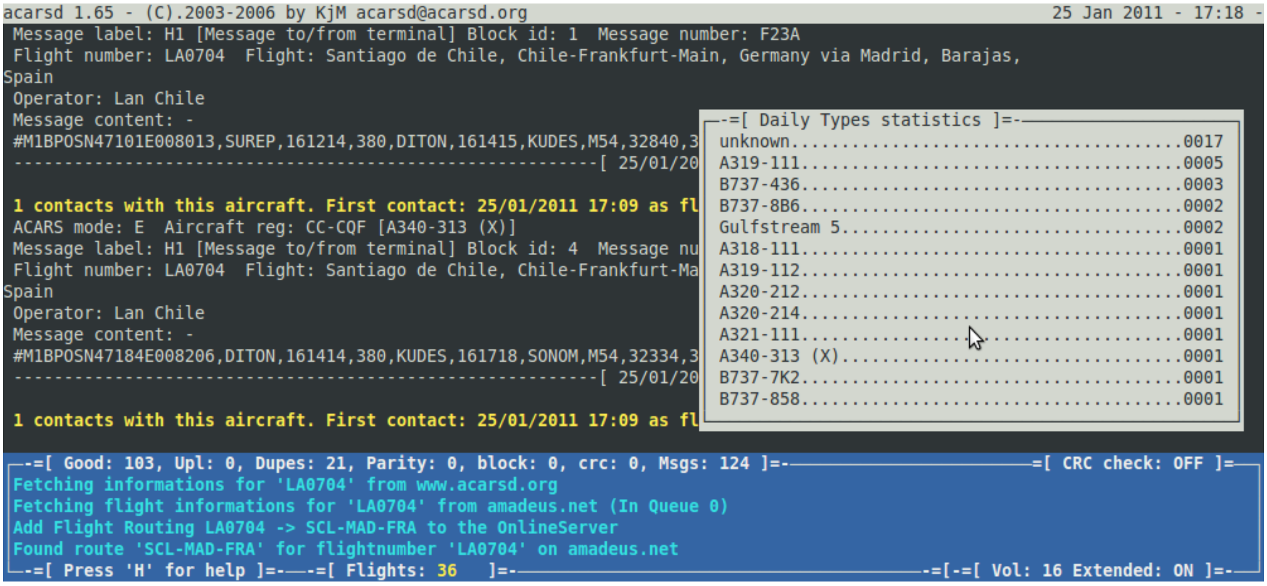

acarsd
Windows
Search
Map
News
Translation
Database
Server
mar., 25 janv. 2011 - 19:32
A320 :- A320-211 [F-GFKZ]
A333 :- A330-322 [D-AERQ]

```
NG WITH AB6777 NUE-DUS
PLS PROVIDE TRANSPORT+AD
VISE CREW
,EOF,
 Flight Distance: appr. 3096 km Flight time: appr. 3.52 h
 ------------------------------------------------------------[ 25/01/2011 19:24 ]-
 First contact with this aircraft

 ACARS mode: E  Aircraft reg: HB-IOF  [ ]
 Message label: Q0 [ACARS link test]  Block id: 4  Msg no: S03A
 Flight id: LX1337  []  [Swiss International Air Lines/Crossair]
 ------------------------------------------------------------[ 25/01/2011 19:24 ]-
 First contact with this aircraft

 ACARS mode: E  Aircraft reg: D-AERQ  [Airbus A330-322]
 Message label: _  [No information to transmit]  Block id: 5  Msg no: S01A
 Flight id: AB913S  [Lanzarote-Arrecife, Spain-Nuernberg, Germany]  [Air Berlin]
 Flight Distance: appr. 3096 km Flight time: appr. 3.52 h
 ------------------------------------------------------------[ 25/01/2011 19:24 ]-
 1 contacts with this aircraft. First contact: 25/01/2011 19:24 as flight AB913S

 ACARS mode: X  Aircraft reg: D-ABOL  [Boeing B757-330]
 Message label: Q0 [ACARS link test]  Block id: 8  Msg no: S35A
 Flight id: DE0539  []  [Condor]
 ------------------------------------------------------------[ 25/01/2011 19:26 ]-
 First contact with this aircraft
```

```
25 janv. 2011 - 19:24:23 Added aircraft HB-IOF to your database
25 janv. 2011 - 19:24:27 Added flightnumber LX1337 DME-GVA to your database
25 janv. 2011 - 19:26:22 First contact with 'D-ABOL [B757-330]'
25 janv. 2011 - 19:26:22 Aircraft 'D-ABOL' was added to your database
25 janv. 2011 - 19:26:26 Added flightnumber DE0539 HEGN-EDDS to your database
```

```
[ Good: 145, Upl: 0, Dupes: 31, Parity: 0, block: 0, crc: 0, Msgs: 176 ]      [ Vol: 20|0|0|0 ]  [Ext: ON]  [ CRC: OFF ]  [56 - 59 Unique flights]  NO SERVER
```

(a) Acarsd Main GUI



```
acarsd 1.65 - (C).2003-2006 by KjM acarsd@acarsd.org                              25 Jan 2011 - 17:18 -
 Message label: H1 [Message to/from terminal] Block id: 1  Message number: F23A
 Flight number: LA0704  Flight: Santiago de Chile, Chile-Frankfurt-Main, Germany via Madrid, Barajas,
Spain
 Operator: Lan Chile
 Message content: -                                             ┌-=[ Daily Types statistics ]=-
 #M1BPOSN47101E008013,SUREP,161214,380,DITON,161415,KUDES,M54,32840,3 │ unknown..................................0017
 -------------------------------------------------------[ 25/01/20 │ A319-111.................................0005
                                                                   │ B737-436.................................0003
 1 contacts with this aircraft. First contact: 25/01/2011 17:09 as fl │ B737-8B6.................................0002
 ACARS mode: E  Aircraft reg: CC-CQF [A340-313 (X)]                │ Gulfstream 5.............................0002
 Message label: H1 [Message to/from terminal] Block id: 4  Message nu │ A318-111.................................0001
 Flight number: LA0704  Flight: Santiago de Chile, Chile-Frankfurt-Ma │ A319-112.................................0001
Spain                                                              │ A320-212.................................0001
 Operator: Lan Chile                                               │ A320-214.................................0001
 Message content: -                                                │ A321-111.................................0001
 #M1BPOSN47184E008206,DITON,161414,380,KUDES,161718,SONOM,M54,32334,3 │ A340-313 (X).............................0001
 -------------------------------------------------------[ 25/01/20 │ B737-7K2.................................0001
                                                                   │ B737-858.................................0001
 1 contacts with this aircraft. First contact: 25/01/2011 17:09 as fl └
┌-=[ Good: 103, Upl: 0, Dupes: 21, Parity: 0, block: 0, crc: 0, Msgs: 124 ]=-──────────────=[ CRC check: OFF ]=┐
│Fetching informations for 'LA0704' from www.acarsd.org
│Fetching flight informations for 'LA0704' from amadeus.net (In Queue 0)
│Add Flight Routing LA0704 -> SCL-MAD-FRA to the OnlineServer
│Found route 'SCL-MAD-FRA' for flightnumber 'LA0704' on amadeus.net
└-=[ Press 'H' for help ]=-──-=[ Flights: 36   ]=-──────────────────────-=[-=[ Vol: 16 Extended: ON ]=-┘
```

(b) Acarsd Aircraft Statistics

Figure 1.2: Acarsd Outputs

This dissertation presents a set of solutions that allow to manage security problems for emerging aeronautical data-based communications. When combined together, these solutions form a global and homogeneous security framework for ATN communications. Also, it is interesting to notice that this thesis is the logical sequel of the work initiated previously in our laboratory regarding the aggregation of aeronautical safety-related services with non operational services on the same SATCOM link.

The current chapter presents an introduction to highlight the concern for security in future aeronautical communications. It shows how security is tightly related to these next generation networks. In the next section, research objectives, contributions, and the dissertation structure are discussed.

1.2 Research Context, Objectives, and Contributions

1.2.1 Research Context

The contributions made in this thesis are supported by two industrial projects that are presented in the following subsections.

1.2.1.1 Overview of the Aerospace Valley FAST Project

The proposed secure infrastructure has been tested and validated within an industrial project entitled FAST (Fiber-like Aircraft Satellite Telecommunications), co-funded by the Aerospace Valley[4] pole and the French government. Recently, concurrent air-ground systems have been proposed to provide Internet and multimedia accesses to passengers in aircraft cabin, with arguable success.

Among the technology candidates, it seems that satellite systems are long-term solutions for IFE and IFC services. The objective of the FAST project is to study the feasibility and reliability of a bi-directional high-capacity airborne satellite antenna for ATS, AOC and APC services. Besides, new airline operational communication services (namely Next Generation AOC, and denoted AOC NG) are introduced in the global design of the system architecture. These are new safety-related services and might be supported by airline companies in a long-term perspective. A medical supervision system has been developed by the MEDES company[5]. The second service is the video surveillance system provided by VODEA[6]. Section 2.1.2.2 gives a brief description of these AOC NG services.

The project is co-funded by the DGCIS (Direction Générale de la Compétitivité, de l'Industrie et des Services), and the FUI (Fonds Unique Interministériel). It federates many research efforts from both industrial partners such as EADS Astrium[7] or Axess Europe[8], and academic/institutional partners: LAAS (Laboratoire d'Analyse et d'Architecture Systèmes)[9], ISAE (Institut Supérieur de l'Aéronautique et de l'Espace)[10], and Télécom Bretagne[11]. The project started in January 2009 and finished in June 2011. Other details about our involvement in the FAST project can be found in section 2.3.

1.2.1.2 Overview of the European SESAR Project

The European collaborative project entitled SESAR (Single European Sky ATM Research) aims to modernize the future European ATM [EUROCONTROL 2011]. The project is still in progress and probably considered as one of the most important European R&D (Research and Development) collaboration ever launched by the European Commission, EUROCONTROL, TEN-T EA (Trans-European transport Network Executive Agency)[12], and other actors from the industry such as Airbus[13] or Thales[14]. The aim of the project is to offer technical and operational solutions to meet future air traffic capacity and air safety needs. The total estimated cost of the development phase of SESAR is 2.1 billion €, to be shared equally between the academic community, EUROCONTROL, and the industry.

[4] http://www.aerospace-valley.com/
[5] http://www.medes.fr/
[6] http://www.vodea.com/
[7] http://www.astrium.eads.net/
[8] http://www.axesseurope.com/
[9] http://www.laas.fr/2-27719-Home.php
[10] http://www.isae.fr/en/index.html
[11] http://international.telecom-bretagne.eu/welcome/
[12] http://tentea.ec.europa.eu/
[13] http://www.airbus.com/
[14] http://www.thalesgroup.com/

Note that a similar project called NEXTGEN (Next Generation Air Transportation System) is undertaken by the FAA (Federal Aviation Administration)[15] to improve the American NAS (National Airspace System) and deal with the same air traffic congestion issues [FAA 2011]. We have been involved in the 15.2.7 SESAR working package (which aims to study airport surface communications) since September 2010, our task in SESAR will be described in section 2.3.

1.2.2 Scope of the Work

This research work deals with the information and network security in the aeronautical communication domain. In this section, we define main concepts and boundaries of our work, as specified in figure 1.3:

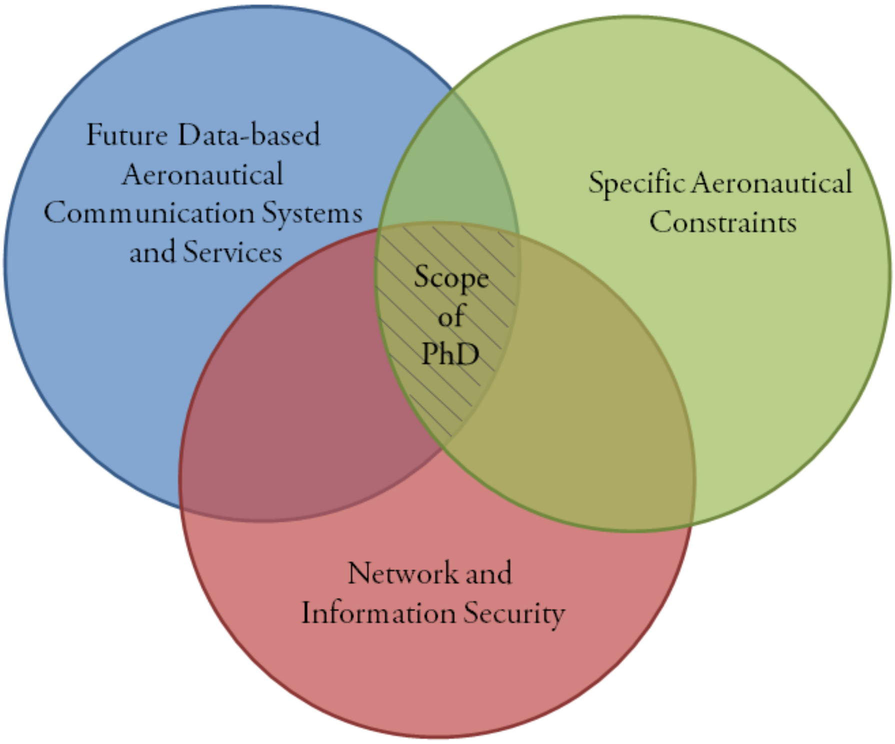

Future Data-based Aeronautical Communication Systems and Services
Specific Aeronautical Constraints
Scope of PhD
Network and Information Security

Figure 1.3: Scope of the Work

Three fundamental axes are explored in the definition of thesis' objectives:

(a) *Future data-based aeronautical communication systems and services*: a lot of work has been done in the frame of future data-based communication and systems, in both civil and military aviation. In this work, we focus mainly on services deployed in both FAST and SESAR projects, namely ATS, AOC, AOC NG, and APC data-based services respectively through SATCOM and AeroMACS (Aeronautical Mobile Airport Communication System) technologies (see chapter 3 for more details). A particular effort has been done at the beginning of our work to model operational datalink services based on international civil aviation organization specifications;

(b) *Specific aeronautical constraints*: RCTPs (Required Communication Technical Performances) [ICAO 2002] have been conducted by ICAO and FAA to define technical performances that every aeronautical communication system must achieve. These RCTPs can be expressed for instance

[15] http://www.faa.gov/

in terms of end-to-end delay, latency, service priority, or security properties. Moreover, network and system resources are generally limited in aeronautical communications: sporadic fading on the communication channel, indeterministic flight level changes, or severe weather conditions for instance could affect available network resources. In such situations, security should be cleverly deployed with low computational and data overhead;

(c) *Network and information security*: security is the keyword of all contributions made in this thesis. Several aspects are covered, such as network security risk assessment design, formal modeling for network security policies, adaptive security in restricted network environment, cryptographic protocol design, formal verification, security protocol classification and optimized key management for future aeronautical applications.

1.2.3 Contributions

This thesis is about bringing solutions to security issues and shortcomings in datalink communications discussed above. Three main contributions are proposed:

1. **Quantitative network security risk assessment model**: a new approach for network security assessment that measures quantitatively the network risk level is designed. The algorithm takes into account the inherent characteristics of any computer network in general (such as interconnection between nodes), in addition to specific requirements for an aeronautical network (such as the service priorities between the network domains). This contribution is motivated, first by shortcomings noticed on qualitative risk assessment methodologies. Secondly, existing quantitative risk assessment models for network security generally missed out with an essential concept in network security risk evaluation, namely the risk propagation. In our model, we fill this gap showing how important risk propagation can be in the network risk computation step;

2. **Adaptive security framework**: considering RCTPs and network resource restrictions in the aeronautical context, adding a security layer to existing communication systems must come with the smallest overhead. Static security policies which are usually costly and heavy, lead to high overhead on the channel communication. QoS and security systems have usually been considered separately with different objectives and implementation architectures. However, recent researches showed that these concepts are highly inter-dependents [Chen *et al.* 2009]: on one hand, security may severely affect the performances of the network (additional data overhead, increased delay, etc). On the other hand, security policies could be improved and wisely chosen, thanks to information obtained about the network state and available resources when security is about to be used. As air-ground channel communication is bandwidth-limited, we therefore propose an adaptive security management framework to mitigate these issues and find an optimal trade-off between network security and performances;

3. **Scalable keys/certificates management model**: because of an expected congestion in the worldwide airspace, some scalability issues in future ATM environment may arise. We therefore illustrate the feasibility of a scalable PKI adapted for the upcoming network-enabled aircraft with particular emphasis on the certificate revocation and verification procedures: many techniques are discussed and their benefits in term of resulting overheads are underlined through a performance assessment study. Also, recommendations and best practices defined by international aeronautical committees such as RTCA (Radio Technical Commission for Aeronautics)[16] are adhered to.

Note that in some cases throughout the thesis work, we have been led to use tools or exploit results brought by different partners in the scope of the FAST and SESAR work packages. In order to avoid

[16] http://www.rtca.org/

misunderstanding and confusion with the thesis contributions described above, results provided by other FAST/SESAR partners are denoted throughout the thesis using the symbol †.

1.3 Thesis Structure

This thesis is structured as follows:

1. **Chapter 2** illustrates rising security challenges to be faced in future aeronautical networks. Characteristics of aeronautical applications and likely future trends in the worldwide ATM are specifically explained, then their impact on security is shown. Also, a traffic profile model for operational aeronautical services is presented. This model is based on communications and operating concepts and requirements defined by EUROCONTROL and the FAA. Then, a brief introduction to main computer and network security concepts is made in order to clarify generic security terms used in the rest of the dissertation. A review of existing security mechanisms for aeronautical datalink communications is provided with an overview of main security activities underway in the aviation industry (working packages, special committees, technical panels, etc). Finally, we present our tasks in both FAST and SESAR projects;

2. **Chapter 3** introduces an original risk assessment approach based on risk propagation for network security. Our approach measures quantitatively the network risk level based on critical aspects such as the impact of a successful attack on a node and the risk propagation of that attack within the network. Our experiments have been conducted using real statistics and vulnerability databases. Each parameter involved in the risk assessment process is quantified then the overall approach is described in detail. As case studies, we consider first the satellite-based system architecture designed for the FAST project. Two network topologies are evaluated: the nominal architecture (*i.e.* without security features) and the secure architecture. The second case study is related to an aeronautical mobile airport communication system defined in SESAR 15.2.7 working package. In this part, we specifically focus on access network vulnerabilities, and a network risk study is conducted for a predefined scenario. Finally, a comparison between AeroMACS authentication/authorization security mechanisms is achieved and some security guidances are given either to enhance security policies or to improve the end to end security using some additional mechanisms such as certificate-based authentication;

3. **Chapter 4** proposes an adaptive security framework for future aircraft communications. A secure system topology for the embedded airborne network is proposed with regards to network and system constraints, services priorities and regulatory recommendations. Firstly, a two-level QoS policy is defined for the system to manage priorities and resource allocation. Then, the design of the new framework, called SecMan (Security Manager), is explained in details and all its processes are formalized. In this chapter, several aspects are covered in the scope of the SecMan framework:

   ◇ a secure negotiation protocol of supported security mechanisms is provided and validated using formal verification techniques;

   ◇ a security protocol classification approach is discussed using a multi-criteria decision algorithm;

   ◇ a hierarchical-oriented formal modeling of network security policies deployed by SecMan is proposed;

   ◇ network and system information processing are described with a particular focus on the selection of an optimal security policy by the algorithm;

- ◇ finally, performance evaluation results are presented in order to assess benefits brought by the adaptive security framework within some critical aircraft communication scenarios;

4. **Chapter 5** presents a performance-aware PKI for next generation connected aircraft. This model uses a combination of hierarchical CAs (Certificate Authorities) in order to minimize the air-ground exchanges caused by any PKI-related operational process (checking and revoking certificates, for instance). The proposed PKI model works across three levels. The first level is relevant to root-CAs interactions. The second level is related to the communications between airline-CAs and subordinate CAs. The last level deals with the on-board users and the subordinate CAs. Different phases of the certification process and key management are also described. OCSP (Online Certificate Status Protocol) [Myers *et al.* 1999] and CRLs (Certificate Revocation Lists) [Cooper *et al.* 2008] servers are discussed to emphasize their benefits in terms of resulting network and computation overheads. In the performance analysis, we used daily air traffic statistics issued from the french ANSP (Aeronautical Network Service Provider) DSNA-DTI (Direction des Services de la Naviguation Aérienne-Direction de la Technique et de l'Innovation)[17] databases. Three experimental scenarios have been defined, depending on the geographic location of both the owner and the verifier of the certificate (*i.e.* on-board or on the ground) and has been used to compare our performance-aware model to the reference PKI model according to the different PKI procedures;

5. **Chapter 6** concludes the manuscript by a summary of contributions made in this thesis. A discussion about limitations highlighted by achieved results is provided. Finally an outlook towards further directions of research is given.

[17] http://www.dsna-dti.aviation-civile.gouv.fr/

Chapter 2

Security Challenges in Future Aeronautical Networks

Contents

This chapter is a general introduction to datalink aeronautical communications and the security challenges to be addressed in future: both operational and non operational data services are depicted and an aggregated traffic profile is generated through a simulation model. The traffic generator presented in this chapter has been used at the simulation campaign (*c.f.* chapter 4) to generate the operational traffic sources. A survey of existing datalink security solutions is provided with a relevant taxonomy, then a set of security issues is identified. Finally, a general picture of mapping between the identified challenges and the thesis contributions is made at the end of the chapter to give the reader a clearer view on the contents of next chapters.

Note: Materials and results presented in this chapter have appeared in [Ben Mahmoud *et al.* 2009a], [Ben Mahmoud *et al.* 2009b], [Ben Mahmoud *et al.* 2009d] and [Ben Mahmoud *et al.* 2009e].

2.1 Datalink Aeronautical Communications

2.1.1 Operational Services

This subsection gives an outlook of both ATS and AOC services. These categories are related to the operational command and control of the aircraft. Both ATS and AOC services are related to the safety and regularity of the flight as defined in Annex 10 of the ICAO Convention [ICAO 2006b]. This document covers the three essential elements of international civil aviation, namely CNS (Communications, Navigation, and Surveillance) concepts through five volumes. Volume III, dedicated to both digital data and voice communication systems, is beyond doubt the most relevant to our work.

2.1.1.1 Air Traffic Services

ATS services support ATC (Air Traffic Control) messages between the pilot and the traffic controller. These services are provided by ATC ground stations performing specific ATS functions such as flight information, flight instructions, or departure/landing clearances. For instance, the DCL (Departure Clearance) service gives the permission to the pilot to take-off. The DYNAV (Dynamic Route Availability) is another ATS service used by an ATC ground station to indicate alternative routes to the crew during the flight. CPDLC is an ATS datalink application that allows for the direct exchange of text-based messages between a controller and a pilot. Figure 2.1 illustrates the pilot interface, known as the DCDU (Datalink Control and Display Unit)[1] for sending and receiving CPDLC messages on an Airbus A330.

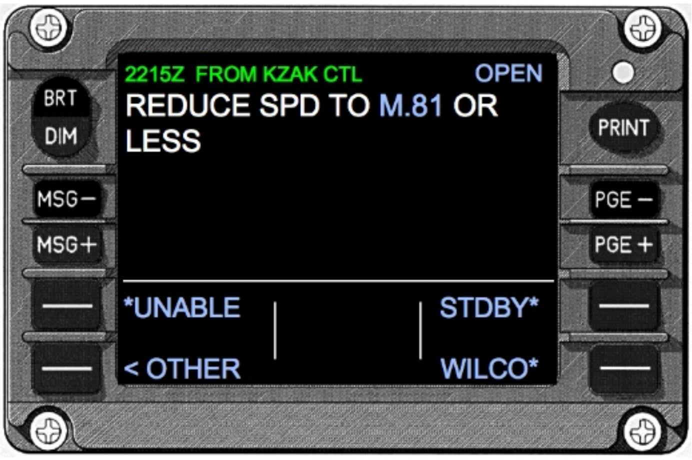

2215Z FROM KZAK CTL OPEN
REDUCE SPD TO M.81 OR LESS

Figure 2.1: DCDU interface for CPDLC Messages

2.1.1.2 Aeronautical Operational Control Services

Besides cockpit ATS services, AOC services are required for efficient CNS and ATM operations. This service category supports operational voice and/or data messages between the aircraft and the airline or airport operational staff. Usually, the business success of a given airline is related to the efficiency of the proprietary airline system, known as AFDC (Airline Flight Dispatch), responsible for the control and the coordination between all airline functions such as flight operations, airport area operations, maintenance or system control. Notice also that some AOC messages might not be safety-related, and would more appropriately be called AAC (Aeronautical Administrative Communications) messages. However, they are often grouped under AOC category in general civil aviation terminology since the same systems and medias are usually used to send both AOC and AAC messages. As an example, the ENGINE (Engine Performance Reports) message is sent from the aircraft to the airline ground entity to provide a technical report about operational performances of engines and embedded avionic systems during the flight.

[1] the term DCDU is proper to Airbus terminology and could be referred differently by another aircraft manufacturers

2.1.2 Non Operational Services

Non operational services are provided by airline companies to satisfy the growing needs of passengers (APC services) and to enhance the safety of the flight (AOC NG services).

2.1.2.1 Passenger Services

Recently, airline companies showed a great concern to develop new APC services for passengers. Indeed, classical IFE services such as "static" multimedia contents (*i.e.* non passenger-friendly contents such as pre-recorded movies, radio, music, etc) are no longer satisfactory. Nowadays, a constant Internet connexion is required no matter where people are (social networking, e-mails, web browsing, video streaming, etc). In the past seven years, many projects tried to propose effective IFC solutions, using either SATCOM or ground coverage technologies. In 2004, the Boeing company[2] launched the very first in-flight system for Internet access and called CCB (Connexion By Boeing) [Jones & de La Chapelle 2001]. The CCB system used a Ku-band satellite link to provide respectively 5 and 1 Mbps throughputs for uplink and downlink (in SATCOM-based aeroanutical communication networks, the uplink means the ground to air direction whereas downlink refers to the air to ground direction), and has been installed on nearly 150 aircraft. However, in 2006, Boeing announced that CCB is definitely stopped, rather for economic than technical reasons[3].

Since the failed Boeing attempt, many other SATCOM solutions have been proposed. In 2008, ARINC provided the OI (On-board Internet)[4] system which used INMARSAT geostationary satellites along with the SBB (Swift Broad Band)[5] service. OI system is supposed to be fully compatible with most IFE existing systems, requiring no additional hardware installations. The system provides cost-effective broadband services including IP-based data up to 432 kbps. In the same year, Panasonic proposed its own Intelsat satellite-based solution and called eXConnect[6] to deliver a reliable passenger experience with high bandwidth performances. Another commercial aviation in-flight communications service called OnAir[7] has been provided jointly by the Airbus company and the telecommunications service provider SITA (Société Internationale de Télécommunications Aéronautiques)[8]. OnAir uses also INMARSAT's SBB fourth generation satellites services, including mobile telephony and IFC services. The in-flight broadband entertainment platform Row44[9] is an IFE/IFC service certified by the FAA, US FCC (Federal Communications Commission)[10], and the European civil aviation authority EASA (European Aviation Safety Agency)[11]. This IFC service proclaims fast broadband to the aircraft using a Ku-band satellite constellation, a high bandwidth capacity, and a flexible and scalable network.

Besides satellite-based systems, some projects relied on direct air-ground connections to overpass SATCOM technology high costs. In 2008, Aircell[12] launched the Gogo In-flight[13] service to provide Internet connectivity to the US fleet. Also, WiSKY[14] proposed a WiMAX-based (Worldwide Interoperability for Microwave Access) system to deliver simultaneously 100 Mbps at 100 miles distance to every equipped airplane in the sky. Ground-based IFC solutions have the advantage to afford higher

[2] http://www.boeing.com/
[3] http://www.boeing.com/news/releases/2006/q3/060817a_nr.html
[4] http://www.arinc.com/products/inflight_entertainment/oi-onboard_internet.html
[5] http://www.inmarsat.com/
[6] http://www.panasonic.aero/Products/GlobalComm.aspx
[7] http://www.onair.aero
[8] http://www.sita.aero/
[9] http://www.row44.com
[10] http://www.fcc.gov/
[11] http://www.easa.europa.eu/
[12] http://www.aircell.com
[13] http://www.gogoinflight.com/
[14] http://www.wi-skyinflight.com

throughput than SATCOM technologies. Still, such ground-based IFC solutions have many drawbacks, including limited service coverage and high deployment costs. Indeed, these solutions are not able to cover oceanic and polar areas. Also, the covered range per ground station is limited, then many base stations are needed to provide a full coverage for an airspace area. As more terrestrial stations are needed, deployment and maintenance costs will likely increase.

2.1.2.2 Next Generation Airline Services

As described in section 1.2.1.1, airlines might introduce new safety-related services in the near future in order to prevent emergency events occurring on-board (*e.g.* unauthorized access to cockpit, on-board terrorist attacks, medical assessment needed for passengers). Two services have been considered in the FAST project, namely the medical supervision and video surveillance systems.

Currently, when a passenger needs a medical diagnosis during the flight, the cabin crew try first to asset the passenger health condition. If the cabin staff has not been trained to face such IFMEs (In-flight Medical Events), a voluntary doctor among passengers may be called to do a reliable medical assessment. For critical cases where the situation is out-of-control (*e.g.* no doctor on-board, extra-medical equipments needed), the pilot may decide to re-route the airplane to the nearest airport. From a financial point of view, this is definitely the worst alternative for airlines, since diverting is a very expensive solution (between 150 and 250 k€)[Franck *et al.* 2010]. Also, several studies showed that up to four IFMEs occur every 10,000 passengers [Jagoda & Pietrzak 1997], which means that the probability of getting these kind of emergency cases still exists.

Therefore, civil aviation authorities such as ECAC (European Civil Aviation Commission) recommended the development of best practices and *"...air/ground/air communications to assist in establishing diagnosis and treatment, and in making decisions to divert the aircraft"* [ECAC 2006]. Then, the key solution is to provide a medical supervision system (also known as Telemedicine) for air-ground communications. Basically, it consists in an embedded medical terminal used when IFMEs occur to transmit live information to ground medical staff. Hence, it is possible to establish an accurate assessment of the passenger's state without any need for physical attendance of a doctor or extra-medical skills for crew members. Figure 2.2 is an example of a portable device used on-board to asset passenger health.

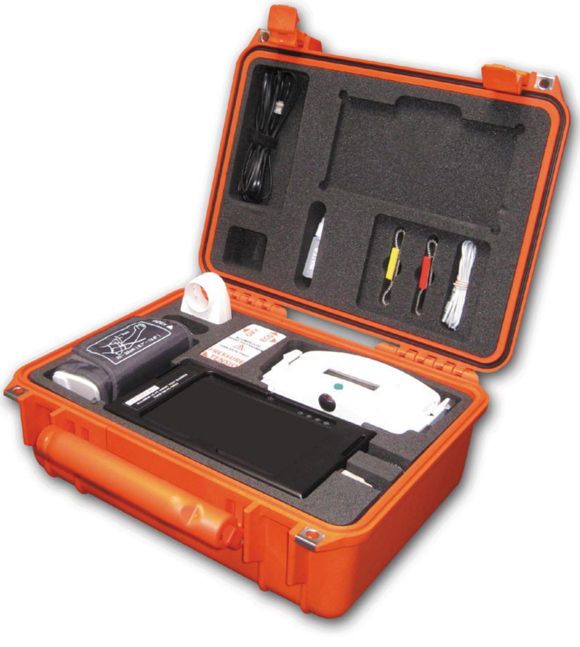


Figure 2.2: Telemedecine Embedded Station

At the same time, past terroristic events (*e.g.* the September 11 2001 airplane hijacking) pointed-out the need for criminal preventive systems for both cockpit and cabin. Also, airlines might need to track the behavior of crew staff (particularly the pilot) in order to clearly identify the origin of an incident or a crash if the black box is still not found for instance (*e.g.* as for the early research stage of the Paris-Rio 447 flight).

In order to mitigate such risks, the latest safety proposal being made in the field of civil aviation is the implementation of a video surveillance system within the aircraft at strategic locations [Thanthry *et al.* 2007]. With the introduction of additional high capacity technologies, it should be even possible to use HD (High Definition) cameras dispatched through the cabin and the cockpit. Videos are streamed to the ground where a monitoring operator try to prevent any behavioral anomaly on-board the aircraft. Figure 2.3 shows the video wrapper proposed in the FAST project. This wrapper is able to record HD videos with an H264 [ITU-T 2010] compression high profile and outputs MPEG/TS [ISO 2007] streams up to 5 Mbps.

Figure 2.3: Video Surveillance Wrapper

2.1.3 Communications Operating Concept and Requirements

In order to cope with the forecast high density in the worldwide airspace, EUROCONTROL and the FAA jointed their efforts to identify candidate data-based technologies able to satisfy safety and regulatory requirements for ATS and AOC communications. In 2002, a document entitled COCR (Communications Operating Concept and Requirements) [EUROCONTROL 2002] has been produced as a result of this collaboration. Basically, the document covers communication systems and operational requirements, referred by FRS (Future Radio System).

ATS and AOC operational voice services are briefly mentioned but the document clearly focuses on emerging data services, with a particular emphasis on message characteristics and specifications (*e.g.* message size, frequency of use per sector). The COCR document also covers safety and security requirements for operational data services using an OSA (Operational Safety Assessment) methodology. These data are used for both SESAR and FAST projects and summarized in appendix B. This preliminary study has been helpful to make a traffic profile of future data-based operational services (see section 2.1.4 for more details).

The COCR considers both current and future ATM specificities. Thereby, the study is divided into two phases:

1. **Phase 1** already started around 2005 in some regions in the world. In this phase, analog voice continues to be used for operational services while digital data serves as a backup for air-ground communications until 2030;

2. **Phase 2** is planned to start around 2020. In this phase, operational data services are expected to be the first means of ATM communications, voice will be used only for exceptional circumstances or for areas that do not have data communications implementation yet.

Figure 2.4 illustrates the COCR phase 1 and 2 expected time-frame [EUROCONTROL 2002]:

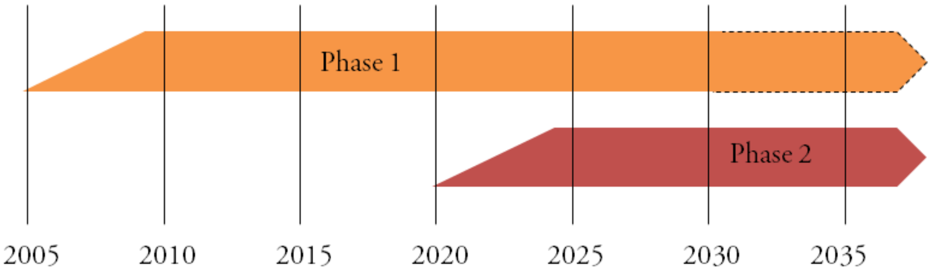


Figure 2.4: COCR Phase 1 & 2 Concept Evolution Over Time

Besides, the document considers the following five representative airspace domains in the specification of operational data services (*e.g.* frequency of use of a given service per domain, total number of ATC ground stations deployed for each domain, support of voice and/or data messages, etc):

1. **APT**: the APT (Airport) domain is the closest airspace to the airport surface. Its consists of a 10 miles cylinder area and up to nearly 5000 ft (feet);

2. **TMA**: the TMA (Terminal Maneuvering Area) domain is the surrounding airport airspace with a typical 50 NM[15] (Nautical Miles) radius from the airport center. The TMA domain starts at 5000 ft and ends at 24500 ft. The COCR assumes the same characteristics for departure and arrival TMA domains;

3. **ENR**: this is the airspace surrounding the TMA domain and starting at FL245 to FL600. ENR (En Route) is the continental/domestic airspace when the airplane is cruising;

4. **ORP**: this airspace domain is identical to the ENR domain, the only difference is that ORP (Oceanic, Remote, and Polar) is associated with oceanic and unmanned areas. The ORP domain has a horizontal limit extending 1000 NM by 2000 NM;

5. **AOA**: this airspace domain will be operational starting from 2020. The AOA (Autonomous Operations Area) domain concerns autonomous operations where aircraft are able to make self-separation without any ATC directives. The COCR assumes that the AOA airspace domain has horizontal limits of 400 NM by 800 NM.

Figure 2.5 illustrates the organization of these five domains in the airspace as defined in the COCR document:

[15] In aeronautics, the FL (Flight Level) is expressed in feet's. For example, FL800 corresponds to 80000 fts. One feet (ft) is equal to 0.3045 m (meters). The Nautical Mile is used in both aeronautics and maritime navigation. One nautical mile corresponds to 1852 m.

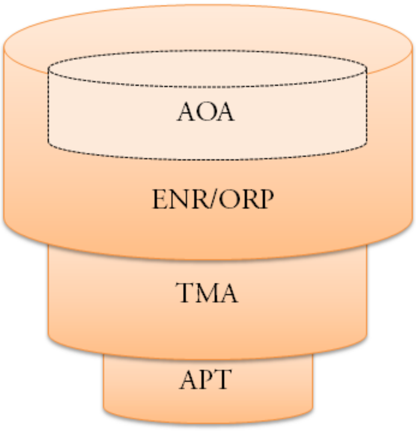


Figure 2.5: General Airspace Decomposition

The COCR introduces a set of flight parameters in order to characterize the data traffic between the aircraft and the ground. Each airspace domain has an average flight duration per phase. For instance, the aircraft is assumed to spend respectively 8 and 14 minutes in the TMA departure and arrival domains in phase 1. Also, each domain is divided into several sectors in order to manage and optimize the air traffic load. Lastly, each domain is assumed to cover a given number of ATC ground stations. For example, the ENR domain is assumed to be divided into 5 sectors and to cover 4 ground stations in phase 2. All these airspace and operational service specifications have to be carefully considered in order to define the message exchange and communication profiles of the future ATM. The ATS and AOC traffic generator resulting from the traffic profile study is used in the final step of the FAST project (see chapter 4 for more details).

2.1.4 Operational Service Traffic Profiles

The objective of this section is to evaluate the operational aeronautical communications profile according to current information available about future data-based air-ground communications. Furthermore, it is important to notice that numerical values used to develop the source model in this section have been estimated and based on *"rational"* assumptions as quoted by EUROCONTROL and the FAA in the document.

2.1.4.1 Assumptions

The COCR document has been adopted as a basis in the definition of ATS and AOC operational service traffic profiles. Several assumptions have been made in order to model these two service categories: some have been imposed by technical restrictions (*e.g.* SATCOM), others are relevant to aimed objectives. Note that the simulation model is designed for one single aircraft and based on assumptions willing to determine the worst case scenario. Indeed, the data service characteristics are given in the COCR on one aircraft basis. Moreover, these data service specifications may differ depending on input parameters as we are about to see in the next subsection (*e.g.* airspace domain, service initiator, etc): as these assumptions could impact the final results, we have chosen the most constraining conditions for our simulations.

Criteria of Operational Services Selection

The first step of the study is to extract the operational services useful and relevant to the network scenarios defined in the FAST project. Indeed, as the system architecture assumes an air-ground datalink based on SATCOM technology (see section 4.3.1), several services have to be discarded. Here are the criteria used in our selection:

- ◇ *The service initiator*: since the design of the air-ground link is focusing on downlink (from the aircraft to the ground), uplink services (*i.e.* initiated by ground entities), such as AMC (ATC Microphone Check) are discarded;

- ◇ *Aircraft to aircraft and broadcast messages*: only air to ground services are considered in our study, and then inter-aircraft and air broadcast messages as ADS-B (Automatic Dependent Surveillance - Broadcast) related services are eliminated;

- ◇ *Irrelevant airspace domains*: as defined in section 2.1.3, the APT domain concerns the immediate airport local area. Services used in this domain are usually supported by an airport ground infrastructure such as the AeroMACS network access technology (see chapter 3 for more details). TMA airspace domain is generally supported by direct datalink ground technologies such as VDL mode 2. Finally, the AOA domain assumes that there is no air traffic control for the aircraft (aircraft are assumed to self-operate). As SATCOM technology has been chosen in FAST project and will be likely used at cruise, only ENR and ORP airspace domains have been considered to model the traffic sources;

- ◇ *Redundant services*: it might seems senseless to have redundant operational services, but some ATS or AOC services redundancy indications have been found in their respective descriptions. For instance, it is explained that the FLINPINT (Flight Path Intent) data can be used to perform the FLIPCY (Flight Plan Consistency) service, superseding the necessity to implement FLIPCY. This means that it will be useless to model both FLININT and FLIPCY in the source traffic model. Consequently, all ATS and AOC services have been studied in depth in order to tell if it is relevant to take into account a service or not in the final simulation;

- ◇ *Infrequent services*: some operational services are used infrequently. For example, the D-ALERT (Data Link Alert) service is used for each aircraft once per year. In order to model most restrictive cases, these services have been considered in the operational traffic simulation.

Simulation Flight Parameters

Tables 2.1 and 2.2 summarize the flight parameters per airspace domain used in the simulation:

Table 2.1: Simulation flight parameters for COCR phase 1

Flight parameters	**Airspace domain**	
	ENR	*ORP*
Duration (min)	90	255
Sector number	8	6
ATC ground station number	4	2

Table 2.2: Simulation flight parameters for COCR phase 2

Flight parameters	**Airspace domain**	
	ENR	*ORP*
Duration (min)	82	255
Sector number	5	4
ATC ground station number	4	2

The fight duration per service, the number of sectors and ATC ground stations traversed in each domain were estimated regarding a high density air traffic profile as defined in the COCR document. High density air traffic profile are evaluated in term of PIAC (Peak Instantaneous Aircraft Count), which is the utmost expected number of aircraft for a given operational volume.

2.1.4.2 Cockpit Traffic Models

Simulations have been conducted using the network simulator OPNET Modeler[16] (a brief description of the tool can be found in appendix C.1). Based on the assumptions made above, operational services have been modeled as stochastic processes using an ON-OFF process:

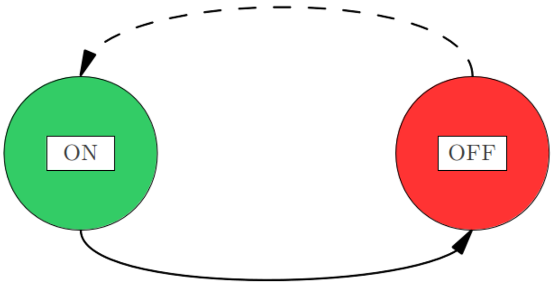


Figure 2.6: General ON-OFF Stochastic Model

The ON-OFF model, illustrated in figure 2.6, is based on a Markov chain with 2 discrete states:

1. **ON** state is used to generate operational service packet according to message specifications made in the COCR document;

2. **OFF** state corresponds to the silent mode (*i.e.* the model does not generate a packet for a specific operational service).

In order to use the ON-OFF model, two distribution laws are essential:

1. The size distribution law which indicates how the packet size is distributed along the simulation;

2. the IAT (Inter-Arrival Time) distribution law which informs on the time between two successive packets in the ON state for each service (*i.e.* the duration of OFF state, as interruptions are automatically generated by ON states to switch to the OFF state).

[16] http://www.opnet.com/

However, there is no clear indication about these type of distribution laws in the COCR document. Therefore, some realistic assumptions have been made in order to model the traffic sources. The provided model is pseudo-deterministic, meaning that:

- ◇ Packets are uniformly distributed on the flight duration. For each service, the domain duration is divided by the number of instances of this service along the domain. One instance arrives in a uniform (*i.e.* sent periodically) random way along each division;
- ◇ No size distribution law is used because packet sizes are already known (*i.e.* constant) as specified the COCR document (*c.f.* section B.1 of appendix B).

These assumptions have been based on a recent study provided by the CNES (Centre National d'Etudes Spatiales)[17] [CNES 2009]. The CNES report aimed to estimate the capacity required for AMS(R)S (Aeronautical Mobile Satellite (Route) Service) communications around 2020 over the European area. The authors used to define an ATM communications profile based on the COCR document as we have done. In their experiments, they implemented three different message generation methods: pseudo-deterministic method, Poisson's instance rate distribution method, and finally an exponential inter-arrival time distribution method. After comparing the results relevant to each method, they concluded that the differences between instance generation laws are not meaningful and all three methods can be used for global simulation without heavy effects on final results. Consequently, we decided to use a pseudo-deterministic method for the IAT packet distribution law.

Figures 2.7 and 2.8 shows respectively the stochastic simulation models developed for the ATS and AOC traffic profiles:

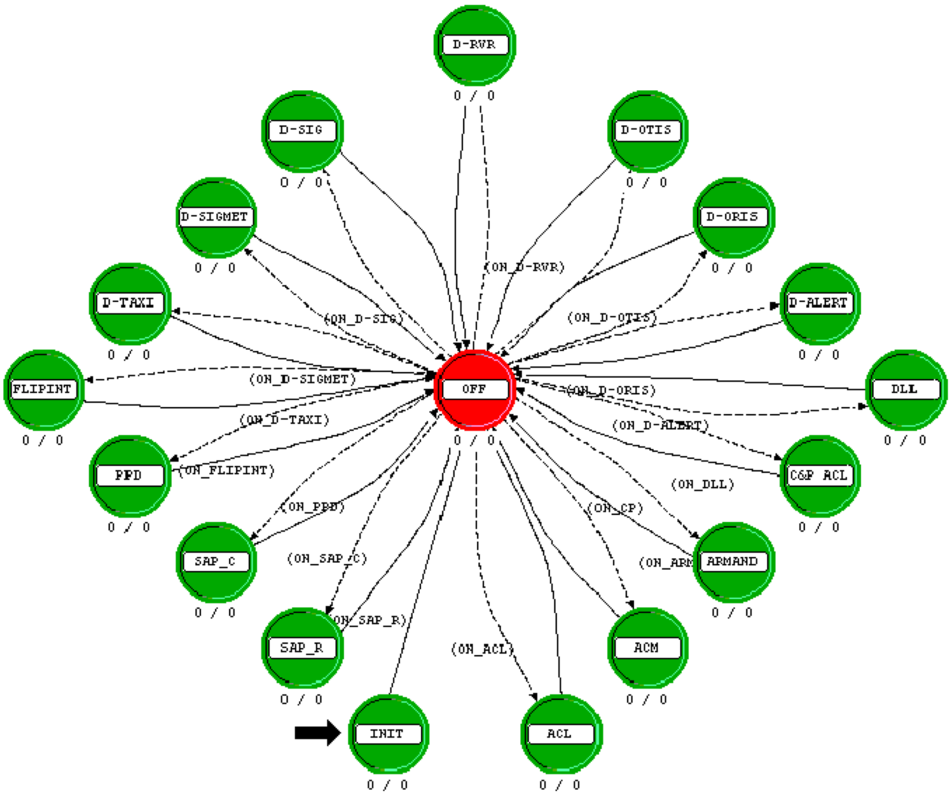


Figure 2.7: Simulation Stochastic Model for ATS Operational Services

[17] http://www.cnes.fr/web/CNES-en/7114-home-cnes.php

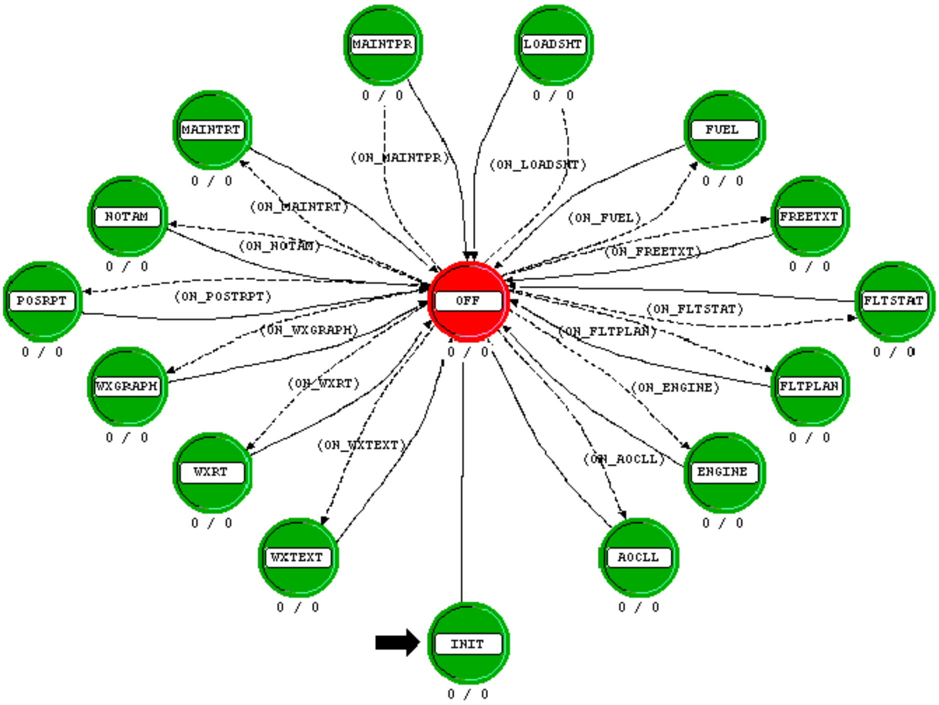


Figure 2.8: Simulation Stochastic Model for AOC Operational Services

The simulation model receives the following inputs for each service and domain, drawn from the COCR: flight duration, number of crossed sectors, transmission duration, instance number (or frequency of use per domain or per sector), number of packets per instance and packet size. Both ATS and AOC models works exactly the same way, meaning:

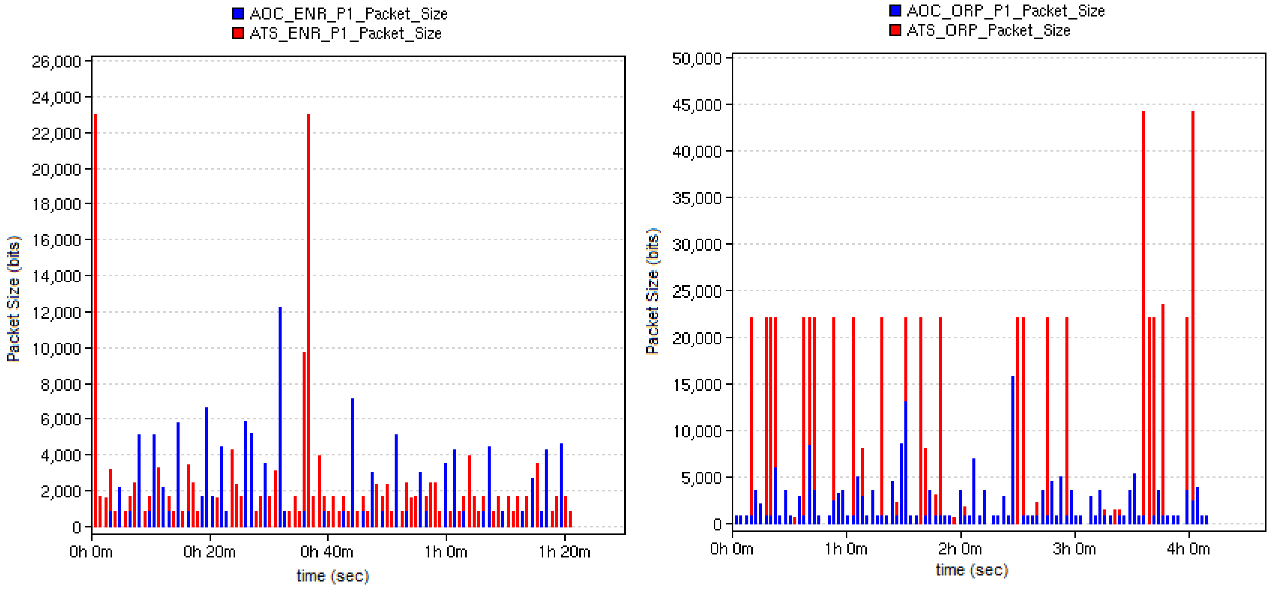
- An INIT state initiates all needed simulation variables and generates the very first interruption before passing to the OFF state (in red);
- As denoted before, no operation is done in the OFF state. The module switches to one active state (*i.e.* ON state in green) when it receives the corresponding interruption code (dotted line) to a given service. For example,the ON_ACL interruption allows to switch to the ACL (ATC Clearance) state, and then generate the relevant packets;
- Every ATS (and respectively AOC) service is represented by one active state, which allows to create packets with the corresponding size, send the message, and generate the next interruption for the next event.

2.1.4.3 Experimental Results and Discussion

In this section, we report results relevant to ENR and ORP domains in phase 1. Both packet profile and traffic bit rates are analyzed and discussed.

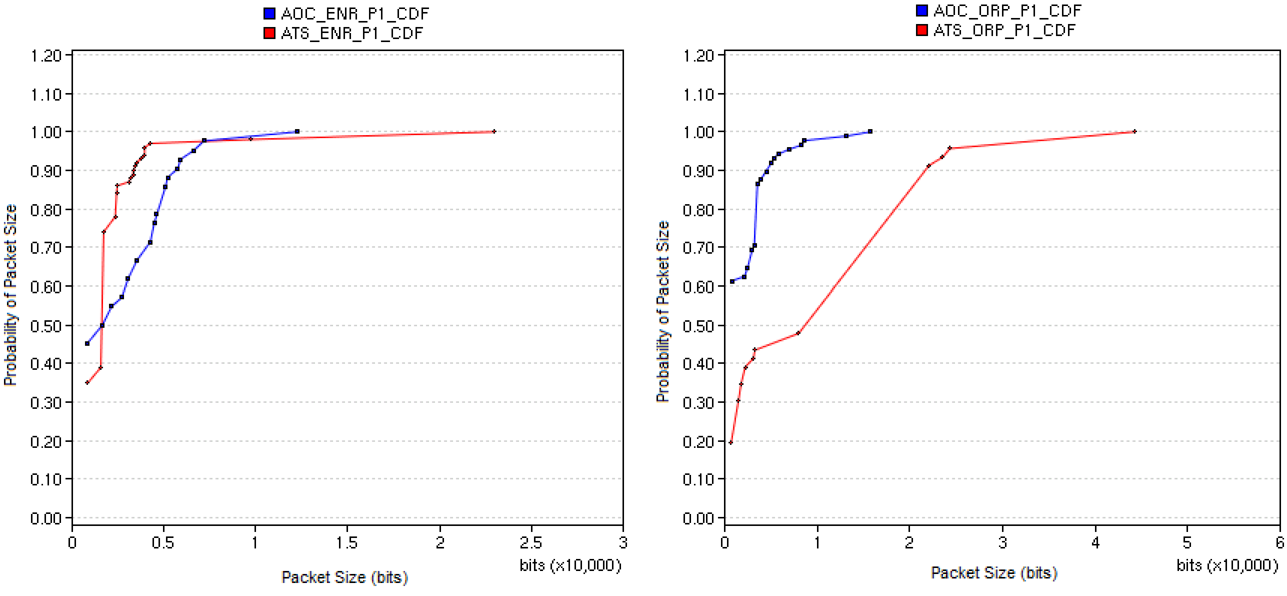
Packet Profile

Figures 2.9 and 2.10 show the generated traffic and corresponding CDFs (Cumulative Distribution Functions) for these two airspace domains:



(a) ENR Airspace Domain (b) ORP Airspace Domain

Figure 2.9: Operational Cockpit Packet Profile in Phase 1 (bits)



(a) ENR Airspace Domain (b) ORP Airspace Domain

Figure 2.10: CDF of Packet Size in Phase 1

As we can see, despite the pseudo-deterministic method used in the model, the operational traffic profile is definitely sporadic (*i.e.* irregular) with significant low loads. This is quite expected given the nature of the operational aeronautical communications. As mentioned in tables 2.3 and 2.4, most of exchanged messages are short and bursty: the minimum and maximum packet sizes are respectively 88 and 2763 bytes. The mean packet size is equal to 176.26 bytes and 109.78 bytes respectively for the ATS and the AOC services in the ENR airspace domain (2052.38 and 186.70 bytes in the ORP airspace domain). The inter-arrival time CDF is not necessary here because the packets are distributed uniformly on the flight duration as mentioned in section 2.1.4.2 (meaning the CDF is likely to be linear).

Table 2.3: Operational Traffic Results Summary for ENR Domain in Phase 1

Results	ATS	AOC
Minimum Packet Size (bytes)	88	93
Maximum Packet Size (bytes)	2763	727
Mean Packet Size (bytes)	176.26	109.78
Mean Bit Rate (bps)	44.76	24.24
Peak Bit Rate (bps)	466	248.78

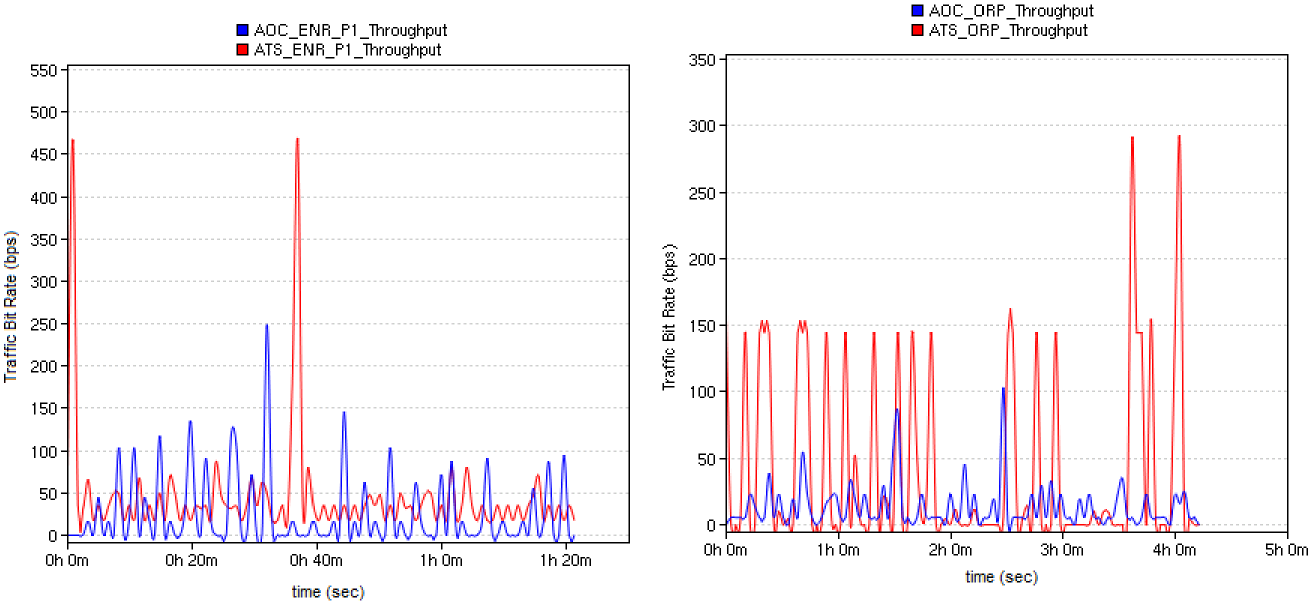
Table 2.4: Operational Traffic Results Summary for ORP Domain in Phase 1

Results	ATS	AOC
Minimum Packet Size (bytes)	88	90
Maximum Packet Size (bytes)	2763	377
Mean Packet Size (bytes)	2052.38	186.70
Mean Bit Rate (bps)	40.68	13.26
Peak Bit Rate (bps)	288.94	103

In the next sub-section, we analyze the traffic bit rate profile for operational services in order to have an estimation of the ATM traffic capacity.

Traffic Bit Rate Profile

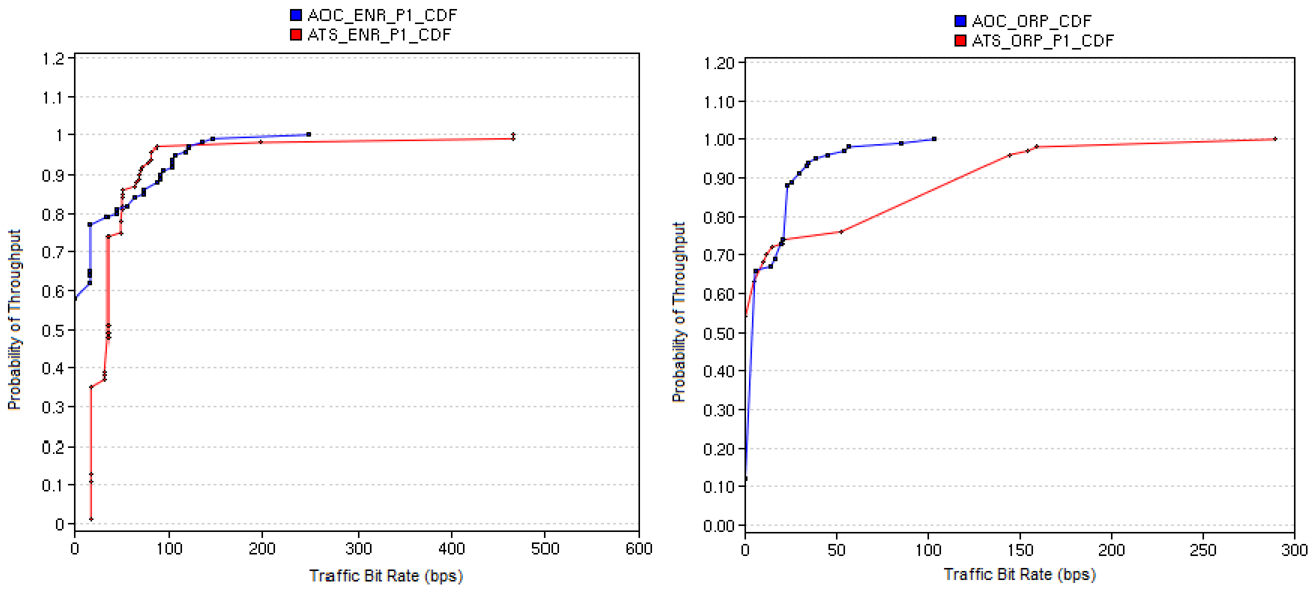
Figures 2.11 and 2.12 show the traffic bit rates for both operational ATS and AOC services:



(a) ENR Airspace Domain

(b) ORP Airspace Domain

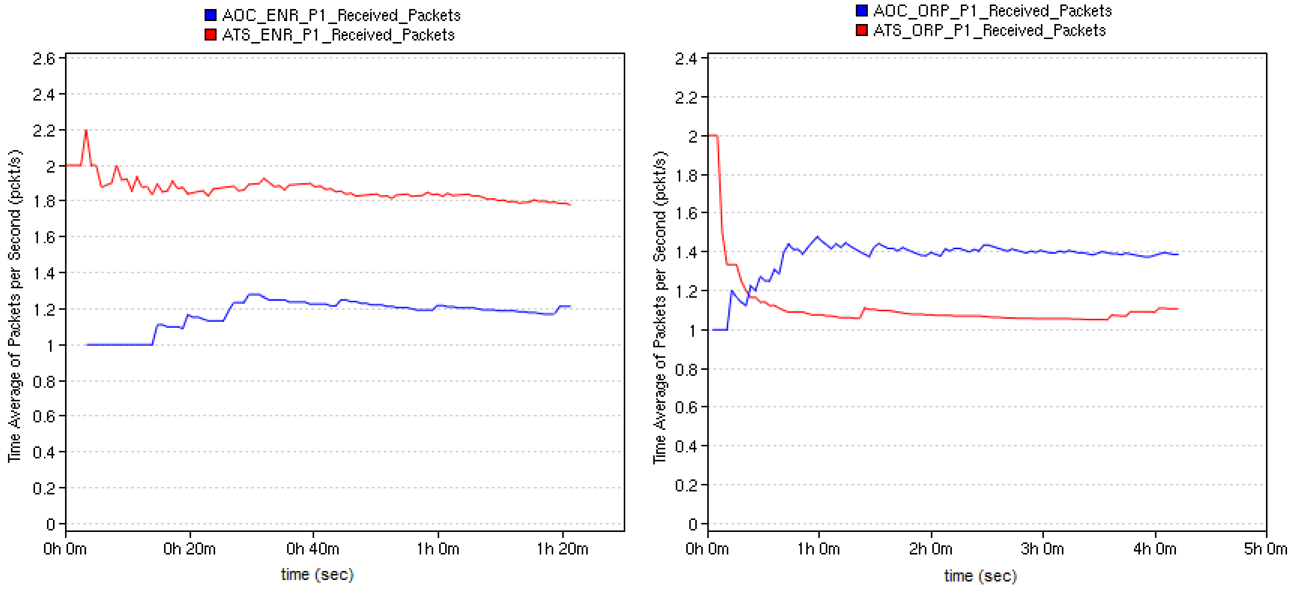
Figure 2.11: Operational Traffic Bit Rate (bps)



(a) ENR Airspace Domain (b) ORP Airspace Domain

Figure 2.12: CDF of Throughput in Phase 1

As illustrated in figures 2.11 and 2.12, the observed peak rate does not exceed 466 bps, which is considerably low, meaning that the cockpit traffic will be likely overflowed by remaining non-operational services (*i.e.* APC traffic). This can be although explained by a very low average number of packet sent per second for both ATS and AOC services as illustrated in figure 2.13. A complete summary of operational service characteristics can be found in section B.1 of Appendix B.



(a) ENR Airspace Domain (b) ORP Airspace Domain

Figure 2.13: Time Average Number of Packet per Second Sent in Phase 1 (Packet/s)

Compared to traffic profile characteristics proposed in other studies, such as in [CNES 2009], results are quite similar with few differences mainly due to particular input assumptions (IAT distribution, selected services, etc) in each study. However, the conclusion is almost the same for every traffic profile study based on the COCR: the cockpit traffic definitely exhibits low loads and a high burstiness.

Nevertheless, the developed model helped us to define the messages exchange and the communications profile of the future ATM system. The results illustrating the operational traffic bit rate and the corresponding CDFs are used to estimate the throughput capacity required for the ATS and AOC traffic in the last test phase of the FAST project (*c.f.* section 4.5 of chapter 4).

As we have seen in this section, the shift from analog voice to digital data greatly contribute in the development of new datalink aeronautical services. Nonetheless, the foreseen evolution of CNS/ATM systems will manifestly depend on other factors affecting the future of aeronautical communications systems. Such factors are introduced in the next section.

2.1.5 Likely Trends in Aeronautical Communications

This section looks at factors that could affect the future ATM infrastructure using datalink systems.

2.1.5.1 Service Integration on a Single SATCOM link

With the paradigm moving progressively towards data communications, SATCOM technology is already pointed as the perfect candidate for both operational and non-operational services. With the imminent use of data for safety-related aeronautical services as seen in the previous section, SATCOM systems should provide an important backup solution to increase the availability and capacity needed for ATM communications. GEO (Geostationary) satellites are an appealing solution able to provide a high coverage (*e.g.* in oceanic and polar regions) and good performances (*e.g.* comfortable throughput) for APC services.

Given the fact that a dedicated satellite system for safety-related applications remains a high-cost and long-term perspective solution, it seems that using a single and seamless satellite network for all aeronautical services is relatively an attractive and cost-effective solution. A plethora of R&D projects has already been created to study the feasibility of an aeronautical SATCOM system able to integrate different communication technologies and application classes into a global heterogeneous airborne network. The ESA (European Space Agency)[18] IRIS program, ANASTASIA (Airborne New Advanced Satellite Techniques and Technologies in a System Integrated Approach) [19], NEWSKY (Networking the Sky)[20], SANDRA (Seamless Aeronautical Networking through integration of Data links, Radios, and Antennas)[21], and more prove that an aggregated aeronautical SATCOM system is currently a hot topic among the industrial and research community.

2.1.5.2 Toward an IP-based ATN Network

With the near evolution of CNS/ATM networks and services, COTS products should play an important role in the next generation aeronautical networks. International standard organizations such as IETF (Internet Engineering Task Force)[22] continue to provide standard technologies for a large range of commercial applications. Plenty of these standards and protocols (*e.g.* IP) are now considered mature enough to be used for critical applications like military communications or exchange markets for instance. Even if COTS products should probably require some tweaking to be tailored for the aeronautical context (as it has been the case for AeroMACS, which is a derived mobile WiMAX technology), the air transport industry has been considering the support of COTS for their future networks since quite

[18] http://www.esa.int/

[19] http://www.anastasia-fp6.org/

[20] http://www.newsky-fp6.eu/

[21] http://www.sandra.aero/

[22] http://www.ietf.org/

a long time. The aviation actors including avionics suppliers/manufacturers, airline companies, and air navigation service providers are now convinced of COTS financial and technological benefits. Overall system development and maintenance costs shall considerably be reduced. Besides, integration with existing terrestrial networks should be made easier.

In the beginning, ICAO has developed the ATN network as a strategy for integrating air/ground and ground/ground data communications into a global network serving ATC and AOC traffic. The ATN network is composed of IS (Intermediate Systems), ES (End Systems) and communication services which allow ground, air-to-ground and avionics data subnetworks to interoperate by adopting common interface services and protocols based on the ISO (International Organization for Standardization) OSI (Open Systems Interconnection) reference model [ICAO 2002]. The legacy ATN/OSI network offers a flexible, reliable, and robust communication system for future ATM communications thanks to two transport layer protocols specifically defined for the use in ATN end systems: the Connection Oriented Transport Protocol (COTP) and the Connectionless Transport Protocol (CLTP) [ICAO 2002] (these protocols are the equivalents of TCP and UDP in the IP-based networks).

Nevertheless, ICAO has defined the ATN/IPS protocol stack as a potential next generation communication networks for future ATM in IP environment. Indeed, the ATN network will be likely used with existing IP networks having very heterogeneous characteristics (different delay, available bandwidth, packet error rates, etc). Therefore, using an ATN/IPS protocol architecture allows a reliable and seamless interoperability needed for such end-to-end IP communications. Figure 2.14 shows the ATN/IPS protocol architecture we rely on in our work as defined in the ICAO "Manual for the ATN using IPS Standards and Protocols" document [ICAO 2008]:

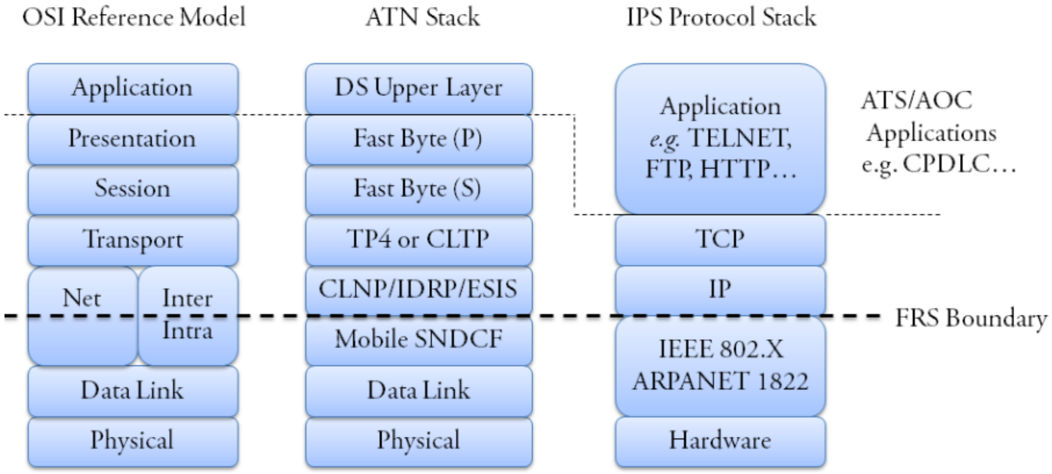


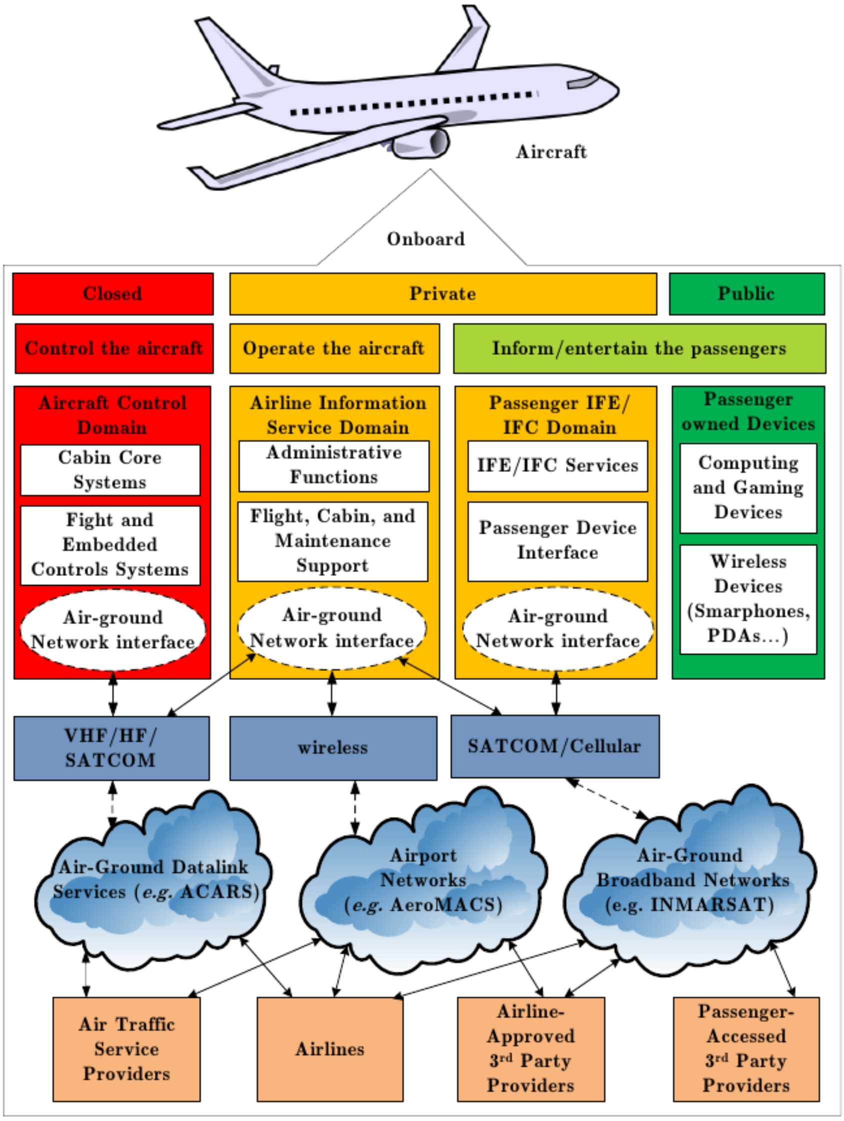
Figure 2.14: ATN/IPS Protocol Architecture

However, despite the aviation community fervor about the future ATM/CNS systems, such evolutions have certainly some consequences that must be considered. Benefits acquired from the use of datalink technologies, COTS products, or IP connectivity, cannot be realized without rigorous security considerations on the background.

2.1.5.3 Impact on Datalink Security

Indeed, many issues rise up when it comes to converge safety-related services with other on-board services: ATS traffic requires high priority and full availability, whereas AOC NG services may have restricting QoS requirements to be satisfied. While it is important for the aircraft to maintain continuous network connectivity with operational ground stations, availability becomes an important aspect of the overall network architecture, specially when considering the usage of a single satellite link for all the aeronautical networks. As usage and dependency on datalink communications increase so do security risks. Security requirements for the communication system will be more complex to fulfill due to additional factors such as traffic heterogeneity, aircraft mobility or scaling issues. Providing IFC services for passengers may be an open invitation for hacking the air-ground connectivity: Internet is an open network where anyone can connect anywhere no matter where they are. With broadband on-board connectivity, hackers may have much more opportunities to cause harm to datalink communications.

Using COTS products heightens the potential security risks involved in air/ground communications. Unlike proprietary security solutions, usually harder to break, COTS technologies are widely used in terrestrial networks such as Internet, meaning that existing vulnerabilities are likely to be publicly known. The civil aviation should then use COTS with great care and a solid vulnerability survey shall be provided in order to track new threats and discover attacks before they get exploited by malicious users. For example, in order to fully benefit from IP technology capabilities for future ATN networks, it is important to understand existing security issues related to the protocol, to analyze how the research community managed to address them, and to consider the environment in which it is going to operate. Figure 2.15 gives a very general picture of network domains, datalink technologies, and entities involved in the air-ground communication as referred in ARINC specification 664 [ARINC 2005]:



Aircraft
Onboard
Closed
Private
Public
Control the aircraft
Operate the aircraft
Inform/entertain the passengers
Aircraft Control Domain
Cabin Core Systems
Fight and Embedded Controls Systems
Air-ground Network interface
Airline Information Service Domain
Administrative Functions
Flight, Cabin, and Maintenance Support
Air-ground Network interface
Passenger IFE/IFC Domain
IFE/IFC Services
Passenger Device Interface
Air-ground Network interface
Passenger owned Devices
Computing and Gaming Devices
Wireless Devices (Smarphones, PDAs...)
VHF/HF/SATCOM
wireless
SATCOM/Cellular
Air-Ground Datalink Services (e.g. ACARS)
Airport Networks (e.g. AeroMACS)
Air-Ground Broadband Networks (e.g. INMARSAT)
Air Traffic Service Providers
Airlines
Airline-Approved 3rd Party Providers
Passenger-Accessed 3rd Party Providers

Figure 2.15: Aircraft Network Domain Model

Aware of these issues, many proposals have been made in order to enhance the security of datalink communications and identify security holes that the industry would probably face soon. In the next section, we provide a brief introduction to basic computer and network security concepts. Then we explain how the industry and research community is actually dealing with datalink security issues.

2.2 Datalink Security

2.2.1 Basic Security Concepts

This section is a brief introduction to generic security terms used in the manuscript. We specifically focus on basic security services and high abstraction level cryptographic primitives used later in more complex security infrastructures.

2.2.1.1 Security Services

Usually, there are six main security services provided by secure communication systems, namely:

1. **Confidentiality**: mostly, exchanged information have to be access-opened only to authorized users. Confidentiality is achieved by encrypting the original message to send making it readable only by the intended receiver. Confidentiality is usually required when communications are classified top secret (*e.g.* military communications) or if data are considered sensitive (*e.g.* passwords, credit card identifiers, etc). The privacy term could also refers to the confidentiality security service in the literature;

2. **Authentication**: the sender or the receiver involved in the communication must be able to prove his identity when required by the other entity he is talking to. In this case, authentication is unidirectional (*i.e.* only one communicating part has to prove his identity). Otherwise, it is bidirectional (also referred by mutual authentication), meaning a strong authentication is required before initializing the transaction. When authentication is proven, malicious intruders cannot impersonate identities to mislead legitimate users;

3. **Integrity**: data should always be amended by users who are authorized to do so. This security feature allows authorized users to verify that received information is identical to the one originally sent;

4. **Non repudiation**: this security property speculates that communicating entities cannot deny the post or the reception of a message (*i.e.* providing a proof that an event actually occurred). Usually, non repudiation is supported by some cryptographic primitives such as digital certificates or signatures. These security mechanisms are described in section 2.2.1.2;

5. **Availability**: network services and computer resources supplied by service providers must be always accessible and available to their customers when needed. In practice, this security feature is quite difficult to guarantee, specifically against DDoS (Distributed Denial of Service) attacks. DDoS attacks are intended to make these resources and services unavailable for a certain amount of time. Until today, there is no efficient way to counter DDoS attacks, such as those targeting websites[23];

[23] Early in 2011, some hacker activist groups started to launch massive DDoS attacks on governmental websites to claim their disagreements with some political choices, putting down main web portals such as the Tunisian Home Office website during the 2011 Tunisian revolution: http://gawker.com/5723104/anonymous-attacks-tunisian-government-over-wikileaks-censorship

6. **Authorization**: (also called access control) is the process of granting particular privileges to a user such as accessing a system resource. Usually, authentication is required before authorization can be given to perform some actions on the system or the network.

Basically, these security features are necessary to ensure the security of the system. Security services are essentially provided using cryptography, which can be considered as the core component of any security protocol.

2.2.1.2 Cryptographic Techniques

Fundamental cryptographic elements that can be used to provide security properties are:

- **Symmetric keys**: in symmetric cryptography, end entities use the same key K to encrypt and decrypt a message M:

  $$\{\{M\}_K\}_K = M \tag{2.1}$$

  This symmetric key has to be shared and always kept secret between both entities. Symmetric keys are either exchanged across a secure channel or derived at both ends using specific methods such as the Diffie-Hellman algorithm [Diffie & Hellman 1976]. Many algorithms have been proposed to provide symmetric cryptography such as DES (Data Encryption Standard) [NIST 1999] and AES (Advanced Encryption Standard)[NIST 2001];

- **Public and private keys**: in asymmetric cryptography, every end entity holds two keys; the public key is made publicly available to all other entities in the network while the private key is always kept secret. Asymmetric keys use one-way mathematical functions, which means it is considerably difficult to decrypt a message if it has been encrypted with one of the two keys. In other words, these keys are strongly related to each other. For example, if a message M is encrypted using the public key K_i^+ of an entity i, only the private key K_i^- allows to reveal the original message:

  $$\{\{M\}_{K_i^+}\}_{K_i^-} = M \tag{2.2}$$

  The reciprocal function remains true: if the message M is encrypted with the private key K_i^-, the public key K_i^+ is used to deduce the original message:

  $$\{\{M\}_{K_i^-}\}_{K_i^+} = M \tag{2.3}$$

  RSA (Rivest, Shamir, Adleman) [Rivest *et al.* 1978] is a widely used asymmetric algorithm in many security protocols dedicated to electronic transactions and even military applications;

- **Digital certificates**: this is a central element in the use of asymmetric cryptography technique. A certificate is a data structure used to bind the public key K_i^+ to its owner i in an authentic way. The certificate has to be signed (see below) by a trusted third party called CA (Certificate Authority) and it ensures that the public key really belongs to the entity stated in the certificate:

  $$Certificate(i) = \{i, K_i^+\}_{K_{CA}^+} \tag{2.4}$$

  A certificate aggregates many information such as a unique certificate number, the issuer identifier, the owner identifier, the public key, the cryptographic algorithm used to generate the signature or a validity period. Other fields can be included, depending on the certificate type or the purpose of use. The ITU-T (International Telecommunication Union - Telecommunication Standardization Sector)[24] X.509 format is probably the most used certificate in Internet applications [Cooper *et al.* 2008];

- ◇ **Digital hash functions**: this is a mathematical one-way function which takes a variable size data M and returns a fixed size value *M'*:

$$Hash(M) = M' \tag{2.5}$$

  A hash function output is called a hash value (also known as a checksum or digest). Hash functions have to be one way, meaning it is easy to compute *M'* if M is known, however if the digest *M'* is known, is it computationally hard to find M. Hash functions should be collision free, *i.e.* knowing M, it is computationally impossible to find $M' \neq M$ such as $Hash(M) = Hash(M')$. Finally the hash digest must always be fixed length, meaning if $M' \neq M$, $|Hash(M)| = |Hash(M')|$. SHA-1 (Secure Hash Algorithm) [NIST 1995] is an example of a hash function which can be used to compute 160 bits length hashes;

- ◇ **Digital signatures**: a digital signature is the result of a cryptographic process used for user authentication and data integrity. A digital signature is produced as follow: a checksum is computed then encrypted using the private key K_i^- of the sender i:

$$Signature(i, M) = \{Hash(M)\}_{K_i^-} \tag{2.6}$$

  The resulting digital signature is then attached to the original data and sent. In order to verify the signature, the receiver decrypts the signature using the public key K_i^+ of the signer i, gets the hash value, computes the hash of the original data and compares the two hashes: if they match then data integrity and sender authentication are proved.

In the next section, we discuss how past and current security activities manage to use these computer and network security basic concepts to provide secure and safe CNS/ATM environments.

2.2.2 Datalink Security Background

2.2.2.1 Working Groups and Industrial Efforts

This subsection describes main activities underway in the aviation industry to address security concerns related to data communications and information technologies.

EUROCAE Working Group 72

The AISS (Aeronautical Information System Security) Working Group 72 of EUROCAE (European Organization for Civil Aviation Equipment)[25] has been created to provide guidelines and methodologies to address security concerns for aeronautical systems (both airborne and ground). EUROCAE WG-72 is not intented to produce effective technical security solutions, but rather recommendations and guidance to support the growing use of highly integrated electronic systems and network technologies used onboard the aircraft. In the scope of WG-72, EUROCAE is also studying the relationship between safety and security in civil aviation: existing safety assessment methodologies related to aircraft environment are constantly revisited in order to include security assessment (*i.e.* threats, vulnerabilities, attacks) in the global framework.

[24] http://www.itu.int/ITU-T/

[25] http://www.eurocae.net/

RTCA Special Committee 216

Beside its own activities, EUROCAE WG-72 is working jointly with RTCA SC-216 (Special Committee 216) to define MASPS (Minimum Aviation System Performance Standards) for aeronautical network security. FIPS (Federal Information Processing Standards) standards provided by the NIST (National Institute of Standards and Technology)[26] are identified to be used as MASPS for the American NAS infrastructure. RTCA SC-216 is also working on a series of documents to define security assurance and assessment processes for safety-related aircraft systems such as the ED-202 "Airworthiness Security Process Specification" document [RTCA 2010].

ATA Activity

The JCG (Joint Coordination Group) have been launched by ATA (Air Transportation Association)[27] to coordinate and harmonise all the computer and network security activities conducted by commercial air transport industrial actors. The claimed goal of this group is not to produce any kind of security standards, but rather gives a global overview and centralizes the ideas and point of interests related to the civil air transport security community. Besides the JCG group, the ATA DSWG (Digital Security Working Group) has been created to provide recommendations and best practices regarding the use of PKI for datalink security. DSWG provides industry specifications to facilitate the implementations of these information security practices and technologies. These recommendations are arranged in a document entitled "ATA Specification 42 - Aviation Industry Standards for Digital Information Security" [ATA 2009]. The ATA Spec 42 document deals with digital identity management and specifies standard digital certificate profiles for the air transport industry. Further discussions about the ATA Spec 42 document are made in section 5.2.3, when PKIs in the future ATM are discussed.

ARINC and AEEC Activity

The AEEC (Airlines Electronic Engineering Committee)[28] organization produces engineering standards and technical solutions for aircraft systems (*e.g.* embedded networks, avionics, cabin systems, etc). AEEC develops and adopts the so called ARINC standards, such as ACARS for instance. AEEC created several subcommittees dealing with information security interests, namely NIS (Network Infrastructure and Security) and DSEC (Datalink Security).

The NIS subcommittee aim is to coordinate the standardization of IP connectivity and security to the aircraft. NIS is some kind of a discussion forum about network and security issues, providing a better coordination between subcommittees working on other aspects of aircraft and non-aircraft systems. The claimed goal is to avoid redundant activities, support interoperable solutions, and provide operational assessments of networking and security issues. Currently, the NIS subcommittee is working on the digital certificate usage in the aircraft environment, specially to provide guidance to aircraft manufacturers and operators about the use of PKI standards for example. NIS subcommittee coordinates permanently with the ATA DSWG to avoid duplicate provisions as they are updating security provisions in the ATA Spec 42 document.

The DSEC subcommittee goal is to develop a standard called ARINC 823 (also known as AMS) to be used as a basis for design and implementation of security mechanisms for ACARS communications. The standard has been defined in a two-part document: part 1 [ARINC 2007b] contains a specification

[26] `http://www.nist.gov/index.html`
[27] `http://www.airlines.org/pages/home.aspx`
[28] `http://www.aviation-ia.com/aeec/`

of the AMS security protocol, and part 2 [ARINC 2007a] specifies life-cycle management of the cryptographic keys necessary to secure AMS operations. Another AEEC contribution to network security in civil aviation is the “Commercial Aircraft Information Security Concepts of Operation and Process Framework” document [Olive *et al.* 2006]. Also known as ARINC Report 811, the document provides additional guidances to deal with both physical and operational security considerations in commercial aircraft. The risk-based framework is assumed to facilitate the development of cost-effective aircraft information security and provide a common language for understanding security needs.

ICAO Aeronautical Communication Panel

ICAO ACPs (Aeronautical Communication Panels)[29] were created in 2003 in order to develop standards and recommended practices for datalink aeronautical communications. Among the four working groups created by ICAO, WG-I is probably the most relevant to security concerns. ICAO WG-I primary goal is to develop guidances to use IPS in future ATN applications. IPS security have been largely addressed for air-ground and ground-ground communications. Several guidance to use and implement IPSec (IP Security) [Kent & Seo 2005] and IKE (Internet Key Exchange) [Harkins & Carrel 1998] protocols have been provided in the scope of WG-I meetings as stated in [Patel & McParland 2008a]. IP mobile security is also discussed [Patel & McParland 2008b]. All these guidelines are meant to update the security requirements listed in the ICAO "Manual for the ATN using IPS Standards and Protocols" document [ICAO 2008] as indicated in [Patel & McParland 2008c].

EUROCONTROL and FAA efforts

Recently, both EUROCONTROL and FAA showed their concern about security aspects. the 16^{th} SESAR WP is a transversal area working package meant to cover common requirements for the rest of WPs. Among other common tasks, three security areas are covered, namely airspace security, ATM security standards and best practices, and CNS security (*e.g.* ADS-B security, ATN security, etc). The FAA NEXTGEN project considers also security as an element inherent in all aspects of NAS operations. Several security aspects are currently covered by NEXTGEN including security threat detection, tracking, and secure integration of heterogeneous information systems. A program called SITS (Security Integrated Tool Set) is underway to seamlessly integrate these security considerations in the future American ATM.

Several researches have been conducted inside and outside the scope of aviation WGs described above. The next subsection gives an overview of proposed datalink security mechanisms.

2.2.2.2 State of the Art of Datalink Security

As described in [ICAO 2002], the general ATN security strategy consists of access control, message integrity, and user authentication functions. Confidentiality has been most of the time considered optional, which is quite logical if we look to the security requirements for operational services (see Appendix B). The main identified threats are data alteration, message replay, and identity masquerading. In order to mitigate these threats and improve the robustness of the ATN network, several security mechanisms have been provided at different layers of the OSI reference model.

[29] http://www.icao.int/anb/panels/acp/

Application Layer Security

Besides the AMS protocol proposed to secure ACARS communications and described in section 2.2.2.1, other work tried to provide security mechanisms at the application layer. A particular emphasis has been set on the CPDLC application. The authors in [Getachew & Griner 2005] investigated an elliptic curve-based authentication protocol for CPDLC communication systems. Mutual authentication between the pilot and ATC ground systems have been provided in order to avoid identity masquerading and spoofing attacks.

Another paper focused on the security of CPDLC over ATN [McParland *et al.* 2001]. The authors presented a set of cryptographic techniques in order to improve the overall security of pilot-controller communications. For the specific key management and agreement schemes, they suggested the use of a dedicated ATN PKI. Similarly, the authors in [Olive 2001] recommended the use of some security mechanisms in order to optimize the ATN security solution such as elliptic curve cryptography or compressed certificates. Key management and distribution have been addressed throughout an ATN PKI using airline certificate authorities. These performance considerations are clearly in line with an optimized security for a resource-restrained ATN network.

Besides the CPDLC application, other aeronautical applications have been addressed. For instance, the authors in [Sampigethaya *et al.* 2007] proposed a security framework for the use of WSNs (Wireless Sensor Networks) in AHMMS (Airplane Health Monitoring and Management System) future applications. The AHMMS system continuously checks health of airplane structures and systems via embedded sensors, providing report to on-board and off-board units. The AHMMS system has been identified by the FAA as a key enabler for current wide-body carrier airplanes, such as Airbus A380 and Boeing 787 models. The authors proposed some security primitives for a secure health data collection by the WSNs. Integrity, authentication, confidentiality, link key establishment and secure routing have been discussed regarding their potentiality to mitigate most critical threats.

[Robinson *et al.* 2007a] presented a security framework for a specific aeronautical network application, namely the EDS (Electronic Distribution of Software). The EDS application aims to distribute information assets such as software and data when the aircraft is in maintenance, in production, or on ground at the terminal. As these software are to be used when the aircraft takes-off, it is essential to ensure the integrity and authenticity of information loaded into the aircraft. Thus, the authors tried to identify main security threats targeting the EDS application, then they proposed a secure EDS system called AADS (Airplane Assets Distribution System), which addressed these threats and served as a guideline for later EDS design and implementation. The AADS system used digital signatures and key management and distribution schemes. As an extension of this work, the same authors presented two security approaches in [Robinson *et al.* 2007b] for their AADS system: an ad-hoc technique without trust chains between certificates, and a structured approach employing a third party PKI for EDS on commercial airplanes and based on the CC (Common Criteria) standard [ISO 1999].

All proposed security solutions are certainly effective in order to improve the security of each application. However, ATN security at application layer may suffer from several weaknesses. First, these solutions are relevant to cockpit communications and specific applications, which usually requires enhancements and modification on an application basis. Besides, critical issues such as services priorities and non-operational communication security remain unsolved. Furthermore, neither interoperability nor scalability issues have been addressed. PKI systems have been mentioned as possible security solutions without a real assessment or adaptation to the specific aeronautical context. These issues may be figured out at lower layers, including the transport layer.

Transport Layer Security

Providing security at the transport layer has a definite advantage over the security at the application layer, as it does not mandate modifications to each application and provides a transparent security for users. SSL (Secure Socket Layer), also known as TLS (Transport Layer Security) [Dierks & Rescorla 2008] has been discussed in several studies like an alternative to application-based security [Stephens 2004, Wargo & Dhas 2003]. However, transport layer security has several drawbacks that make it inefficient in some circumstances. For example, TLS is only able to secure TCP flows and does not provide any security mechanisms for UDP (User Datagram Protocol) flows [Postel 1980]. Indeed, TLS needs to maintain context for a connection and is not implemented over UDP. This could be inconvenient if we consider real-time applications such as telemedecine or video surveillance systems that likely use UDP instead of TCP. In this case, network layer security could be an alternative to secure upper packet layers without any considerations to the nature of transport layer flows (*i.e.* TCP or UDP).

Network Layer Security

Another aspect that has been covered by the aviation community is the IP network connectivity between the aircraft and the ground stations. In [Ehammer *et al.* 2008], the authors provided an overview of IP-based threats against aeronautical networks. They focused mainly on network logical separation using network tunnels. In [Ali *et al.* 2004], the authors depicted the scalability issues related to the use of IPSec in the scope of ATN networks. They revealed the computational high consumption related to various encryption and decryption processes within IPSec. Then they proposed a backup solution, namely the use of an anomaly detection engine within an NIDS (Network Intrusion Detection System) in order to monitor malicious acts within the operational and non-operational network domains.

Still, IPSec should play a major role in ATN security as it has been quoted in the ICAO ACPs. However, the protocol is unadapted to the use of SATCOM datalink communications. Indeed, PEPs (Performance Enhancing Proxies) [Lacamera 2007] are often used at strategic locations in order to improve the performances of satellite links, usually suffering from high latency issues. As PEP agents needs to access the TCP header in order to work, IPSec fails in delivering such information because the TCP flow is encapsulated then encrypted at the network layer. This issue has been identified in [Thanthry *et al.* 2006] and the author suggested the use of SSL/TLS-based security mechanism for aircraft data networks.

Currently, there is no datalink SATCOM security framework that has been developed specifically for aeronautical communications. Nevertheless, a lot of work has been done in the scope of IP-based satellite communications. In [Iyengar *et al.* 2007], an overview of existing threats on IP-based satellite communications has been presented. The authors made a security analysis of these threats and provided a set of security requirements for IP over satellite DVB networks. SatIPSec (Satellite IPSec) [Duquerroy *et al.* 2004], which is a derived version of the IPSec protocol for satellite communications, has been proposed in the scope of the SATSIX (Satellite-based Communications Systems within IPv6 Networks) project[30]. SatIPSec is similar to IPSec while it addresses the PEP compatibility issue of the original IPSec protocol.

Access Layer Security

The DVB-RCS (Digital Video Broadcasting - Return Channel via Satellite) [ETSI 2009], which is currently the standard proposed by the ETSI (European Telecommunications Standards Institute)[31]

[30] http://www.ist-satsix.org/

[31] http://www.etsi.org/WebSite/homepage.aspx

and used for the return satellite link in FAST project, has a link layer security framework. Unfortunately, this access layer security suffers from attacks taking advantage from upper layer vulnerabilities (*e.g.* IP spoofing attack[32], rogue attacks, etc). Another illustrative case of security weaknesses at the access layer is the AeroMACS system. In the scope of the EUROCONTROL/FAA Action Plan 17 [EUROCONTROL & FAA 2007], the AeroMACS has been identified as the *C*-band technology candidate that suits the best for the provision of dedicated aeronautical communication services on the airport surface (this technology will be detailed in the next chapter). As the AeroMACS has been based on the IEEE 802.16e/802.16-2009[33] standard, it logically uses the same layer-2 security mechanisms of the WiMAX technology.

Meanwhile, several studies [Eren & Detken 2008, Barbeau 2005] discovered critical WiMAX security weaknesses such as unauthenticated signaling messages or unencrypted management messages (these vulnerabilities are detailed in chapter 3 in the scope of SESAR WP 15.2.7). Therefore, AeroMACS security is already considered obsolete again attacks like eavesdropping or MITM (Man In The Middle) attacks. Security at upper layers seems in this case essential to mitigate these issues.

The survey of proposed security mechanisms for networked aircraft points out to similarities between traditional terrestrial networks and aeronautical datalink networks. A relevant taxonomy is provided in the next subsection in order to give an general overview of these contributions and deduce the security challenges that remain unsolved.

2.2.2.3 A Taxonomy for Aeronautical Datalink Security

Figure 2.16 exposes a taxonomy for security mechanisms already identified and used in the scope of datalink communications:

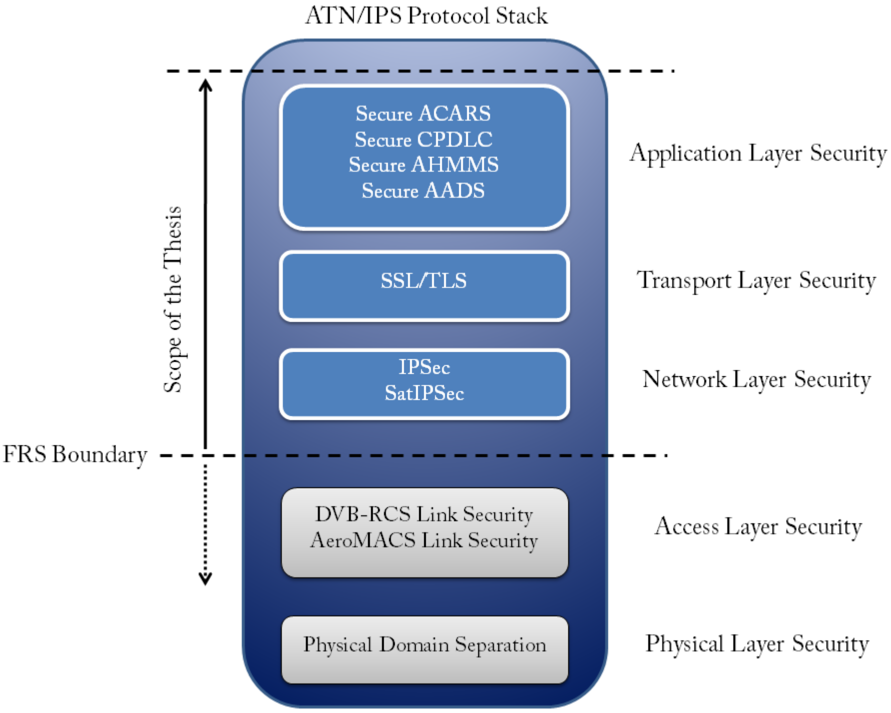


Figure 2.16: Taxonomy for Aeronautical Datalink Security

[32] IP Spoofing refers to the a type of network attack that forges IP packets with false IP addresses in order to impersonate legitimate users.

[33] http://standards.ieee.org/about/get/802/802.16.html

At the sight of this taxonomy, we can clearly see that intrinsic datalink characteristics have almost never been considered when security has been tackled. Identified security mechanisms have been roughly deployed, just to secure the air-ground communications. Thus, a challenging point relies in identifying the datalink security challenges that should be imperatively addressed.

2.2.3 Datalink Security Challenges

2.2.3.1 Logical Separation, QoS and Priority management

Currently, ATS communications have to be strictly separated from other types of communications, because of safety and regulatory reasons, as required by ICAO SARPs (Standards and Recommended Practices) [ICAO 2005]. ICAO SARPs define a set of end-to-end protocols and operational access procedures that permit both safety and non-safety aeronautical applications to use datalink technologies independently of air-ground and ground-ground sub-networks. In the perspective of a network architecture which allows the coexistence of all aeronautical services in the same infrastructure (like in the FAST project), an exciting challenge from a security point of view emerges. Indeed, the security architecture should, not only satisfy different security requirements for each traffic class, but also provide a logical and efficient segregation of ATS services in order to be fully compliant with ICAO SARPs.

Besides, efficient QoS policies (*e.g.* traffic shaping, priority queuing) have to be designed for a preferred treatment of operational data in front of non-operational data. This is a key point to prevent an excessive bandwidth consumption by non-operational applications when the network or system link capacity does no more provide enough resources for all the applications, which can be considered as a critical security issue.

2.2.3.2 Differentiated Security Requirements

In operational aeronautical communications, some security properties are more or less required than others. For instance, integrity and availability need to be highly provided for ATS services as these are vital for the safety of the flight. In this case, confidentiality is not mandatory as carried ATS data are not really sensitive. However, it could be important for airline companies to protect some of their AOC data (*e.g.* kerosene consumption) from unauthorized access for competitive matters and prevent themselves from industrial spies for instance. Appendix B.2 summarizes the security requirements for both ATS and AOC operational services in terms of confidentiality, integrity and availability as described in the COCR document.

One security mechanism would probably not be able to satisfy all the security requirements for all traffic classes, unless the strongest security mechanism is deployed. For instance, the AH (Authentication Header) [Kent 2005] mode for IPSec does not provide any confidentiality for network packets. If IPsec is configured to use the AH mode for all traffic classes, passengers would likely hesitate (even refuse) to access a pay-per-view service provided by the airline (which could have an impact on the business plan of the company by the way).

Besides, despite the fact that security requirements would be warranted, using the strongest security mechanism is an expensive and absolutely not the most performing solution considering the limited bandwidth for air-ground communications and the data overhead generated by such a heavy infrastructure. Consequently, as for differentiated QoS, the deployed security framework has to take into account that security requirements are not necessarily the same for all traffic classes.

2.2.3.3 ATM Scaling Issues

As discussed in chapter 1, the growing air traffic and passenger load is now well established. In order to manage all the security primitives (see section 2.2.1.2 for more details) that might be needed by aircraft, passengers, or avionic systems, a PKI has to be deployed. PKI is usually defined as a set of practices, technologies, and policies involved in several process such as deployment, management, storage, and revocation of certificates when cryptography is used. The aim is to create a ”chain of trust“ between heterogeneous entities in order to provide basic security services such as authentication. In the aeronautical environment, the management of PKI credentials gets more complicated because of the typical constricted network capacity of air-ground technologies: both signaling and data messages induced by the PKI have to be performed at lower cost. For instance, retrieving large CRLs repeatedly can be a bottleneck if the aircraft does not use caching mechanisms. Another restricting factor, but not the least, is the aircraft mobility. Indeed, since the aircraft and passengers should get seamless services during the flight, onboard entities should be able to authenticate or exchange required certificates when needed, even if they are not registered to the same airline for instance.

Because different aviation organizations may have different security policies in their own PKIs, interworking and roaming schemes between aircraft, end entities, or airlines are required. In such conditions, deploying a PKI regardless of these considerations becomes a tough task. Thus, a great challenge lies in finding adequate mechanisms and a well-suited PKI for datalink communications under such constraints. ATM scaling issues will be addressed in chapter 5 of the manuscript.

2.2.3.4 Minimizing Security Overhead

Security has an undeniable impact on the network performances. Experimental studies have been carried out in the past and demonstrated the performance impact due to security protocols such as IPSec [Elkeelany *et al.* 2002] or SSL (Secure Socket Layer) [Gupta *et al.* 2005]. Generally, security should be optimized, otherwise it may induce heavy overhead to data transmission and could deteriorate the system performances. For instance, public-key authentication may have an impact on the end-to-end delay [Liang & Wang 2005] whereas encryption/decryption schemes consume computational and network resources. In terrestrial networks such as LANs (Local Area Networks), large bandwidths are usually available. This makes easier the deployment of security mechanisms, regardless of the induced overhead. In contrast, air-ground datalink systems provide limited capacities. On top of being limited, these bandwidth channels are shared among several traffic classes as a consequence of service integration on the same SATCOM link (see section 2.1.5.1) and operational services RCTPs are very strict. Therefore, specific techniques shall be used to optimize the security solution for the mobile bandwidth-limited aeronautical communication environment.

Meanwhile, QoS information should play a key role in order to minimize the security overhead. Indeed, information about available network and computational resources are precious in view of choosing the security algorithm that fits the best to the current context. For instance, knowing how much bandwidth capacity is left for a given traffic class could be an interesting information in order to secure the communication. Thereby, the security could be adjusted to the network state and available resources while satisfying security requirements for each traffic class. In such a case, designing an adaptive security framework should be a valuable solution for the aeronautical context. Finding the best trade-off between required security services and system performances may be an interesting challenge to meet. From a conceptual point of view, this is equivalent to a knapsack problem: the aim is to maximize security while the induced data overhead is minimized as much as possible. This is one principle we used in the SecMan adaptive security manager framework (*c.f.* chapter 4).

2.2.3.5 Defense in Depth Security

The datalink security mechanisms overview we previously provided showed that each security layer has it own advantages and shortcomings. In most cases, these advantages and shortcomings are complementary when these security layers are compared (*e.g.* TLS vs IPSec PEP compatibility). Then, an alternative solution is to use a defense in depth security to manage datalink communications. The idea behind defense in depth is to use multiple security strategies at several layers, so that if one security layer falls down, there will be always a security backup layer to prevent a full breach into the communication system. Consequently, having a security framework that takes the advantages of each security layer and uses it to deal with the drawbacks of another security layers will be perfect. Nevertheless, such in depth security framework should not be used roughly, and must provide enough accuracy to choose the security layer mechanism only whenever needed. Defense in depth security has been considered in the SecMan framework through the selection of several security mechanisms (see chapter 4 for more details).

2.2.3.6 Risk Assessment Design

Risk assessment is generally considered as the core of the computational framework in a risk management process for a network information system. This process is mandatory and crucial for the protection of interconnected systems that provide various services to their clients or users. Traditionally, system vulnerabilities are identified, determining the likelihood of occurrence of threats being exploited, and evaluating the consequences of attacks that could take advantages of these security holes. Having assessed the risks, security measures (which could be technical or operational) are identified then implemented in order to mitigate those risks.

With the rapid introduction of datalink systems, a risk assessment phase should be conducted when the security of the network is analyzed. For the specific aeronautical context, there is no mandatory methodology imposed by aviation standard organizations, withal some risk assessment studies have been carried out and followed nearly the same principles. For instance, the ARINC 811 report provided a security process framework to meet airline specific needs. This process framework, depicted in figure 2.17, is a risk-based security assessment process composed of 4 steps, very similar to existing risk assessment methodologies for terrestrial networks:

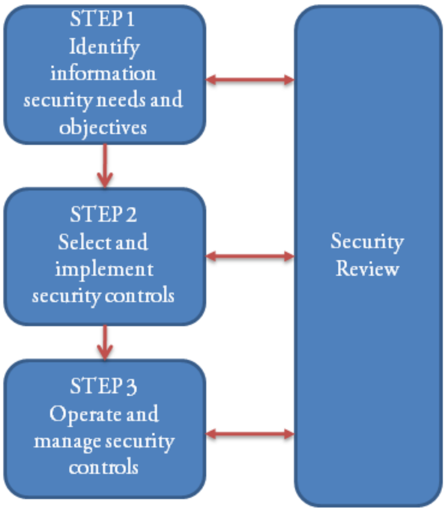

STEP 1
Identify information security needs and objectives
STEP 2
Select and implement security controls
STEP 3
Operate and manage security controls
Security Review

Figure 2.17: ARINC 811 Risk-based Security Process Framework

(a) *Step 1*: Information security needs and objectives identification;

(b) *Step 2*: Security controls selection and implementation;

(c) *Step 3*: Security controls operation and management;

(d) *Step 4*: a security review process allowing a rollback to any previous step if needed (*e.g.* if new security requirements are identified in the first step).

Another example of a risk assessment analysis in the field of aeronautical communications has been conducted throughout the NEWSKY project. Indeed, the threat analysis approach used in the project relied on a domain-based security methodology, known as DBSy (QinetQ's Domain Based Security) [34]. This approach is based on data exchanges between different security domains. In the scope of the NEWSKY threat analysis, ATS, AOC, AAC, and APC communication classes have been identified as separated security domains for the study. As long as the on-board network domain is organized in several entity clusters that have different security needs and objectives, a domain-based security analysis seemed to be a sound strategy to assess the risk in an aeronautical information system architecture.

However, these two risk assessment methodologies share the same shortcomings. First, the security analysis in both methodologies is qualitative. Indeed, security experts evaluated the potential likelihood and impact of each of identified threats to determine the risk level for the network system. This is an essential issue as security expertise is expensive financially speaking, and relatively slow compared to an automated risk assessment procedure: several workshops and appointments have to be held in order to catch the subjective assessment of the experts. Finally, the measured risks are based on a ranking scale. For instance, in the case of the NEWSKY study, the likelihood of occurrence of a threat has been ranked from *"extremely improbable"*, to *"highly probable"*. Using such an evaluation methodology allows possibly to compare two different risk levels (for instance, between high and low), but it is impossible to estimate the distance between them (for instance, between two security levels ranked as high). This can be confusing for a security network administrator willing to improve the overall security level of the network.

Besides qualitative issues, these risk assessment methodologies omit that an intrusion is mostly a transitive attack. Indeed, when an attack occurs on a network node, it is highly likely that the intruder will try to attack the interconnected nodes when this is allowed by the network topology. The attacker would be able to do so if there were some system assets that could help him to break into a connected node. These assets could be applications, services (intruded on an associated port), user logins (*e.g.* root privilege access), or database access accounts. Therefore, when the network risk level has to be estimated, it should include the risk resulting from the interconnections between the node within the network (*i.e.* the propagated risk between two or many interconnected nodes).

A comparison between quantitative and qualitative security risk assessment methodologies is provided in the next chapter before we introduce our network security risk assessment model. Also, a detailed risk assessment methods overview is depicted in order to appreciate the benefits brought by our technical solution. The designed methodology is part of our work in the scope of both FAST and SESAR projects (*c.f.* chapter 3).

2.3 FAST and SESAR Assigned Tasks

As we mentioned in chapter 1, FAST and SESAR projects have been the main scientific framework where we massively cooperated throughout these three years research work.

[34] http://www.qinetiq.com/Pages/default.aspx

2.3.1 FAST Work Packages and Tasks

In the FAST project, we collaborated in three different work packages:

1. **WP1** has been dedicated to the satellite-based system architecture. Our task was to define (jointly to the ISAE partner) the nominal system architecture and protocol stack specifications. We particularly focused on QoS policy and resource management. Finally, we have been asked to develop the simulation model for ATS and AOC data-based services in order to use it later in WP4 (see below);

2. **WP3** is clearly the biggest part of our work in FAST. In the scope of this WP, our first task was to identify potential applications used in the project. After, we evaluated the security requirements for each application in terms of confidentiality, integrity, authentication, and availability. Considering network and system restrictions discussed above, we added some security features to the nominal system architecture defined in WP1. Thus, the SecMan framework has been formally designed and implemented according to the rest of on-board network services. All the details about this work are presented in chapter 4;

3. **WP4** concerned the last step development of the project. The different partners jointly produced a tested platform, the goal was to put their applications and defined framework together in order to see how these different components interact with each other and evaluate the performances of the global system. Therefore, our last but not least task was to seamlessly interconnect the security module proxy to the rest of the system architecture and see how it secures the different services (namely ATS, AOC, APC and AOC NG).

There are two other work packages in the FAST project where we were not involved: the WP0 dealt with the management of the project and the WP2 concerned the development of the bidirectional satellite antenna.

2.3.2 SESAR Work Packages and Tasks

The SESAR project has been initially divided into 16 working packages. We participated to the WP 15 which aims to address *"...CNS technologies development and validation considering their compatibility with the Military and General Aviation user needs"*[35]. The WP 15 has been further organized into 15 projects. 15.2 sub-work package dealt with future aeronautical data-based communications. Specifically, we have been assigned to the 15.2.7 WP for airport surface datalink communications. We have been asked to evaluate the network security risk resulting from the deployment of the AeroMACS technology in the airport surface network area. Our first task was to make a state-of-the-art of existing network security risk assessment methods. As we identified few shortcomings not yet addressed, we proposed our own risk assessment methodology, implemented it, and tested it through several scenarios we defined in close collaboration with our partners. All details about the work done in the scope of SESAR are given in the next chapter.

2.3.3 Research Contributions vs. Projects Breakdown

For both projects, we have been asked to write and submit deliverables and updated technical reports, the complete document list can be found at the end of the manuscript, in the publication dedicated section D. Considering the main datalink security challenges and FAST/SESAR tasks described above,

[35] http://www.sesarju.eu/programme/workpackages/wp-15-non-avionic-cns-system--202

figure 2.18 gives a general picture of following chapter contents and how we managed to address each security issues within the thesis contributions.

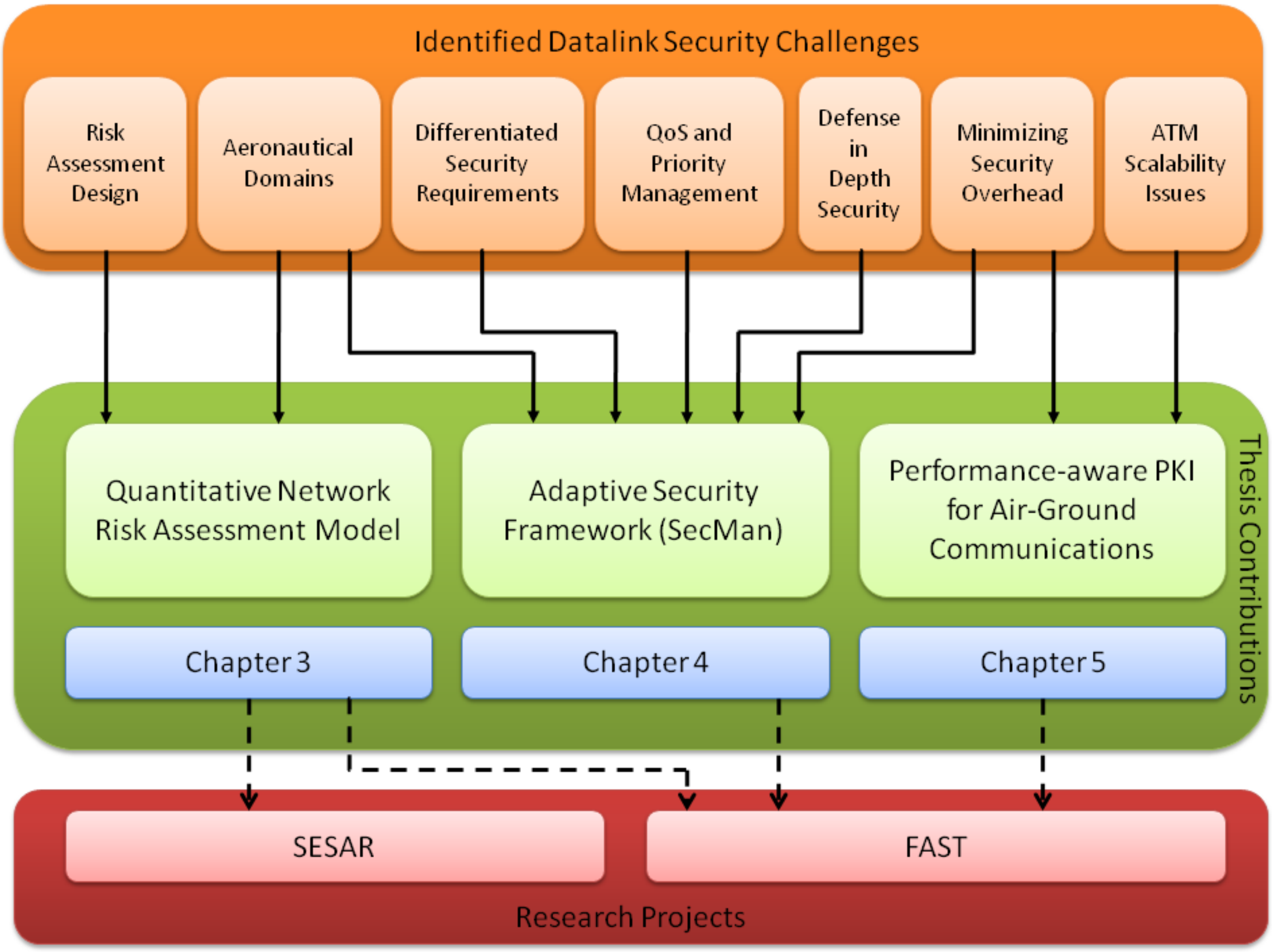

Identified Datalink Security Challenges
Risk Assessment Design
Aeronautical Domains
Differentiated Security Requirements
QoS and Priority Management
Defense in Depth Security
Minimizing Security Overhead
ATM Scalability Issues
Quantitative Network Risk Assessment Model
Adaptive Security Framework (SecMan)
Performance-aware PKI for Air-Ground Communications
Thesis Contributions
Chapter 3
Chapter 4
Chapter 5
SESAR
FAST
Research Projects

Figure 2.18: Datalink Security Challenges - Thesis Contributions Relationship

As presented in the figure 2.18, identified security challenges are sometimes addressed in more than one contribution (which is the case of the security overhead issue or the aeronautical domains for instance). The following mapping has been then made:

- Chapter 3 presents the quantitative network risk assessment model we mentioned in 2.2.3.6 where the aeronautical domains and the risk assessment design issue are going to be deeply addressed. As depicted in the figure 2.18, both SESAR and FAST projects are going to be case studies for the validation of our approach;
- Chapter 4 concerns the adaptive security framework (SecMan) we developed in the scope of the FAST project. Many security issues are addressed in this part such as the security overhead issue, the defense in depth security policy, QoS and priority management at access and network layers, heterogeneous security needs and aeronautical domains;
- Chapter 5 focuses on the ATM scaling issues, the cryptographic credentials distribution/revocation issues while minimizing the induced overhead. As for the SecMan security framework, the scalable PKI presented here is an exclusive contribution made in the scope of the FAST project.

Also, since the FAST project has been our first concern during these three thesis work, all thesis contributions are relevant to the FAST project. We have been asked to join the SESAR 15.2.7 WP lately in 2010 and thus only the quantitative network risk assessment model introduced in chapter 3 is relevant to this project.

2.4 Summary

In this chapter, we provided an overview of datalink security and future ATM likely trends. All data-based aeronautical service categories have been discussed and a traffic profile model for operational data services has been presented with relevant results. The core of this chapter was dedicated to a survey of different security mechanisms used in existing aeronautical datalink networks. A conclusion was finally drawn with regards to this survey in order to highlight main ATM security challenges.

It is clear that in the long term future, regulatory and mandatory policies imposed by the aviation industry, which still exist todays, will be relaxed only if a secure infrastructure that deals with these security issues is provided and proved to work efficiently, starting from risk assessment, which is a critical step in the perspective of security countermeasures implementation. As it is the very first step in a network and information security analysis, we dedicate the next chapter to our first contribution, namely a quantitative risk assessment methodology for network security. We investigate particularly in depth two main concepts: quantitative risk parameters and risk propagation among the network.

Chapter 3

Risk Propagation-based Quantitative Assessment Methodology for Network Security

Contents

Note: Materials and results presented in this chapter have appeared in [Ben Mahmoud *et al.* 2011c], [Ben Mahmoud *et al.* 2011d], [Ben Mahmoud *et al.* 2011e], and [Ben Mahmoud *et al.* 2011f].

3.1 Introduction to Information System Security Risk Management Process

Usually, ISSRM (Information System Security Risk Management) processes follow an overall framework composed of classical and common steps. Nevertheless, these steps can be different from one method to another and do not put necessarily the same weight on each step. For instance, some methods focus on security controls and countermeasures whereas others put more efforts on risk assessment and treatment procedures.

However, a general ISSRM framework can be drawn and considered as a basis for any information security management related work, as illustrated in figure 3.1:

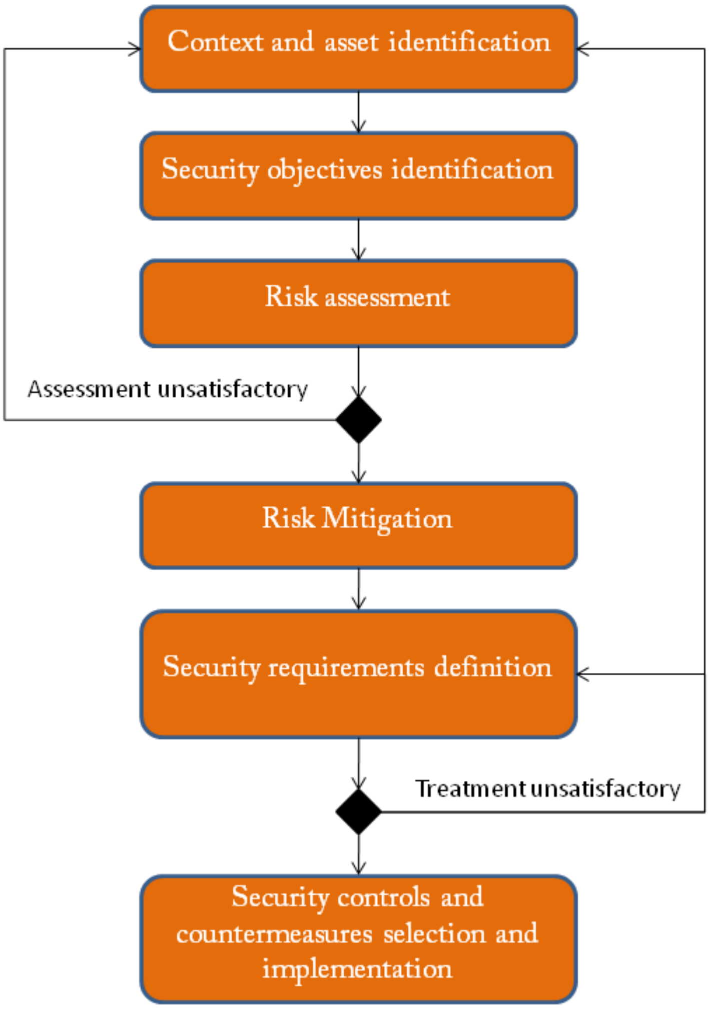

Context and asset identification
Security objectives identification
Risk assessment
Assessment unsatisfactory
Risk Mitigation
Security requirements definition
Treatment unsatisfactory
Security controls and countermeasures selection and implementation

Figure 3.1: General Information System Security Risk Management Process

This general ISSRM framework is composed of six steps:

(a) *Context and asset identification*: first, the system and its environment are described with an emphasis on the sensitive assets (*e.g.* devices, data, etc) to protect;

(b) *Security objectives identification*: security needs are then defined. Based on the previous step, security objectives to be warranted, usually expressed in terms of basic security services (as defined in section 2.2.1.1 of chapter 2), are identified;

(c) *Risk assessment*: this step consists in estimating potential risks that can harm the assets identified in step (a) and threaten security objectives of step (b). The risk assessment procedure can be based on a qualitative or quantitative study. Note that if the risk assessment is unsatisfactory, it could be possible to go back to previous steps and restart again the analysis;

(d) *Risk mitigation*: once the risk has been clearly identified, risk treatment measures can be taken. For instance, such a measure could be to decide to retain the risk (*e.g.* accept the risk because it is considered low enough), reduce the risk (*e.g.* reinforce security policies), or avoid the risk (*e.g.* deactivate a network device with a high risk);

(e) *Security requirements definition*: security requirements are determined as security solutions to mitigate the identified risks, mainly if the risk reduction strategy has been fixed;

(f) *Security controls and countermeasures selection and implementation*: finally, security requirements are instantiated into explicit security controls and countermeasures. For instance, a stateful inspection firewall has been selected and implemented to protect a DMZ (Demilitarized Zone) in the FAST project where the SecMan proxy has been deployed (chapter 4).

This very general picture of the ISSRM process highlights the importance of the intermediate risk assessment step, generally considered as the nucleus of risk management processes lifecyle. As a matter of fact, next steps, such as security controls and countermeasures implementation, highly depend on the success of the risk assessment step. For instance, if the risk is over estimated, administrators will likely implement high-cost protection devices to mitigate a risk which actually necessitate cheaper equipments.

Many approaches can be used to evaluate the risk related to information and network security systems. Most often, the security risk is expressed as:

$$Risk = Likelihood * Impact \tag{3.1}$$

Indeed, risk assessment is usually conducted based on threat likelihood and impact, which are respectively the probability of occurrence of a threat and potential damages resulting from it on the system. A threat is defined as the possibility for an intruder to attack a system by exploiting existing vulnerabilities. However, this is one general expression among others, and as involved factors (*e.g.* likelihood, impact) could be modeled in many ways, numerous security risk assessment methods have already been proposed.

As we mentioned in step (c) of figure 3.1, these techniques can be done in a quantitative or qualitative manner.

3.1.1 Quantitative vs. Qualitative Risk Assessment Approaches

Typically, qualitative risk assessment approaches rely on security specialist expertise and, most of the time questionnaires are used to gather their opinions, like in [Bennett & Kailay 1992]. This can be a financial issue as security expertise costs money to companies. Also, data collection process is considered complex as it requires much time and effort, and might induce some computation errors (because they are performed manually). Besides, the qualitative results are mostly based on a ranking scale, and then cannot be substantially evaluated because of their subjective nature.

For instance, it is possible to compare two security risk levels (*e.g.* between high and low) but impossible to estimate the distance between these measures (*e.g.* between two levels ranked as high). Moreover, the security expertise is generally based on the expert intuitiveness and past experiences in the field, which does not always reflect the current and real situation. Then, qualitative risk assessment techniques likely suffer from a lack of sound theoretical bases, which does not give a concrete knowledge about the information security risk.

Quantitative risk assessment allows a more accurate analysis of risk events, and to some degree, solves the issues related to qualitative techniques. In fact, a plethora of parameters involved in the risk assessment process can be used and are designed in many ways thanks to mathematical and theoretical models. For instance, some designers might focus on modeling the impact of threats on business assets, whereas others decide to concentrate their efforts on attack progression modeling using Petri networks [Jing 2009]. This allows a sharper analysis of risk events compared to qualitative techniques.

Besides, the quantitative results are mostly accurate and can be expressed either in business or technical languages. On the one hand, this makes it easier for enterprises in reaching their financial objectives. In the other hand, it could be helpful for administrators willing to enhance the security of their networks. Quantitative risk assessment methods are usually supported by automated tools, which has the advantage of accelerating the assessment process and avoiding some computation errors.

Furthermore, quantitative risk assessment techniques can be used either for preventive risk analysis, or reactive risk analysis depending on the context of the study. Preventive risk analysis often relies on the ALE (Annual Loss Expectancy) index [Microsoft 2004], which is the annual monetary loss that can be expected by a company according to the identified likely risk events. From a financial point of interest, ALE is an important metric that can be used directly in cost-benefit risk analysis.

Quantitative risk assessment techniques support also reactive analysis, which are generally conducted to identify security countermeasures when an alert corresponding to an attack is triggered by a monitoring system. This could be done using, for instance, a NIDS system. For this purpose, several decision criteria are used and modeled in various ways. The most prominent models are detection and reaction cost models (*e.g.* number of security countermeasures to deploy, percentage of intrusions into the supervised network, monetary or processing resources required to cope with an attack) [Barth *et al.* 2009], attack models (*e.g.* scenarios-based or tree-based graphs) [Wing 2008], and threat impact models (*e.g.* impact distribution laws, impact progression over the network) [Lao & Wang 2008].

Succinctly, qualitative and quantitative information security risk assessment approaches could be compared from three points of view: subjectivity, efficiency, and cost. Table 3.1 depicts a summary of the advantages (denoted by +) and drawbacks (denoted by −) of each approach according to these three axes:

Table 3.1: Qualitative vs. Quantitative Risk Assessment Approaches

Criteria	**Quantitative Approaches**	**Qualitative Approaches**
Subjectivity	− At a design level + Solid theoretical models + Several factors are modeled	− Security experts intuitiveness and past experiences − Pedestrian risk evaluation
Efficiency	+ Numerical risk estimation (comparison is always possible) + Automated procedures/tools (less errors) − Based on advanced aspects (not adapted for beginners)	− Ranking scale (difficult to compare) − Computation errors (human in the loop) − Preventive/reactive analysis are difficult to conduct
Cost	+ Relatively fast (only time needed by the tool) + No extra-expense	− Time-consuming procedures (*e.g.* questionnaires collection) − Financially expensive

Looking to the comparison made above, it makes sense to confirm that a quantitative risk assessment approach is strongly desired. A lot of work has been done in this area and we provide further a summary of major researches in this field (section 3.2.3).

Consequently, our methodology is based on a quantitative assessment of each parameter involved in the global risk processing. However, the survey of these quantitative risk assessment methods emphasized another point that should be considered when the so-called risk assessment methodology has to be designed for an information system network, namely the network security risk propagation.

3.1.2 Network Security Risk Propagation Concept

3.1.2.1 Impact of Node Correlation

In order to understand the importance of the network security risk propagation concept in the design of a risk assessment methodology, let us see what could be the simplest definition of the word "network". According to the Online Cambridge Dictionary, it is *"a large system consisting of many similar **parts** that are **connected** together to allow movement or **communication** between or along the parts or between the parts and a control centre"*[1]. Starting from this definition, we can deduce three important concepts that must be considered carefully when a risk assessment method has to be designed for a network information system:

- ◇ *Nodes*: these are the main components of a network information system, such as end systems (terminals, servers) and intermediate systems (hubs, switches, gateways). Every node has its own set of vulnerabilities that can be related to hardware, software, protocol stack, etc;
- ◇ *Physical interconnection between nodes*: as we have seen in the definition, nodes are interconnected by physical supports in a network. For example, nodes can be connected with cables (shielded twisted pair cable for instance) in a wired LAN or using radio waves (WIFI for instance) in a WLAN (Wireless LAN);
- ◇ *Communication (i.e. data flows) between nodes*: some nodes are able to provide services (FTP transfer, HTTP browsing, database access, etc). If two nodes want to communicate together, they must be interconnected physically and logically.

Considering all these factors, it is not easy to deduce exactly the total risk of a large network, even if we can evaluate this risk node by node. In fact, apart from individual vulnerabilities, the global network security can be seriously compromised by the interconnected nodes. Indeed, many endogenous and exogenous factors have to be analyzed in order to determine as accurately as possible the risk level for the whole network.

On the one hand, the global network risk can be very low even if the risk related to a single node is very high (*e.g.* this node is isolated from the rest of the network and does not communicate with many other nodes). On the other hand, the security of the whole network can be heavily compromised by nodes which have strong interconnections and data flow exchanges with the rest of the network, even if those nodes have individually a low network risk.

Therefore, network security risk should no longer be evaluated individually, but rather globally taking into account the service dependencies and node correlation. The security risk propagation within an information system network consolidates the idea that network intrusions are likely transitive processes.

3.1.2.2 Network Security Risk Transitivity

When an attack occurs on a network node, it is highly likely that the intruder will try to attack the interconnected nodes when this is allowed by the network topology. The attacker would be able to do so if there are some system assets that could help him to break into a connected node. These assets could be applications, services (intruded on the associated port), user logins (*e.g.* root privilege access), or database access accounts. Strong dependencies between these system facilities imply some kind of transitivity in the network risk propagation process.

[1] http://dictionary.cambridge.org/dictionary/british/network_1

By way of example, let i and j be two correlated nodes in the network and t an exploitable vulnerability on node j as shown in figure 3.2. Since node j has some vulnerabilities that could be exploited by an attack (step B) , it might transmit its correlative risk to the connected node i (step C). This risk will propagate to the different nodes connected with node j. Besides, as long as the risk has been propagated from node j to the correlated node i, there is a strong probability that the intruder continues his way and tries to break in nodes connected with i (node k in step D). To provide deeper understanding of the network security risk transitivity, we illustrate the risk propagation concept through a practical intrusion scenario in the following subsection.

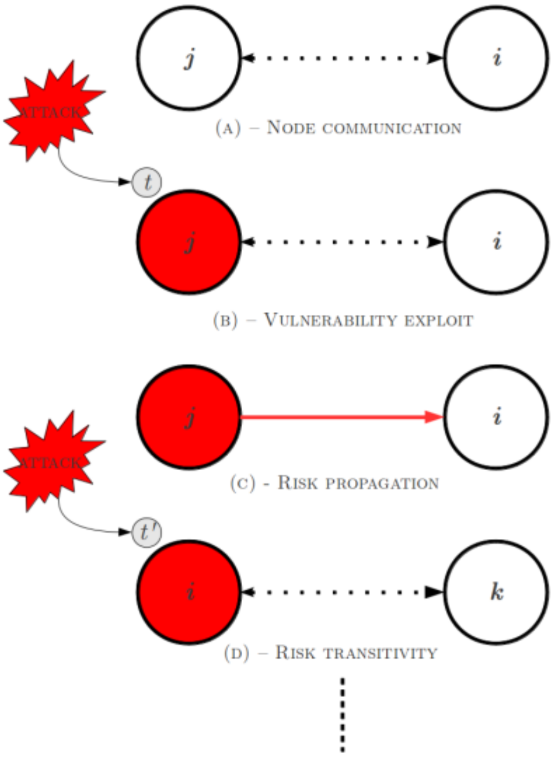


Figure 3.2: Risk Transitivity Between Correlated Nodes

3.1.2.3 Network Security Risk Propagation Illustrative Case

Figure 3.3 illustrates an example of a step-by-step network security risk propagation into a simple LAN network. Let's say administrator users on node A are allowed to log on a web server (node B) using the SSH (Secure SHell) [Ylonen & Lonvick 2006] service in order to manage a website and refresh its content. Users possessing root privileges on node B are allowed to access a database (node C) that contains confidential data (*e.g.* website user informations like emails, credit cards, addresses, etc). Only root users on node B are allowed to access the database on node C: for this purpose a firewall (node D) is deployed and configured to filter the access to node C, meaning users from node A (even those with administrator privileges) are prevented from login the database.

However, node A could suffers from a vulnerability that is still exploitable (*i.e.* not already fixed). An intruder may first exploit this specific vulnerability to node A (*e.g.* OS vulnerability) to get administrator privileges. He would probably face some issues trying to access directly node C from node A, but he could gain access to node B using the SSH service. In a second phase, he may try to grant root privileges on node B, then access the confidential data on node C without being intercepted.

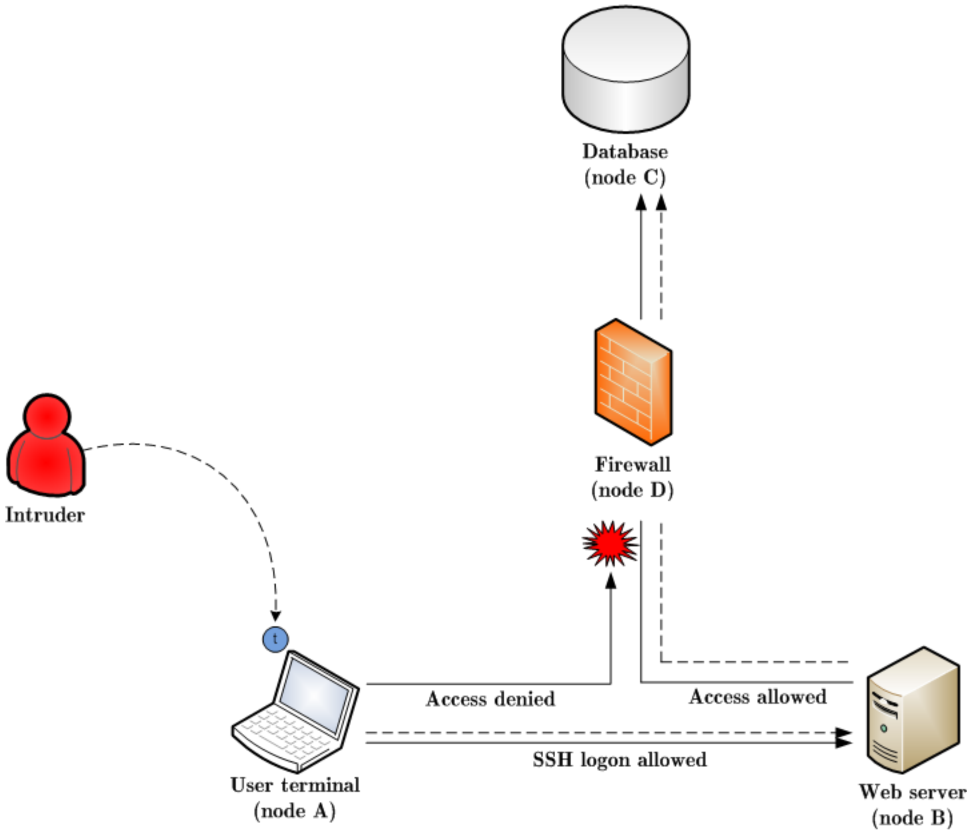


Figure 3.3: Network Security Risk Propagation Example

In this subsection, we turned our attention to a second point of interest in network security risk assessment, namely the risk propagation. We showed that risk should not be considered under a classic perspective (*i.e.* individually node by node), but instead at a higher level such that the impact of node correlation is taken into account in the risk computation.

The methodology presented later takes into consideration both quantitative assessment and risk propagation concepts. The proposed approach could help administrators willing to compare different security policies and find a cost-effective and secure policy. Besides, they will be able to evaluate the impact of any topological change in the network architecture (*e.g.* adding or deleting a node) on the network security. All the parameters involved in the network risk measurement are explained and quantified: threat likelihood, risk impact (*i.e.* cost of damages), individual network risk (*i.e.* specific to a single node), and the total risk induced by the interconnection between the network components.

Furthermore, this methodology can be applied to any computer network and is not specific to a particular environment. Since we are able to quantify the logical interconnection between network nodes, the security assessment framework would fit to measure the risk on that network. While specific characteristics (such as priority between network domains) have been included to support datalink communications, the aeronautical network remains a case study of the presented methodology as we will see in the dedicated subsection.

In the last section of this chapter, we apply our approach on both FAST and SESAR network architectures. However the aim of the study is not exactly the same. In the FAST context, we compare the network risk relevant to two different network architectures. The goal is to validate the methodology showing that the network risk is higher in a nominal architecture (*i.e.* without security features) than in a secure architecture (*i.e.* with security-dedicated devices).

In the SESAR context, we focus on the AeroMACS access network topology for airport communications. The goal is to discuss the risk results for the isolated AeroMACS scenario and to compare them regarding the authentication/authorization intrinsic security mechanisms in order to finally find which scenario holds the lower network risk and to provide at the end some security guidance for future AeroMACS implementations. The validation experiments relied on vulnerability statistics issued from the NVD (National Vulnerability Database)[2] CVE (Common Vulnerabilities and Exposures) database published by the NIST. The NVD provides information about vulnerabilities such as type, severity class and scores, extended descriptions, products or versions affected. Other vulnerability and statistical reports exist like Secunia[3] or OSVDB (Open Source Vulnerability Database)[4] databases: we picked up the NVD database because it provides the CVSS (Common Vulnerability Scoring System) [Schiffman 2005] severity score of a vulnerability, which is an essential quantitative parameter in our methodology.

In the following section, we provide a network security risk management methodology overview with a particular emphasis on quantitative security risk assessment approaches.

3.2 Security Risk Management Background

ISSRM is a very active domain and many methods have already been proposed in the last decade. These methods are mainly driven by standards and guidance provided by professional and standardization organizations.

3.2.1 Security Risk Management Standards

Two major security standards has been proposed, namely ISO/IEC 13335 [ISO 2004] and ISO/IEC 15408 - CC (Common Criteria) [ISO 1999] standards. ISO/IEC 13335 is designed to define the basis for the information security management, whereas the CC standard provides tools to define a set of security requirements and for evaluating the security specifications of an information system. These standards are widely used and referenced in many other standards and methods. However, despites the security concern, both standards do not specifically focus on risk management and assessment activities.

Other standards dealing with security are focusing on risk management field such as the NIST SP 800-30 “Risk Management Guide for Information Technology Systems” standard [NIST 2002], or The ISO/IEC 2700x series of standards, which is probably the most referred in the literature. As shown in figure 3.4, the ISO/IEC 2700x series is composed of 8 standards dedicated to information security, where ISO/27005 standard [ISO 2008a] is the most relevant to our work.

The ISO/27005 standard provides best practices and guidance to perform security risk management, as required by the ISO/IEC 27001 [ISO 2005a]. The ISO/27005 process is composed of 7 steps: context establishment, risk assessment, risk treatment, risk acceptance, risk communication, risk monitoring and a continuous review step, as it appears in [ISO 2008a]. These steps are quite the same as the ones presented in figure 3.1. Risk assessment is obviously an intermediate step in the standard and holds an important place in the overall ISSRM process.

These standards are related to information security in general and are not provided for a specific context such as the aeronautical field. Therefore, ARINC provided in 2005 the ARINC 811 standard discussed in chapter 2, which includes additional guidance to deal with physical and operational security of aeronautical hardware and software information assets.

[2] http://nvd.nist.gov/
[3] http://secunia.com/
[4] http://osvdb.org/

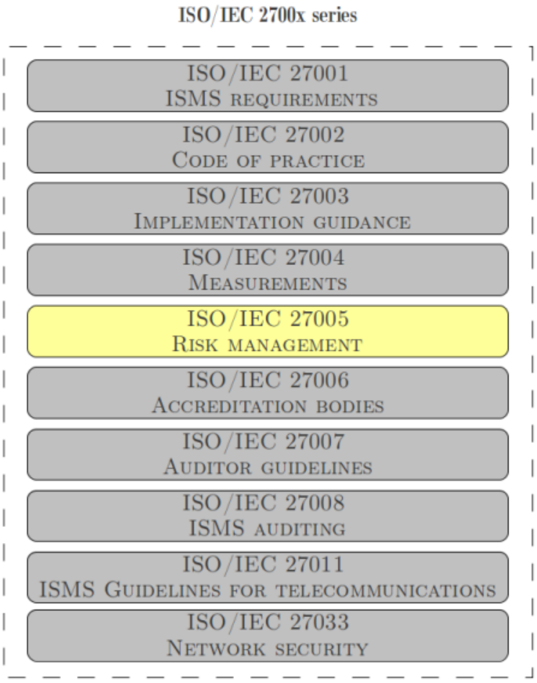


Figure 3.4: The IS/IEC 2700x Series of Standards

All these standards serve as a basis for the design and implementation of security risk management methods, defined for specific contexts and compliant with international standards such as [ISO 2008a, NIST 2002].

3.2.2 Security Risk Management Methods

In this section, we discuss the most known security risk management methods in the security community. Indeed, there are more than 200 ISSRM methods in the literature, however plenty of them are private and used in the scope of a single company or enterprise. In the literature, the main methods that are usually quoted are : CCTA (Central Computer and Telecommunications Agency) CRAMM (CCTA Risk Analysis and Management Method)[5], OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) [Bennett & Kailay 1992], EBIOS (Expression des Besoins et Identification des Objectifs de Sécurité) [DCSSI 2004], MEHARI (Méthode Harmonisée d'Analyse du Risque Informatique)[6], and CORAS (Risk Assessment of Security Critical Systems) [Fredriksen *et al.* 2002].

3.2.2.1 CRAMM

The CRAMM method has been developed by the CCTA of the United Kingdom in 1985. Currently, the method is owned by the SIEMENS company and is supported by a software tool that provides guidance to collect needed data and exploit adequately the method. CRAMM is built around three steps: asset identification and evaluation, threat and vulnerability assessment, and countermeasures selection and recommendation. CRAMM is the only ISSRM method that explicitly recommends the use of quantitative risk assessment but it uses a pedestrian scale values ranging from very low to high.

[5] http://www.cramm.com/

[6] http://www.clusif.asso.fr/en/clusif/present/#mehari_link

3.2.2.2 OCTAVE

OCTAVE risk-based security assessment method has been published in 2001 by the Software Engineering Institute of The Carnegie Mellon University of Pittsburgh, USA. The method has an original approach to decompose the risk assessment into three phases: building asset-based profiles, identifying infrastructures vulnerabilities, and developing security strategy and plans. OCTAVE aims to examine organisational and technologial issues and define the best security strategy to face the risk events.

3.2.2.3 EBIOS

EBIOS has been created in 1995 by the French DCSSI (Direction Centrale de la Sécurité des Systèmes d'Information)[7] under the authority of the FNISA (French Network and Information Security Agency). EBIOS is assumed to support all the ISSRM process (*e.g.* risk assessment, risk mitigation, etc) and is strongly recommended for military and governmental information systems use. Still, the method is also commonly used in the industry by several companies. EBIOS is composed of 5 steps: context of study, expression of security needs, threat study, identification of security objectives, and finally identification of security requirements. EBIOS is also supported by an open-source software tool that makes easier the use of the method by non-initiated users.

3.2.2.4 MEHARI

MEHARI has been created in 1996 by the CLUSIF (Club de la Sécurité de l'Information Français)[8], a French association of information security professionals. MEHARI has been build upon two methods called MARION (Méthodologie d'Analyse de Risques Informatiques Orientée par Niveaux) and MELISA (Méthode d'Evaluation de la Vulnérabilité Résiduelle des Sytèmes d'Armement), which are now obsolete and not maintained anymore. The MEHARI method is composed of several modules (security analysis and classification, evaluation guide for security services, and risk analysis guide) which are centered around risk assessment and management processes. The advantage is that each module can be used independently without necessary running the other modules. As for EBIOS, MEHARI is supported by a software tool[9].

3.2.2.5 CORAS

The CORAS ISSRM method results from an European project led by research institutes (*e.g.* Queen Mary University of London) and commercial industries (*e.g.* German Solinet[10]). The aim of the project was to develop a security framework supporting risk analysis and assessment for critical security systems (*e.g.* telemedecine and e-commerce applications). The method provides an integrated risk management and development process and an XML (Extensible Markup Language)[11] based platform.

3.2.2.6 Discussion

The approaches used in the risk management methods mentioned above evaluate damages produced by threats qualitatively, making results somewhat subjective as we discussed in section 3.1.1.

[7] http://www.ssi.gouv.fr/en/

[8] http://www.clusif.asso.fr/en/clusif/present/

[9] http://www.clusif.asso.fr/en/production/mehari/download.asp

[10] http://www.solinet.com/

[11] http://www.w3.org/TR/2008/REC-xml-20081126/

Table 3.2 summarizes these methods and highlights the lack of quantitative-oriented approach among the existing ISSRM methodologies:

Table 3.2: ISSRM Standards and Methods Summary

Standards and Methods	Security oriented	Risk oriented	Aeronautical oriented	Quantitative based approach
ISO 13335	✓	×	×	×
ISO 15408	✓	×	×	×
ISO 27001	✓	×	×	×
ISO 27005	✓	✓	×	×
NIST SP 800-30	✓	✓	×	×
ARINC 811	✓	✓	✓	×
CRAMM	✓	✓	×	*R*
OCTAVE	✓	✓	×	×
EBIOS	✓	✓	×	×
MEHARI	✓	✓	×	×
CORAS	✓	✓	×	×

Legend

✓: covered

×: not covered

R : recommended but not used

In this perspective, the scientific community worked a lot on this specific point. In the next subsection, we cover the main contributions made in the field of quantitative and formal representation of network security risk assessment.

3.2.3 Quantitative Security Risk Assessment Approaches

There are lots of studies on network and information security risk quantification. [Chen & Zheng 2006] outlined hierarchical threat assessment models for network security and quantified the information system security parameters. Instead of focusing on quantifying the risk, this model is more threat-oriented, meaning that the impact and probability of occurrence of potential vulnerabilities are assessed.

[Danfeng *et al.* 2009] prompted a service-based risk quantitative calculation method called SRQC (Service-based Risk Quantitative Calculation) for NGN (Next Generation Networks), which included a layered risk assessment model (quantification of assets, vulnerabilities, threats and risk). The calculation model centers on NGN services and considers the relationship between services. The authors claimed more valuable and objective results compared to the qualitative calculation. They proposed another three dimensional security architecture model dedicated to NGN networks and focused this time on quantifying threat, vulnerability, stability, and survivability parameters [Huang *et al.* 2009].

[Yongli *et al.* 2008] proposed an hybrid assessment method that combined qualitative analysis with quantitative computation. The method is assumed to be ISO/IEC 27002 standard [ISO 2005b] compliant and the authors used a multi-criteria decision making algorithm to weight the security parameters in the risk evaluation process. In the qualitative part of this research work, security experts have been asked to give different values to these weights, which bring us to the subjectivity and qualitative related issues.

[Ahmed *et al.* 2008] proposed a novel security metric framework combined with the NVD vulnerability database to identify and quantify objectively the most significant security risk factors: existing vulnerabilities, historical trend of known vulnerabilities, prediction of potential new vulnerabilities, and the associated impact severity. The authors made a considerable effort to model many specific risk parameters, however the risk has been considered individually and the network risk propagation has not been included in the global network risk evaluation.

[Lv 2009] proposed a quantitative risk evaluation method for network security. The method analyzes the process usually used by attackers to enter the network. The analysis was based on modeling attack activities and processes by tracking the transferring of safety states. Three key risk factors (assets, threats and vulnerabilities) have been identified and quantified. Especially, the attack probability indexes have been recurrently estimated by a layering approach based on the intruding process.

[Meng *et al.* 2009] proposed a risk evaluation method by formalizing and modeling attacks to find out how intruders proceed, and illustrated all the likely paths from origins to their targets. The work is quite similar to [Lv 2009] as the authors adopted also a graph-tree based attack modelization.

Some other quantitative studies used specific Bayesian network-based attack graphs to quantify the interdependency between existing vulnerabilities on each network node [Frigault & Wang 2008]. The authors proposed a probability model on attack graphs as a Bayesian network. The method provides the capabilities of using conditional probabilities to address the general cases of interdependency between vulnerabilities.

Authors in [Kondakci 2010] presented a causal model to encode logical relations inside the information system network. This contribution consisted in a probabilistic approach to model causal relationships among various threat sources and victim systems in order to facilitate quantitative and relational security assessment of information systems. The model is claimed to be context-free, meaning it makes no a-priori assumptions regarding the application domain.

An approach for detection, assessment and prevention of intrusions in a computer network has been presented in [Yau & Zhang 1999]. The approach uses audit data from multiple network nodes and services. To achieve accurate results, inherent security relations among different network nodes has been considered: an SDR (Security Dependency Relation) index has been defined to describe these relations, and a ripple effect analysis has been used to detect, assess, and prevent intrusions based on SDRs. Several agents have been also used to improve the scalability and efficiency of the assessment approach.

[Zhang *et al.* 2004] proposed a risk assessment approach for network information systems based on node correlation, and an enhanced NNC (Network Node Correlation) index has been introduced and based on the SDR index presented in [Yau & Zhang 1999]. Finally a quantitative taxonomy of network node has been provided to support the quantitative calculation.

Table 3.3 resumes the methods discussed above and how their authors managed to quantify the risk parameters. The first thing noticeable from this quantitative information security risk assessment approach survey is that the authors provided different methods and formal techniques for reaching generally the same goal: assessing as accurately as possible the risk related to network and information security.

However, these approaches perform this task in different manners (*e.g.* service-oriented, threat-oriented, dedicated to specific networks like NGNs, etc) and consequently have different coverage levels of the generic risk assessment process, which undeniably has an impact on the final risk scores. Each approach has its own positive points and weaknesses, depending from what perspective the user sees the network risk.

Table 3.3: Quantitative Risk Assessment Approaches Summary

Approach Reference	Quantitative assessment	Propagation oriented	Domain oriented	CVSS based
[Chen & Zheng 2006]	✓	×	×	×
[Danfeng *et al.* 2009]	✓	×	×	✓
[Huang *et al.* 2009]	✓	×	×	✓
[Yongli *et al.* 2008]	*H*	×	×	×
[Ahmed *et al.* 2008]	✓	×	×	✓
[Meng *et al.* 2009]	✓	×	×	×
[Lv 2009]	✓	×	×	×
[Frigault & Wang 2008]	✓	✓	×	×
[Kondakci 2010]	✓	✓	×	×
[Zhang *et al.* 2004]	✓	✓	×	×
[Yau & Zhang 1999]	✓	✓	×	×

Legend

✓: covered

×: not covered

H: hybrid approach (qualitative and quantitative analysis)

For sure, if we try to discuss these methods from a network security risk propagation point of view, there is a blatant lack of consideration for this particular issue. Even when risk propagation has been introduced in few work [Frigault & Wang 2008, Kondakci 2010, Zhang *et al.* 2004, Yau & Zhang 1999], some parameters such as the attack severity has been roughly quantified (or assessed using a qualitative technique). Most often, there is no indication on how the risk parameters are computed, which leads to some misunderstanding of the global assessment process. For instance, it has been pointed out in [Zhang *et al.* 2004] that likelihood of occurrence of threats is based on the analysis of attack frequency and complexity without any additional details. Also, the same authors claimed that attack impacts are evaluated based on analysis of severity results. They have chosen to let the administrators decide about the ranking of these parameters, which is totally subjective and at the opposite of quantitative risk assessment methods baselines.

Besides, network domain priority issues has been simply ignored. This is a surprising fact as far as several field of applications in information and network systems deploy network domains and sub-domains that do not have necessary the same priority level or security needs (*e.g.* public network domain vs. top secret or unclassified domain networks). A network domain is generally controlled by single operator or administrator authority who implements a uniform security policy within the domain. Hence, the level of security and the available security services are the sames for all the nodes belonging to that network domain. When a network is composed of an interconnection of network domains, whether in a small network area (*e.g.* LAN) or a wider network (*e.g.* WAN), multi-domain security should be considered. [Sher & Magedanz 2006] is an example of a network multi-domain security for IP multimedia subsystems.

Another example is the Internet where cross-domain services are constantly provided. In such a case, depending on the delimited perimeter of the risk assessment study, the network node and the carried information should likely have a different value, meaning that the impact of a given attack will not be the same. Obviously, the aeronautical field is sensitive to these network domain priorities and should benefit from a methodology that has enough considerations regarding this particular need.

In order to address these issues (full quantitative assessment, network domain priority, risk propagation), we present a new quantitative methodology based on network security risk propagation in the next subsection. To allow a better integration of aeronautical network domain considerations throughout our approach, we take these concepts into account at an early phase of the methodology design (see section 3.4).

3.3 A Quantitative Network Risk Assessment Model based on Risk Propagation

3.3.1 Quantifying Methodology Parameters

In this section, we explain how we compute every involved network parameter in the risk assessment process. First, we clarify how the risk is decomposed in our model, then we present every parameter and its respective formula.

3.3.1.1 Network Risk Decomposition

In our methodology, there are 4 types of risks, namely:

1. **The risk per node** is computed for each node depending on its own vulnerabilities and its connections with correlated nodes. As a node is connected to other nodes in the network, we evaluate the total risk for a given node i as the product of node value $Value_i$ and the sum of its individual and propagated risks (respectively denoted $Risk_i^+$ and $Risk_i^-$):

$$Risk_i = Value_i * (Risk_i^+ + Risk_i^-) \tag{3.2}$$

2. **The individual risk** is the intrinsic risk computed for each node, meaning it takes into account only the vulnerabilities associated with the node itself. The individual risk $Risk_i^+$ is computed as the sum on the number of existing vulnerabilities T_i of the product between the likelihood of occurrence of a threat $P_t(i)$ and its impact $I_t(i)$, which is fully compliant with the basic expression of the risk mentioned in formula 3.1:

$$Risk_i^- = \sum_{t=0}^{T_i} P_t(i) * I_t(i) \tag{3.3}$$

3. **The propagated risk** is the risk inherited from the dependency between correlated nodes (*e.g.* data flow exchanges, client-server architectures, etc). The propagated risk $Risk_i^+$, is estimated as the following:

$$Risk_i^+ = \sum_{j=0}^{n_i} \sum_{t=0}^{T_j} P_t(i,j) * I_t(i,j) \tag{3.4}$$

   Compared to equation 3.3, the idea is quite the same except the difference that the propagated likelihood $P_t(i,j)$ and the propagated impact $I_t(i,j)$ are induced by all the vulnerable nodes connected with node i (and denoted n_i).

4. **The network risk** is the total risk computed for all the nodes composing the network. It is calculated as the sum of all the risks relevant to each node in the network (where n denotes the total number of hosts on the network):

$$Risk_{net} = \sum_{i=0}^{n} Risk_i \tag{3.5}$$

In the next section, we explain how the node value used in the formula 3.2 is computed.

3.3.1.2 Node Value

Considering that network nodes have not the same functionalities, we can assume their degree of importance in the network may vary. Thus, the value of a node i is given by:

$$Value_i = n_i * FunctionValue_i \tag{3.6}$$

Indeed, we take into account the node functionality $FunctionValue_i$, which reflects the importance of a host from a security point of view. For instance, it is clear that a firewall is more critical for the security of the network than a user terminal. Besides, the value of a node increases when it is connected to a large number of nodes (*i.e.* when the node correlation is high). For instance, server nodes (*e.g.* email servers, web servers, DNS servers, etc) or proxies are highly connected to other nodes in the network.

This node value expression could be sufficient if the risk assessment is dedicated to an intra-network domain perimeter where only the node functionality may vary from a host to another one. Nevertheless, as presented in equation 3.6, it is certainly unsatisfactory if we want to introduce the priority between network domains as discussed in section 3.2.3.

3.3.1.3 Enhanced Node Value

In order to introduce the priority between network domains, we slightly modify the equation 3.6:

$$Value_i = n_i * FunctionValue_i * ClassValue_i \tag{3.7}$$

where $ClassValue_i$ expresses the class value of a node, depending on the network domain it belongs to. Both node function and class values are the only parameters requiring a *"human in the loop"* since there are no means to quantify them in practice. It is usually the duty of the administrator to assign these values depending on his needs and objectives. For instance, if the monetary value is an important parameter in the risk evaluation process, he could instantiate the function values to the cost of each host. For such a purpose, the risk assessment tool must give enough freedom to the user that he can assign the node values as he wants before initiating the evaluation process. In our model, both functions and class values are ranged between 0.0 and 1.0 (which is an arbitrary choice and could be modified as we said). For the aeronautical case studies, we instantiate $ClassValue_i$ and $FunctionValue_i$ in section 3.4 for the specific needs of FAST and SESAR projects.

In the following section, we focus on quantifying the individual and propagated impact of threat on network nodes.

3.3.1.4 Impact of Threats

As seen in equations 3.3 and 3.4, we actually define two types of impact of threats:

1. $I_t(i)$ is the impact of threat caused by the exploit of a specific vulnerability t to the node i. For each node, the estimated impact $I_t(i)$ is retrieved from the CVE public vulnerability database. To be more specific, $I_t(i)$ is referred to the CVSS severity (*i.e.* impact) score associated to each vulnerability occurring on that node, which is basically a numerical score ranged between 1.0 and 10.0. Practically, this information can be obtained by accessing manually the CVE database or by scanning the network using the NESSUS[12] vulnerability scanning tool: as far NESSUS is connected to the CVE database, the results remain the same. The number of vulnerabilities T_i, used in both equations 3.3 and 3.4, is a simple addition on the existing vulnerabilities for that node;

2. $I_t(i,j)$ is the propagated impact of a threat caused by the exploit of a vulnerability t from a node j to a node i:

$$I_t(i,j) = Value_i * I_t(j) * \sigma(i,s) \tag{3.8}$$

   The propagated impact depends on the affected node value, namely $Value_i$ (as expressed in formula 3.7), the impact of t on the issuing node j (*cf.* CVSS score as we defined above), and the targeted service s. $\sigma(i,s)$ is a scalar value deduced as the following:

$$\sigma(i,s) = SPV_i * (SOV_s)^{tr} \tag{3.9}$$

   where SPV_i is a binary *Security Protection Vector* that defines the security features (namely the security services introduced in section 2.2.1.1 of chapter 2) provided by security mechanisms and countermeasures used to protect the node i. The dimension of the vector depends on how many security services are provided. For instance, if confidentiality, integrity, and authentication are considered, SPV_i would be a 3-dimension vector. Moreover, let's assume that only confidentiality is provided for the data flows issued from the node i, the associated binary indicator vector will be equal to [0 1 1]. It could seem meaningless to associate the zero binary value to express a *"YES"*, but this is done in order to respect the impact function behavior. Indeed, the impact grows when less security features are available. Thus, mapping the one binary value to a *"YES"* is inadequate. In this specific case, the more security features is, the bigger would be the propagated impact, which would not be logical regarding the impact function previously defined.

   The second part of the equation 3.9 is the *Security Objective Vector* SOV_s that defines the security objectives per service (note that the transpose of the vector is used here in order to obtain a scalar result). For instance, if we take again the example of a service s provided by the node i where high security objectives are expressed for the the confidentiality, integrity, and authentication security services, the SOV_s vector could be equal to [5 5 5], where the value 5 expresses the highest security requirement. These values depend on the security objectives expressed beforehand at a previous step of the ISSRM process (see step (b) of figure 3.1).

   In section 3.4, we instantiate the SOV_s vector according to the security objectives expressed in the COCR document for the aeronautical operational services.

In the following section, we explain how to quantify the likelihood of occurrence and propagation of threats.

[12] Note that any scanning tool providing the same features as NESSUS can be used.

3.3.1.5 Likelihood of Threats

Similarly to the impact of threats, we define two types of likelihood of threats:

1. $P_t(i)$ is the likelihood of occurrence of a threat caused by the exploit of a specific vulnerability t to the node i. $P_t(i)$ represents the possibility that an attack associated with a specific vulnerability t is conducted. The likelihood of occurrence evaluation is driven by the TYPHON (Telecommunications and Internet Protocol Harmonization Over Networks) threat analysis methodology [ETSI 2003] proposed by the ETSI. However, as the likelihood values are qualitative, we extend this part of the existing ETSI methodology in order to quantify the involved parameters. Indeed, as described in [ETSI 2003], the evaluation of the likelihood is based on two behavioral factors:

   ◇ The technical difficulties that an attacker may face in order to achieve his goal;

   ◇ The motivation for an attacker to carry out a given attack.

   As it appears in [ETSI 2003], the likelihood of threats can be:

   (a) *Unlikely*: if the motivation for conducting an attack is low (*e.g.* no financial interest or technical challenges) and there are strong technical difficulties to overcome (*e.g.* insufficient knowledge to conduct the attack);

   (b) *Possible*: if the motivation is moderate (*e.g.* reasonable financial gains) and the technical difficulties are solvable (*e.g.* information required to exploit the vulnerability are available);

   (c) *Likely*: if there is a high attacker motivation (*e.g.* inducing a denial of service on the network, important financial gains) and technical difficulties are almost nonexistent (*e.g.* no security protection).

   Despite the fact that these behavioral factors seem adapted and should be logically considered to evaluate the likelihood associated to an attack, they can not be used directly in our model as we need quantitative values. Besides, the ETSI methodology does not explain how the technical difficulties and motivation factors are combined together in order to deduce the likelihood of occurrence of a threat. To address these issues, we brought some modifications in order to use these behavioral factors in our risk assessment process. First, the likelihood is computed using the motivation and technical difficulties values (respectively denoted $Motivation_t(i)$ and $TechnicalDifficulty_t(i)$) as shown in the following formula:

$$P_t(i) = \frac{Motivation_t(i)}{TechnicalDifficulty_t(i)} \tag{3.10}$$

   In fact, $P_t(i)$ should increase when the motivation gets higher; otherwise, $P_t(i)$ decreases when the technical difficulties that must be overcome increase. The motivation for an attacker to exploit a vulnerability t on a node i is:

$$Motivation_t(i) = Value_i * T_i \tag{3.11}$$

   Equation 3.11 shows that the motivation increases as the node value or the number of known vulnerabilities increases. Technical difficulties becomes more significant when security features (*e.g.* Firewalls) are reinforced (*e.g.* increasing their number or enhancing the security policies) or the amount of information required to exploit a vulnerability t is high:

$$TechnicalDifficulty_t(i) = S_i + B_t \tag{3.12}$$

In equation 3.12, S_i expresses the number of security mechanisms used to protect the node i. B_t is the amount of elementary information needed by an attacker to exploit the vulnerability t. We assume that $B_t > 0$, meaning that at least one elementary information has to be available in order to conduct an attack exploiting t. Indeed, the attacker will be probably unable to exploit a vulnerability if a minimum of data is not available to start the attacking process (*e.g.* opened port IDs, user logins, target addresses, etc). As the resulting probability value must be ranged between 0.0 and 1.0, both motivation and technical difficulties values are normalized between 0.0 and 1.0;

2. $P_t(i,j)$ is the likelihood of propagation of a threat caused by the exploit of a vulnerability t on a node j to node i given by:

$$P_t(i,j) = P_t(j) * P(i,j) \tag{3.13}$$

In fact, the propagation likelihood depends on the likelihood of vulnerability t on the issuing node j and the likelihood of correlation $P(i,j)$ between the two nodes, given by:

$$P(i,j) = \frac{f_{ij}}{F_{ij}} \tag{3.14}$$

$P(i,j)$ depends on the ratio between the number of detected and total data flows exchanged (*i.e.* the sum of all detected data flows) between two nodes i and j and relative to the attacked service. Practically, f_{ij} and F_{ij} can be directly deduced using some network statistic tools like NETSTAT[13] or raw data from the */proc/net/dev* repository.

Finally, since all the parameters involved in the network risk computation process have been defined (and resumed in table 3.4), the risk assessment algorithm is presented in the next subsection. Each step of the overall network risk assessment process is explained regarding the relevant speudo-code algorithm.

3.3.2 Network Security Risk Assessment Process

In this section, we describe the 6 steps leading to the final network risk evaluation using our approach. In each step, we give the corresponding pseudo-code algorithm:

1. *Initiation step*: first, we begin simply by initiating a set of vulnerable nodes and a set of processed nodes to null. All risk values (individual, propagated, and node risks) are initiated for each node in the network. Also, we create a set of correlated nodes for each node;

Algorithm 3.1 Variables Initiation

```
$V \leftarrow \{\varnothing\}$; //initiate a set of vulnerable nodes
$NV \leftarrow \{\varnothing\}$; //initiate a set of processed nodes
for all $i \in network$ do
    $Risk_i^- \leftarrow 0$;
    $Risk_i^+ \leftarrow 0$;
    $Risk_i \leftarrow 0$;
    $C_i \leftarrow \{\varnothing\}$; //initiate a set of correlated nodes with node i
end for
```

[13] http://linux-ip.net/html/tools-netstat.html

Table 3.4: Risk Parameter Notations

Notation	**Description**
$Risk_i$	Node risk evaluated on node i
$Risk_i^-$	Individual risk evaluated on node i
$Risk_i^+$	Propagated risk evaluated on node i
$Risk_{net}$	Network risk
$Value_i$	Value of node i
$FunctionValue_i$	Function value of node i
$ClassValue_i$	Class value of node i
t	An exploitable vulnerability
n	Total number of nodes in the network
n_i	Number of nodes connected with node i
T_i	Number of vulnerabilities exploitable on node i
S_i	Number of security features deployed to protect i
B_t	Number of information needed to exploit t
$P_t(i)$	Likelihood of occurrence of a threat exploiting t
$I_t(i)$	Impact of threat exploiting t
$P_t(i,j)$	Likelihood of propagation of a threat exploiting t
$I_t(i,j)$	Propagated impact of a threat exploiting t
f_{ij}	Number of flows detected between nodes i and j
F_{ij}	Total number of flows exchanged between i and j
$Motivation_t(i)$	Motivation of an attacker to exploit t
$TechnicalDifficulty_t(i)$	Technical difficulty level to exploit t
$\sigma(i,s)$	Scalar value as the product of SPV_i and SOV_s
SPV_i	Security protection vector for the node i
SOV_s	Security objective vector for the service s

2. *Scan and identify vulnerable nodes*: the second step is to identify all the vulnerabilities specific to each node in the network. If any vulnerability is detected, the node is marked as vulnerable and added to V. The node values are also computed at this step and the corresponding CVSS severity scores are stored for a later use in the algorithm (see algorithms 3.3 and 3.4);

Algorithm 3.2 Scan and identify vulnerable nodes

```
for all $i \in network$ do
    identify vulnerabilities;
    $Value_i \leftarrow n_i * FunctionValue_i * ClassValue_i$;
    if any vulnerability is detected then
        add node $i$ to $V$;
    end if
    for all vulnerability t do
        store $t$ and associated $CVSS$ score;
    end for
end for
```

3. *Compute the individual risk for each vulnerable node*: for each vulnerable node identified in the previous step, we keep track of the nodes that could be impacted by the network correlation. The obtained set of correlated nodes will be used in the next step. Then for each vulnerable node, we

compute its individual risk according to the formula 3.3;

Algorithm 3.3 Compute the individual risk for each vulnerable node

```
for all node $i \in V$ do
    store correlated nodes with node $i$ in $C_i$;
    for all vulnerability $t$ do
        $TechnicalDifficulty_t(i) \leftarrow S_i + B_t$;
        $Motivation_t(i) \leftarrow Value_i * T_i$;
        $P_t(i) \leftarrow Motivation_t(i)/TechnicalDifficulty_t(i)$;
        $Risk_i^- \leftarrow Risk_i^- + (P_t(i) * I_t(i))$;
    end for
end for
```

4. *Compute the propagation risk for nodes correlated with vulnerable nodes*: for each vulnerable node, we take the correlated nodes one by one, and increment the propagated risk of the targeted node. The infected node is then tested: if it was not considered as vulnerable, we update the set of vulnerable nodes and the likelihood of occurrence of threat is updated;

Algorithm 3.4 Compute the propagation risk for nodes correlated with vulnerable nodes

```
while $V \neq \{\varnothing\}$ do
    for all $j \in V$ do
        for all $i \in C_j$ do
            for all vulnerability $t$ do
                $\sigma(i, s) \leftarrow SPV_i * (SOV_s)^{tr}$;
                $I_t(i, j) \leftarrow Value_i * \sigma(i, s) * I_t(j)$; //s is the targeted service by t
                $P(i, j) \leftarrow f_{ij}/F_{ij}$;
                $TechnicalDifficulty_t(j) \leftarrow S_j + B_t$;
                $Motivation_t(j) \leftarrow Value_j * T_j$;
                $P_t(i) \leftarrow Motivation_t(i)/TechnicalDifficulty_t(i)$;
                $P_t(i, j) \leftarrow P_t(j) * P(i, j)$;
                $Risk_i^+ \leftarrow Risk_i^+ + (P_t(i, j) * I_t(i, j))$;
                $P_t(i) \leftarrow P_t(i) + P_t(i, j)$; //update the likelihood of threat
                if $P_t(i) > 1$ then
                    $P_t(i) \leftarrow 1$; //the likelihood of threat should not exceed 1
                end if
            end for
            if node $i \notin V and \notin NV$ then
                store node $i$ in $V$; //the node is now vulnerable
            end if
        end for
        copy node $j$ to $NV$ and remove it from $V$; //this node has been processed
    end for
end while
```

5. *Compute the risk for each node in the network*: at this phase, we take the output of algorithms 3.3 and 3.4 and calculate the risk per node according to the expression 3.2;

Algorithm 3.5 Compute the total risk for each node in the network

```
for all node $i \in network$ do
    $Risk_i \leftarrow Value_i * (Risk_i^- + Risk_i^+)$;
end for
```

6. *Compute the whole network risk level*: finally, we estimate the total network risk according to formula 3.5;

Algorithm 3.6 Compute the whole network risk level

```
for all node $i \in network$ do
    $Risk_{net} \leftarrow Risk_{net} + Risk_i$;
end for
```

In order to test and validate our approach, we apply the risk assessment methodology to the specific aeronautical FAST and SESAR case studies in the following section.

3.4 Aeronautical Network Case Studies

3.4.1 Aeronautical Node Values Instantiation

In order to instantiate the $ClassValue_i$ and $FunctionValue_i$ parameters for the aeronautical context, the following decomposition is considered:

- ◇ 5 function nodes with their respective values as shown in table 3.5:

Table 3.5: Node Function Values

	Firewall Gateway	**Router**	**Switch Hub**	**Server**	**Terminal**
Function value	1.0	0.7	0.5	0.3	0.1

As mentioned in section 3.3.1.2, the node function values point out the importance of the node functionalities from a security point of view. In the studied topologies, Firewall and Gateway are the most important security nodes because they provide security capabilities to protect the whole network, then they have been given the maximum *FunctionValue* (*i.e.* equal to 1.0). Routers come in a second position because they provide essential features to the operability of the network (*e.g.* routing, packet filtering). Switches and hubs are obviously less significant than routers (*i.e.* *FunctionValue*=0.5) but remain more important than servers and terminals (that have respectively 0.3 and 0.1 as a *functionValue*) because they support several advanced features that might be relevant to the security of the network (*e.g.* MAC address filtering, flow control).

- ◇ In order to be compliant with the ICAO SARPs, we respected the priorities between the 4 network domains as shown in table 3.6:

Table 3.6: Aeronautical Class Values

	ATS	**AOC**	**AOC NG**	**APC**
Class value	1.0	0.7	0.4	0.1

As for the function values indicated in table 3.5, the choices made for the aeronautical class values shown in table 3.6 are driven by security considerations: the more the aeronautical class is high, the more the corresponding class value increases (aeronautical classes have been discussed in chapter 2). This said, ATS is the most critical traffic class, then it has the maximum *ClassValue* (*i.e.* equal to 1.0). Even if AOC and AOC NG are both relevant to airline services, AOC should have a higher *ClassValue* simply because this traffic class refers to some operational services that are critical to the safety of the flight. For instance, NOTAM (Notice to Airmen) service provides information to alert the flight crew about abnormal events such as closed runways or inoperable radio navigation aids). Finally, APC traffic class gets the lowest *ClassValue* (*i.e.* equal to 0.1).

The matrix resulting from the combination of these function and class values is:

Table 3.7: Node Values for Aeronautical Network Domains

	ATS	**AOC**	**AOC NG**	**APC**
Firewall - Gateway	1.0	0.7	0.4	0.1
Router	0.7	0.49	0.28	0.07
Switch - Hub	0.5	0.35	0.2	0.05
Server	0.3	0.21	0.12	0.03
Terminal	0.1	0.07	0.04	0.01

The indicated values in table 3.7 do not integrate the number of connected nodes (namely n_i) as required in the formula 3.7 because this parameter depends on the considered network topology. However, we will integrate the specific n_i in the value node computation depending on the network topology scenario we want to test.

3.4.2 Aeronautical Services Instantiation

The *Security Objective Vector* SOV_s is deduced from the COCR document where security objectives are expressed for three security services (confidentiality, integrity, and availability) using a qualitative scale. In order to compute the network risk and get quantitative values, we perform the following mapping:

Table 3.8: From Qualitative to Quantitative COCR Values Mapping

Qualitative Values	**Quantitative Values**
None	0
Low	1
Medium	2
High	3
High-Severe	4
High-Catastrophic	5

For instance, if the service s is the ATS data-based FLIPCY (Flight Plan Consistency) service, the corresponding *Security Objective Vector* is equal to SOV_{FLIPCY}=[1 4 2] as indicated by the COCR document.

3.4.3 Testbed Infrastructure

In this section, we introduce the testbed platform used to validate the risk assessment algorithm detailed previously. Figure 3.5 illustrates the emulated risk assessment framework. It has been based on the emulation MARIONNET tool [Loddo & L.Saiu 2008] to run different virtual machines for each network node.

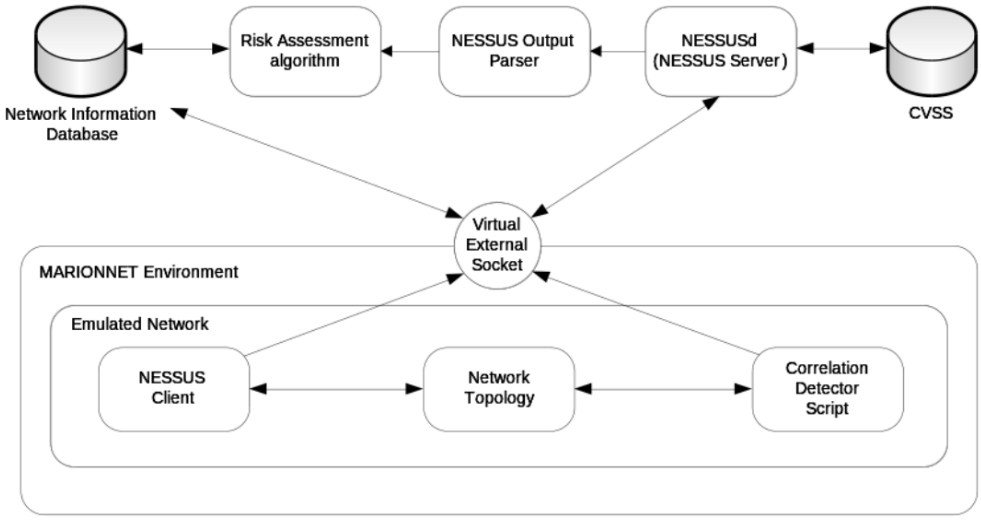


Figure 3.5: Risk Assessment framework using MARIONNET

MARIONNET allows users to configure and run complex networks without the need for physical devices. The vulnerability scanning tool NESSUS has been installed and configured in a client mode on each node. A NESSUS daemon (the server part) has been also configured on the physical terminal (basically, this is the host where the entire software framework is installed) and used to emulate the network. When the risk assessment algorithm starts, the NESSUS daemon scans every node to get the set of individual vulnerabilities that are detected (*cf.* step 3.2 of the risk assessment process).

As far as NESSUS is connected with the NVD vulnerability database, such a platform has the advantage to detect new vulnerabilities if some are newly added to the database and to provide an automated risk assessment tool. The communication between the client and the server parts is made possible thanks to a special virtualized item in MARIONNET called Virtual External Socket. The external socket device provides opportunities to connect virtual nodes to non-virtual networks (Internet for instance). This device allows us to transfer information from the emulated network to the physical terminal. As we used this risk assessment framework for aeronautical network purposes, we identified each aeronautical class using a different network IP address pool. A correlation detector script has been written and distributed over the network to gather correlation information for each node. NESSUS outputs are parsed and all collected network data are then stored in a Network Information Database (IP addresses, CVSS scores, correlation information, etc). Finally, the risk assessment algorithm accesses this database to compute and display information on the network risk level.

In the next section, we discuss the behavior of the risk assessment methodology regarding the network topologies relevant to the FAST project.

3.4.4 Case Study 1: Air-Ground Network in the FAST Project

3.4.4.1 Network Architectures

Figure 3.6 depicts the network scenarios used in the FAST project. The red framed nodes are the security features added to the nominal network scenario (*i.e.* without any security mechanisms). The left-hand side shows the aircraft on-board system: a satellite terminal is connected to an ATN router for ATS cockpit services, where an FMS (Flight Management System)[14] is considered as an ATS client host. A New Generation (NG) router is connected to passenger laptops, an EFB (Electronic Flight Bag)[15] device for AOC, the telemedicine workstation, and a video surveillance terminal. The right-hand side represents the ground segment where a satellite gateway is connected to an ATN router, an Internet Router, and four servers used to deliver services to the on-board clients. Nodes FW_1 and FW_2 are two Firewalls and nodes SMP_1 and SMP_2 are two Security Manager (SecMan) proxies, located in DMZs, and deployed to establish secure communications with ground entities.

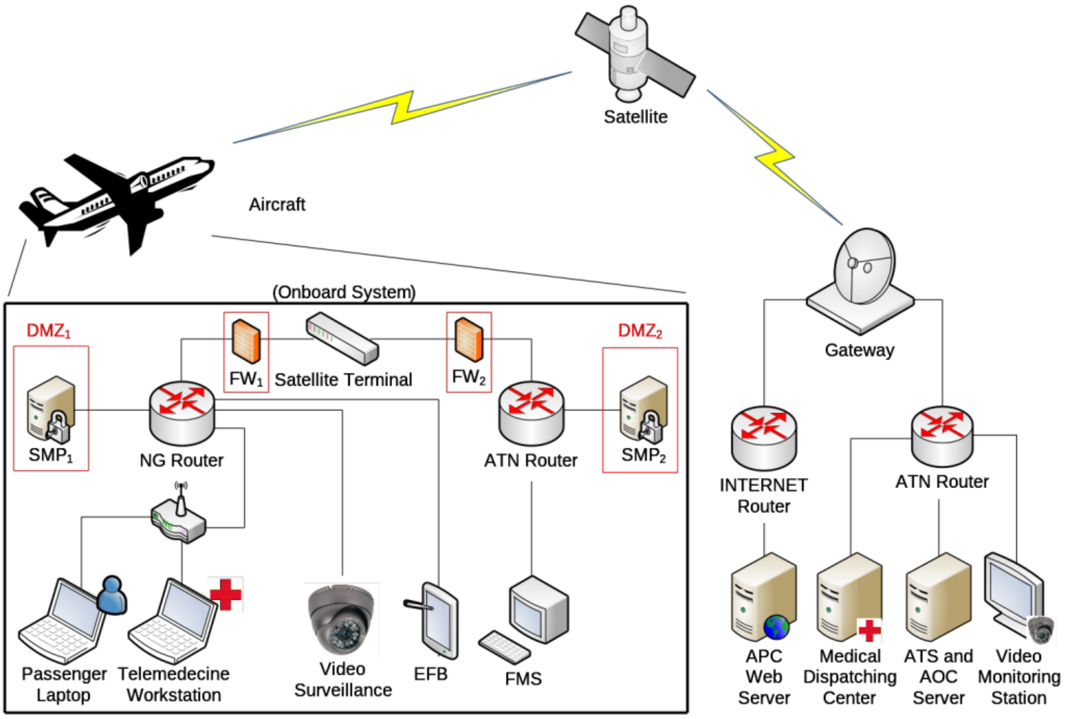


Figure 3.6: Nominal and Secure Network Architecture Scenarios

More details about the interconnection policies and the SecMan proxies are given in the next chapter dedicated to the adaptive security infrastructure we developed in the scope of the FAST project.

3.4.4.2 Experimental Results

Embedded Nodes

Since security improvements have been made for the on-board part, we focus first our analysis on the embedded nodes into the aircraft. Statistics on vulnerabilities and CVSS scores show that the total

[14] The FMS is the equivalent of the DCDU interface for the CPDLC messages illustrated in 2.1.

[15] EFB is a display system used to perform AOC flight management tasks and intended to replace crew paper-based document.

number of vulnerabilities per network node has slightly changed since the intrinsic vulnerabilities (at a node basis) of security devices added in the second scenario have few exploitable vulnerabilities.

Figures 3.7 and 3.8 show the CVSS score distribution for both network topologies:

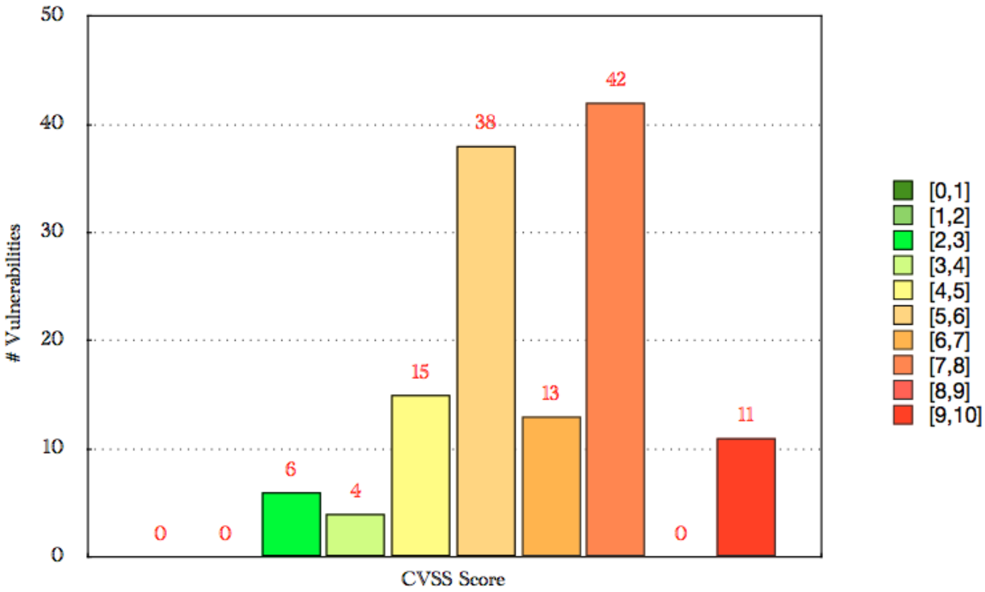


Figure 3.7: CVSS Score Distribution for the Nominal Architecture

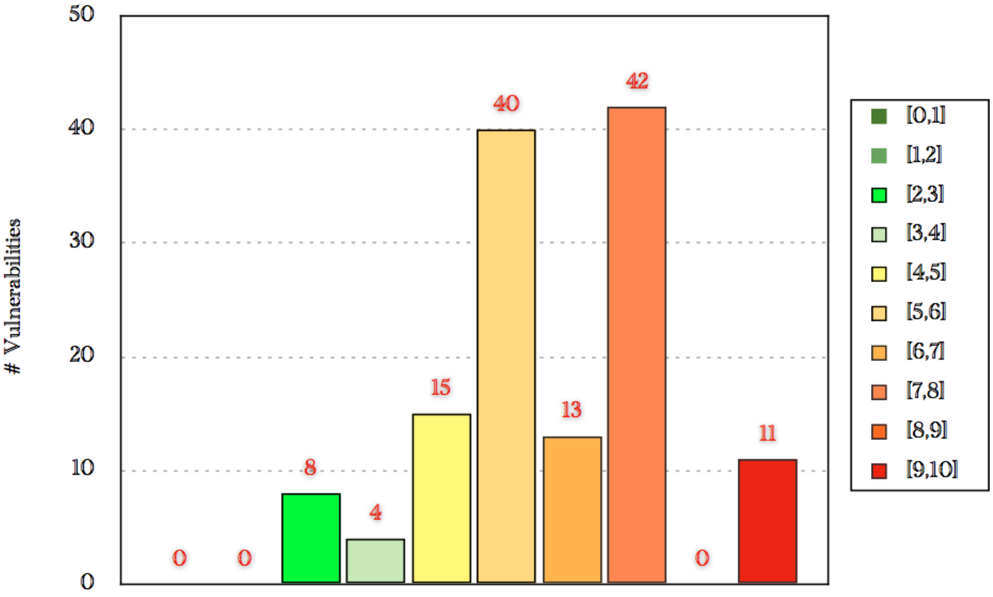


Figure 3.8: CVSS Score Distribution for the Secure Architecture

The different assessed risks on both nominal and secure aeronautical network architectures are resumed in table 3.9. As expected, the total network risk is higher in the nominal architecture than in the secure architecture: the network risk value is 8.959 in the first scenario against 6.312 in the second scenario and the difference is equal to 29.54%:

Table 3.9: Onboard Risk values for Nominal and Secure FAST Architectures

	Nominal Architecture		**Secure Architecture**	
	$Risk_i^-$	$Risk_i^+$	$Risk_i^-$	$Risk_i^+$
NG Router	0.910	2.205	0.758	1.330
ATN on-board Router	0.650	0.588	0.121	0.007
Satellite terminal	0.047	0.033	0.028	0.020
FMS	0.023	0.875	0.007	0.844
EFB	0.114	0.022	0.093	0.002
Video Surveillance	0.011	0.062	0.001	0.052
Telemedecine workstation	0.125	0.078	0.114	0.068
Passenger Laptop	0.701	1.554	0.690	1.543
Full Network	**8.959**		**6.312**	

Figures 3.9 and 3.10 show the individual and propagated risks of the embedded nodes under another perspectives.

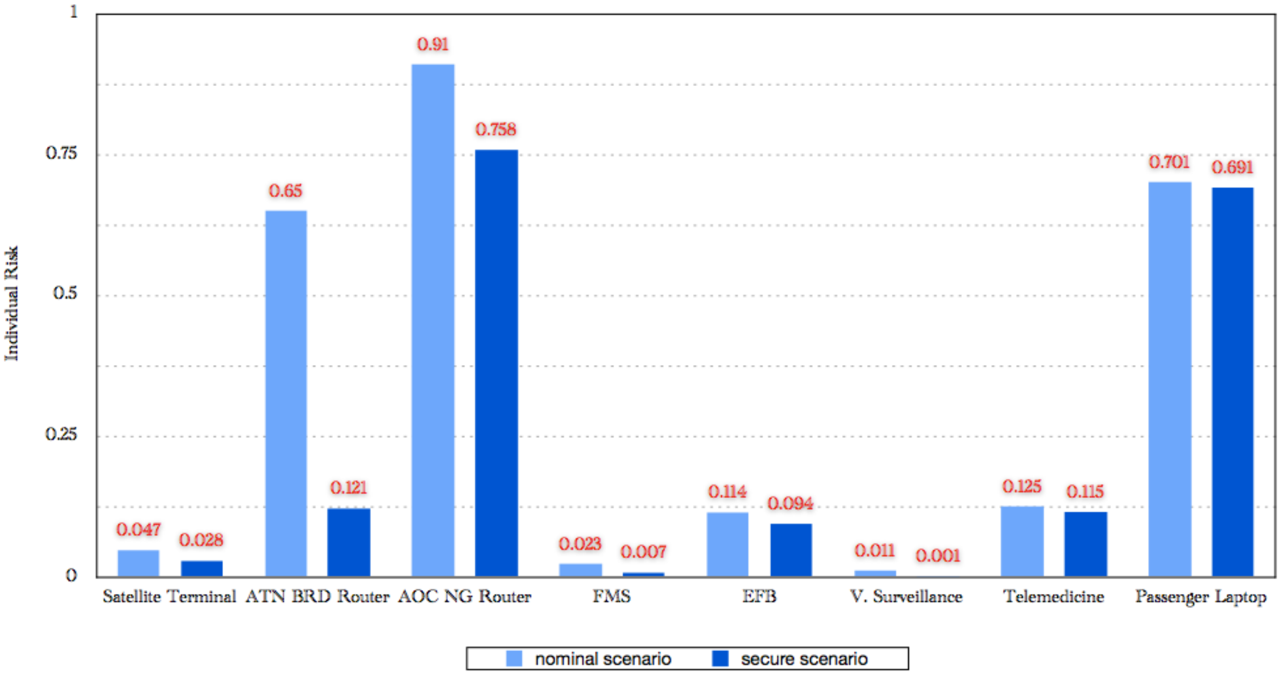


Figure 3.9: Individual Risks for Onboard Nodes

The improvement noticed in the second scenario is mainly due to the decrease of individual risk values $Risk_i^-$. In fact, individual node risks remain nearly the sames except for nodes directly connected to the security devices added in the second scenario. This decrease is due to the fact that threat likelihoods are affected by the number of security protections for a concerned node as expressed in formula 3.12. For instance, the individual risk of the NG router is assessed to 0.910 and 0.758 respectively in the nominal and the secure architectures, which represents a 16.66% decrease.

Moreover, adding four network devices (two Firewalls and two security proxies) has not increased the propagated risk of the other nodes as much as expected. Indeed, the firewalls considered in our simulations (Netfilter Iptables) have only two vulnerabilities with low CVSS scores (2.1 and 5 respectively for vulnerability CVE-2001-1388 and CVE-2001-1387). Besides, since the security manager proxies SMP_1

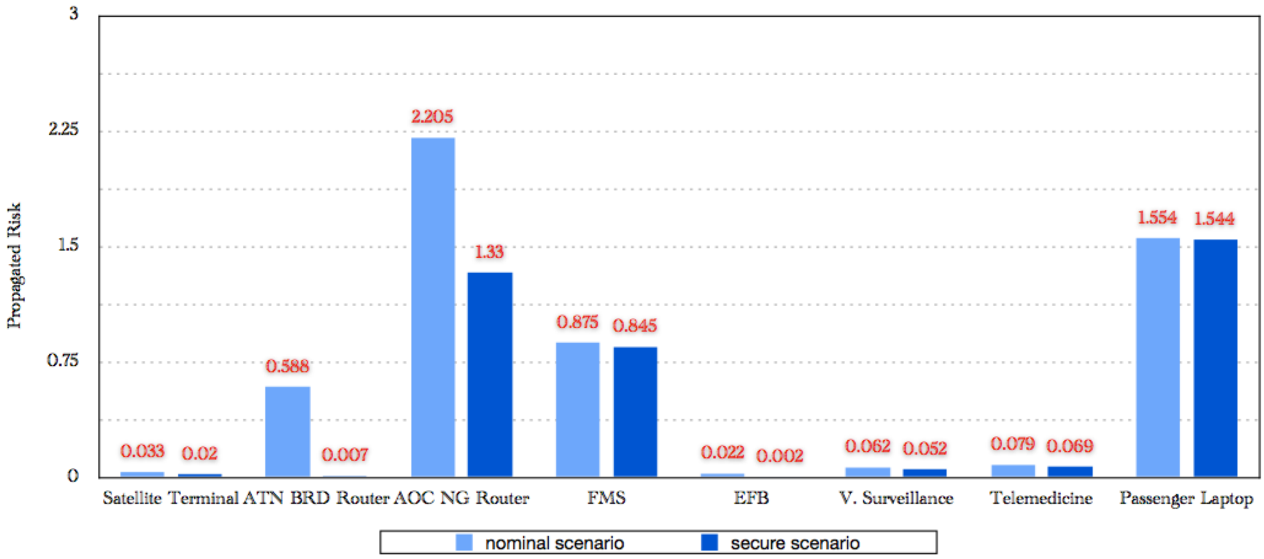


Figure 3.10: Propagated Risks for Onboard Nodes

and SMP_2 are only used for security purposes, traditional services (*i.e.* not related to security) have been turned off. Consequently, both SMP_1 and SMP_2 are considered as safe nodes (*i.e.* no intrinsic vulnerabilities on them). This made the propagated risk values of nodes such as the satellite terminal or on-board routers (ATN and NG) lower than the one in the first scenario. For instance, the propagated risk for the ATN router has been evaluated respectively to 0.588 and 0.007 in the nominal and the secure architectures.

Furthermore, these security devices have been isolated in DMZs and users have been forced to go through the SMPs in order to communicate with the ground servers. As this affects the volume of the traffic (number of data flows in equation 3.14) exchanged between intermediate and end systems, the propagation likelihood between vulnerable nodes has considerably decreased. For instance, the propagation likelihood between the NG router and the passenger laptop node has decreased by 32.4% if we compare the two scenarios (practically, the propagation likelihood has been evaluated respectively to 0.540 and 0.365 in the nominal and the secure architectures).

Ground Nodes

Table 3.10 illustrates the quantitative individual and propagated risk values obtained for ground nodes (these results are presented also in figures 3.11 and 3.12). As we can see, the propagation risk of the satellite gateway remains high in both scenarios and is not really impacted by the security features added in the secure architecture since these are dedicated to the on-board embedded part. The satellite gateway being connected to 3 critical nodes (two ground routers and the satellite terminal), we tried to identify how it is possible to reduce its node risk value. For this purpose, we specially focused our attention on the APC server case which is by far, the most vulnerable node in both nominal and secure architectures: 21 vulnerabilities have found in the NVD database and the individual risk has been assessed to 2.454 (which is the highest value among all the network nodes connected to the satellite gateway).

Table 3.10: Ground Risk values for Nominal and Secure FAST Architectures

	Nominal Architecture		**Secure Architecture**	
	$Risk_i^-$	$Risk_i^+$	$Risk_i^-$	$Risk_i^+$
Gateway	1.954	1.875	1.789	0.451
Internet Router	0.222	1.214	0.178	1.170
ATN Router	0.578	0.821	0.355	0.112
Video Monitor. station	0.054	0.078	0.017	0.041
ATC/AOC Server	0.454	0.087	0.357	0.009
Medical Dispatch. Center	0.019	0.152	0.017	0.115
APC Web Server	2.454	0.065	2.447	0.058

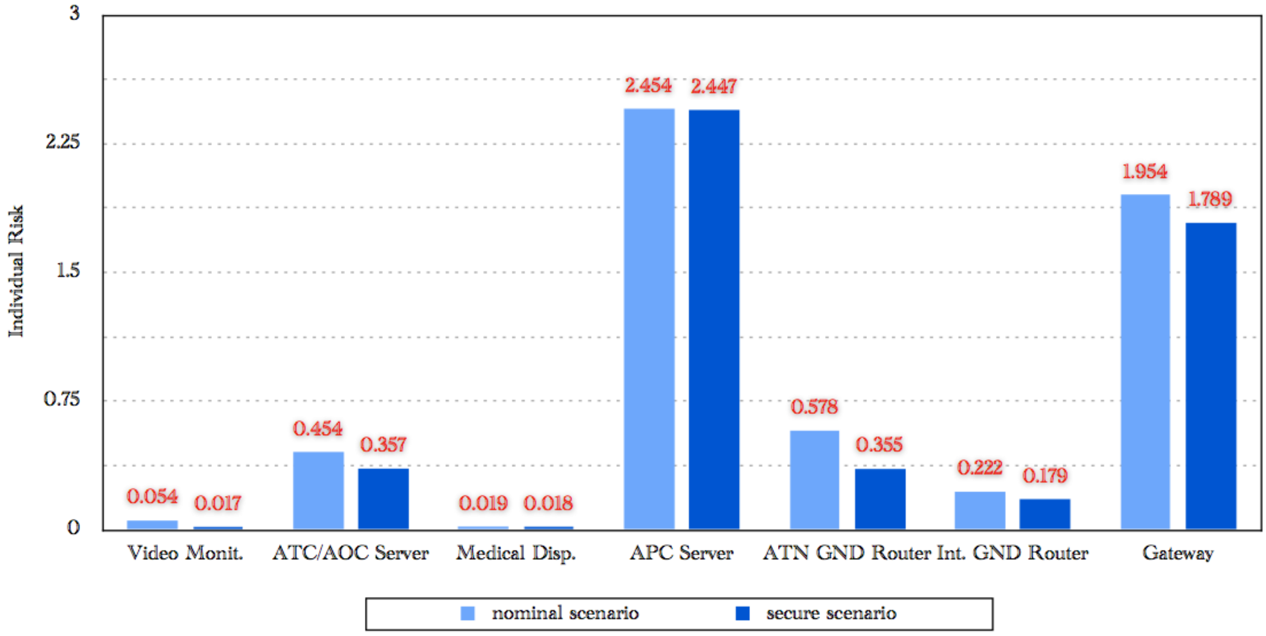


Figure 3.11: Individual Risks for ground Nodes

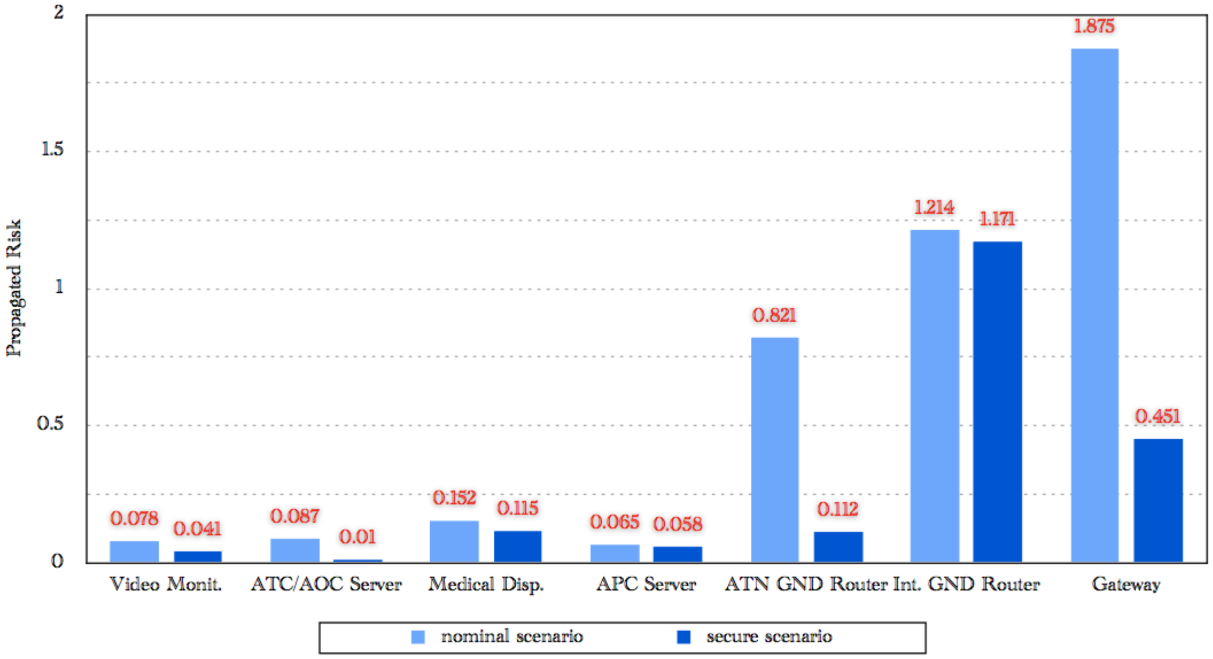


Figure 3.12: Propagated Risks for ground Nodes

In order to find out how the APC server impacts the gateway node risk, we managed to estimate the propagated risk of the satellite gateway using different web server implementations as shown in table 3.11:

Table 3.11: CVSS Score and Vulnerability Statistics for the APC Web Server

Web Server Type	**# Vulnerabilities**	**Average CVSS Score**
Apache Tomcat HTTP Server	89	5.6
SUN Java Web Server	33	6.7
Lighttpd Web Server	21	6.3
Microsoft IIS Web Server	17	7.4
Zeus Web Server	4	6

The results for these different web servers show that there is no linearity between the number of vulnerabilities and the average CVSS scores: a high number of vulnerabilities does not mean necessarily a high average CVSS score, meanwhile a low number of vulnerabilities does not always generates a low CVSS score. As an example, the Apache HTTP server has the biggest number of vulnerabilities (89 in total) in the NVD database (which is quite normal since it is the most used web server worldwide) but, compared to the Zeus web server (which has the lowest total of vulnerabilities), it has a lower average CVSS score (5.6 against 6 respectively for the Apache and Zeus web servers). Even if these two parameters are important in the risk analysis and should be considered together for later guidance and implementation considerations, the number of vulnerabilities seems to have a larger impact on final results since it clearly varies from a web server to another.

Figure 3.13 shows the propagated risk of the gateway as a function of web server vulnerabilities in both network topologies:

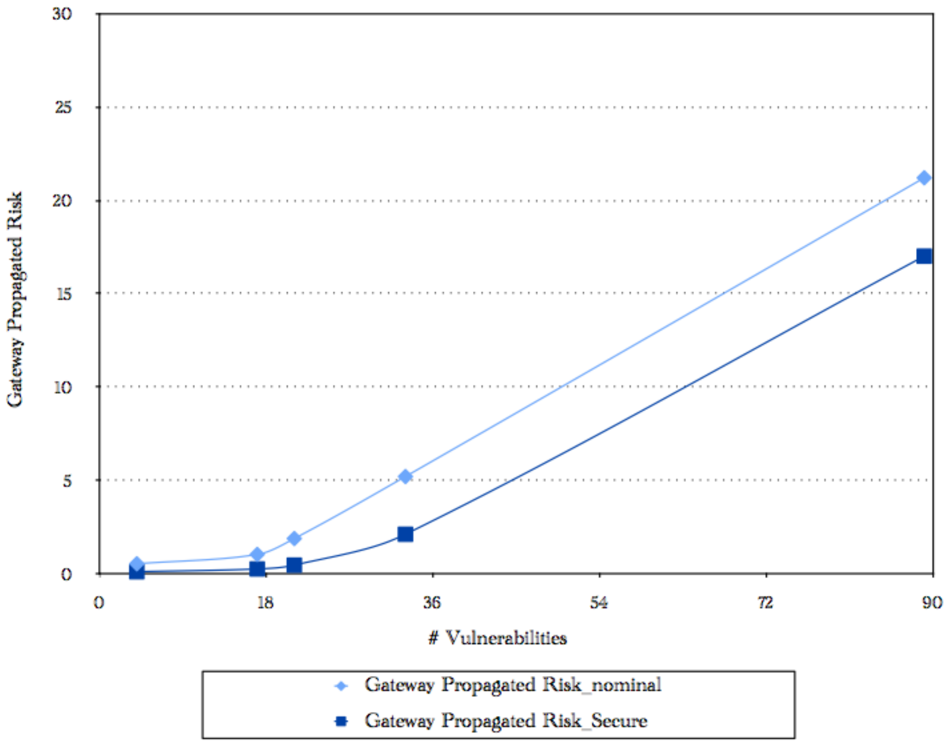


Figure 3.13: Gateway Propagated Risk as a Function of Web Server Vulnerabilities

As we can see, the gateway propagated risk increases when the web server total vulnerability number grows. Moreover, the propagated risk in the secure scenario remains always under the propagated risk in the nominal scenario even if the difference between them is not as important as for the on-board nodes where security features have been deployed (the average gain is equal to 1.984 for the gateway propagated risk level between the nominal and the secure architectures).

As a conclusion, the risk assessment study and the results obtained for both network topologies helped us to make some important design decisions in the scope of the FAST project regarding the node interconnection and security policies for both on-board and ground part. These security controls and their principles are depicted in chapter 4 of the thesis.

The second simulation campaign is dedicated to the AeroMACS system. As previously mentioned in this chapter, the risk assessment algorithm has been primarily designed in the scope of the SESAR project, consequently more results and experiments are presented for the AeroMACS system compared to the FAST case study.

3.4.5 Case Study 2: The AeroMACS System in SESAR Project

3.4.5.1 Introduction to The AeroMACS Technology

AeroMACS is a new aviation-dedicated transmission technology based on the WiMAX IEEE 802.16e standard[16]. The aim is to support fixed and mobile ground-ground and air-ground data communications at the airport surface. The AeroMACS technology allows MSs (Mobile Stations) such as aircraft or surface vehicles to communicate with airline operators and airport staffs at three different surface zones: RAMP (where the aircraft is at the gate before departure), GROUND (the aircraft is taxing to the runway), and TOWER (until the aircraft takes-off). These three zones constitute the APT domain introduced in the chapter 2. Using a WiMAX-based technology standard is profitable for the aviation industry for many reasons. First, the standardization and deployment processes are fast and cost-effective at the opposite of a newly developed standard for the sake of airport communications.

Moreover, the scientific community has been working on IEEE 802.16 standards since many years: highly qualified certification agencies such as the WiMAX Forum[17] are continuously looking after interoperability and technical issues related to the standard. The AeroMACS standard is currently a hot topic in datalink communications and many tests are already running their way for a future deployment. For instance, an AeroMACS profile was recently developed jointly by the RTCA SC-223[18] and EUROCAE WG-82[19] and intended to provide performance requirements for the system implementation.

By way of example, figure 3.14 shows a testbed AeroMACS network layout on the Cleveland-Hopkins International Airport in Ohio, USA as it appears in [Hall *et al.* 2010]. Several SSs (Subscriber Stations)[20] and BSs (Base Stations) have been deployed and tested by the NASA Glenn Research Center[21] for the sake of the NextGen project under the supervision of the FAA. BSs are responsible for providing air interface and additional functions (*e.g.* DHCP proxies, handoff triggering, etc) to these MSs.

[16] http://standards.ieee.org/about/get/802/802.16.html

[17] http://www.wimaxforum.org/

[18] http://www.rtca.org/comm/Committee.cfm?id=133

[19] http://www.eurocae.net/working-groups/wg-list/50-wg-82.html

[20] SS is the initial term used in the early versions of the WiMAX standard: after the IEEE 802.16e amendment, the MS term is used to refer to the station mobility.

[21] http://www.nasa.gov/centers/glenn/home/index.html

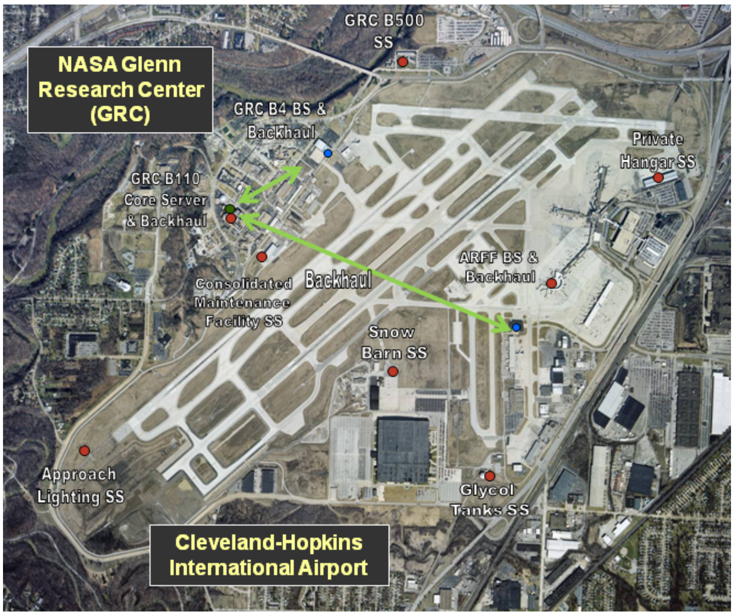

NASA Glenn Research Center (GRC)
GRC B500 SS
GRC B4 BS & Backhaul
Private Hangar SS
GRC B110 Core Server & Backhaul
ARFF BS & Backhaul
Consolidated Maintenance Facility SS
Backhaul
Snow Barn SS
Approach Lighting SS
Glycol Tanks SS
Cleveland-Hopkins International Airport

Figure 3.14: Cleveland-Hopkins Airport AeroMACS Testbed [Hall *et al.* 2010]

3.4.5.2 AeroMACS Protocol Stack

The AeroMACS protocol stack is composed of two main layers: the PHY and the MAC layers, which is itself composed of three sub-layers as shown in figure 3.15. The first layer is the service specific CS (Convergence Sublayer) which communicates with higher layers through the CS SAP (Service Access Point), acquires external network data and transforms them into MAC SDUs (Segment Data Units). The second layer is the CPS (Common Part Sublayer) responsible for the system access, bandwidth allocation, connection management, and MAC SDUs fragmentation into MAC PDUs (Protocol Data Units).

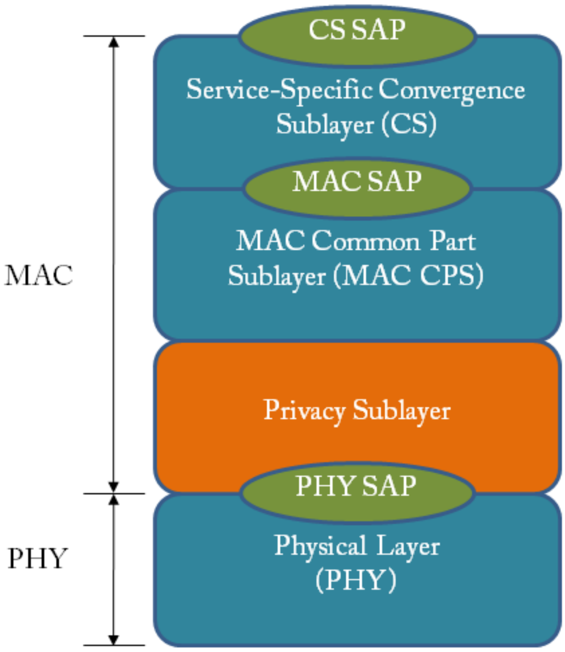

CS SAP
Service-Specific Convergence Sublayer (CS)
MAC SAP
MAC Common Part Sublayer (MAC CPS)
Privacy Sublayer
PHY SAP
Physical Layer (PHY)
MAC
PHY

Figure 3.15: AeroMACS Protocol Stack

Security is handled at the privacy sublayer. It addresses many security services such as authentication, authorization, key establishment, and encryption/decryption of data between the PHY and MAC layers. In fact, after the failures that restricted the early IEEE 802.11 networks, security has been seriously considered in the WiMAX standards (and consequently AeroMACS) and was built in (rather than built on) the 802.16 protocol specifications since the first standard release in 2001. The privacy sublayer provides several mechanisms designed to protect both service providers and users/customers from unauthorized access or information disclosure. However, the details of these security features are out of scope of the thesis, but roughly speaking, the standard uses a PKM (Privacy Key Management) protocol for secure key distribution and X.509 certificates for BSs and MSs identification. Security connections are maintained using SAs (Security Associations), advanced algorithms such as AES (for encryption), EAP (Extensible Authentication Protocol) [Aboba *et al.* 2004] or RSA (authentication, authorization) are used to protect the data exchanges.

The specification of the AeroMACS standard focuses on the physical and the access layer of the radio link as shown in figure 3.15. However, this is not sufficient to built in a broadband wireless airport network: end-to-end services such as QoS, security, mobility management or IP connectivity are a requisite and should be provided beyond the AeroMACS scope (*i.e.* at higher layers of the protocol stack). In this context, an AeroMACS network reference model has been developed.

3.4.5.3 AeroMACS Network Reference Architecture

In order to understand the interconnection concepts and topology considerations we made in the scope of the SESAR project, a logical representation of an AeroMACS reference network must be first introduced. Such a scheme distinguishes between the logical domains, the functional entities, and the RPs as reported in figure 3.16:

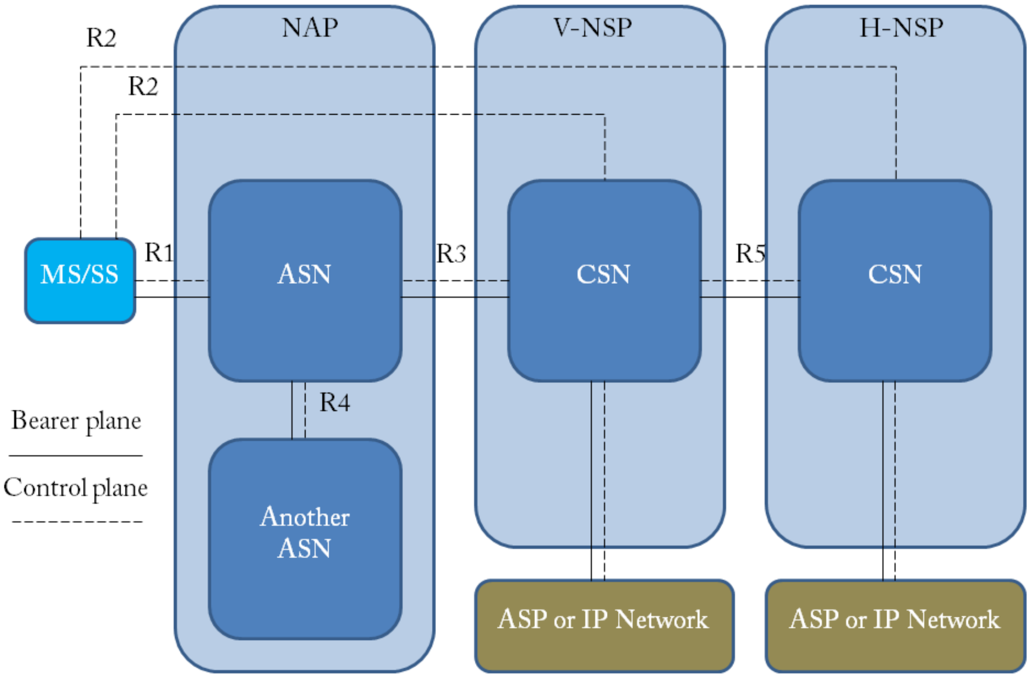


Figure 3.16: AeroMACS Network Reference Architecture

The main depicted functional entities are:

- ◇ MSs and SSs which could be aircraft, surface vehicles, or passenger personal generic devices;
- ◇ The ASN (Access Service Network) network represents the boundary for functional interoperability between MSs and AeroMACS connectivity services. The ASN integrates many functions such as forwarding AAA (Authorization, Authentication, Accounting) messages between MSs and the H-NSP (Home Network service Provider), relaying network service messages (*e.g.* DHCP request/response), etc;
- ◇ The CSN (Connectivity Service Network) network provides connectivity to public networks such as the Internet.

The logical domains, which are basically set of functions associated to a single domain, and considered in the network reference architecture are:

- ◇ The NAP (Network Access Point) is the physical point used by MSs to access the network;
- ◇ The H-NSP is the AeroMACS service provider which provides SLA (Service Level Agreement) to the MSs such as IP connectivity and core network services. These NSPs could be for instance, SITA, ARINC, or even the airlines depending on the provided service;
- ◇ The V-NSP (Visited Network service Provider) is visited by the MSs to access the network in a roaming scenario (which usually depends on the roaming agreement made between the MS's H-NSP and the V-NSP).

Referring to figure 3.16, RPs (Reference Points) are the communication end-points between functional entities and represent the interfaces that ensure the interoperability between AeroMACS-related components. Table 3.12 gives a description of all the reference network architecture interfaces:

Table 3.12: AeroMACS Reference Point Interface Description

RP	**Interface**	**Functionality**
R1	between MSs and BSs	Air interface
R2	between MSs and the CSN	IP host configuration, IP mobility, Authentication, Authorization
R3	between the ASN and the CSN	Mobility management, Authentication, Authorization, tunneling
R4	between ASNs	Mobility management, ASNs interworking
R5	between CSNs	Roaming and interworking between the V-NSP and the H-NSP

The AeroMACS network reference model provides a unified and flexible network for airport data communications. However, due to the sensitive nature of operational data and the priority issues in aeronautical communications, the AeroMACS network should provide enough security to protect exchanged data.

3.4.5.4 AeroMACS Security Considerations

The privacy sublayer has been defined to deal with several attacks targeting the authentication, the authorization, or the confidentiality of exchanged data. Despite these security mechanisms, many security weaknesses have been already discovered. Attacks ranging from physical layer-based attacks (*e.g.* jamming, scrambling, etc) [Barbeau 2005], attacks on the key management (*e.g.* mutual authentication, PKM failure, replay attacks) [Xu & Huang 2006], brute force attacks on privacy (*e.g.* DES in CBC mode is now considered insecure after operating $2^{\frac{n}{2}}$ blocks where n is the block size) [Johnston & Walker 2004], attacks on availability (*e.g.* DoS on PKM authorization phase) [Eren & Detken 2008] have been identified. To mitigate them, many amendments have been conducted and the privacy sublayer has been redefined for the mobile IEEE 802.16e. Still, some of these issues exist and are exploitable in the network as we will see in the next subsection.

Moreover, an end-to-end connected network (*e.g.* adding application servers, service provider nodes, etc) induces an additional risk due to the node correlation and service dependencies as we have previously discussed. In the next section, we focus on the inherited AeroMACS vulnerabilities from the mobile version of the WiMAX standard. These security issues are mainly related to the integrity of some management communication messages and the key sharing scheme used in multicast and broadcast services.

3.4.5.5 Analysis of AeroMACS Security Weaknesses

The AeroMACS security weaknesses can be grouped into three categories as follows:

- ◇ **Unauthenticated management messages**: most of the management messages defined in AeroMACS are integrity protected using a HMAC (Hash-based Message Authentication Code) [Krawczyk *et al.* 1997] or alternatively a CMAC (Cipher-based Message Authentication Code) [NIST 2005]. However, some messages (*e.g.* MOB_TRF-IND, MOB_NBR-ADV, RNG-REQ) are not authenticated which leads to some vulnerabilities. Also, some management messages are sent over the broadcast management channel: authenticating this type of messages becomes difficult since there is no common key to generate the message digest. Furthermore, a common key would not completely provide message integrity as mobile stations sharing the same key can arbitrary forge these management messages and generate false authentication digests;
- ◇ **Unencrypted management messages**: when initial network entry procedure begins between a MS and a BS, many control information (*e.g.* mobility parameters, power settings, security capabilities) are sent in clear. An adversary may eavesdrop these messages just by listening to the channel in order to establish detailed profiles of MS or BS. More specifically, when MBRA (Multicast and Broadcast Rekeying Algorithm) is used, group encryption keys called GTEKs (Group Traffic Encryption Keys) are distributed to all group members and encrypted using another key, namely GKEK (Group Key Encryption Key). This GKEK symmetric key is shared and known by all groups which means that a malicious group member may use a new GKEK key to update request for the GTEK key and generate its own GTEK;
- ◇ **Shared keys in multicast and broadcast services**: multicast and broadcast messages are encrypted and authenticated using a symmetric shared key between a BS and all MSs belonging to the same group: this is an issue in the sense that any MS may impersonate the original BS by forging false multicast or broadcast messages and sending them to other MSs in the same group.

These security weaknesses have already been under inspection in recent work [Hall *et al.* 2010, NASA 2009, Wilson 2011]. The authors clearly stated that these security holes should be quickly fixed:

the network risk assessment methodology we developed is a valuable requisite in the sense that it could help us to determine which network topology or security policy (*i.e.* with the lowest network risk) provides the most secure configuration of the AeroMACS network.

In the next section, we depict the AeroMACS network topology defined in the scope of the SESAR 15.2.7 WP project. The different simulation results are discussed then, a sub-scenario is presented in order to compare two authentication/authorization security protocols supported by the AeroMACS privacy sublayer, namely RSA and EAP, in terms of vulnerability, individual and propagated risks.

3.4.5.6 AeroMACS Network Architecture

Figure 3.17 depicts the AeroMACS network topology[†], three main portions can be identified, namely the ASN, the CSN, and the mobile stations (aircraft or surface vehicles). This basic topology assumes the existence of a standalone service network supported by an AeroMACS access network in the airport. Consequently, all the services are provided by components inside the airport network (AAA, DHCP and application servers) and placed within the airport backbone.

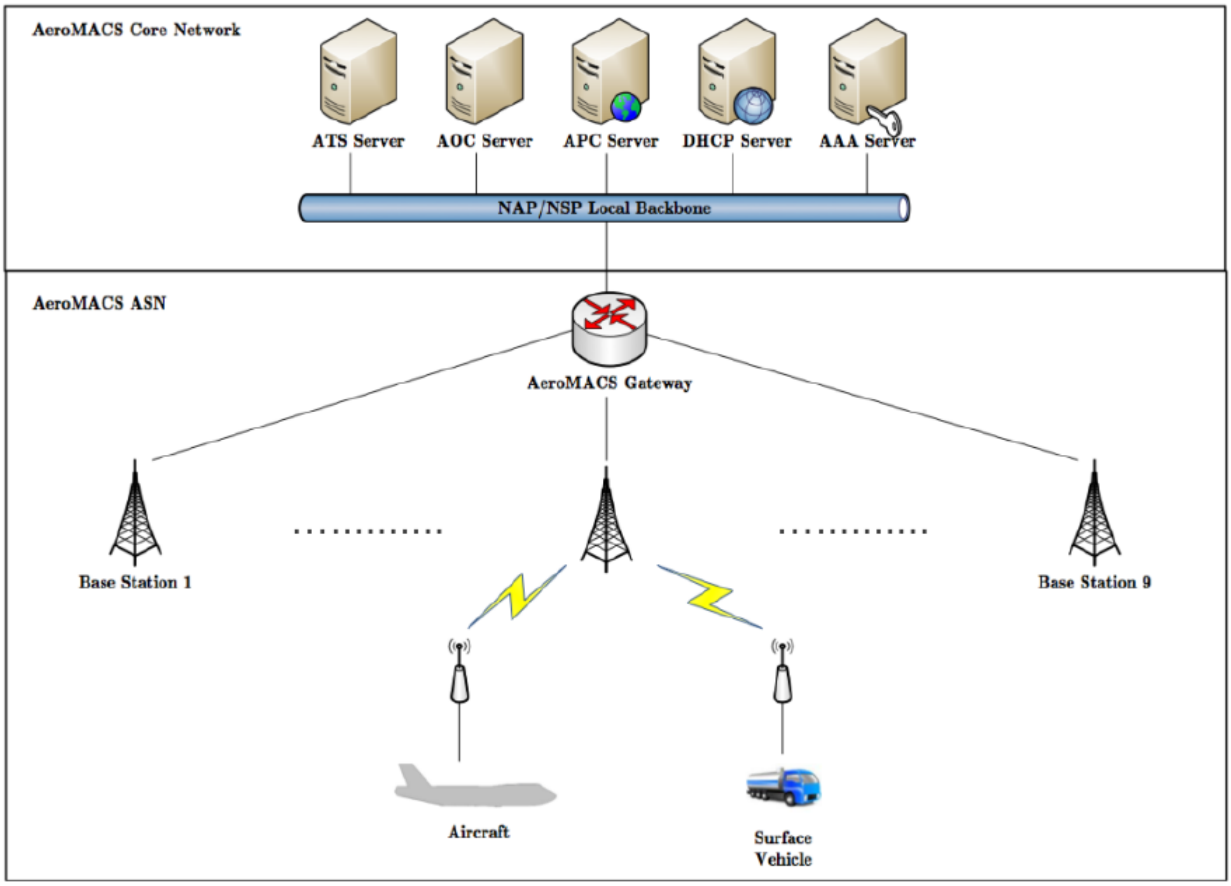


Figure 3.17: AeroMACS Network Topology Scenario

The AeroMACS (additionally to the AAA server) segment is the only system supporting security features and the AAA server will be directly reachable through a dedicated gateway between the AeroMACS network and the others Airport networks. The APC server of the figure 3.17 refers to the AirPort Communications server (and not to the Airline Passenger Communication Server as it is the case in the FAST project for instance).

[†] The AeroMACS topology has been defined jointly to the Thalés, Airbus, Indra, the French DSNA, and Selex partners.

3.4.5.7 Experimental Results

Table 3.13 summarizes the main simulation inputs[†] used for the AeroMACS network scenario:

Table 3.13: AeroMACS Topology Simulation Parameters

Node ID	Connected Nodes	Security Protection	Vulnerabilities
Base Stations (1 to 6)	3	8	1
Base Stations (7 and 8)	4	8	1
Base Station 9	5	8	1
Aircraft	1	2	0
Surface vehicles	1	2	0
AAA Server	1	3	20
DHCP Server	1	1	64
ASN Gateway	14	2	1
ATS Server	1	1	47
AOC Server	1	1	47
APC Server	1	1	13

9 base stations, 10 aircraft and 12 surface vehicles have been considered in the simulation. It is important to note that the number of connected nodes for each base stations may vary, which has an impact on the risk value results discussed below.

Individual Risks and Vulnerability Statistics

Figure 3.18 illustrates the individual risks assessed for each node in the network. Base stations and the ASN gateway individual risks are relatively low because there is only a single vulnerability detected for these nodes. Despite having the same specific vulnerability (CVE-2008-1542)[22], there is a difference (which varies from 0.075 to 0.15) between Base Station 8, Base Station 9, and the seven first base station individual risks. This is mainly due to the higher number of connected nodes of Base Station 8 and Base Station 9 which increases their node values and consequently their individual risks.

Another interesting fact is that the APC server and the ASN gateway individual risks are nearly close (respectively 0.46 and 0.5) despite a big difference in the number of intrinsic vulnerabilities of each of them (respectively 13 and 1). Indeed, giving these values, we shall expect a higher individual risk for the APC Server node as long as it has more vulnerabilities, however the ASN gateway compensates the gap thanks to its higher node value (equal to 14). The APC Server, giving its functionality and traffic class values, has the lowest node value (equal to 0.03) which affects the final individual risk result.

The DHCP server node is the most vulnerable node in the network and has the highest individual risk value (which is assessed to 6.62): 64 vulnerabilities have been detected for that node with very high CVSS scores: 92.12% of them have the top CVSS score (equal to 10). Among these DHCP server vulnerabilities, The lowest score has been assessed to 9.3, which is relatively high in the CVSS scale. With such as high individual risk, the DHCP server node should be considered carefully when the

[†] Simulation inputs of the SESAR case study have been partially provided by the Thalés, Airbus, and Indra partners.

[22] Vulnerability details as it appears in the NVD: base station has ”top secret“ as its password for the root account, which allows remote attackers to obtain administrative access via a telnet login.

propagated risk results are discussed specially when it is connected to a node with a high connectivity in the network (*c.f.* the next subsection).

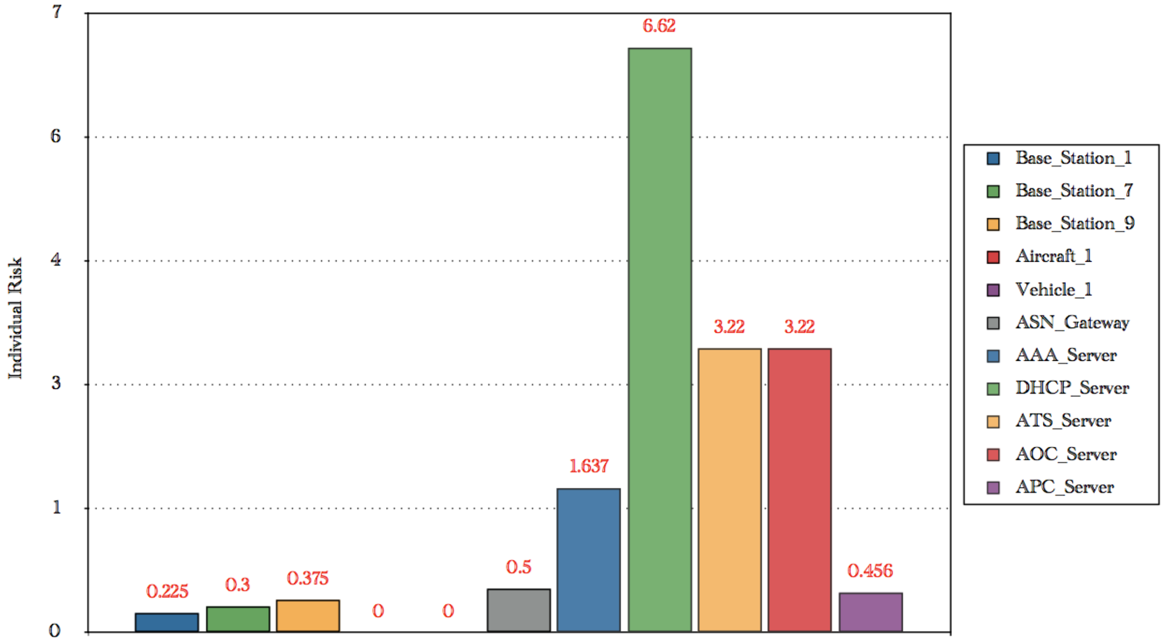


Figure 3.18: Individual Risks for all Network Nodes

Finally the AAA, ATS and AOC servers, regarding the assumptions made in the simulation inputs, get medium individual risk values. Except the ASN Gateway individual risk value which is considerably impacted by its high node value, all the individual risk values we measured seem to increase linearly as a function of the number of exploitable vulnerabilities as shown in figure 3.19:

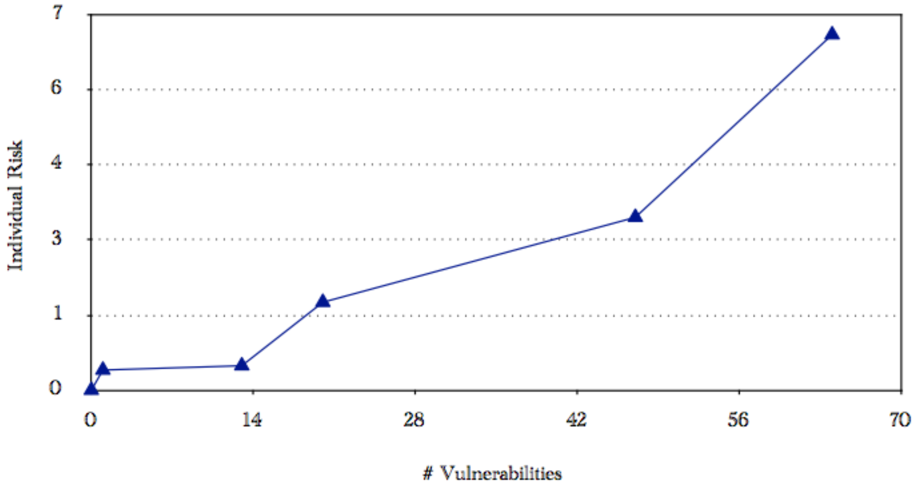


Figure 3.19: Individual Risk Evolution as a Function of Vulnerabilities for All Nodes

The individual risk and CVSS vulnerability score are then strongly related. Table 3.14 illustrates the repartition of the CVSS score for all nodes in the network. The majority of the scores are ranked in the [7,8] NVD CVSS scale (and represents 55.224% of the total vulnerability scores). The maximum CVSS scores ranked between 9 and 10 are in most cases relative to the DHCP server node, which explains why this node has the highest individual risk value inside the network. The average CVSS score has been measured to 7.938.

Table 3.14: Vulnerability CVSS Statistics

CVSS Score	Number of Vulnerabilities	Percentage
$[0,1]$	0	0%
$[1,2]$	2	0.995%
$[2,3]$	1	0.498%
$[3,4]$	0	0%
$[4,5]$	7	3.483%
$[5,6]$	11	5.473%
$[6,7]$	4	1.990%
$[7,8]$	111	55.224%
$[8,9]$	0	0%
$[9,10]$	65	32.338%
Total	**201**	-
Average Score	**7.938**	

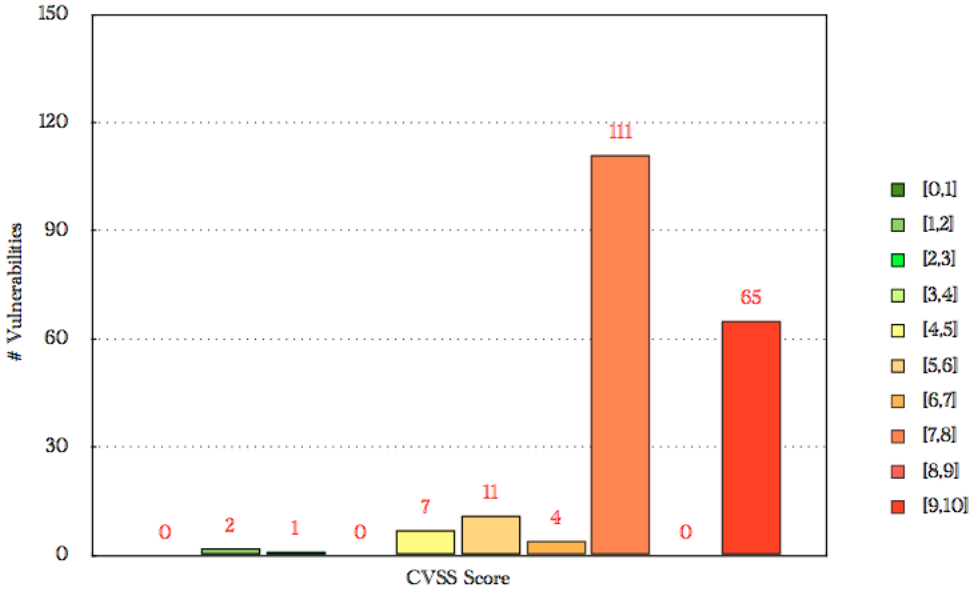


Figure 3.20: Vulnerability CVSS Statistics

Propagated Risks

Table 3.15 summarizes the propagated risk values for each node in the network. The propagated risk results are mainly impacted by the importance of the connected node number parameter in the algorithm. For instance, as depicted at the beginning of section 3.4.5.7, we made different assumptions for the AeroMACS base stations: the first 6 base stations are connected to three nodes (*i.e.* one aircraft, one vehicle, and the ASN gateway), base stations 7 and 8 are connected to four nodes (plus another vehicle) and the last base station to five nodes. The remaining parameters (*i.e.* security protection, offered service, exchanged data, NVD vulnerabilities) remain the sames. However, the propagated risk values are slightly different (ranging from 7.474 to 12.456) mainly because of a different correlation density for each base station in the network.

The propagated risk values for the aircraft also varies and are not the sames for the 6 first aircraft (equal to 0.812) and the 6 last ones (equal to 1.082). This time, the reason is not the connected node

Table 3.15: Propagated Risks for all Network Nodes

Node ID	Propagated Risk
Base Stations (1 to 6)	7.474
Base Stations 7 and 8	9.965
Base Station 9	12.456
Aircraft (1 to 6)	0.812
Aircraft (7 to 12)	1.082
Vehicle (1 to 6)	0.812
Vehicle 7 and 8	1.082
Vehicle 9 and 10	1.353
ASN Gateway	538.998
DHCP and AAA Servers	1.2
ATS Server	0.398
AOC Server	0.750
APC Server	0.135

parameter as far as all aircraft are connected to a single base station: the difference between aircraft propagated risks (which has been evaluated to 0.27) is due to the individual risk value specific to each particular base station connected to every aircraft. DHCP, AAA, ATS, AOC, and APC servers have all low propagated risk values (ranging from 0.135 to 1.2) because all of them are connected to a single node (the ASN gateway) which has a low individual risk (equal to 0.5).

The most important result to discuss in this simulation is probably the propagated risk value of the ASN gateway, which supersedes all the remaining nodes. This is likely due to the high node correlation of the ASN gateway. As far as it is the *"corner stone"* of the topology where all node exchanges have to pass through, the ASN gateway propagated risk value is logically impacted by the connected nodes. The propagated risk is clearly impacted by the number of connected nodes and increases exponentially as illustrated in figure 3.21:

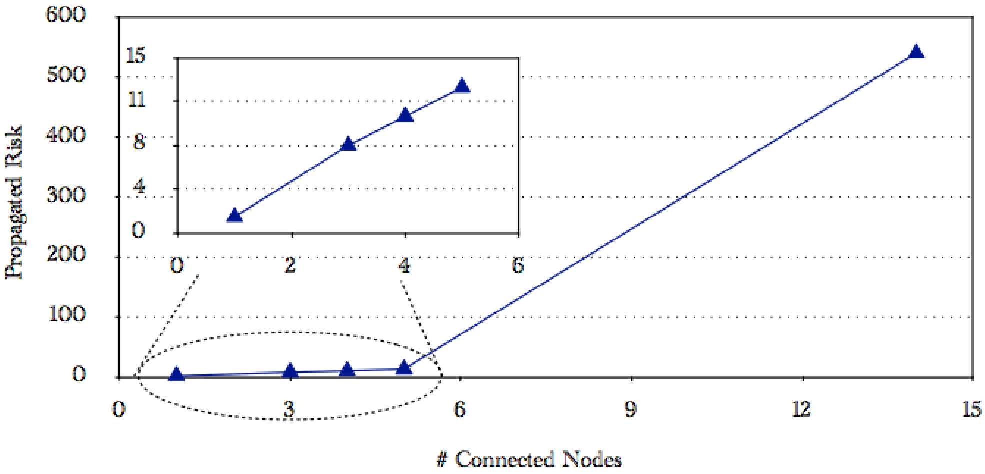


Figure 3.21: Propagated Risk Evolution as Function of Connected Nodes

Node and Network Risks

The node risk of the ASN Gateway is hardly impacted by the high propagated risk of the node (*cf.* the formula 3.2 of node risk evaluation). Besides, the high node value of the ASN Gateway plays a major role in the growth of the node risk value: even the highest node risk (which is relevant to the Base Station 9) is 117 times smaller than the ASN Gateway node risk. Consequently, as the network risk is given by the sum of the node risks, the network risk is mainly represented by the ASN Gateway node risk as we can see in figure 3.22:

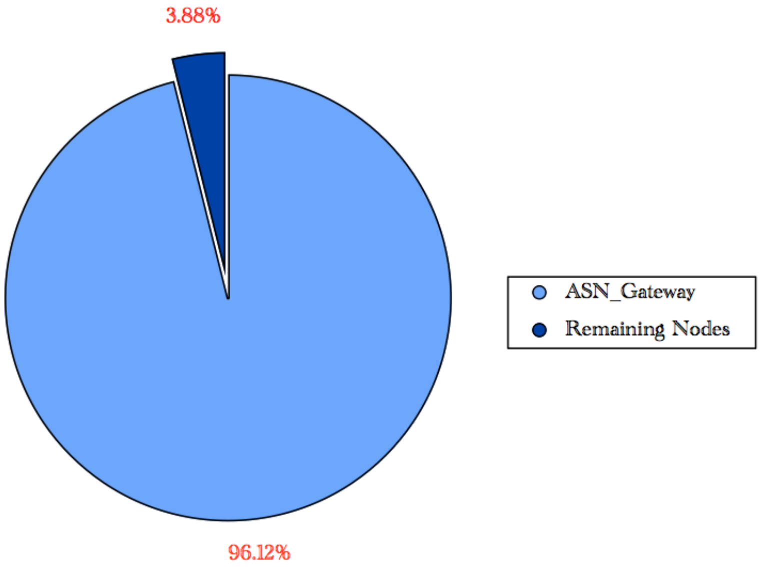


Figure 3.22: Percentage of Impact of Node Risks on the Network Risk

The detailed node risk impact (in percentage) on the network risk is given in table 3.16:

Table 3.16: Node Risk Results

Node ID	Node Risk	% of Network risk
All Base Stations	284.866	3.63%
All Aircraft	8.337	0.11%
All Vehicle	6.821	0.09%
ASN Gateway	7552.98	96.12%
DHCP Server	2.346	0.03%
AAA Server	0.851	0.01%
ATS Server	1.085	0.01%
AOC Server	0.833	0.01%
APC Server	0.017	0%

In order to figure out how the ASN gateway node risk could be lowered, the percentage of impact of connected nodes on the ASN gateway propagated risk is presented in figure 3.23. As seen in the individual risk results, the DHCP server high individual risk value has an important effect on the ASN gateway propagated risk.

This means that if we want to get a lower ASN gateway node risk (and consequently a lower network risk) we should adjust the following related parameters:

1. the number of connected nodes which is very important and should be limited per node as much as possible. For instance, this can be done after some topological considerations that allows the network risk to be less higher than in this first simulation;

2. the ASN gateway is the point of failure of this topology as seen in the previous results. Despites its high node connectivity, its high propagated risk value is not directly due to the number of interconnected nodes, but rather to the high DHCP server individual risk, the high correlation that exists between the two nodes (the ASN gateway is the only node connected with the DHCP server node) and the high node value of the ASN gateway.

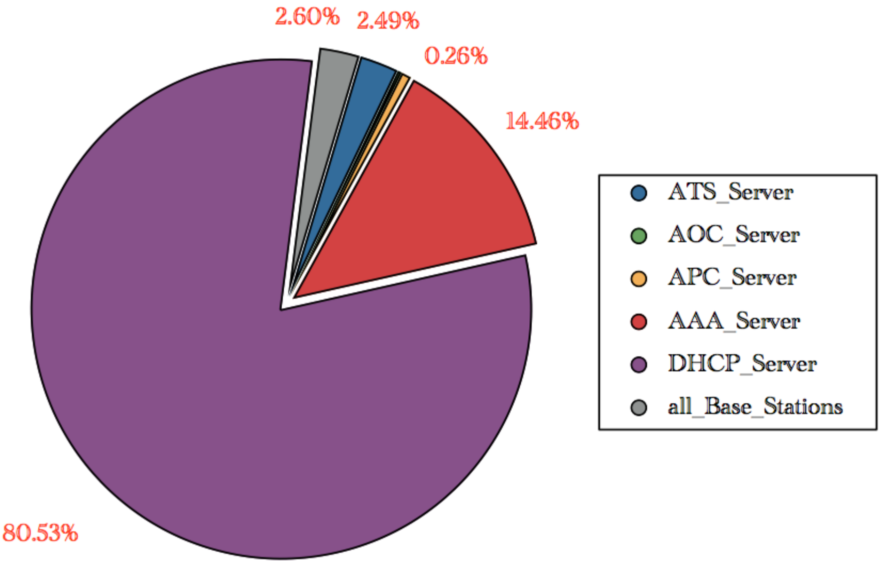


Figure 3.23: Percentage of Impact of Connected Nodes on the ASN Gateway Propagated Risk

In the next subsection, we discuss two authentication/authorization schemes implemented in the AeroMACS privacy sublayer: the aim is to analyze how RSA and EAP protocol can affect the risk results discussed in the previous sections.

EAP vs. RSA Sub-scenario

As it has been previously mentioned, the AeroMACS privacy sublayer is able to support both EAP and RSA for authentication and authorization security services. The aim of this simulation scenario is to compare the effects of these security options on the AeroMACS network risk. Many vulnerabilities have been found for both security mechanisms: the NVD database clearly indicates a higher number of vulnerabilities for RSA (*i.e.* 33 vulnerabilities) compared to EAP (only 4 vulnerabilities). Indeed, RSA is much more known and used over all IT systems in the world. Then, it is quite logical to find more vulnerability inputs in the database compared to EAP.

Moreover, the vulnerability statistics we made in the simulation shows again that the number of vulnerabilities is not the only indicator on what security mechanism should be privileged. Indeed, despite a higher number of vulnerabilities (+29 vulnerabilities), RSA remains more secure than EAP: as we have seen in the previous simulation, the average CVSS score is a safer criteria if we want to compare two or more security mechanisms in the NVD database. The average CVSS score (on the total vulnerabilities in the AeroMACS network) has been evaluated to 6.325 for RSA and 7.795 for EAP.

Table 3.17 gives more details on the vulnerabilities statistics for the EAP vs. RSA scenario simulations:

Table 3.17: EAP vs. RSA Vulnerability Statistics (all nodes)

CVSS Score	EAP		RSA	
	Vulnerabilities	Percentage	Vulnerabilities	Percentage
$[0,1]$	0	0%	0	0%
$[1,2]$	2	0.83%	12	2.26%
$[2,3]$	1	0.415%	11	2.072%
$[3,4]$	0	0%	10	1.883%
$[4,5]$	7	2.905%	107	20.151%
$[5,6]$	31	12.863%	121	22.787%
$[6,7]$	4	1.66%	24	4.52%
$[7,8]$	121	50.207%	161	30.32%
$[8,9]$	0	0%	0	0%
$[9,10]$	75	31.12%	85	16.008%
Total	**241**	-	**531**	-
Average Score	**7.795**		**6.325**	

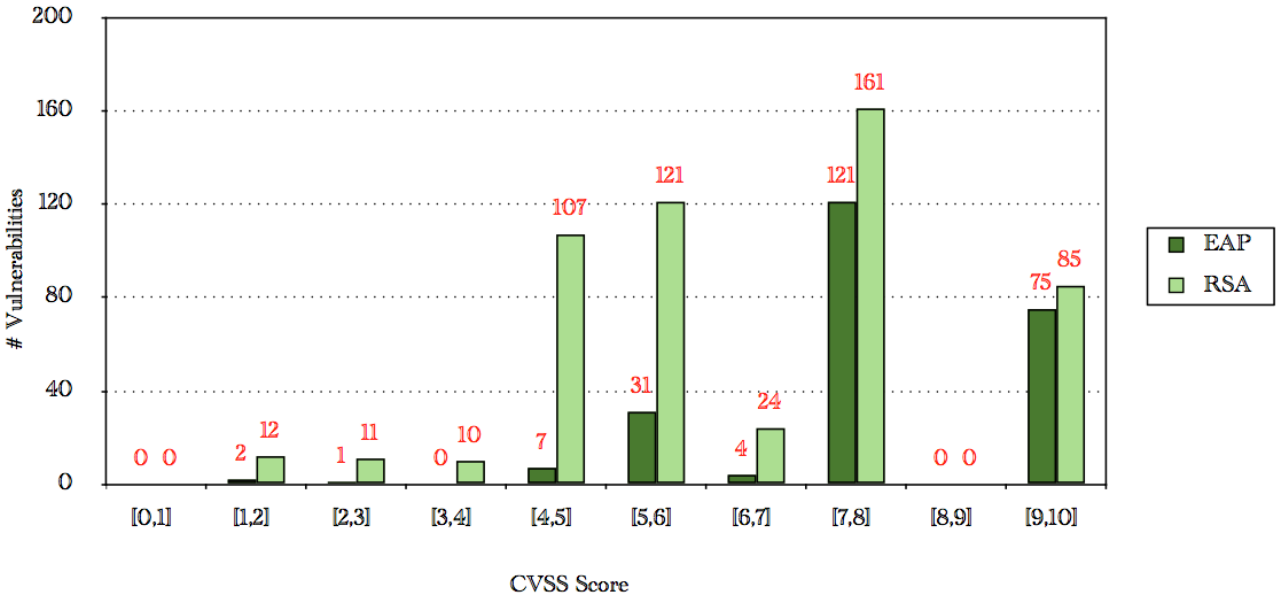


Figure 3.24: Vulnerability CVSS Score Distribution for EAP and RSA

However, the average CVSS score should be weighted according to the individual risk values obtained after the simulation. Indeed, Figure 3.25 shows the updated individual risk values for the base stations and the ASN gateway (all the remaining nodes are not represented since there is no change to notice on them). The higher number of vulnerabilities for RSA makes naturally the individual risk values higher than those obtained using EAP for both base stations and the ASN gateway (+16.35 and +14.8 respectively using RSA and EAP). These results suggest that the number of vulnerabilities remains an important parameter because the individual risk is computed as a sum of likelihood of occurrence of a threat and its impact on the total number of vulnerabilities. Since RSA has much more inputs in the NVD database, the individual risk relevant to EAP is lower.

As a conclusion, if we want to take the risk individually by node, it is clear that EAP should be used for the authentication and authorization procedures on the AeroMACS nodes.

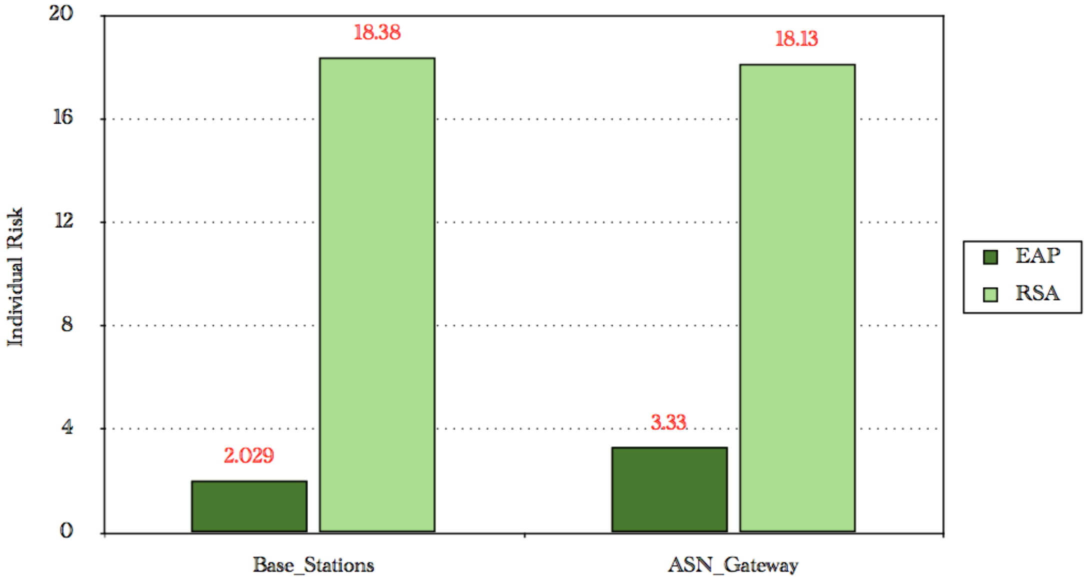


Figure 3.25: Individual Risks for Base Stations and the ASN Gateway

Nevertheless, the propagated risk results should be also considered to effectively make final guidance on the use of EAP or RSA protocols. Figure 3.26 shows the propagated risk values using EAP or RSA protocols for all network nodes (except the ASN gateway which has a very big propagated risk value and is not represented for clarity matters). The same comments made for the individual risks remain true here: the EAP authentication protocol induces a lower propagated risk compared to the RSA protocol. The ASN gateway is still the point of failure in both sub-scenarios since it has the largest propagated risk among all the network nodes (1042.64 and 2499.87 respectively using EAP and RSA). It still has the biggest impact on the global network risk (either for EAP or RSA) as illustrated in table 3.18.

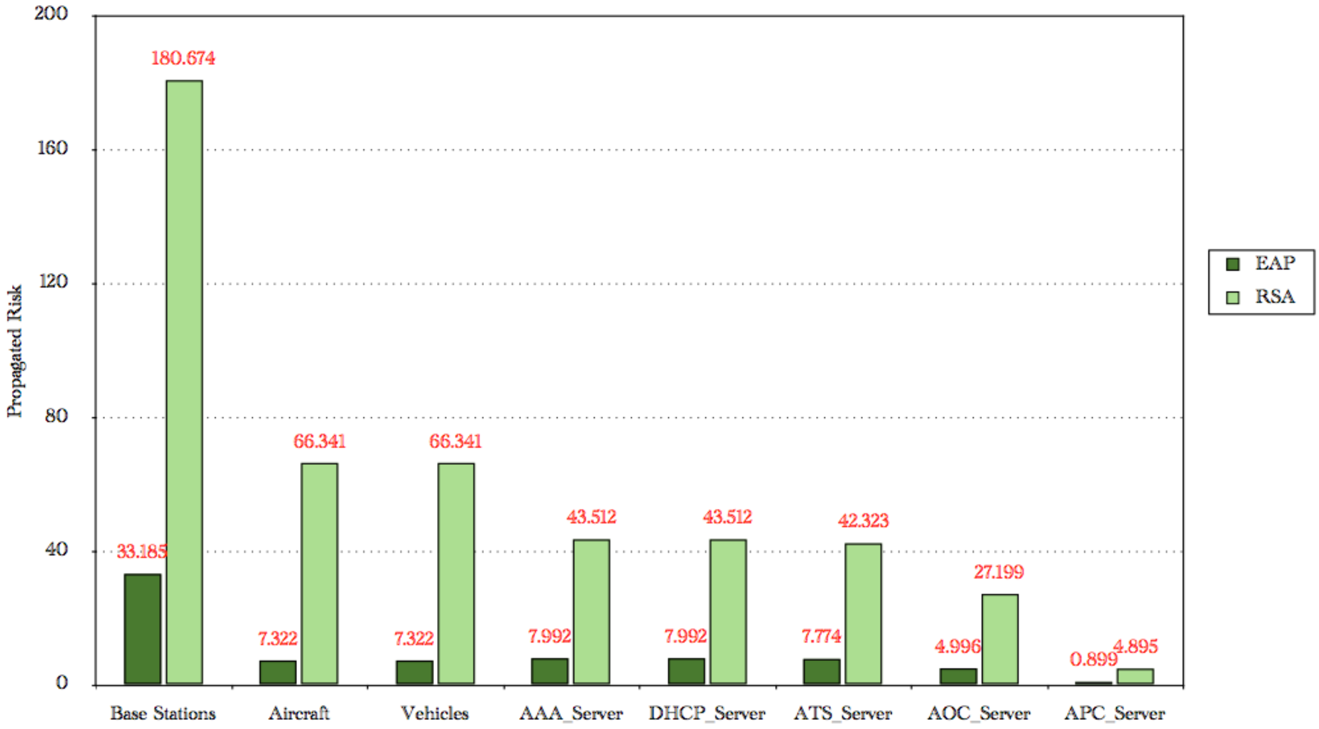


Figure 3.26: Propagated Risks for all Nodes (EAP vs. RSA)

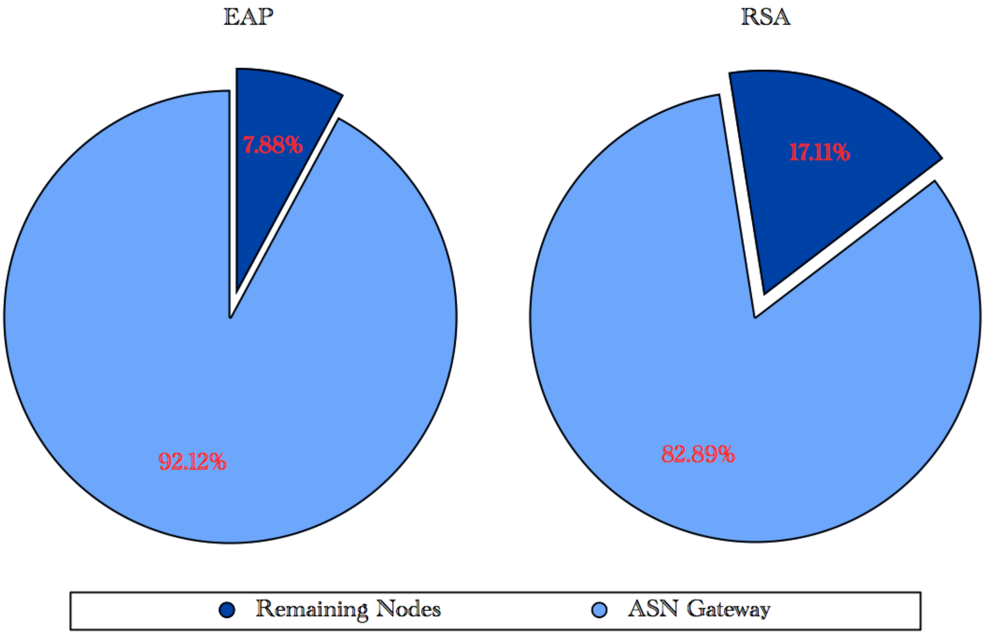


Figure 3.27: Percentage of network risk per node risk (EAP vs. RSA)

Table 3.18: Node risk Results (EAP vs. RSA)

Node ID	**% of Network risk**	
	EAP	RSA
All Base Stations	6.867%	14.509%
All Aircraft	0.386%	1.310%
All Vehicle	0.322%	1.091%
ASN Gateway	92.118%	82.888%
DHCP Server	0.03%	0.038%
AAA Server	0.053%	0.030%
ATS Server	0.135%	0.0755%
AOC Server	0.081%	0.0546%
APC Server	0%	0%

Regarding the simulation results we discussed above, we can already draw the big lines of the guidance that should allow us to decrease the risk level for the different network nodes.

3.4.5.8 Security Guidance

The security guidances we propose to enhance the end-to-end AeroMACS network security are divided into three categories:

1. *Implementation guidance*: Network nodes should be chosen wisely with a minimum of intrinsic vulnerabilities. IP COTS nodes (AAA Server, DHCP Server) should be discussed regarding the number of the exploitable vulnerabilities and their respective CVSS scores. It could be interesting to establish a state-of-the-art of the potentially usable IP nodes (particularly the DHCP server node), classify them by number of vulnerabilities and CVSS scores and see how the individual

risk per node is affected. The nodes to be preferred are obviously those with the lowest individual risks;

2. *Topological guidance*: as we have seen, the global network risk is highly impacted by the propagated risk values (more than the individual risk values) because the node connectivity is taken into account at this step of the risk assessment process. It is clear that the ASN gateway is the main point of failure in this topology and some countermeasures should be taken to avoid this problem. For instance, according to the AeroMACS network reference architecture depicted in figure 3.16, it could be interesting to implement two ASN gateways as NAPs, each one connected to a set of base stations and IP nodes. This will likely provide a less important node correlation between the ASN gateway (then a lower likelihood of correlation) and highly impacted IP nodes (such as the DHCP server) allowing the network risk to decrease;

3. *AeroMACS security guidance*: in order to get a lower global network risk and limit the impact of high risk nodes, some security mechanisms should be deployed. A particular attention should be given to the connectivity between the ASN gateway and the IP nodes such as the DHCP server. To deal with this connectivity problem, firewalls should be privileged. Indeed, they can limit the data exchanges between highly vulnerable nodes and the ASN gateway. Also, maximizing security protections at a layer-2 (typically AeroMACS security) should also decrease the propagated node risk decrease for AeroMACS-based nodes (*i.e.* base stations and mobile stations). In this context, the EAP vs. RSA simulations showed that EAP induces lower risks (individual, propagated, and network risks) compared to RSA.

3.5 Summary

In this chapter, we exposed a new quantitative risk assessment method for network security based on risk propagation. We defined the main model components and detailed the consistency of each of them. We discussed the drawbacks of existing methods and then we palliated the identified theoretical lacks. The major part of this chapter was dedicated to quantify the main risk parameters used in our methodology. finally, we specified the risk assessment algorithm and applied it on both SESAR and FAST projects. Particularly, experimental results relative to the case study devoted to the FAST project shown that the global network risk in the nominal system architecture is relatively high, meaning the system needs more consideration from a security point of view. Thus, the next chapter presents a network security framework dedicated to lower as much as possible the network security risk identified in this chapter. The adaptive security policy is discussed according to aeronautical constraints mentioned in chapter 2.

Chapter 4

SecMan: an Adaptive Security Architecture for Future Aircraft Communications

Contents

In the previous chapter, the network security risk results associated to the FAST study case highlights the need for an efficient security framework onboard the aircraft. The so-called security architecture must fulfill at least the datalink security requirements identified in chapter 2. Moreover, QoS and service priorities have to be well-managed through an accurate and secure logical separation between operational and non-operational aeronautical services. Finally, differentiated security requirements per traffic class shall be provided with low network overheads particularly when datalink network resources become limited. Thus, in this chapter, the SecMan adaptive security framework, developed in the scope of the FAST project, is introduced. As mentioned in chapter 2, we have been assigned to several work packages, our first contribution consists in the definition of the nominal satellite architecture and the QoS policy jointly to the ISAE[1] partner within the WP1.

[1] SCAN (Signal Communication, Antennas, and Navigation) research team.

Note: Materials and results presented in this chapter have appeared in [Ben Mahmoud *et al.* 2009c], [Ben Mahmoud *et al.* 2010a, Ben Mahmoud *et al.* 2010b, Ben Mahmoud *et al.* 2010d, Ben Mahmoud *et al.* 2010e], [Ben Mahmoud *et al.* 2010f], [Ben Mahmoud *et al.* 2011a], and [Ben Mahmoud *et al.* 2011b].

4.1 System Description

The FAST project aims to design a new low-cost and extra-flat antenna for bi-directional satellite communications on commercial and business aircraft worldwide. The foreseen system should allow high data rates and cover high density traffic routes (*i.e.* Europe, North Pacific, North Atlantic, USA, and the East of Asia). Figure 4.1 illustrates the FAST satellite transmission system architecture:

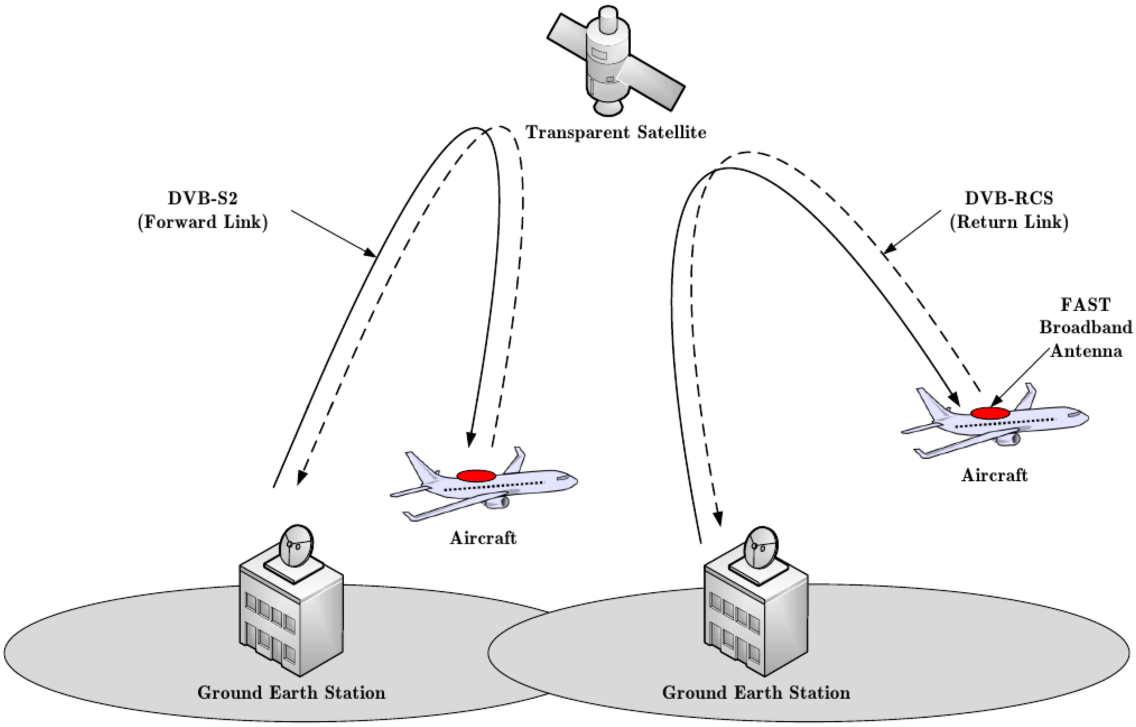


Figure 4.1: FAST Transmission System Architecture Overview

The system is composed of three main segments:

1. the air segment is composed of the antenna and the embedded satellite terminal responsible for the transmission/reception between the aircraft and the ground servers via the satellite link. Using the AMSS (Aeronautical Mobile Satellite Service) terminology [ICAO 2006b], the aircraft segment is composed by mobile AES (Aeronautical Earth Station);
2. the space segment is composed of *Ku* band (10.7Ghz - 14.5Ghz) geostationary satellites, assumed to be completely transparent (*i.e.* without embedded processing). Air interfaces are based on the DVB-S2 standard [ETSI 2005] for the forward link (*i.e.* from the ground to the air) and the DVB-RCS standard [ETSI 2009] for the return link (*i.e.* from the air to the ground). The network architecture and satellite terminal are designed according to the ETSI BSM (Broadband Satellite Multimedia) architecture for IP-based satellite links [ETSI 2006] as described in the next section;
3. the ground segment is composed of several GESs (Ground Earth Station), in charge of handling air-ground data traffic and providing an interface with the ground networks. Every GES is associated to a satellite and a NCC (Network Control Center), responsible for performing the satellite network control functions (*e.g.* synchronization, resource management).

4.2 Network Architecture and Terminal Design

In this section, we provide an overview of the global system architecture defined in the scope of the first work package WP1 of the FAST project. We mainly focus on three key points: network architecture, QoS policy, and resource management.

4.2.1 Satellite-based System Architecture

Figure 4.2 represents the FAST satellite-based aeronautical system architecture[†]:

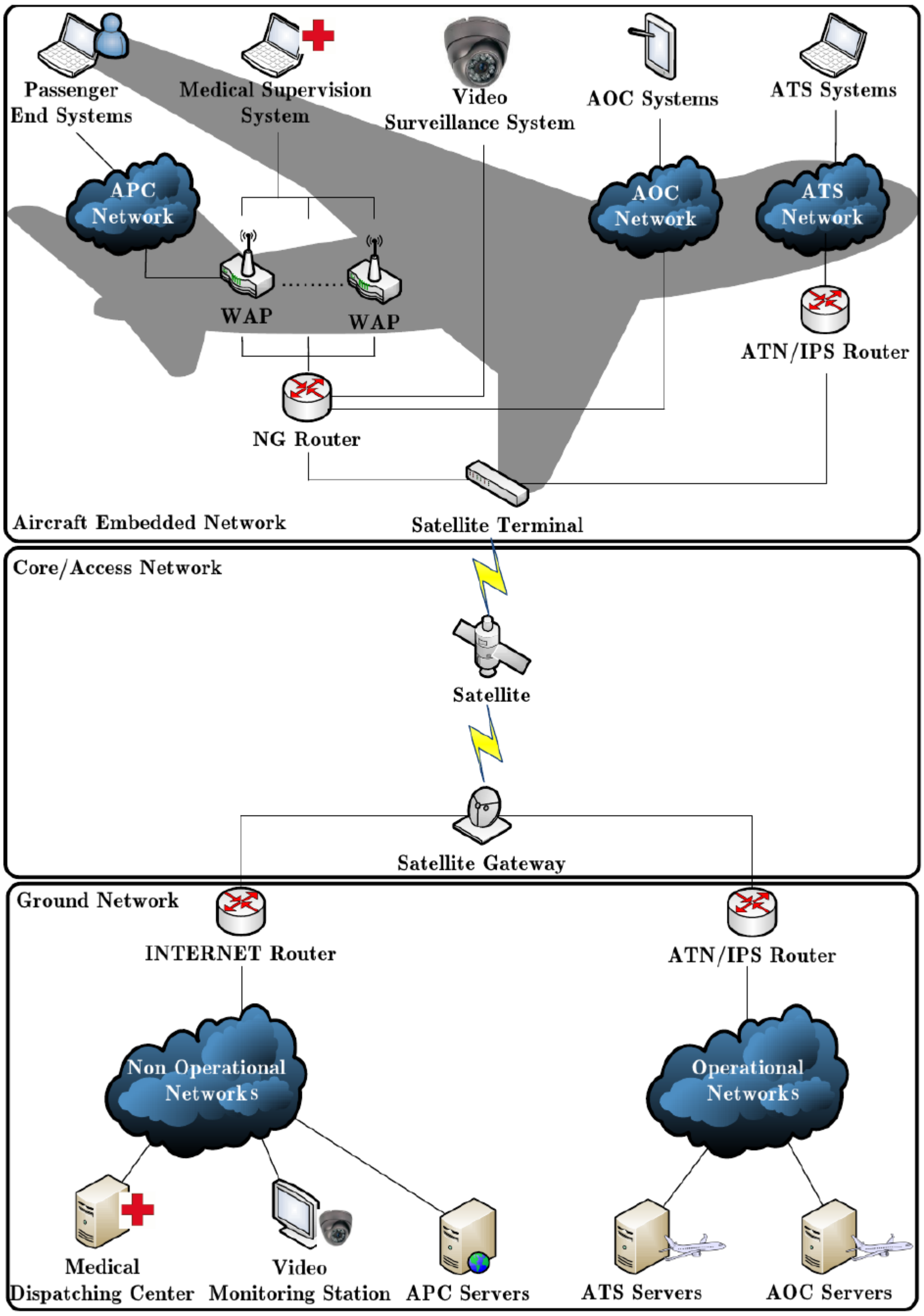


Figure 4.2: Satellite-based Aeronautical System Architecture

[†] The satellite-based aeronautical system architecture shown in figure 4.2 has been provided defined jointly to the ISAE partner in the scope of the FAST work package WP1.

The proposed architecture is decomposed into three networks:

1. The ground network is a relatively traditional system formed by two different routers:
   - ◇ An ATN/IPS router connected to the operational aeronautical networks (intermediate nodes, ATS and AOC servers, etc);
   - ◇ An Internet router for non-operational services (APC content servers, video monitoring stations, medical disptaching center, etc);
2. The access/core network is composed of the bent pipe satellite and the access gateway connected to both the ATN/IP and Internet ground routers;
3. The most challenging aspect was to design the aircraft embedded network, which is the upper part of figure 4.2. Two routers are deployed and connected to a satellite terminal:
   - ◇ An ATN/IPS router for operational ATS traffic connected to onboard ATS devices (*e.g.* CPDLC client, FMS, etc);
   - ◇ A NG (Next Generation) router for the emerging onboard services proposed in the FAST project. This router is connected to the operational AOC sub-network (*i.e.* based on the airline data services listed in the COCR document), the video surveillance system, and several WAPs (Wireless Access Points), dispatched across the aircraft. These WAPs provide a full coverage for the on-board Internet service and the medical supervision terminal, which requires, by the way, an absolute mobility when a medical emergency is declared during the flight. In order to provide a logical separation between the APC and medical supervision traffics when being connected to the same WAP, multi-SSID (Service Set IDentifier) access points are configured to provide several virtual access points within the same hardware device.

4.2.2 Satellite Terminal Architecture

Figure 4.3 shows the satellite terminal protocol stack used onboard the aircraft:

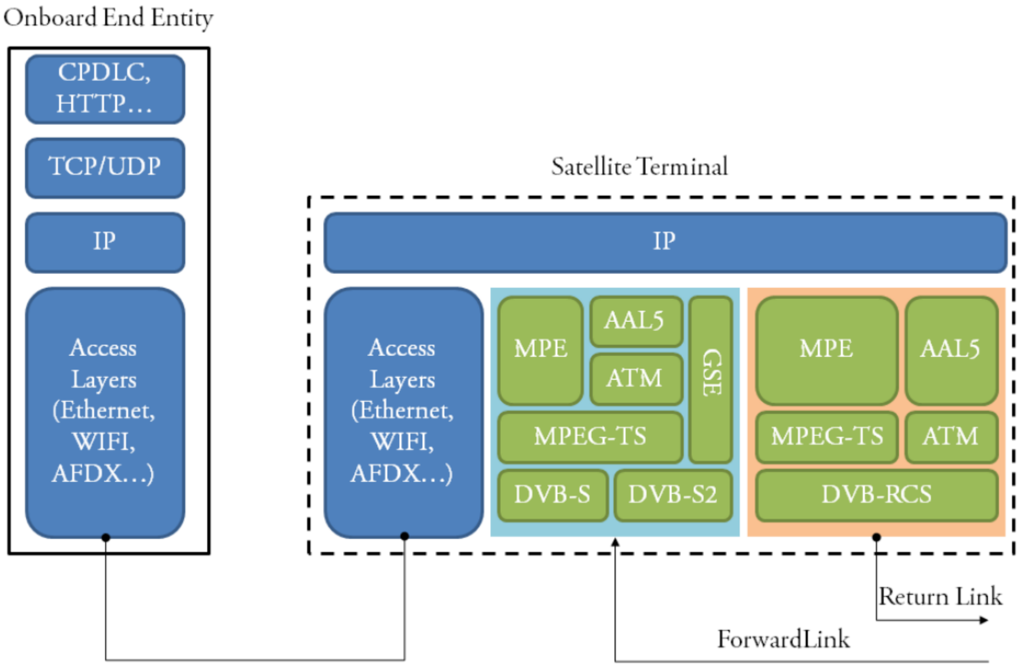


Figure 4.3: Satellite Terminal Protocol Stack

The proposed stack design is fully compliant with the BSM standard defined by the ETSI:

- the air interfaces rely on the DVB-S2 and DVB-RCS standards as defined in [ETSI 2006];
- it uses the MPEG (Moving Picture Experts Group 2) packet format at the MAC (Medium Access Control) level. Another option is to use the ATM (Asynchronous Transfer Mode) however the MPEG packet format has been chosen because of its better interoperability and protection against errors;
- it uses the MPE (Multi Protocol Encapsulation) protocol for the segmentation/reassembly schemes at the LCC (Logical Link Control) level;
- the network layer is based on the IP protocol;
- the separation between SD (Satellite Dependent) layers (*i.e.* physical and access layers) and SI (Satellite Independent) layers (*i.e.* network and layers above) is respected as recommended in [ETSI 2006].

After the terminal architecture description, we focus in the next subsection on the QoS policies defined to handle the service priorities and system resources.

4.2.3 Quality of Service and Resource Management

Figure 4.4 illustrates the QoS scheme used onboard in the FAST project. It is based on a two-level QoS policy[†] through a differentiation scheme for both network resource assignment and service priorities:

1. The first QoS level is deployed at the satellite independent side (*i.e.* bottom half of figure 4.4) within the NG router using a DiffServ (Differentiated Services) architecture [Babiarz *et al.* 2006]. DiffServ is a scalable and efficient set of network mechanisms used to classify and manage traffic QoS on IP networks. As several traffic classes are aggregated at the NG router, an IP classifier is used in order to associate each traffic class to a different and prioritized queue using a unique PID (Priority IDentifier). Unlike the NG router, the ATN router is not concerned by any QoS policy. In fact, there is no need to differentiate or prioritize the traffic on the ATN router as far as ATS is the only type of service passing through this router;

2. At the satellite dependent side (*i.e.* top half of figure 4.4), a second QoS policy is provided within the satellite terminal. A classifier, in charge of the input traffic, associates each traffic flow to the corresponding QID (Queuing IDentifier). In order to differentiate the traffic coming from the NG router and the ATN router, two ports are used, providing in this way a physical separation between the ATS traffic and the other types of traffic. This separation is mandatory and required by the ICAO SARPs (Standards and Recommended Practices) [ICAO 2002] for every aeronautical system design. The mapping established between the routers' output traffic and the corresponding QID's allows to manage the priorities between the different coming traffic flows. The IP traffic is segregated in several queues, each one with a different DSCP (Differentiated Services Code Point) code and associated to a unique QID. Finally, the MPEG packets are segmented, encapsulated at the SLC (Satellite Link Control) layer using the MPE protocol, then transferred to the MAC server block.

[†] As for the nominal satellite system architecture shown in figure 4.2, the QoS policy presented in this section is the result of our collaboration with the ISAE partner in the scope of the FAST work package WP1.

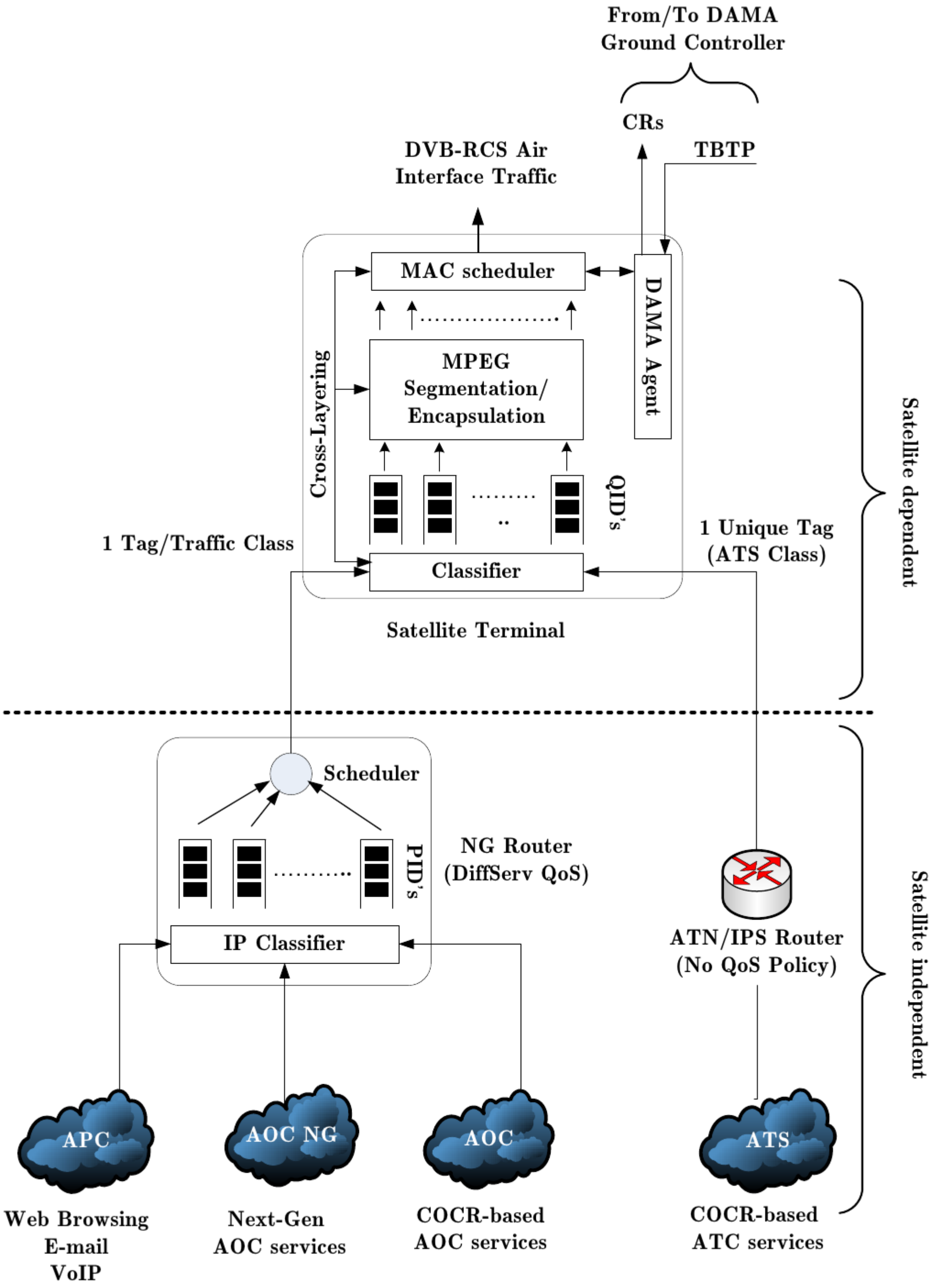


Figure 4.4: Embedded QoS Management Policy

Capacity and bandwidth assignments[†] have been provided at the SMAC (Satellite Medium Access) layer using the DAMA (Demand Assigned Multiple Access) protocol [ETSI 2006]. The DAMA protocol provides a BoD (Bandwidth on Demand) access to the system resources, which allows an easy and accurate usage of these resources when different traffic classes have to be managed.

The DAMA protocol works as follow:

[†] Capacity and bandwidth assignments have been defined by the ISAE partner. We have not participated directly in this specific task, however it is described here in order to give the reader more details about the mapping between the capacity assignments and QoS policy defined in figure 4.4.

- ◇ A DAMA controller is configured at the ground NCC and manages the shared capacity on the return link (*i.e.* the DVB-RCS link) among all the registered satellite terminals using a time slot allocation from the available frequencies within the transponder bandwidth;
- ◇ The satellite terminal assigns the resources allocated by the DAMA controller and sends the adequate CRs (Capacity Requests) to the ground.

According to the DVB-RCS standard [ETSI 2009], these CRs are estimated and categorized into four different classes depending on the required QoS:

1. *CRA (Continuous Rate Assignment)*: this CR class allows a fixed and static rate assignment by the satellite terminal. Therefore, there is no need for control exchanges between the NCC and the satellite terminal (*i.e.* fixed number of time slots). CRA capacity requests are usually used for a traffic which requires a fixed guaranteed rate and a minimum delay such as CBR (Constant Bit Rate) applications;
2. *RBDC (Rate Based Dynamic Capacity)*: this CR class provides a dynamic allocation at variable data rate. A maximum threshold is defined and provides a warranty for the satellite terminal in order to get the capacity needed when a connection is established. RBDC capacity requests are said to be absolute (*i.e.* corresponding to the full rate currently being requested). This class of CR could be combined with CRA capacity requests for typical ABR (Available Bit Rate) applications;
3. *(A)VBDC ((Absolute)Volume Based Dynamic Capacity)*: this CR class provides a "Best Effort" assignment on a volume basis (*i.e.* using either MPEG packets or ATM cells) after allocation of resources for CRA and RBDC CR classes. VBDC capacity requests are cumulative (*i.e.* each request is added to previous capacity requests from the same satellite terminal) whereas AVBDC capacity requests are absolute (*i.e.* the current request replaces the previous one from the same satellite terminal). The VBDC CRs could be used for delay tolerant traffic such as UBR (Unspecified Bit Rate) or IP traffic. AVBDC and VBDC are likely to support the same traffic classes;
4. *FCA (Free Capacity Assignment)*: the FCA CR class provides a capacity assignment without the need for any BoD beforehand and uses the capacity left after the assignment of higher CR classes (*i.e.* CRA, RBDC, and VBDC).

These QoS CR classes are ranked depending on the priority level of each traffic class as the following:

$$CRA > RBDC > (A)VBDC > FCA \tag{4.1}$$

The satellite terminal estimates the CRs depending on the queue sizes at the MAC level then sends them to the DAMA controller. At reception, the DAMA controller generates a TBTP (Terminal Burst Time Plan) and broadcasts it to the connected satellite terminals. Each satellite terminal takes the global assigned capacity sent by the DAMA controller and shares it among the different service classes, depending on the defined priority and QoS requirements.

In order to carry out an optimized estimation of the capacity needed for each service class, a cross-layer scheme [Srivastava & Motani 2005] is used between the MAC layer and the adaptation layer as shown in figure 4.4. The idea is the following: instead of sending directly the CRs to the DAMA controller, additional information (*e.g.* data rate estimation, IP queue sizes, etc) are gathered and used to estimate more accurately the CRs. Then, these information are sent to the MAC server block which generates the final CRs to the ground NCC. Finally, five ChID (Channel IDentifiers) are used within the SAC (Satellite Access Control) MPEG field and associated to a QID relevant to a given traffic class. These ChID's are assigned as follows:

- $ChID_0$ is assigned to the ATS traffic and has consequently the highest priority. As already depicted in chapter 2, the ATS traffic is sporadic and has low data rates, meaning there is no need for a continuous rate assignment for this specific channel. Therefore, the VBDC CR class is the best adapted. Moreover, having the highest priority guarantees the highest availability of the operational services requested in the COCR document;

- $ChID_1$ is assigned to the AOC traffic and has almost the same characteristics as $ChID_0$ except a lower priority;

- $ChID_2$ is a reserved channel for extreme situations such as medical emergencies or aircraft hijacking (*e.g.* need for HD video streaming). If there is no emergencies onboard the aircraft, the $ChID_2$ is not assigned and remains inactive. Considering the AOC NG traffic characteristics (*i.e.* real time VBR), both RBDC and VBDC CRs are associated to this channel ID;

- $ChID_3$ is assigned to the AOC NG traffic when normal situations occur, meaning when there is no critical events onboard the aircraft. As the telemedicine application is usually inactive in normal situations, the $ChID_3$ is then dedicated to the video surveillance application with low or medium quality definition. When a medical emergency or hijacking situation occurs, the corresponding CRs are assigned to the $ChID_2$ reserved channel;

- $ChID_4$ is assigned to the APC traffic which has the lowest priority level. Three CR classes are used, depending on the type of the running application: RBDC CRs are used for ABR applications which are tolerant to scheduling delay and jitter (*e.g.* best effort web browsing), VBDC CRs are used for UBR applications (*e.g.* e-mail, FTP), and finally FCA CRs are used when there is some capacity left after assigning all the other CRs.

When the priorities between the CRs shown in 4.1 are mapped with these channel ID's, a new priority ranking is defined:

$$RBDC[ChID_0] > VBDC[ChID_0] > RBDC[ChID_1] > VBDC[ChID_1]... \tag{4.2}$$

As we can see, priorities between the allocated channels drive the overall priority for the resource allocation, meaning that an VBDC CR which normally has a lower priority than a RBDC CR could be privileged if it is associated to a channel ID with a higher priority. Table 4.1 provides the final mapping between the capacity requests and the channel identifiers:

Table 4.1: Channel IDs and Capacity Requests Mapping

Channel ID	**Service**	**Assignment**	**Capacity Request**
$ChID_0$	ATS	Always	VBDC
$ChID_1$	AOC	Always	VBDC
$ChID_2$	Reserved	Video Surveillance	RBDC
		Telemedecine	RBDC+VBDC
$ChID_3$	AOC (NG)	Video Surveillance	RBDC
		Telemedecine	RBDC+VBDC
$ChID_4$	APC	Always	RBDC+VBDC+FCA

The work undertaken in the WP1 of the FAST project represents a first step toward a system architecture compliant with a differentiated QoS and resource management in the ATM system, where priorities shall be managed at several points of the architecture. This first contribution can be summarized as follows:

- ◇ Aeronautical priorities between operational and non operational traffic classes are strictly respected throughout a two level QoS policy at both the NG router and the satellite terminal. A mapping between QoS outputs of the NG router (using Diffserv) and QoS inputs at the satellite terminal (using a cross layered approach and the DAMA protocol) is provided for an optimized assignment of the system resources;

- ◇ The architecture design is fully compliant with the BSM reference architecture for IP-based satellite links with a clear segregation between SD and SI layers as depicted in figure 4.5;

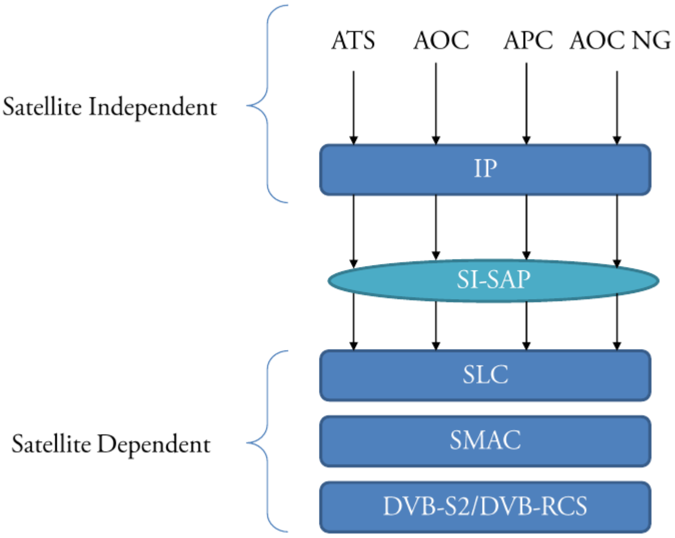


Figure 4.5: Seperation between SD and SI Layers

- ◇ The system architecture is also compliant with the ICAO standards and recommended practices SARPs with a physical separation between the ATS traffic and the other traffic classes.

At this point of the study, the QoS and priority management issues have been partially addressed. Additionally, the service priorities shall also be managed within the security framework defined in WP3. Indeed, at the end of the WP1, all the partners have been asked to provide the security requirements for each onboard service (*i.e.* operational services, telemedecine, video surveillance, passenger services). These security requirements are obviously different from one application to another, meaning that one security mechanism configuration would probably not cover the security requirements for all the traffic classes. Consequently, several security solutions have been considered and discussed as shown in the next subsection.

4.2.4 Toward an Adaptive Security Framework

In order to provide a secure satellite-based aeronautical system architecture, three different solutions have been inspected:

1. *One security policy for all the onboard services*: this solution is the one that willingly comes to designer's minds when they have to deal with a similar situation. The idea is to use a single security policy (meaning a fixed set of security algorithms) to provide the strongest security level and therefore cover all the security requirements (*i.e.* confidentiality, authentication, integrity, availability, non repudiation) for all the considered traffic classes. Note that it is possible to use one security policy with different entries (*i.e.* set of security algorithms) for each service. For

instance, when the IPSec protocol is used, it is possible to specifically use one entry for each Security Association (SA) in the SPD (Security Policy Database). Each SA is defined for a single unidirectional flow of data, covering traffic distinguishable by a unique selector using the triplet (IP destination address, Security Parameter Index, Security Protocol). However, this solution uses a single IPSec security sub-protocol (*i.e.* namely AH or ESP) which does not cover necessary all the security requirements while providing the strongest security level. Besides, for some onboard services, it is more adequate to use security protocols at higher layers of the protocol stack (*e.g.* TLS for web browsing for instance). This solution might be the easiest to implement in a non-constrained bandwidth environment where resources are widely available. However, as mentioned in chapter 2, this is absolutely not the case for the air-ground link, and this solution will likely induces a large overhead load. Therefore, this alternative has been discarded;

2. *One security mechanism for each onboard service*: this alternative has been rejected too because it requires the use of a security feature on a client basis. Moreover, some onboard end entities are unprotected, meaning they do not provide a security layer. Besides, as for the previous security solution, implementing one security mechanism on each end entity is not cost-effective and may induce an increasing overhead load in a presence of a high number of connections between onboard entities and ground servers;

3. *An adaptive security for all the onboard services*: this solution represents the final security architecture onboard the aircraft. As one infrastructure is meant to deal with several traffic flows coming from different embedded entities, a security proxy is useful to manage all the secure communications between the air and the ground. This adhoc solution has the advantage to provide a differentiated security as requested by the onboard operational and non operational services. Moreover, it allows a security defense in depth through the use of several security mechanisms, covering many security services. Besides, the priority between the traffic classes could be integrated at a security design level additionally to the QoS policy already provided in section 4.2.3. Finally, an adaptive security solution provides enough granularity and accuracy to find the optimal trade-off between the required security level and the induced network and system overhead.

Table 4.2 resumes the advantages and drawbacks of these alternative security solutions regarding the challenges introduced in chapter 2:

Table 4.2: Comparison of Security Architectures for the FAST Project

	Differentiated Security	**Priority Management**	**Defense in Depth**	**Minimize Overhead**
- One security policy for all services	×	×	×	×
- One security policy per service	✓	×	×	×
- Adaptive Security	✓	✓	✓	✓

Legend
✓: covered
×: not covered

Therefore, an adaptive secure satellite-based system architecture is chosen. The next section illustrates how the nominal satellite-based system architecture is improved and reinforced using the SecMan proxies.

4.3 SecMan: An Adaptive Security Architecture

In this section, the adaptive security management proposal is depicted with a particular emphasis on the protocol stack and design of the SecMan module. First, the secure satellite-based system architecture is introduced.

4.3.1 Secure Satellite-based System Architecture

Figure 4.6 gives an overview of the onboard secure system architecture built upon the nominal architecture proposed in the previous section:

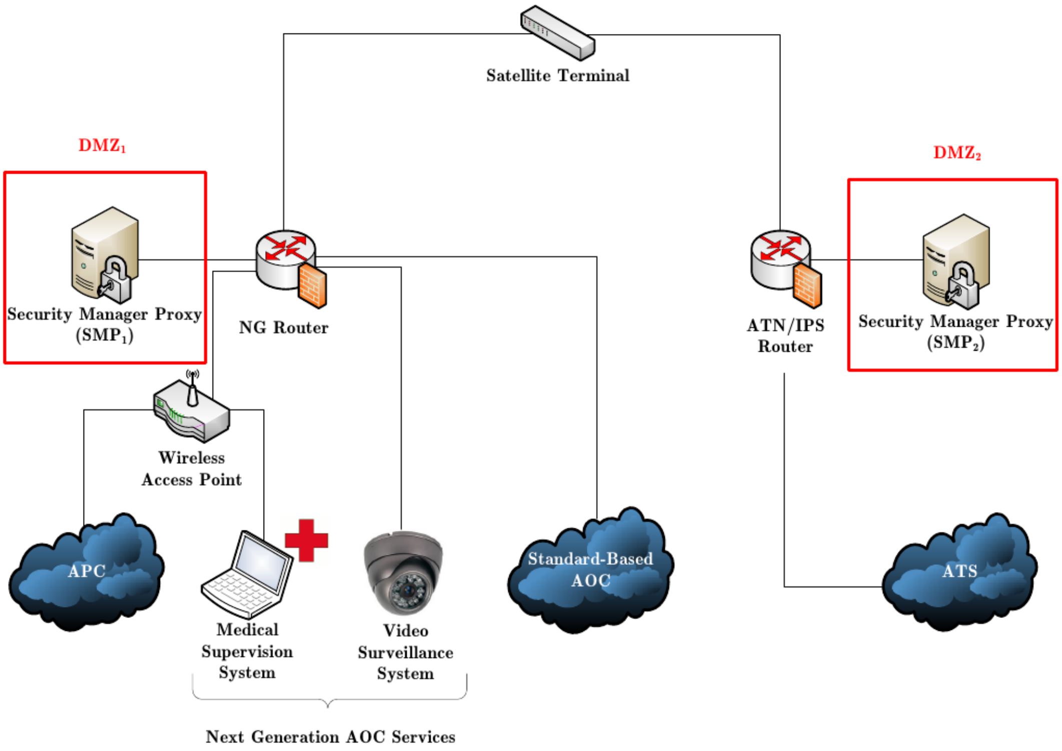

Satellite Terminal
DMZ₁
Security Manager Proxy (SMP₁)
NG Router
DMZ₂
ATN/IPS Router
Security Manager Proxy (SMP₂)
Wireless Access Point
APC
Medical Supervision System
Video Surveillance System
Standard-Based AOC
ATS
Next Generation AOC Services

Figure 4.6: Onboard Secure Aeronautical System Architecture

Four nodes are added to the onboard network:

- Two SMPs (Security Manager Proxy) are connected respectively to the NG and ATS routers and located into DMZs (Demilitarized Zones). Request connections are redirected to the SMPs where the SecMan module treats individually each request and establishes the adapted security policy. SMP_1 uses an inter-class operational mode (*c.f.* section 4.3.4) to handle three types of traffic (APC, AOC, and AOC NG), so it uses both the priorities and the system state (*i.e.* available network and system resources) to define the security policies. On the other side, SMP_2 uses an intra-class operational mode (*c.f.* section 4.3.4) where only the ATS traffic is handled. In this case, it is useless to consider the priority as only one type of traffic is going through SMP_2. Thus, the security policy will be defined only according to the system state;
- Two stateful inspection firewalls are added on both the NG and ATN routers in order to keep track of TCP and UDP flows. These firewalls are renowned to be efficient and powerful against complex attacks. Indeed, stateless packet filtering firewalls do not keep track of connection's state, they simply look at the IP header fields (*e.g.* source and destination addresses or ports) or options/flags

in the transport headers. Therefore, they are not sufficient to avoid advanced intrusion attempts: for instance, a malicious remote host could send packets which seems to be part of an established TCP connection (*i.e.* with the adequate TCP ACK flag set), but which in reality are not. This allows an attacker later to map the local network as if the firewall did not even existed. In this way, attacks against weaknesses in the TCP/IP protocol stack (*e.g.* reverse TCP attacks, DoS attacks on TCP) [Schuba *et al.* 1997] can pass through the firewall by claiming to be part of an established TCP session. These problems are avoided using stateful inspection firewalls, which look at both the network and transport level data. They keep track of the initial TCP packets with the SYN flag set and allow the packets to be forwarded back only until the FIN packet is sent and acknowledged. Application layer firewalls have been envisaged, but their high computation load and heavy filtering processing slow down the air-ground communications and would induce a supplementary delay which cannot be tolerated by operational services. Whereas application layer firewalls require each datagram to make a trip up and down the whole protocol stack, packet filtering firewalls are much faster because they do not require any knowledge or cooperation of users at the application layer [Ingham & Forrest 2002].

In the next section, the SecMan architecture design is introduced.

4.3.2 SecMan Architecture

Figure 4.7 depicts the SecMan architecture:

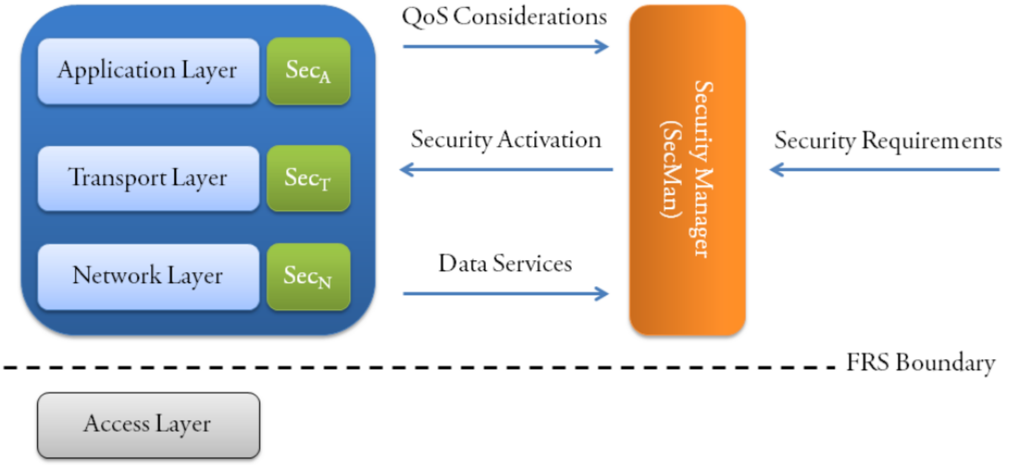


Figure 4.7: SecMan Protocol Stack

The SecMan module is intended to work above the FRS boundary on the control plan. This has the advantage to provide a complete independence regarding the access layers(*e.g.* AFDX[2], Ethernet, etc).

The SecMan module inputs are:

- *Data services*: every data stream is redirected to the SMP with the priority associated to its traffic class (ATS, AOC, AOC NG, or APC);
- *Security requirements*: minimum security requirements[†] are needed for each onboard service. Security requirements for operational services have been extracted from the COCR document (section

[2] AFDX (Avionics Full-Duplex Switched Ethernet) is a deterministic Ethernet technology developed for safety-critical avionics.

[†] The security requirements for the telemedicine and video surveillance applications have been provided by the MEDES and VODEA partners in the scope of the WP3 work package.

B.1 of Appendix B). The security requirements for the new services have been expressed by each partner in the scope of the WP3 (section B.3 of Appendix B);

- *QoS considerations*: information relevant to the available system resources are sent to the SecMan module and used when a security policy has to be established by triggering security mechanisms of the TCP/IP protocol stack (denoted Sec_A, Sec_T, and Sec_N in figure 4.7).

Besides, the SecMan design relies on a cross-layer approach in order to exchange the QoS, security, and data service information, collected at different layers of the TCP/IP protocol stack. The SecMan module can be seen as a shared component by all the layers where information received from the protocol stack can be regrouped and used properly. Unlike to other cross-layer design techniques [Srivastava & Motani 2005] (*e.g.* upward/downward information flow, merging of adjacent layers, etc), using SecMan as a transverse control module has the advantage to not bring any modification on the classical layered design of the TCP/IP model. If we take a closer look to the design of the SecMan module, we can distinguish five different and interdependent submodules working together in order to establish the final security policy for each data flow.

4.3.3 SecMan Algorithm Design

The flow chart of figure 4.8 gives an overview of the SecMan components:

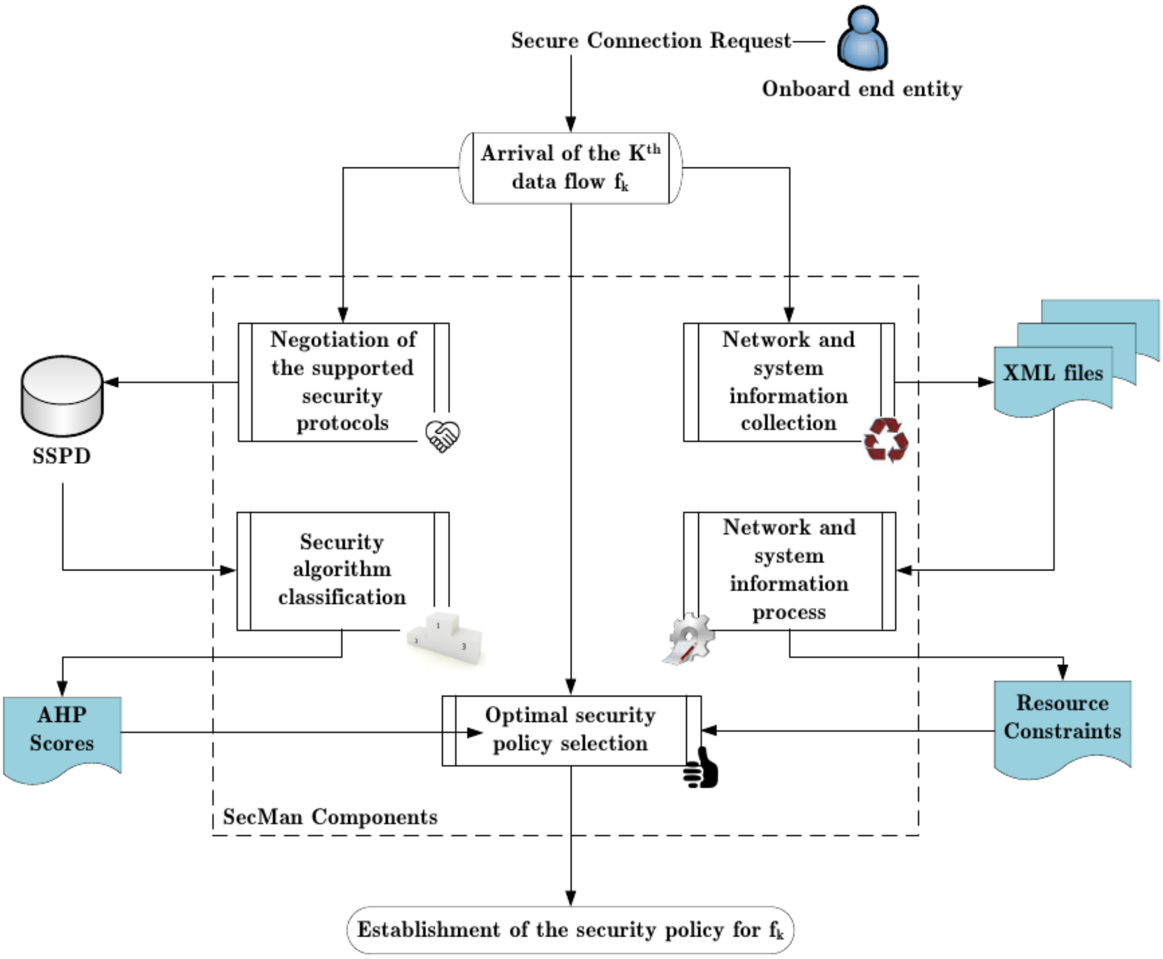

Secure Connection Request
Onboard end entity
Arrival of the Kth data flow f_k
Negotiation of the supported security protocols
Network and system information collection
SSPD
XML files
Security algorithm classification
Network and system information process
AHP Scores
Optimal security policy selection
Resource Constraints
SecMan Components
Establishment of the security policy for f_k

Figure 4.8: SecMan Flow Chart

When a secure connection between an onboard client and a ground server is requested, five components are sequentially triggered:

1. *Negotiation of supported security protocols*: this is the very first step toward the selection of the optimal security policy. The objective is to establish a common database which contains a list of the supported security protocols and algorithms between the onboard SMP and the ground server. For this purpose, a network protocol called SNSSP (Secure Negotiation of Supported Security Protocols) is defined and formally verified in section 4.4.1;

2. *Security protocol and algorithm classification*: the next step is to classify the set of security protocols and algorithms previously negotiated according to:

   ◇ the provided security level using several criteria such as cipher or signature key lengths (*c.f.* section 4.4.2.2);

   ◇ the system and network costs they may have (*e.g.* packet size overhead);

   As many criteria are considered for this specific purpose, a MCDMA (Multi-Criteria Decision Making Algorithm) is used to establish a ranking among the supported security mechanisms according to the selected criterion. MCDMA is a particular class of algorithm used to support decision making processes in complex systems (*i.e.* social, economic, management domains) where various and conflicting criteria are involved. Most of these methods use a series of mathematical computations to finally select the best option from a set of several alternatives based on an overall score. In our work, the adopted MCDMA approach is called AHP (Analytical Hierarchical Process) [Saaty 2000]. The SecMan classification processing step is described in section 4.4.2;

3. *Network and system information collection*: at this step, the SecMan module is informed about the system state and available resources thanks to a client-server adhoc application[†] developed jointly to the ASTRIUM partner. Details are provided in section 4.4.3;

4. *Network and system information processing*: the information received at the previous step are used to estimate the network and system resources available at this point. Depending on the AHP scores computed at the second step of the framework and the priority relevant to the data flow f_k, an optimal security policy is selected in the next phase. The network and system information processing step is depicted in section 4.4.4;

5. *Optimal security policy selection*: the final step is to select the best security policy. In this case, best does not mean the strongest security level, but the highest trade-off between the security level and network/system costs. The selected policy can use one or several security mechanisms as explained in section 4.4.2.1 in order to fulfill the security defense in depth requirement. This step can be seen as a typical *"knapsack problem"* where the available resources (received in step 3) should be respected according to the network and system costs of the selected security policy (steps 2 and 4).

In order to provide enough flexibility for connection requests coming from onboard clients, several operational modes have been integrated into the SecMan module.

[†] This application has been developed jointly to the ASTRIUM partner using information stored by the SATEM satellite emulator.

4.3.4 SecMan Operational Modes

SecMan is designed to work in two operational modes:

1. *Intra-class mode*: used when only one type of traffic is passing through the SMP. In this mode, the priority is not considered and the security policy is established under QoS constraints. In figure 4.6, SMP_2 located in the DMZ_2 works in the intra-class mode;
2. *Inter-class mode*: used when different traffic classes (*e.g.* AOC NG, APC) are passing through SMP. In this case, service priorities and QoS constraints are considered by SecMan to establish the adequate security policy. In figure 4.6, SMP_1 located in the DMZ_1 works in inter-class mode.

Besides these operational modes, SecMan can be compliant with any kind of connection requests through four security modes:

1. *Insecure mode*: in this mode, packets are routed and transmitted without being secured using a `ByPass` policy. This mode is useful for data flows which do not require any security treatment. This mode can be also useful when available resources are too low and do not allow the establishment of secure communications (*e.g.* high attenuation of the satellite signal + emergency situation);
2. *Secure transparent mode*: in this configuration, everything remains transparent for end entities given that security mechanisms are deployed at the network level (*e.g.* IPSec);
3. *Secure transport proxy mode*: in this mode, an end-to-end transport level secure connection is established (*e.g.* TLS). In this mode, every application is secured thanks to a secure socket connection (*e.g.* https, imaps, ftps). Note that this mode is only compatible with TCP-based services, then it does not support UDP-based services (*e.g.* AOC NG services). In fact, the secure transport proxy mode mainly uses the TLS protocol which is not compatible with UDP flows. More details can be found in appendix C.8 dedicated to TLS and the validation and experiment section 4.5 dedicated to the FAST testbed;
4. *Secure application proxy mode*: SecMan established an end-to-end application level secure connection using dedicated security protocols (*e.g.* SSH).

These security modes can be combined with the intra-class and inter-class modes as we will see in the last section of this chapter. They are also very useful in order to measure the benefits and gain of the adaptive security policy. In section 4.5.3 these security modes are tested in the scope of four reference scenarios. Now that the SecMan components and functional modes have been introduced, the very first step of the framework, namely the SNSSP protocol, is explained and formally proofed using a formal model checking tool.

4.4 SecMan Components

4.4.1 Negotiation of Supported Security Mechanisms

In order to use the AHP algorithm and evaluate the security level of each policy, it is important to establish a set of security mechanisms supported at the same time by the onboard SMP and the ground server involved in the exchange. These security alternatives are negotiated beforehand using the SNSSP protocol to establish a common SSPD (Supported Security Protocol Database). The reason why we need such a protocol is that the SMP may provide some ciphers or protocols which are not supported by the ground server (and vice versa), making impossible the establishment of the requested secure connection. Similar negotiation schemes can be found in the literature such as ISAKMP (Internet Security Association and Key Management Protocol) [Maughan *et al.* 1998] for establishing security associations (*e.g.* IKE for IPSec), however these schemes are hardly reusable because they have been originally designed for specific protocols. Besides, the security of this negotiation protocol is fundamental. As long as this is the first step driving all the remaining SecMan components, it cannot not be compromised. That is why a formal model checking of this protocol is conducted in section 4.4.1.3.

4.4.1.1 SNSSP Security Requirements

The SNSSP protocol must provide at least the following security properties:

- *Protection against replay attacks*: replay attacks can be confusing when a new SSPD has to be established between the same communicating parties. An attacker can for example replay old messages containing out-of-date supported security protocols on the ground side, leading to false entries in the SSPD database. Nonces (Number used Once) shall be useful to provide protection against this specific class of attack;
- *Entity authentication*: impersonation attacks are very likely to occur, particularly at the ground side where the server could be connected to a public network such as the Internet. The entity authentication can be easily provided using digital signatures and public key certificates;
- *Message integrity*: the negotiated set of supported security protocols has to be protected against any alteration or modification when being transfered between the SecMan and the ground server. Cryptographic primitives such as hashes and digital signatures should be used to provide message integrity.

Confidentiality is not mandatory here as there is no sensitive data carried out through the SNSSP messages. The cryptographic credentials mentioned above (*e.g.* certificates, public/privates keys) are distributed using the PKI defined in chapter 5. In the next subsection, we explain how the SNSSP messages are exchanged.

4.4.1.2 SNSSP Protocol Description

Figure 4.9 shows a sequence diagram relevant to the SNSSP protocol. Three entities are involved in the negotiation procedure:

- E_1 is the onboard entity (*e.g.* an ATS terminal) requesting a secure connection with the ground;
- E_2 is the ground entity (*e.g.* an operational server);
- SMP is the SecMan proxy.

E_1 asks for a secure connection with E_2, and SMP initiates the negotiation in order to establish the common SSPD by sending a $Request\_SSP$ message containing a random number $Nonce_1$ to avoid replay attacks. The SSPD is stored onboard and has the following structure:

$$<IP_d, Port_d, Protocol, SSP_{negotiated}, h_{E_2}, Expiration\_date> \tag{4.3}$$

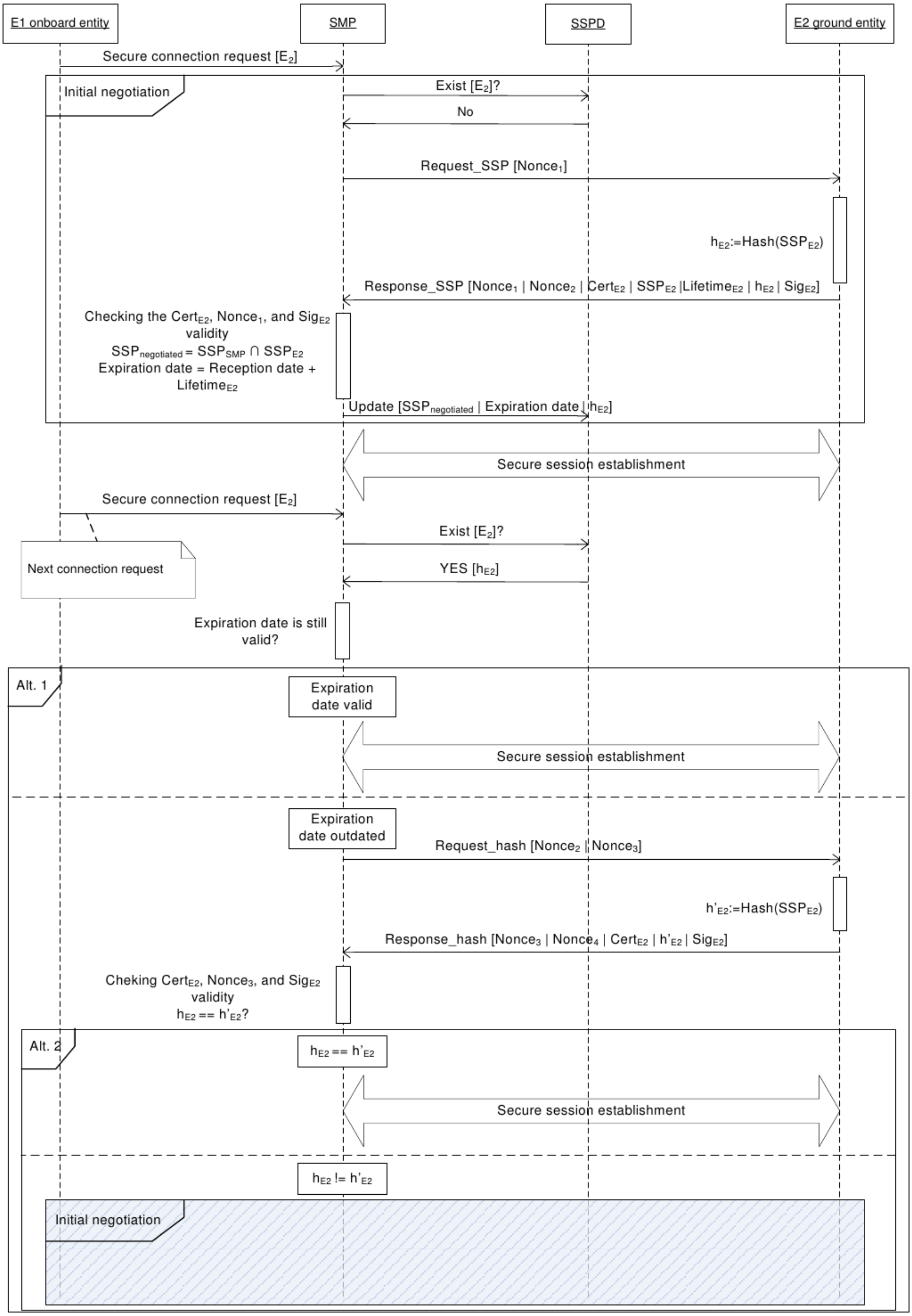

E1 onboard entity
SMP
SSPD
E2 ground entity
Secure connection request [E_2]
Initial negotiation
Exist [E_2]?
No
Request_SSP [$Nonce_1$]
h_{E2}:=Hash(SSP_{E2})
Response_SSP [$Nonce_1$ | $Nonce_2$ | $Cert_{E2}$ | SSP_{E2} |$Lifetime_{E2}$ | h_{E2} | Sig_{E2}]
Checking the $Cert_{E2}$, $Nonce_1$, and Sig_{E2} validity
$SSP_{negotiated}$ = SSP_{SMP} ∩ SSP_{E2}
Expiration date = Reception date + $Lifetime_{E2}$
Update [$SSP_{negotiated}$ | Expiration date | h_{E2}]
Secure session establishment
Secure connection request [E_2]
Next connection request
Exist [E_2]?
YES [h_{E2}]
Expiration date is still valid?
Alt. 1
Expiration date valid
Secure session establishment
Expiration date outdated
Request_hash [$Nonce_2$ | $Nonce_3$]
h'_{E2}:=Hash(SSP_{E2})
Response_hash [$Nonce_3$ | $Nonce_4$ | $Cert_{E2}$ | h'_{E2} | Sig_{E2}]
Cheking $Cert_{E2}$, $Nonce_3$, and Sig_{E2} validity
h_{E2} == h'_{E2}?
Alt. 2
h_{E2} == h'_{E2}
Secure session establishment
h_{E2} != h'_{E2}
Initial negotiation

Figure 4.9: SNSSP Protocol Specifications

Each record in the SSPD contains the following information:

- IP_d is the IP address of the server entity E_2;
- $Port_d$ is the destination port of the server entity E_2;
- $Protocol$ is the used protocol;
- $SSP_{negotiated}$ is the negotiated set of Supported Security Protocols;
- h_{E_2} is the hash of SSP_{E_2} (E_2 supported security protocols) used to avoid later negotiation as described below, meaning that: $h_{E_2} = Hash(SSP_{E_2})$. $Hash$ could be any hash function that produces a HMAC of the SSP_{E_2} record. For instance, the SHA-1 hash function can be used to produce a 160 bits message digest. The use of SHA-1 is even recommended and preferred instead of the MD5 hash function which is now considered as weak again cryptanalysis and collision attacks [Yu *et al.* 2009]. Besides, there is no need for a shared secret here because E_1 does not compute any hash on its own, it just compares the hashes sent by E_1 after sending a request for a new hash (*c.f.* the SNSSP pseudo-code algorithm at the end of the section from **17:** code line);
- $Expiration\_date$ is the expected expiration date (*i.e.* when the negotiated set of security protocols is no longer valid) of the corresponding record.

At the reception of the request, E_2 computes a hash h_{E_2} of the supported security mechanisms SSP_{E_2} then it sends a *Response_SSP* message containing:

- the certificate of the server $Cert_{E_2}$ retrieved from the adequate certificate authority and provided by the PKI defined in chapter 5. This certificate provides a proof of identity to the onboard SMP;
- $Nonce_1$ received in the *Request_SSP* message plus a new generated random number $Nonce_2$ used to guarantee the freshness of the response;
- the list of supported security mechanisms SSP_{E_2} at the server side;
- the generated hash h_{E_2};
- a lifetime $Lifetime_{E_2}$ used by the SMP to define the expiration date of the security mechanisms supported by the ground server. Thanks to $Lifetime_{E_2}$ and h_{E_2}, SMP can avoid later negotiations if the supported security mechanisms have not been modified on the server side;
- a signature of the *Response_SSP* message Sig_{E_2} generated using the private key of the server E_2. Signature provides the message integrity expressed in security requirements (*c.f.* section 4.4.1.1).

When the SMP receives the *Response_SSP* message:

- it checks the validity of the server certificate using a certificate revocation scheme as described in the last chapter of the thesis;
- it compares the received $Nonce_1$ with the nonce it actually generated in *Request_SSP*;
- it verifies the signature Sig_{E_2} using the public key binded to the digital certificate.

If these three operations are completed successfully, the SMP establishes a new entry in the SSPD database by combining the received set of supported security protocols with its own set:

$$SSP_{negotiated} = SSP_{SMP} \bigcap SSP_{E_2} \tag{4.4}$$

The expiration date is deduced as the sum of the reception date and the lifetime $Lifetime_{E_2}$ contained in the *Response_SSP* message. Finally, SecMan is able to proceed to the next step and classify the security protocols in $SSP_{negotiated}$. For later negotiations with the same ground server E_2, there are two cases:

1. *Expiration_date is valid*: this means that the server always uses the same security protocols, then $SSP_{negotiated}$ is always valid and can be used by SecMan in the next step;

2. *Expiration_date is no longer valid*: this means that the server has may be changed its supported security protocol set, and then the SMP has to check if there are some modifications or not at the server side. For this purpose, the SMP sends a *Request_hash* message to the ground entity and asks for a second hash h'_{E_2} of the supported security protocol set. The nonce $Nonce_2$ and a new nonce $Nonce_3$ are added to the request message for freshness purposes. The server sends back a *Response_hash* message containing $Nonce_3$, $Nonce_4$, $Cert_{E_2}$, the new hash h'_{E_2} and the signature. The SMP receives the response, proceeds to the same verification as for the initial negotiation and then compares the hash stored in the SSPD and the newly received hash:

   ◇ *if* $h'_{E_2} \neq h_{E_2}$: a new negotiation is started from the beginning;

   ◇ *if* $h'_{E_2} = h_{E_2}$: a secure connection is directly established. Indeed, there is no need to repeat the negotiation procedure as far as the SSP_{E_2} is still the same, even if *Expiration_date* is out of date. As noted before, the lifetime and hash are used to avoid excessive waste of resources. The lifetime metric is sufficient to know if a new negotiation procedure is needed or not. But, quantitatively, the *Request_hash* and *Response_hash* are lightweight control messages compared to the *Response_SSP* message which carries out the SSP_{E_2} data. For a matter of comparison, the SHA-1 hash sizes measured in the FAST experiments are equal to 20 bytes whereas the SSP_{E_2} are stored in 1000 bytes size XML files.

When the negotiation phase is finished, the $SSP_{negotiated}$ is made available to the multi-criteria classification functional block. The following pseudo-code shows all the steps relevant to SNSSP:

Algorithm 4.1 Negotiation of Supported Security Protocols

```
if first negotiation with ground entity $E_2$ then
    SMP sends $Request\_SSP[Nonce_1]$ to $E_2$;
    $E_2$ computes a hash of $SSP_{E_2}$;
    $E_2$ asks for a public certificate $Cert_{E_2}$;
    $E_2$ sends $Response\_SSP[Nonce_1|Nonce_2|SSP_{E_2}|Lifetime_{E_2}|Cert_{E_2}|h_{E_2}|Sig_{E_2}]$;
    if ($Nonce_1$, $Cert_{E_2}$, and $Sig_{E_2}$ are valid) then
        SMP computes $SSP_{negotiated}$;
        SMP stores $SSP_{negotiated}$ in the SSPD database;
    else
        Negotiation fails;
    end if
else
```

```
13:   **if** expiration_date is valid **then**
14:     SMP uses the same $SSP_{negotiated}$ in the SSPD database;
15:   **else**
16:     SMP sends $Request\_hash[Nonce_2, Nonce_3]$;
17:     $E_2$ computes a new hash of $SSP_{E_2}$;
18:     $E_2$ sends $Response\_hash[Nonce_3|Nonce_4|Cert_{E_2}|h'_{E_2}|Sig_{E_2}]$ to SMP;
19:     **if** ($Nonce_3$, $Cert_{E_2}$, and $Sig_{E_2}$ are valid) **then**
20:       **if** ($h_{E_2} == h'_{E_2}$) **then**
21:         SMP uses the same $SSP_{negotiated}$ in the SSPD database;
22:       **else**
23:         Go to (2:) ;
24:       **end if**
25:     **else**
26:       Negotiation fails;
27:     **end if**
28:   **end if**
29: **end if**
```

The SNSSP protocol is designed in a way it is immunized again the attacks identified in section 4.4.1.1. However, there is no formal proof that the protocol actually fulfills all the security properties as it should. Thus, we must check if all the security goals are effectively covered by the protocol before being implemented. In the next section, we provide a formal verification of the SNSSP protocol using a publicly available automatic model checking tool called AVISPA (Automated Validation of Internet Security Protocols and Applications) [Armando *et al.* 2005].

4.4.1.3 Formal Verification of the SNSSP Protocol

It has been shown in the past that security protocols can never be considered 100% safe even after many years of deployment. The best example is the Needham-Schroeder authentication protocol [Needham & Schroeder 1978] which has been used in public and private networks for 16 years before Lowes discovered a security hole in the protocol specification [Lowe 1995].

Motivation

In order to make these protocols more robust and secure, security protocol verification techniques allow the designer to verify if the security specification actually reaches the security requirements fixed beforehand. Automatic tools provide systematic approaches to verify the security properties without loosing much time and being confronted to computation errors. Generally, these tools use two approaches:

1. *Computational approaches*: rely on Turing machine-based intruder models. Security properties are considered as a string of bits and are verified regarding the computational resources used by the intruder to find the decryption key;

2. *Formal method approaches*: assumes that the intruder cannot conduct cryptanalysis attack like in computational approaches, meaning that they assume perfect cryptography in the verification process (*i.e.* the original message can only be decrypted if the intruder has the appropriate decryption key). Several formal model checking tools have been provided to automate the verification procedure. Usually they use either the BAN (Burrows Abadi Needham) logic [Burrows *et al.* 1990], or a state exploration approach where all possible execution paths are reached and each visited state

is tested regarding a set of preconditions. If security properties are not reached at a particular state, an exploit trace of the attack is built from the initial state to the vulnerable state. The verification process can be bounded (fixed and finite number of states and sessions) or unbounded (infinite number of users or sessions) depending on the purpose of the verification.

In order to verify the robustness of the SNSSP protocol, the formal automatic security analyzer AVISPA is used. The formal verification of the SNSSP protocol is divided into two steps:

1. *Protocol specification using high level languages*: in this phase, two high level languages are used, namely HLPSL (High Level Protocol Specification Language) [Chevalier *et al.* 2004] and CAS+ [Saillard & Genet 2011] to specify the SNSSP messages described in section 4.4.1.2. These languages offer a high level of abstraction and allow a very detailed specification of every security protocol aspects (*e.g.* roles, cryptographic operators, intruder models, etc). The specifications are then automatically translated into a lower level language, named IF (Intermediate Format) using the HLPSL2IF translator (*cf.* section C.4 of Appendix C);
2. *Verification of SNSSP specifications using the AVISPA back-ends*: in this step, the SNSSP specifications are used by the AVISPA tool to check if the security requirements are respected or not. AVISPA uses 4 different verification back-ends and we tested them all in order to check if there are some inconsistencies in the SNSSP protocol specifications. In software architecture, *"back-end"* is a generic term used to refer to programs called indirectly by users. In the case of AVISPA, these back-ends are formal checking model tools receiving the translated protocol specifications for security verification.

In the next subsection we provide a brief comparison between the existing formal model checking tools found in the literature and explain why we have chosen AVISPA instead of other tools.

Overview of Formal Model Checking Tools of Security Protocols

Previous studies provided comparative analysis of existing formal model checking tools of security protocols. Table 4.3 shows a comparison of these tools as it appears in [Patel *et al.* 2010]:

Table 4.3: Comparison of Existing Formal Model Checking Tools [Patel *et al.* 2010]

Reference	Availability	Falsification	Verification	Termination
OFMC	✓	✓	bounded	✓
CL-Atse	✓	✓	bounded	✓
SATMC	✓	✓	bounded	✓
TA4SP	✓	✓	unbounded	✓
AVISPA	✓	✓	both	✓
FDR/Casper	✓	✓	bounded	✓
HERMES	✓	✓	unbounded	✓
Interrogator	×	✓	bounded	✓
NRL Analyzer	×	×	unbounded	✓
Brutus	×	✓	bounded	✓
ProVerif	✓	✓	unbounded	✓
Athena	×	✓	both	✓
Scyther	✓	✓	both	✓

Legend: ✓: covered, ×: not covered

Four criteria have been used in the comparison process:

◇ *Availability*: is the tool public/free?

◇ *Falsification*: is the tool able to provide a report when attack traces are detected?

◇ *Verification*: is the tool able to provide bounded and/or unbounded verification of the protocol? This is an important criterion as some attacks can use parallel sessions to gain more knowledge;

◇ *Termination*: is the tool able to indicate if the verification procedure ended successfully or not?

For instance, Athena [Song *et al.* 2001] is a strong formal checking tool and it uses bounded and unbounded verification techniques, but it is not publicly available. Publicly available tools such as HERMES does not provide unbounded verification. From table 4.3, two verification tools arise: AVISPA and Scyther [Cremers 2006]. We have finally chosen AVISPA because:

◇ It is freely and publicly available;

◇ The tool provides a friendly GUI tool called SPAN (Security Protocol ANimator for AVISPA)[3] which has been very useful in the intruder simulation of the SNSSP protocol;

◇ It is supported by 4 different back-ends, which provide several possibilities to verify one security protocol from different perspectives: OFMC (On the Fly Model Checker) [Basin *et al.* 2004], CL-Atse (Constraint Logic based Attack Searcher) [Basin 1999], SATMC (SAT based Model Checker) [Armando & Compagna 2004], and TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols) [Boichut & Oehl 2004].

◇ It supports two different high-level and modular languages, namely CAS+ and HLPSL.

In the protocol validation phase, we used the Dolev-Yao Intruder model since it is adequate to evaluate the correctness of the SNSSP protocol and supported by AVISPA back-ends. Complete descriptions of the automatic verification tool AVISPA, the security animator SPAN, and the Dolev-Yao intruder model are provided in appendix C. In the next subsection, the SNSSP CAS+ specifications are provided.

SNSSP Initial Negotiation Specification

The CAS+ specification relative to the initial negotiation phase of the SNSSP protocol is the following (details on the CAS+ syntax are provided in appendix C):

Spec 4.2 CAS+ Specification - SNSSP Initial Negotiation

```
identifiers
P, S, Ca :  user;
Kp, Ks, Kca :  public_key;
SSPs, Nonce1, Lifetime, CertRequest, IdS : number;
F, H : function;

messages
1.  P -> Ca :  CertRequest
2.  Ca -> P : {IdS}Kca'
```

[3] http://www.irisa.fr/celtique/genet/span/

```
3.  P -> S : Nonce1
4.  S -> P : SSPs,H(SSPs), IdS, Nonce1,Lifetime,{F(H(SSPs),IdS, Nonce1,Lifetime)}Ks’

knowledge
P : S,F,Ca;
S : P,F,H,Ca;
Ca :  S,P;

session_instances
[P:secman,Ca:authority,Ks:serverkey,Kca:authoritykey,Kp:secmankey,S:server];

intruder_knowledge
secman,authority,server;

goal
P authenticates S on IdS;
```

In the CAS+ specification provided above, P denotes the SecMan Proxy, S the ground server, and Ca the certificate authority issuing the digital certificates. Keys and credentials have been provided and requested by the SNSSP protocol and two different hash functions have been used. The F hash function is used to provide message integrity whereas the H hash function is used to provide a hash of the server SSP. One session is established between the three main identifiers (*i.e.* P, Ca, and S) and the intruder is assumed to have knowledge of all public keys and identifier roles.

The security goal is expressed as an authentication required between SecMan (*i.e.* P) and the ground server (*i.e.* S) using the signed number IdS (which is equivalent to a digital certificate provided by the certificate authority Ca). The corresponding HLPSL specification is provided in appendix C.

SNSSP Renegotiation Specification

The CAS+ specification relative to the renegotiation phase (*i.e.* after the lifetime expiration date) is:

Spec 4.3 CAS+ Specification - SNSSP Initial Negotiation

```
identifiers
P, S : user;
Kp, Ks :  public_key;
SSPs, Nonce2, Nonce3, Nonce4, IdS : number;
F, H : function;

messages
1.  P -> S : Nonce2, Nonce3
2.  S -> P : H(SSPs), IdS, Nonce3, Nonce4,{F(H(SSPs),IdS, Nonce3,Nonce4)}Ks’

knowledge
P : S,F;
S : P,F,H;
```

```
session_instances
[P:secman,Ks:serverkey,Kp:secmankey,S:server];

intruder_knowledge
secman,server;

goal
P authenticates S on IdS;
```

The security goal is the same as the one described for the initial negotiation specification (another step of the SNSSP protocol but the goal remains the same).The corresponding HLPSL specification is also provided in appendix C. These specifications have been translated by AVISPA using the IF format then converted to the backends for verification and eventually, intruder trace (if weaknesses are found).

Verification and Intruder Simulation Results

The final back-end results shown in tables 4.4 and 4.5 are classified into 5 sections:

- ◇ SUMMARY section tells if security properties are provided, or if the verification analysis was not finished (*e.g.* syntax errors);
- ◇ DETAILS section describes the simulation conditions used to find an attack trace or to declare a protocol as secure (*e.g.* bounded or unbounded number of sessions);
- ◇ PROTOCOL section reminds the protocol name as mentioned in the protocol specifications;
- ◇ GOAL section reminds the security goals specified by the user (*e.g.* secrecy of a key, etc);
- ◇ BACKEND section provides the verification tool used by AVISPA to conduct the security analysis (*i.e.* OFMC, CL-Atse, SATMC, and TA4SP).

As depicted in the back-end outputs, we can confirm that the SNSSP protocol entirely satisfies the security goals required and expressed in the CAS+ and HLPSL specifications. Back to the SecMan components, now that the supported security mechanisms set has been securely established, SecMan is able to classify them using a multi-criteria hierarchical approach as described in the next subsection.

Table 4.4: AVISPA Backends - Negotiation SNSSP Results

OFMC Output	**ATse Ouput**
`% OFMC`	`%ATSE`
`SUMMARY`	`SUMMARY`
`SAFE`	`SAFE`
`DETAILS`	`DETAILS`
`BOUNDED_NUMBER_OF_SESSIONS`	`BOUNDED_NUMBER_OF_SESSIONS`
`PROTOCOL`	`TYPED_MODEL`
`/home/Slim/span/results/init.if`	`PROTOCOL`
`GOAL`	`/home/Slim/span/results/init.if`
`as_specified`	`GOAL`
`BACKEND`	`As Specified`
`OFMC`	`BACKEND`
`COMMENTS`	`CL-AtSe`
`STATISTICS`	`STATISTICS`
`parseTime: 0.00s`	`Analysed : 3 states`
`searchTime: 0.04s`	`Reachable : 1 states`
`visitedNodes: 19 nodes`	`Translation: 0.00 seconds`
`depth: 4 plies`	`Computation: 0.00 seconds`
SATMC Output	**TA4SP Ouput**
`SUMMARY`	`SUMMARY`
`SAFE`	`SAFE`
`DETAILS`	`DETAILS:`
`STRONGLY_TYPED_MODEL`	`BOUNDED_NUMBER_OF_SESSIONS`
`BOUNDED_NUMBER_OF_SESSIONS`	`TYPED_MODEL`
`BOUNDED_MESSAGE_DEPTH`	`PROTOCOL:`
`PROTOCOL`	`/home/Slim/span/results/init.if`
`/home/Slim/span/results/init.if`	`GOAL:`
`GOAL`	`SECRECY`
`%% see the HLPSL specification`	`BACKEND:`
`BACKEND`	`TA4SP`
`SATMC`	
`COMMENTS`	
`STATISTICS`	
`attackFound false boolean`	
`stopConditionReached true boolean`	
`fixedpointReached 5 steps`	
`stepsNumber 5 steps`	
`atomsNumber 0 atoms`	
`clausesNumber 0 clauses`	
`encodingTime 0.04 seconds`	
`solvingTime 0 seconds`	
`if2sateCompilationTime 0.31 seconds`	
`ATTACK TRACE`	
`%% no attacks have been found`	

Table 4.5: AVISPA Backends - Re-negotiation SNSSP Results

OFMC Output	**ATse Ouput**
% OFMC SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/Slim/span/results/init.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s searchTime: 0.01s visitedNodes: 4 nodes depth: 2 plies	%ATSE SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/Slim/span/results/init.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 4 states Reachable : 2 states Translation: 0.01 seconds Computation: 0.00 seconds
SATMC Output	**TA4SP Ouput**
SUMMARY SAFE DETAILS STRONGLY_TYPED_MODEL BOUNDED_NUMBER_OF_SESSIONS BOUNDED_MESSAGE_DEPTH PROTOCOL /home/Slim/span/results/init.if GOAL %% see the HLPSL specification BACKEND SATMC COMMENTS STATISTICS attackFound false boolean stopConditionReached true boolean fixedpointReached 6 steps stepsNumber 6 steps atomsNumber 0 atoms clausesNumber 0 clauses encodingTime 0.13 seconds solvingTime 0 seconds if2sateCompilationTime 0.25 seconds ATTACK TRACE %% no attacks have been found	SUMMARY SAFE DETAILS: BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL: /home/Slim/span/results/init.if GOAL: SECRECY BACKEND: TA4SP

4.4.2 A Multi-Criteria Hierarchical Classification of Security Protocols

The security protocols and mechanisms negotiated between the SecMan proxies and the ground servers forms a very heterogeneous set with many independent properties. For instance, it is always difficult to tell which security protocol is the strongest one between IPSec and TLS. Indeed, such an affirmation can be very subjective, depending on the criteria used in the decision process (*e.g.* experiments, existing attacks, key lengths, provided security services, etc). Similarly, how is it possible to quantitatively establish a link between the robustness of these security protocols and the related network and system costs?

As the adaptation process in the SecMan framework needs a clear answer for these questions, we propose to use a multi-criteria hierarchical approach to classify the negotiated security protocols beforehand. As mentioned in section 4.3.3, we rely on the AHP technique in this specific task (*c.f.* appendix C for a complete description of AHP). In order to establish the hierarchy structure and adapt it to the security features negotiated thanks to the SNSSP protocol, we provide in the next subsection a formal hierarchical oriented model for network security policies.

4.4.2.1 Formal Hierarchical oriented Model for Network Security Policies

The security features negotiated using the SNSSP protocol consists in heterogeneous security mechanisms, protocols, and algorithms. The complexity behind classifying such a set of parameters resides in the number, the nature, and the interdependence of the mechanisms to consider. For instance, the DES(128)-CBC[4] algorithm provides the confidentiality security service and can be used at the network layer within the IPsec protocol or the transport layer within the TLS protocol. At a higher abstraction layer, a network security policy can use IPsec, TLS, or even both to protect the network. Moreover, it is senseless to compare a security algorithm providing a confidentiality security service with a security algorithm providing the integrity security service.

As the combination between these security features can be confusing, we provide a mathematical hierarchical oriented modeling approach for network security policies. The formal model defined in this section will be used later throughout the remaining SecMan components. The nomenclature and relationships established between the security algorithms, security mechanisms, and security policies are depicted in figure 4.10:

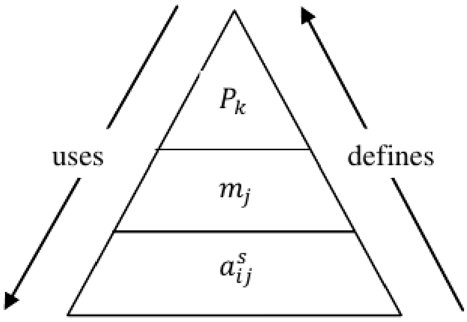


Figure 4.10: Pyramidal Security Feature Relationship Representation

[4] Data Encryption Standard using a 128 bit key in the Cipher Block Chaining mode

As shown in figure 4.10, there is a bi-directional relationship between the represented security features:

- ◇ a_{ij}^{s} denotes the i^{th} security algorithm providing the security service s and being supported by the security mechanism m_j. For instance, if the considered security services are data confidentiality, message integrity, and authentication, then we write:

  $$s \in S = \{conf, integ, auth\} \tag{4.5}$$

  AES(192)-CBC and SHA-1(160) are examples of security algorithms providing respectively the confidentiality and message integrity security services;

- ◇ m_j denotes the j^{th} security mechanism of the $SSP_{negotiated}$ set. A security mechanism m_j uses a set of security algorithms and can be formally represented as:

  $$m_j =< a_{0j}^{s_0}, a_{1j}^{s_1}, ..., a_{yj}^{s_y} > \tag{4.6}$$

  where $\forall s_y \in S$, y denotes the number of security algorithms supported by the security mechanism m_j. SSH, TLS, and IPSec are security mechanisms defined respectively at Sec_A, Sec_T, and Sec_N as illustrated in figure 4.7;

- ◇ P_k denotes the security policy for the k^{th} data flow:

  $$P_k =< p_{k,0}, p_{k,1}, ..., p_{k,n} > \tag{4.7}$$

  where $p_{k,j}$ refers to the j^{th} security mechanism m_j selected in the security policy P_k and $n \in \{0, ..., card(SSP_{negotiated})\}$:

  $$p_{k,j} \begin{cases} = 1 & \text{if the security mechanism } m_j \text{ is selected in } P_k; \\ = 0 & \text{if the security mechanism } m_j \text{ is not selected in } P_k. \end{cases}$$

  For instance, if m_0 and m_1, and m_2 denote respectively the IPsec, TLS, and SSH security mechanisms, then the <1, 1, 0> security policy refers to the <IPsec<SHA-1(160)>, TLS<AES(192)-CBC> > hybrid security policy in order to provide respectively the integrity (using SHA-1(160)) and confidentiality (using AES(192)-CBC) security services.

At the AHP functional block, the security algorithms represent the alternatives to be ranked by the algorithm. When the common security algorithms are defined, the AHP process is launched for all the security algorithms targeting the same security service s. For instance, if $S = \{conf, integ, auth\}$, then the AHP block is launched three times for each security service (*i.e.* confidentiality, integrity, and authentication).

In the next subsection, we explain how the AHP method is used in the scope of the SecMan framework.

4.4.2.2 Using AHP in the SecMan Framework

Figure 4.11 shows the hierarchy used for ranking the security algorithms in the scope of the SecMan framework:

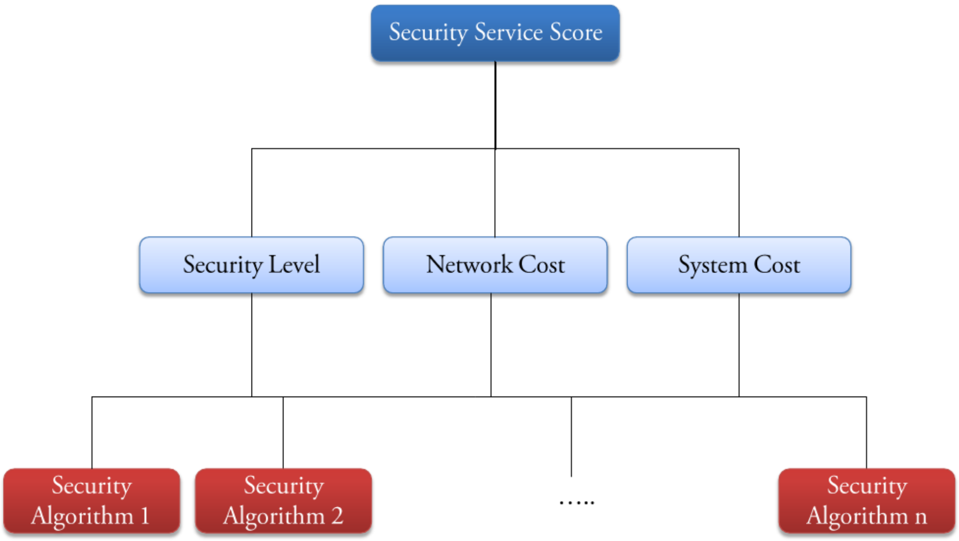


Figure 4.11: AHP Hierarchical Structure adapted to the SecMan Framework

The hierarchy is established for each security service (*i.e.* confidentiality, integrity, and authentication). The first level of criteria is composed by the security level (*i.e.* the robustness provided by the security algorithm), the network cost (*i.e.* the network overload cost), and the system cost (*i.e.* the computational cost). Each of these criteria is evaluated using several sub-criteria as depicted in table 4.6:

Table 4.6: AHP Criteria and sub-Criteria List

Security Service	**Security Level**	**Network Cost**	**System Cost**
Confidentiality	Cipher Key Length Block Size # rounds	Encryption Delay Decryption Delay Packet Overhead	Computational Throughput
Integrity	Hash Length Block Size # rounds	Hash Generation Delay Packet Overhead	
Authentication	Signature Key Length	Hash Generation Delay Signature Verification Delay Packet Overhead (bytes)	

The security level corresponding to each security algorithm is evaluated according the following criteria:

- ◇ *The cipher key/hash/signature length*: the more the key is long, the more the algorithm security level increases. For instance, it is harder to crack a 1024-bits RSA key than a 768-bits key (*i.e.* factoring the RSA public key). For the integrity service, the more the hash is long, the more the algorithm is robust. Finally, the signature length is considered for the authentication service;

- ◇ *Block size and number of rounds*: a block cipher is a fixed length string of plain-text bits used by security algorithms to make a series of finite iterations (or rounds) such as permutation, transformation, or XOR, to provide the adequate cipher output. For instance, the AES encryption algorithm uses 128 bits blocks with 10, 12, or 14 rounds, depending on the length of the cipher key. When the size of the processed blocks and the number of iterations increase, the algorithm is able to provide a higher level of security as the inverted operation (*i.e.* finding the original message using the encrypted message) will be longer and more difficult [Preneel *et al.* 1998].

The criteria used for the network cost are the following:

- ◇ *Delay*: the end-to-end delay is an important parameter that could impact the network performances of the air-ground link. In the case of the security algorithms, there are several types of delay:
  - ▷ *Encryption/Decryption delay*: when a message has to be encrypted by the sender and decrypted by the receiver, an additional delay due to the cryptographic operations is added. This parameter is used for the security algorithms providing the confidentiality security service;
  - ▷ *Hash generation delay*: as for the encryption/decryption operations, there is a supplementary delay to be considered when a hash has to be calculated. This parameter is used for both the integrity and the authentication algorithms (*c.f.* equations 2.5 and 2.6 of chapter 2);
  - ▷ *Signature verification delay*: this parameter is exclusive to the authentication procedure, when a signature has to be computed. Indeed, at the signed message reception, the receiver has to verify the validity of the signature using the certificate/signature verification techniques described in chapter 5.
- ◇ *Packet overhead*: this is the overhead in bits induced by the security algorithms in addition to the initial size of the original data message.

Finally, for the system cost score, the cryptographic computational throughput (in MB/s) is used as a criterion to rank the computational performances of each security algorithm (which can be assimilated to the speed of the security algorithm).

As mentioned before, the AHP process is applied to each security algorithm a_{ij}^s of the $SSP_{negotiated}$ set in order to find a score for the associated security service s. The outputs of this step are an AHP security level score, a network cost score, and a system cost score for each security service s (normalized to 1). This way, the SecMan knows for each security mechanism m_j, which are the security algorithms that have the best scores for each security service s. The following pseudo-code algorithm resumes the security classification step of the SecMan framework as described in this section:

Algorithm 4.4 Security Mechanism Classification

```
for each security mechanism $m_j$ do
    for each security service $s$ do
        Apply AHP on the set $\{a_{ij}^s\}$;
        for each security algorithm $a_{ij}^s$ do
            get the security level score;
            get the network cost score;
            get the system cost score;
        end for
    end for
end for
```

After the classification of all the security features negotiated between the air and the ground entities, the SecMan framework collects all the information relative to the system and the network available resources. This step is described in the next subsection.

4.4.3 Network and System Information Collection

In order to provide the SecMan with information about the network available resources, a client-server application is provided as shown in figure 4.12:

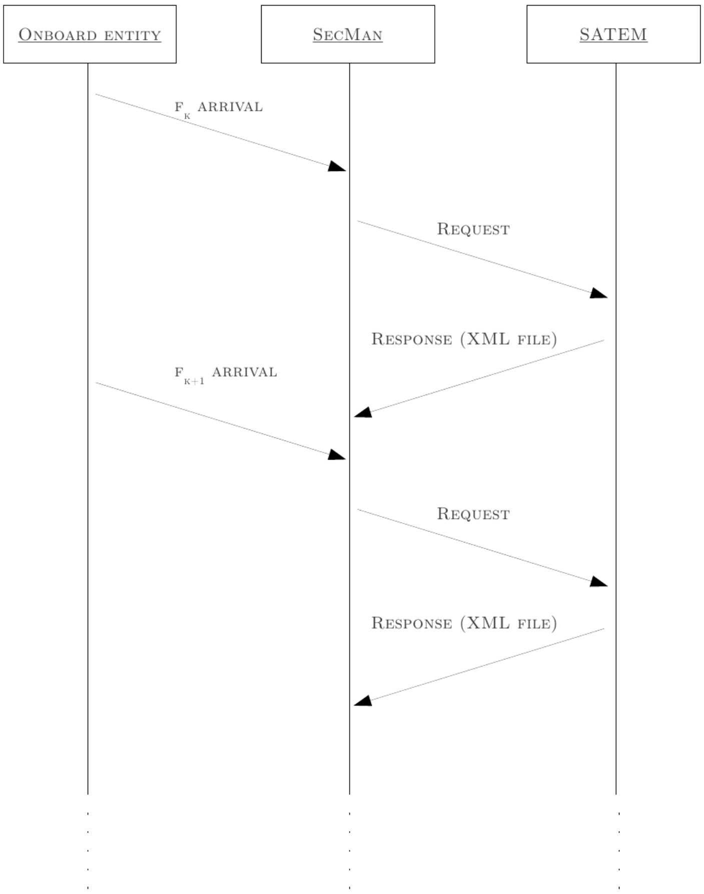


Figure 4.12: Network and System Information Collection Adhoc Protocol

The end entities concerned by the exchanges are:

1. *The onboard entity*: who requests a secure connection with the ground server;
2. *SecMan*: who plays the role of the client requesting these information when it has to establish a security policy for a data flow f_k;
3. *The SATEM (Satellite Emulator)*: this is a proprietary system developed by the ASTRIUM partner (*c.f.* appendix C for more details). The aim is to emulate the behavior of a real IP-based satellite link (*e.g.* QoS, link capacity, packet loss, jitter, delay, etc).

When the SecMan receives a request for a data flow f_k, it sends a request to the SATEM node in order to get the information about the available network resources and the satellite link condition. When the request is received, the SATEM sends an XML-based response to the SecMan where all the information are assembled. The following XML tags are used:

- ▷ *a DVB-RCS tag*: includes information about the return link such as the S/N (Signal to Noise) ratio or the MODCOD (Modulation Code);

- ▷ *a Satlink tag*: includes information about the jitter and the delay measured at the SATEM emulator input/output interfaces;
- ▷ *a Traffic tag*: includes information about used and available resources per traffic class.

Figure 4.13 shows an example of the formated XML files containing the information required by the SecMan to adapt and optimize the security policy at the end of the global process:

```
<DVB-RCS>
  <MODCOD>8-PSK 2/3</MODCOD>
  <CN>0.8</CN>
</DVB-RCS>
<SATLINK>
  <DELAY>0.337</DELAY>
  <JITTER>0.027</JITTER>
</SATLINK>
<TRAFFIC>
  <USED>
    <ATC>15222</ATC>
    <AOCNG>78100</AOCNG>
    <AOC>22005</AOC>
    <APC>155792</APC>
  </USED>
  <FREE>
    <ATC>34888</ATC>
    <AOCNG>1202900</AOCNG>
    <AOC>5780995</AOC>
    <APC>5543000</APC>
  </FREE>
</TRAFFIC>
```

Figure 4.13: XML File Structure for SecMan-SATEM Exchanges

In order to avoid overflooding issues, a lifetime parameter τ is defined in the case many data flow requests arrive in a short period of time to the SecMan proxy. τ gives an information on when the SecMan should send a request to the SATEM emulator. In order to define the lifetime value, there are two options:

1. choose a fixed τ: this means that the SecMan is going to send periodically a request to the SATEM. However, there is no accurate indication on how to choose such a fixed value. On the one hand, if τ is too small, this could cause an excessive number of consecutive requests. On the other hand, if τ is too big, the SecMan could miss some important information about the network state in extreme situations (*e.g.* sudden decrease of the satellite signal);

2. choose an adaptive τ: this option is chosen in the design of the client-server tool. As the SecMan sends the requests for each data flow f_k, we defined the lifetime parameter as a function of the data flow arrival rate (*i.e.* $\frac{1}{\Delta t}$). The arrival rate is evaluated for two successive data flows: if it decreases then the lifetime value is incremented in order to not over flood the SATEM with consecutive requests, otherwise the lifetime value is decreased.

Algorithm 4.5 Network and System Information Collection

```
if $(t_k - t_{k-1}) > (t_{k-1} - t_{k-2})$ then //$\Delta t$ calculation
    $\tau \leftarrow \tau + 1$;
else
    $\tau \leftarrow \tau - 1$;
end if
if $reception\_date + \tau > current\_date$ then
    Send a request to the SATEM;
else
    Use the information retrieved for the data flow $f_{k-1}$;
end if
```

In the next step, the collected information are used to define the network and system constraints which need to be respected when the optimal security policy is established.

4.4.4 Network and System Information Process

At this point, the information received by SecMan and issued from the SATEM satellite emulator are used to define the constraints under which the security policy is going to be determined. Let θ_c^{used} and θ_c^{free} be respectively the used and available network resources issued from the XML files of the previous step, where $c \in C = \{ATS, AOC, AOCNG, APC\}$. Let $\omega_{c,net}^{used} \in [0,1]$ be the used network resource ratio and defined as:

$$\omega_{c,net}^{used} = \frac{\theta_c^{used}}{\theta_c^{used} + \theta_c^{free}} \tag{4.8}$$

The available network resource ratio $\omega_{c,net}^{free}$ is then directly deduced from the $\omega_{c,net}^{used}$ ratios relevant to traffic classes with higher priorities. $\omega_{c,net}^{free}$ depicts the available network resource ratio for all the data flows issued from any service relevant to traffic class c and defined as:

$$\omega_{c,net}^{free} = 1 - \sum_{i \in C, i \geq c} \omega_{i,net}^{used} \tag{4.9}$$

For instance, if the information received and processed from the SATEM satellite emulator give the following values for the different traffic classes:

◇ $\omega_{ATS,net}^{used} = 0.1$ (10 % of overall network resources);

◇ $\omega_{AOC,net}^{used} = 0.2$ (20 % of overall network resources);

◇ $\omega_{APC,net}^{used} = 0.3$ (30 % of overall network resources);

then, using the formula 4.9, available network resource ratios per traffic class are defined as:

◇ $\omega_{ATS,net}^{free} = 1 - \omega_{ATS,net}^{used} = 0.9$;

◇ $\omega_{AOC,net}^{free} = 1 - (\omega_{ATS,net}^{used} + \omega_{AOC,net}^{used}) = 0.7$;

◇ $\omega_{APC,net}^{free} = 1 - (\omega_{ATS,net}^{used} + \omega_{AOC,net}^{used} + \omega_{APC,net}^{used}) = 0.4$;

In this example, we can clearly see that the priorities between the different traffic classes are respected according to the inter-class operational mode of the SecMan framework (*c.f.* section 4.3.4). The used and available system resource ratios are also defined on the basis of each traffic class c. However, as the system resources are relevant to the SecMan computation capabilities, we define the associated ratio locally as the following:

$$\omega_{c,syst}^{used} = \%CPU * \frac{N_c}{N_{total}} \tag{4.10}$$

where N_c is the number of data flows relative to the traffic class c, and N_{total} is the total number of all traffic flows. The percentage of the CPU usage is extracted using a SHell script based on the LINUX command `top`[5]. The available system resource ratio is calculated similarly to formula 4.9:

$$\omega_{c,syst}^{free} = 1 - \sum_{i \in C, i \geq c} \omega_{i,syst}^{used} \tag{4.11}$$

The pseudo-code algorithm relevant to a data flow f_k of a traffic class c is finally provided as:

Algorithm 4.6 Network and system Information Process

```
for all traffic class $i \geq c$ do
    $\omega_{i,net}^{used} \leftarrow \frac{\theta_i^{used}}{\theta_i^{used}+\theta_i^{free}}$;
    $\omega_{i,syst}^{used} \leftarrow \%CPU * \frac{N_i}{N_{total}}$
end for
$\omega_{c,net}^{free} \leftarrow 1 - \sum_{i \in C, i \geq c} \omega_{i,net}^{used}$
$\omega_{c,syst}^{free} \leftarrow 1 - \sum_{i \in C, i \geq c} \omega_{i,syst}^{used}$
```

The constraints ($\omega_{c,net}^{free}$ and $\omega_{c,syst}^{free}$) established in this step, jointly to the AHP scores determined in section 4.4.2.2, are used by the SecMan framework to determine the optimal security policy for the data flow f_k as depicted in the next subsection.

4.4.5 Optimal Security Policy Selection

The aim of SecMan at this final step is to establish an optimal security policy P_k for the data flow f_k. In order to map the security services provided in the security policy P_k and the security requirements for the data flow f_k, we define R_k as a 3D vector:

$$R_k = [r_k^{conf}, r_k^{integ}, r_k^{auth}] \tag{4.12}$$

where r_k^s is the elementary security requirement for the security service s of the data flow f_k. r_k^s values are ranked from 1 to 5 as we mentioned in the security risk assessment study of chapter 3. The optimal security policy, noted P_k^* should verify the following constraints:

- ◇ The security requirement vector R_k must be at least verified, meaning that the security policy level of P_k^* has to satisfy these requirements. We define the PSL (Policy Security Level) of a policy P_k as:

[5] The `top` program provides a dynamic real-time view of a running system. It can display system summary information as well as a list of tasks being managed by the Linux kernel.

$$PSL(P_k) = R_k^{tr} * V_{sec}(P_k) \tag{4.13}$$

where R_k^{tr} is the transpose of the vector R_k and $V_{sec}(P_k)$ is the security level value of the policy P_k. $V_{sec}(P_k)$ is defined according to the formal hierarchical oriented model for network security policies defined in section 4.4.2.1 as:

$$V_{sec}(P_k) = \sum_{j=0}^{n} p_{k,j} * V_{sec}(m_j) \tag{4.14}$$

$V_{sec}(m_j)$ is calculated according to the AHP scores of each security algorithm a_{ij}^s composing the security mechanism m_j as depicted in equation 4.6;

◇ The available network and system resource ratios computed at the previous step must be taken into account, meaning that the global cost of the optimal security policy P_k^* should respect the available network and system resource ratios. The overall cost $Cost(P_k)$ of a security policy P_k is defined as:

$$Cost(P_k) = Cost_{net}(P_k) * (1 - \omega_{c,net}^{free}) + Cost_{syst}(P_k) * (1 - \omega_{c,syst}^{free}) \tag{4.15}$$

where $\omega_{c,net}^{free}$ and $\omega_{c,syst}^{free}$ are calculated in the fourth step of the SecMan framework and used in equation 4.15 to weight $Cost_{net}(P_k)$ and $Cost_{syst}(P_k)$. These costs are established according to the AHP scores of the security mechanisms composing the security policy P_k:

$$Cost_{net}(P_k) = \sum_{j=0}^{n} p_{k,j} * V_{netCost}(m_j) \tag{4.16}$$

$$Cost_{syst}(P_k) = \sum_{j=0}^{n} p_{k,j} * V_{systCost}(m_j) \tag{4.17}$$

$V_{netCost}(m_j)$ and $V_{systCost}(m_j)$ are calculated according to the AHP scores of each security algorithm a_{ij}^s composing the security mechanism m_j as depicted in equation 4.6.

The optimal security policy P_k^* for the data flow f_k corresponds to the security policy which maximizes the security level while network/system costs are minimized:

$$P_k^* = \arg\max_{P_k}(RBI(P_k)) \tag{4.18}$$

where RBI (Relative Balance Index) is defined as:

$$RBI(P_k) = PSL(P_k) - \beta * Cost(P_k) \tag{4.19}$$

where $\beta \in \mathbb{R}_+^* \backslash \{1\}$ is a positive coefficient used to balance the security policy level and the cost of P_k. As the minimum security requirements are already ensured in the PSL function (*c.f.* equation 4.13), the β value is chosen such as the cost score has more weight than the security level of the policy P_k. Indeed, beyond the minimum security level required for each data flow f_k, adding a supplementary level of security is not mandatory, whereas the network and system resources should be privileged. Then, in practice, it is sufficient to choose $\beta > 1$.

Finally, the pseudo-code algorithm for the optimal security policy step is:

Algorithm 4.7 Optimal Security Policy Selection

```
Establish a list of all possible security policies $P_k$;
$RBI(P_k^*) \leftarrow -\infty$;
for all $P_k$ do
    if $P_k$ satisfies $R_k$ then
        Compute $PSL(P_k)$;
        Compute $Cost(P_k)$;
        $RBI(P_k) \leftarrow PSL(P_k) - \beta * Cost(P_k)$;
        if $(Cost_{net}(P_k) \leq \omega_{c,net}^{free}) \wedge (Cost_{syst}(P_k) \leq \omega_{c,syst}^{free})$ then
            $P_k^* \leftarrow P_k$;
            $RBI(P_k^*) \leftarrow RBI(P_k)$;
        end if
    end if
end for
```

As described in chapter 4, we have been involved in three different work packages of the FAST project. Once the SecMan design finished, our last task in the project consisted in integrating the SecMan proxies within the remaining partner contributions for the validation phase. In the next section, the final FAST testbed infrastructure is presented and SecMan performances are discussed.

4.5 Validation and Experiments

The work accomplished in the scope of the WP4 of the FAST project has been divided into three steps[†]:

1. Specification of the evaluation scenarios;

2. Network testbed architecture and addressing plan;

3. Experiments and performance evaluation.

4.5.1 Evaluation Scenarios

Several evaluation scenarios have been drawn, depending on the FAST partner objectives (*e.g.* testing the SATEM functionality, TCP optimizations, etc). For these scenarios, we used the insecure mode (`Bypass` policy) in order to not alter the results. The SecMan evaluation scenario has been divided into 6 different sub-scenarios, according to the SecMan operational modes mentioned in section 4.3.4:

[†] Note that except the last step which is exclusively relevant to the SecMan performance evaluation, almost all the FAST partners (*i.e.* ASTRIUM, Télécom Bretagne, MEDES, VODEA) have been involved in the specification of the evaluation scenarios and the definition of the testbed platform.

Table 4.7: SecMan Integration Test Scenarios

Scenario	SecMan Mode	Traffic
Section 4.5.3.1	Insecure (Bypass) mode	All
Section 4.5.3.2	Secure transport mode	ATS, AOC, and APC
Section 4.5.3.3	Secure application mode	ATS, AOC, and APC
Section 4.5.3.4	Secure transparent mode	All
Section 4.5.3.5	Intra-class mode	All
Section 4.5.3.6	Inter-class mode	All

All traffic classes have been tested among the different sub-scenarios except the secure transport and application modes where only the TCP-based traffic classes (*i.e.* ATS, AOC, and APC) have been evaluated. Indeed, the telemedicine and video surveillance traffics include UDP flows and can not be secured using neither TLS nor SSH. For each sub-scenario, a signal attenuation (or fading) was triggered using the SATEM satellite emulator in order to test the SecMan adaptive capabilities. In the next section, we present the network testbed architecture integrating all the partner contributions.

4.5.2 Network Testbed Architecture

Figure 4.14 shows the network testbed architecture deployed in the final phase of the FAST project:

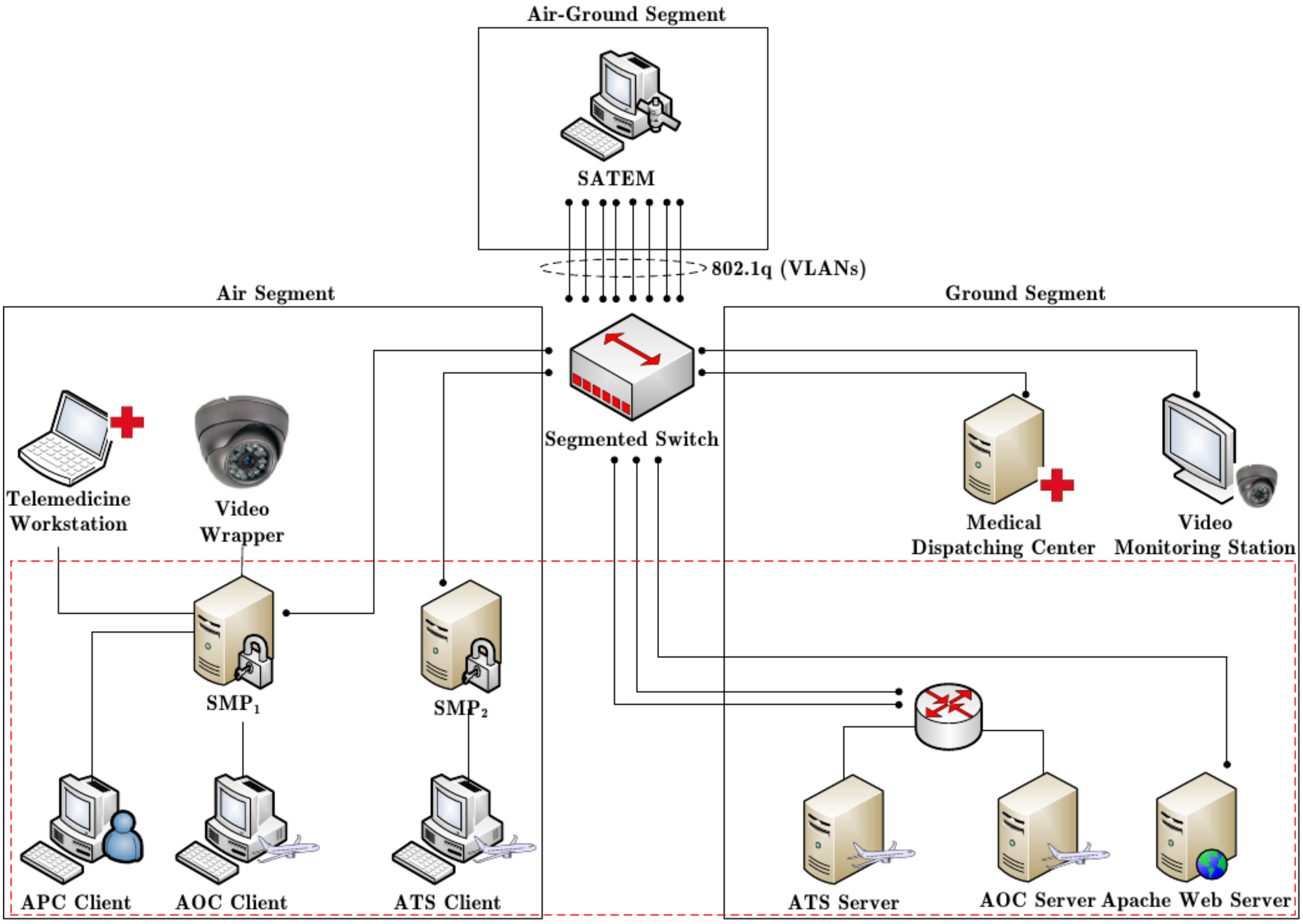


Figure 4.14: FAST Network Testbed Architecture

The testbed topology[†] is composed of an air-ground segment (SATEM satellite emulator), an air segment (all the onboard clients and the SecMan proxies), and a ground segment (all the servers). Given the high number of network devices to be connected (clients, servers, routers, switches, etc), a part of the network architecture has been virtualized on one specific physical machine using VirtualBox[6] (dotted red line in figure 4.14). Besides, a switch has been segmented and used as an interface between the three segments. The IEEE 802.1q [Cisco 2006] technique has been used in order to tag and configure different VLANs (Virtual Local Area Network) within the same physical switch.

The topology provided in figure 4.14 reduced the number of physical network devices from 18 to only 7. To summarize, here are the devices deployed in the final testbed architecture:

1. The machine dedicated to the SATEM satellite emulator;
2. The VLAN enabled switch;
3. The proprietary VODEA video surveillance wrapper and the corresponding server;
4. The proprietary MEDES telemedicine embedded station and the corresponding server;
5. One physical machine where several virtual machines have been deployed:
    - ◇ An ATS (respectively AOC) operational traffic generator: the ATS and AOC packet and bit rate profiles established in chapter 2 have been used to generate the operational traffic thanks to the IPERF[7] network testing tool. The time average of traffic generation varied from 1 to 5 minutes, depending on the evaluation scenario (*c.f.* section 4.5.1). Indeed, we observed that a 5 minutes traffic generation period was sufficient enough to analyze the impact of the adaptive security policy on the ATS/AOC traffic profile;
    - ◇ An ATS (respectively AOC) server machine which is a simple sink recovering and dropping the traffic coming from the operational client machines;
    - ◇ An APC client machine: we used the WGET[8] network tool to retrieve web contents from an application server. In order to generate a high number of TCP connections, we used the recursive mode of the WGET tool without the keep-alive option (which allowed us to avoid persistent connections);
    - ◇ An APC server machine has been configured using the APACHE HTTP server[9] where several type of media contents (HTML files, JPEG pictures, RSS feeds, etc) have been stored;
    - ◇ the SecMan SMP proxies.

On each node, the Wireshark[10] network analyzer has been installed in order to collect the traffic statistics on each specific network interface. At the end of the simulation campaign, all the statistics have been gathered then shared between all the partners for later analysis. In the next subsection, we provide the results obtained for each sub-scenario described above.

[†] The testbed architecture shown in 4.14 has been designed jointly to the FAST partners involved in the validation phase of the project. Our tasks were to provide the addressing plan, configuration of the segmented switch, and virtualization of the network components framed by the dotted red line in figure 4.14.

[6] https://www.virtualbox.org/

[7] http://sourceforge.net/projects/iperf/

[8] http://www.gnu.org/s/wget/

[9] http://www.apache.org/

[10] http://www.wireshark.org/

4.5.3 Experimental Results

4.5.3.1 Insecure Bypass Mode

In this mode, the SecMan machine plays the role of a router (*i.e.* the `Forwarding IP` option has been activated[11]) which allows all the coming data flows to pass the output network interface without being secured. The experiments conducted in this first sub-scenario are very useful as long as they are used later as reference results to evaluate the benefits and advantages of the SecMan adaptive capabilities. Figures 4.15 and 4.16 show the average bit rates of all traffic classes using the `Bypass` policy with and without an attenuation of the satellite signal (*i.e.* Fading triggered on the SATEM satellite emulator).

Seemingly, when the `Bypass` policy is activated, the SecMan machine just relays the packets from one network interface to another. However, the results shown in figures 4.15 and 4.16 illustrate that the priority between the traffic classes is totally respected, even after an important mitigation of the air-ground satellite signal. It is apparent that the ATS and AOC traffic classes kept almost the same performances with and without the fading whereas the bit rates of traffic classes with lower priorities (*i.e.* AOC NG and the APC classes) have considerably decreased.

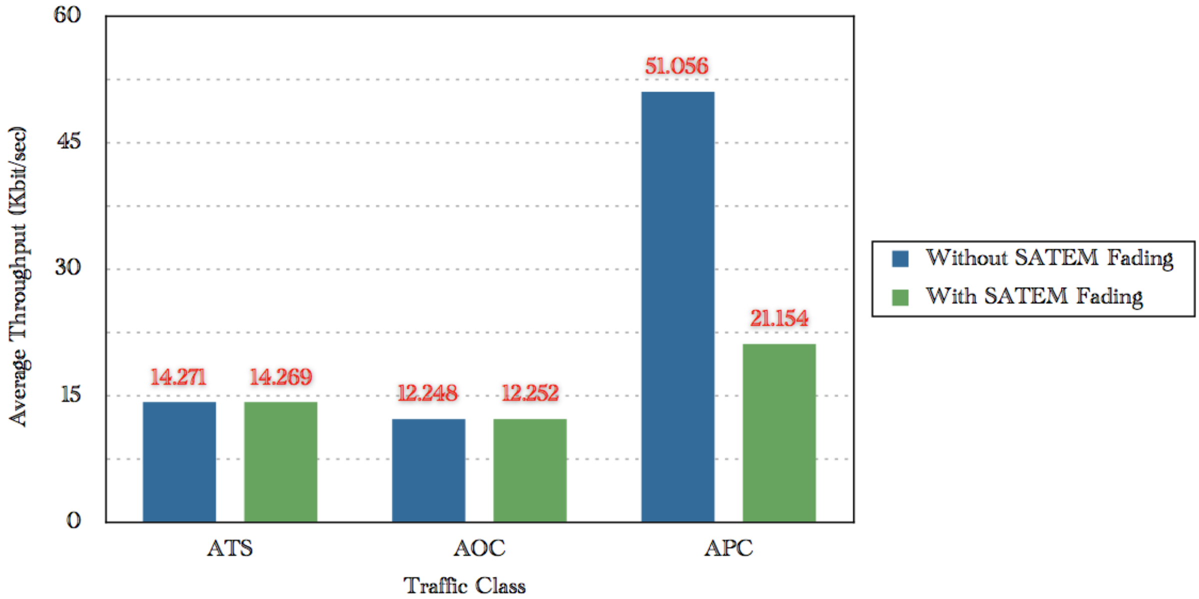


Figure 4.15: ATS, AOC, and APC Average Throughputs (Kbit/sec) (`Bypass` Policy)

Although, we can observe that the variation of the average throughputs relevant to the APC and AOC NG traffic classes are more meaningful compared to the operational traffic classes. Figure 4.17 quantifies these throughput variations as a function of the traffic classes' priorities. We can see that when the traffic class priority increases, the gap between the throughputs measured with and without the signal attenuation decreases: the measured variations (in percentage) are 0.01%, 0.03%, 29.10%, 49.12%, and 58.57% respectively for ATS, AOC, telemedicine, video surveillance, and APC traffic classes.

[11] `echo 1 > /proc/sys/net/ipv4/ip_forward`

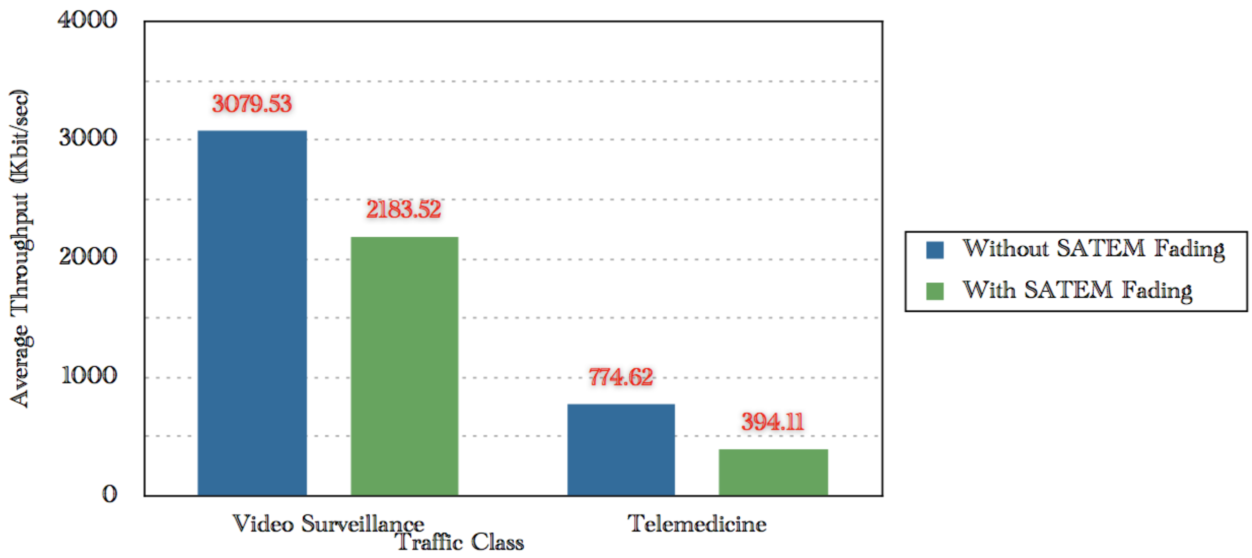


Figure 4.16: AOC NG Average Throughputs (Kbit/sec) (`Bypass` Policy)

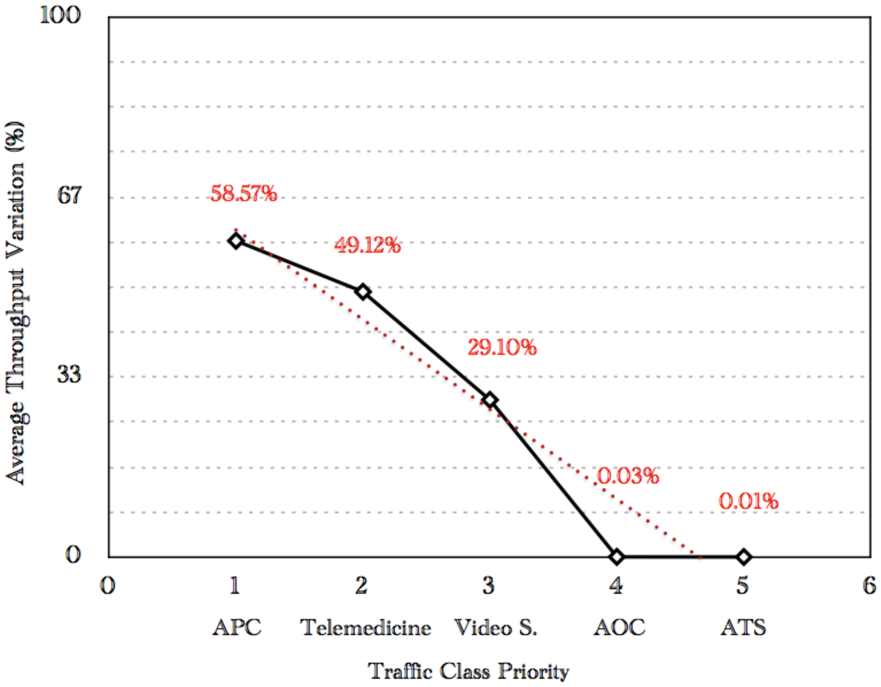


Figure 4.17: Average Bit Rate Variation as a Function of Traffic Classes' Priorities

The difference observed between the telemedicine and video surveillance average throughput variations is explained by a higher volume of data for the video surveillance application (3.07 and 0.77 Mbits respectively for the video surveillance and the telemedicine applications). Also, it is worthy to note that the satellite link capacity in this scenario allows the traffic classes with lower priorities to be served even after the signal attenuation. However, when the fading becomes severe, these traffic classes (*e.g.* APC) are simply rejected and denied from accessing to the network resources. This can be important for instance in critical situations (*e.g.* emergency onboard) when most of the satellite capacity should be dedicated for traffic classes with high priorities (mainly operational services).

Table 4.8 resumes the observed average throughputs for all traffic classes using the `insecure` Bypass mode with and without the SATEM fading. The results obtained in this first scenario are compared later with those relevant to the intra-class and inter-class adaptive modes:

Table 4.8: SecMan Insecure Mode Results

Traffic	**Average Throughput (Kbits/sec)**		**Variation**
Class	**Without SATEM Fading**	**With SATEM Fading**	**(Percentage)**
ATS	14.271	14.269	0.01 %
AOC	12.248	12.252	0.03%
APC	51.056	21.154	58.57%
Telemedicine	774.62	394.11	29.10%
Video Surveillance	3079.53	2183.52	49.12%

In the next subsection, we present the results relevant to the SecMan secure transport mode.

4.5.3.2 Secure Transport Mode

In this mode, the TLS protocol is used at the transport layer of the TCP/IP protocol stack to secure all the TCP data flows. As mentioned earlier, the secure transport mode is used only for the ATS, AOC, and APC traffic classes. For the telemedicine application, the secure transport mode worked few seconds at the beginning when HTTP (relevant to the graphical user interface of the embedded telemedicine station) and SIP (Session Initiation Protocol) data flows have been transmitted for management and Electrocardiography reports. However, when the onboard medical agent started using the auricular camera to broadcast medical assessment of the passengers (which are video UDP-based data flows) to the ground medical dispatch center, the secure transport mode failed because it is adapted only for TCP-based data flows. Similarly, this mode was not adapted for the video surveillance application because it uses a UDP-based traffic. Therefore, the secure transport mode has been used only for the ATS, AOC, and APC TCP-based traffic classes.

The Stunnel[12] network program has been used to establish the secure transport communication between the SecMan machine and the ground servers. Stunnel is designed to work as an SSL/TLS encryption wrapper between remote clients and local or remote servers that do not support SSL/TLS protocols natively. As operational servers do not have SSL-aware daemons running originally on they systems, Stunnel is used here to set them up easily in order to communicate with clients over secure SSL channels. For instance, figure 4.18 shows how Stunnel has been configured to secure AOC data flows:

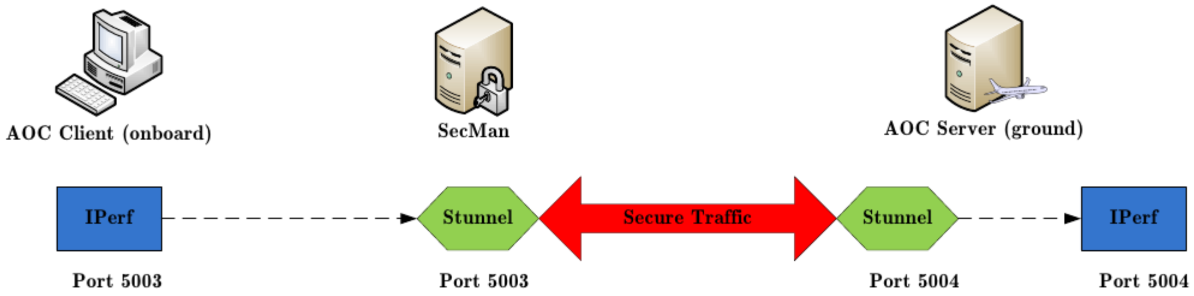


Figure 4.18: Stunnel - IPERF Mapping to Secure AOC Data Flows

[12] Stunnel is `http://www.stunnel.org/`

The IPERF tool has been used to generate the ATS and AOC operational traffic according to the simulation traffic profiles established in chapter 2, and the WGET downloading tool to get the APC multimedia contents. Stunnel has been configured using the `server mode` in order to listen to an arbitrary port on each server virtual machine (*i.e.* the ATS, AOC, and APC server virtual machines). For instance, as shown in figure 4.18, in order to secure the AOC data flows, IPERF has been configured in a `client mode` using the 5003 port. Stunnel (on the SecMan machine side) listened on that specific port and forwarded the secured data flows to the 5004 port, which was listened by IPERF in a `server mode` on the AOC server virtual machine side.

The results obtained in this second simulation scenario are relevant to a by default static security policy established by Stunnel. Additionally, this security policy (*i.e.* <TLS<AES(256)-CBC, RSA, SHA> >) has the highest security level according to the AHP security score (equal to 0.99), which is promising for later comparisons with the adaptive SecMan modes. As for the results shown in section 4.5.3.1, they are used as reference results and compared to the adaptive security policy. Figure 4.19 shows the average data overhead, deduced as the difference between the insecure `Bypass` mode data and the secure transport mode data, for each traffic class:

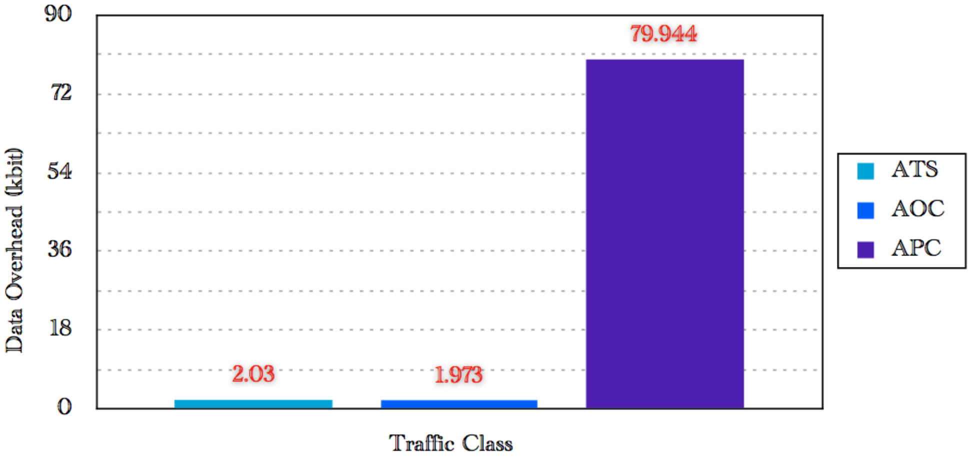


Figure 4.19: Average Data Overhead per Traffic Class in the Secure Transport Mode

The `Bypass` mode results used to calculate the data overhead are those measured without the signal attenuation. Indeed, as far as the secure transport mode uses a static security policy and has no adaptive capabilities, it is fair to compare the results under the same environmental conditions (*i.e.* without the SATEM fading). Experimentally, the average data overhead (denoted *average_data_overhead*) has been deduced for each simulation then averaged on the total number of simulations.

The resulting data overhead shown in figure 4.19 is explained by the signalization extra data induced in the TLS handshake initial phase and the TLS additional protocol sub-layers, namely the Record Layer Protocol and the Alert Layer Protocol. The ATS and AOC data overheads are significantly low. This is explained by the fact that the generated traffic for operational services carry out a low number of TCP data flows compared to the APC traffic. Indeed, given the nature of the APC application (HTTP) and the configuration used in WGET (no keep-alive option), the APC traffic encompassed a high number of TCP independent and non-persistent connections between the client and the server. As SecMan applied a security policy for each different connection, the average data overhead over all the simulation periods logically increases: 79.944 kbits for the APC against 2.03 and 1.973 kbits respectively for the ATS and the AOC traffic. The average data overheads represent 14.22%, 16.10%, and 156.58% respectively of the ATS, the AOC, and the APC traffic in the insecure `Bypass` mode ($\frac{average\_data\_overhead}{data\_in\_the\_\texttt{Bypass}\_mode} * 100$). *average_data_overhead* is deduced as the sum of data overheads in each simulation averaged on the total number of simulations (*i.e.* 3 simulations). *data_in_the_*`Bypass`*_mode* refers to the sum of data in the Bypass mode (there is no average data deduced here because only one simulation has been conducted in this mode). Table 4.9 resumes the average data overhead measured for the ATS, AOC, and APC traffic classes using the secure transport mode.

Table 4.9: Secure Transport Mode Results

Traffic Class	**Average Data Overhead (Kbits)**	**Average Data Overhead vs. Bypass Mode (%)**
ATS	2.03	14.22%
AOC	1.973	16.10%
APC	79.944	156.58%

The third sub-scenario dedicated to the SecMan performances is relevant to the secure application mode which uses the SSH protocol.

4.5.3.3 Secure Application Mode

The SSH protocol has been initially designed to provide safe and secure remote connections as a substitute of older tools such as TELNET (Terminal Network) [Postel & Reynolds 1983] or the rlogin Linux command [Kantor 1991], but now it is widely used as an alternative to secure TCP-based applications like the ATS, AOC, and APC traffic in our case. Figure 4.20 shows the experimental configuration used for the SecMan secure application mode.

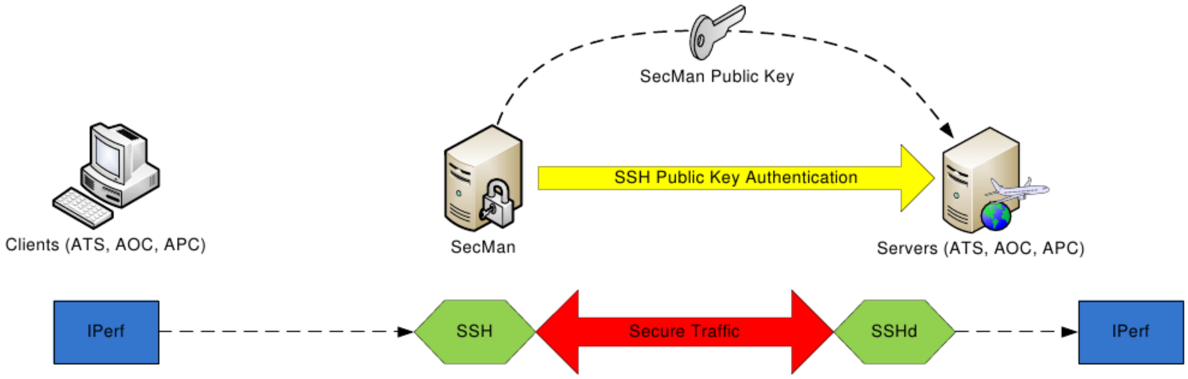


Figure 4.20: SSH - IPERF Mapping to Secure TCP-based Applications

Similarly to the secure transport mode, IPERF and WGET tools have been used to generate an adequate traffic. The SSHd daemon has been configured on the server machines in order to receive the secure connections from the SecMan proxy. Figure 4.21 shows the average data overhead using the secure application mode. As for the previous scenario, a by default static security policy has been established by the SSH protocol (*i.e.* <SSH<AES(256)-CBC, RSA, SHA-1> > with an AHP security level score equal to 0.97). The average data overheads shown in figure 4.21 represent (in percentage) 90.45%, 92.08%, and 95.08% respectively of the initial ATS, AOC, and APC traffic in the Bypass mode. Compared to the results obtained in the previous secure transport mode, the data overheads in the secure application mode are quite different, which is a good indicator for the relevance of an adaptive security policy. Indeed, this concretely means that SecMan likely has a large range of security policies to choose from with very different implications on the network performances (*i.e.* according to the AHP scores for each security policy).

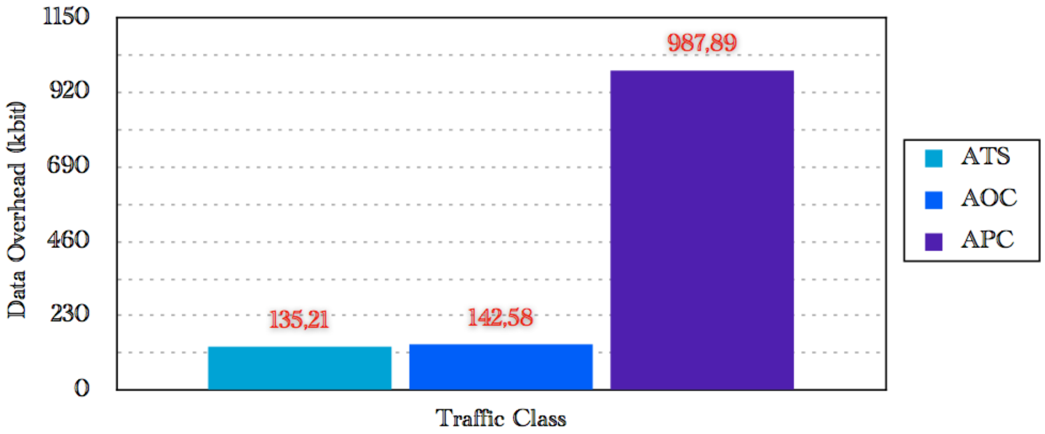


Figure 4.21: Average Data Overhead per Traffic Class in the Secure Application Mode

Table 4.10 resumes the average data overhead measured for the ATS, AOC, and APC traffic classes using the secure application mode:

Table 4.10: Secure Application Mode Results

Traffic Class	**Average Data Overhead (Kbits)**	**Average Data Overhead vs. `Bypass` Mode (%)**
ATS	135.21	90.45%
AOC	142.58	92.08%
APC	987.89	95.08%

The secure transparent mode performances presented in the next subsection should reinforce the heterogeneity of the security policy combinations and their respective security levels/performance effects.

4.5.3.4 Secure Transparent Mode

The secure transparent mode relies on the IPsec network layer security protocol. Consequently, this mode is used to provide security services for all traffic classes. The KAME[13] IPsec utility, native to the Linux OS, has been used jointly to the Racoon[14] tool, which is an IKE (Internet Key Exchange) [Harkins & Carrel 1998] based key management daemon required to establish the security associations between the different hosts or IPSec gateways. The KAME IPSec implementation greatly differs from the usually used StrongSwan[15] implementation and allows an easier configuration and modification of the security inner primitives needed later in the adaptive modes (*e.g.* **`encryption_algorithm`**, **`hash_algorithm`**, or **`authentication_method`** directives in RACOON for instance). The ESP tunnel mode has been used instead of the AH mode because it provides all security services (*i.e.* confidentiality, integrity, and authentication) required by operational and non operational traffic classes, whereas the AH encapsulation mode does not provide the confidentiality security service.

As for the secure transport and secure application modes, the average data overheads are measured for all traffic classes. Simulation results are illustrated in figure 4.22:

[13] `www.kame.net`

[14] `http://netbsd.gw.com/cgi-bin/man-cgi?racoon++NetBSD-current`

[15] `http://www.strongswan.org/`

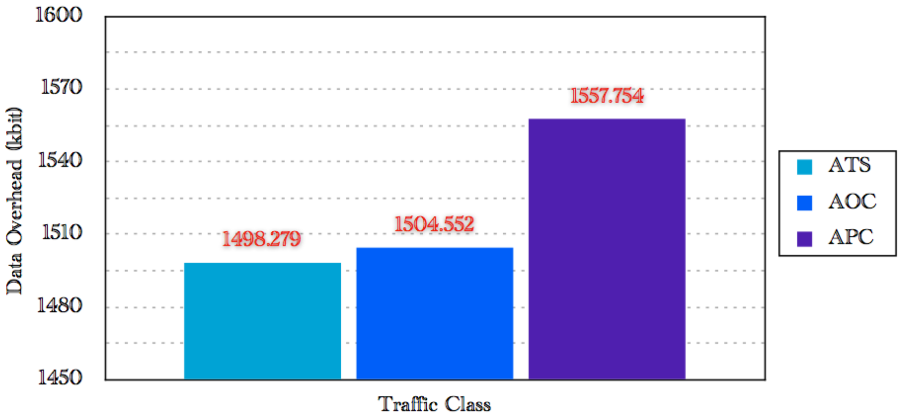


Figure 4.22: ATS, AOC, and APC Average Data Overhead in the Secure Transparent Mode

Compared to the secure transport and application modes, the SecMan secure transparent mode induces a higher data overhead which is due to the initial IKE key exchange phase and the extra ESP packet headers (*i.e.* a new IP header, an ESP header, an ESP trailer, and the authenticated data). The average data overheads in the secure transparent mode represent (in percentage) respectively 99.05%, 99.19%, and 96.82% of the initial ATS, the AOC, and the APC traffic in the `Bypass` mode. Besides, unlike the two previous modes which secure each TCP data flow, the secure transparent mode provides a security on a packet basis, which explains why the difference between induced data overheads for each traffic class is not as important as in the secure transport and application modes. Figure 4.23 shows the telemedicine and the video surveillance application throughputs using the secure transparent mode (with an average data bit rate equal to 4.6 and 2.26 Mbits/sec respectively for the video surveillance and the telemedicine applications):

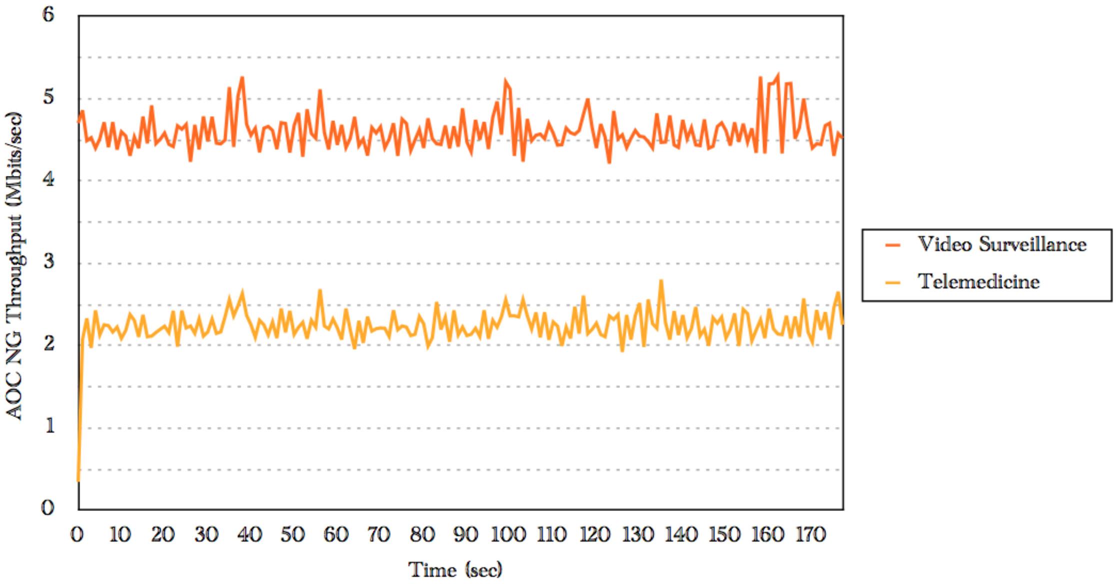


Figure 4.23: AOC NG Throughputs (Secure Transparent Mode)

The average overhead represents (in percentage) 33.06% and 65.81% respectively of the initial video surveillance and telemedicine applications in the `Bypass` mode. Compared to the ATS, AOC, and APC results in the secure transparent mode, the average data overhead relative to the AOC NG traffic is lower which could be explained by a higher data rate for these type of traffic. Figure 4.24 shows the average data overhead for the AOC NG traffic class. The shown results represent a supplementary perquisite for the adaptive security policy. Even if the number of security policies allowed in this mode is lower compared to the secure transport and secure application modes, the good point that has to be underlined here is that the secure transparent mode has a different impact of the network performances: with such a variety in the security level and cost scores, the SecMan inter-class and intra-class modes emphasizes the perceptiveness of bringing adaptive capabilities in such an heterogeneous environment.

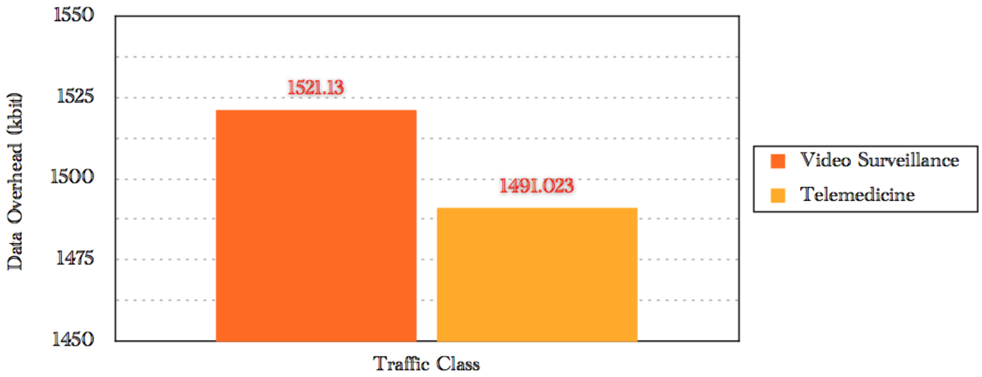


Figure 4.24: AOC NG Average Data Overhead in the Secure Transport Mode

Table 4.11 resumes the average data overhead results measured for all traffic classes using the secure transparent mode:

Table 4.11: Secure Transparent Mode Results

Traffic Class	**Average Data Overhead (Kbits)**	**Average Data Overhead vs. Bypass Mode** (%)
ATS	1498.279	99.05%
AOC	1504.552	99.19%
APC	1557.754	96.82%
Telemedicine	1491.023	33.06%
Video Surveillance	1521.13	65.81%

4.5.3.5 Intra-class Mode

Table 4.12 depicts the behavior of the SecMan adaptive framework using the intra-class mode. For recall, the intra-class mode is used to secure data flows from the same traffic class, meaning it is unable to handle more that one traffic class at the same time. Initially, this mode was meant to be used only for the ATS traffic, but we decided to test it will all the traffic classes in order to investigate the behavior of the SecMan framework with and without the SATEM fading. Two different tests have been conducted for each traffic class separately as we can see in table 4.12. In all performed tests, we observed that without the SATEM fading, SecMan initially applies security policies with high AHP security level scores, which is due to the fact that plenty of network and system resources are available at the beginning, which allows the establishment of a high-cost security policy. When the SATEM fading is triggered, SecMan changes the applied security policy according to the AHP triplet scores. The newly selected security policy belongs to the same operational mode (*i.e.* using either the secure transport, transparent, or application modes). The security level is kept as high as possible in order to satisfy the minimum security requirements for each traffic class with the lowest network/system costs. For instance, in the case of the ATS test (#2), the <SSH<BLOWFISH, RSA, SHA(96)> > security policy has been chosen with the SATEM fading because it has a lower network cost (equal to 0.52) compared to the initial <SSH<AES(256)-CBC, RSA, SHA-1> > security policy.

Table 4.12: SecMan Security Policies (Intra-class Mode)

Traffic Class	Test Number	SATEM Fading	Security Policy
ATS	Test(#1)	×	<TLS<AES(256)-CBC, RSA, SHA> >
		✓	<TLS<RC4(128), DHE-DSS, SHA> >
	Test(#2)	×	<SSH<AES(256)-CBC, RSA, SHA> >
		✓	<SSH<BLOWFISH, RSA, SHA(96)> >
AOC	Test(#3)	×	<TLS<3DES-EDE, DHE-DSS, SHA> >
		✓	<TLS<RC4(128), DHE-DSS, SHA> >
	Test(#4)	×	<SSH<ARCFOUR, RSA, MD5(96)> >
		✓	<SSH<CAST(128), DSA, MD5(128)> >
APC	Test(#5)	×	<TLS<3DES-EDE, DHE-DSS, SHA> >
		✓	<TLS<DES(40), RSA, SHA> >
	Test(#6)	×	<SSH<AES(192), RSA, UMAC(64)> >
		✓	<SSH<CAST(128), RSA, RIPEMD(160)> >
V. Surveillance	Test(#7)	×	<IPSEC<AES(256)-CTR, RSA, SHA(96)> >
		✓	<IPSEC<AES(128), DSA, MD5> >
	Test(#8)	×	<IPSEC<3DES(192), RSA, SHA(96)> >
		✓	<IPSEC<IDEA(128), RSA, MD5> >
Telemedicine	Test(#9)	×	<IPSEC<AES(256)-CBC, RSA, SHA(96)> >
		✓	<IPSEC<BLOWFISH, RSA, RIPEMD> >
	Test(#10)	×	<IPSEC<3DES(192), DSA, MD5(96)> >
		✓	<IPSEC<CAST(40), DSA, RIPEMD> >

Figure 4.25 illustrates the evolution of the applied security levels for each performed test with and without the signal attenuation:

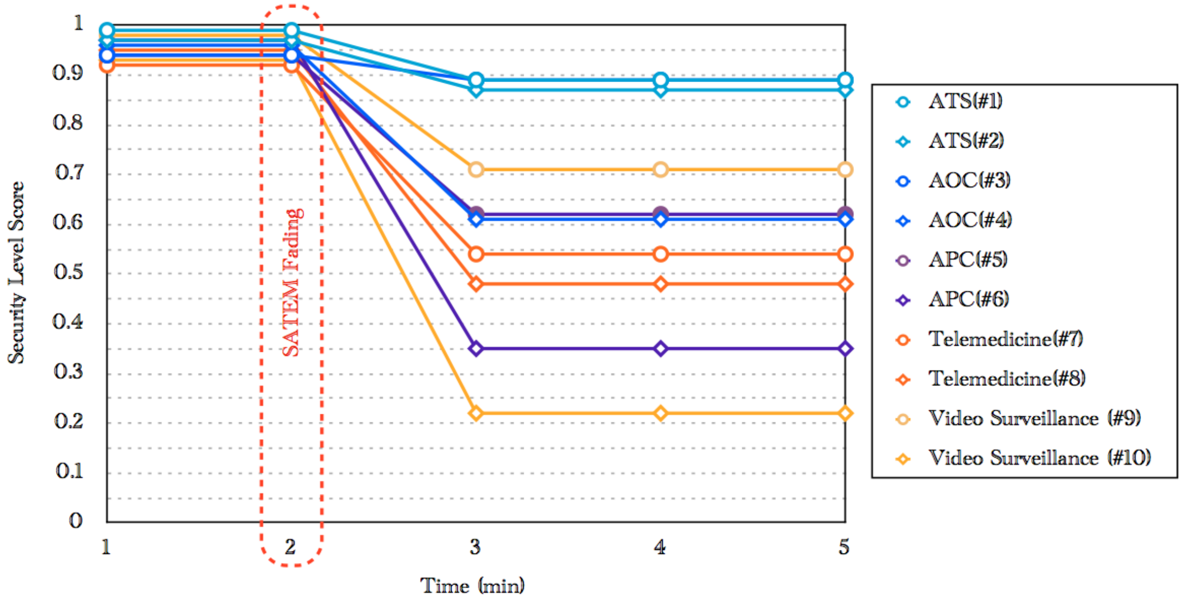


Figure 4.25: Evolution of the Security level Scores (Intra-class Mode)

We can see that the security level of the initial security policies are more or less high (ranging from 0.99 which is top security level score to 0.92) no matter the nature of the considered traffic class.

However, when the SATEM fading is triggered (at minute 2), we can see that the highest the traffic class priority is, the best SecMan tries to privilege the security level regarding the network and system scores.

Figures 4.26 and 4.27 depict the average data overheads for all traffic classes using the intra-class mode with and without the satellite signal attenuation (these results correspond to the first test of each traffic class). Depending on the considered traffic class, the average data overhead measured without the SATEM fading matches with the average data overhead measured for each traffic class using either the secure transport, transparent, or application modes. Indeed, this behavior concords with our expectations: the security policy applied before the signal attenuation has the highest security score, which corresponds generally to the by default static security policy in one of the SecMan operational modes mentioned in section 4.3.4. However, the data overhead measured after the signal attenuation is lower because SecMan tries to minimize the network and system costs while maximizing the security policy robustness.

The average data overhead (in percentage) represents 10.34% for the ATS, 21.79% for the AOC, 34.37% for the APC, 32.86% for the video surveillance, and 37.96% for the telemedicine initial traffics in the `Bypass` mode.

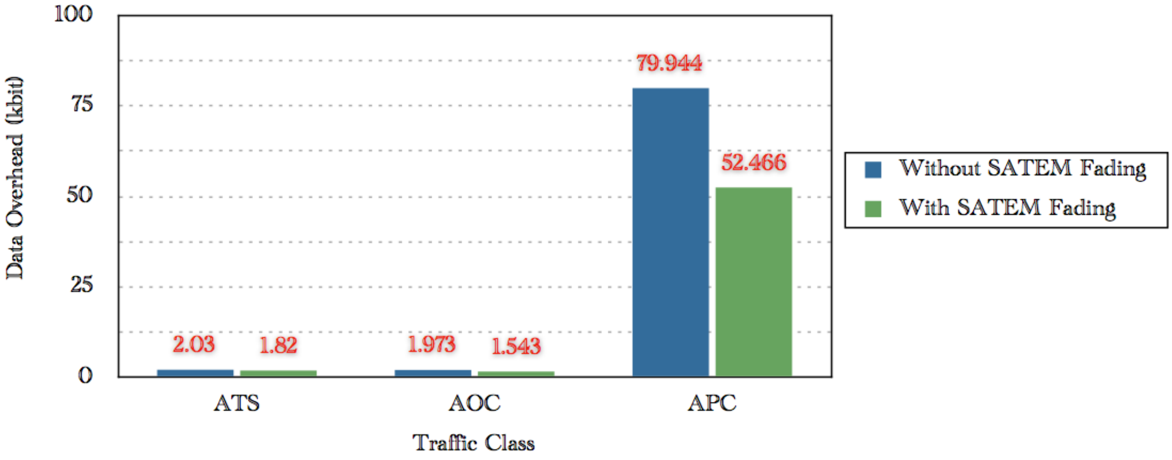


Figure 4.26: ATS, AOC, and APC Overheads in the Intra-class Mode

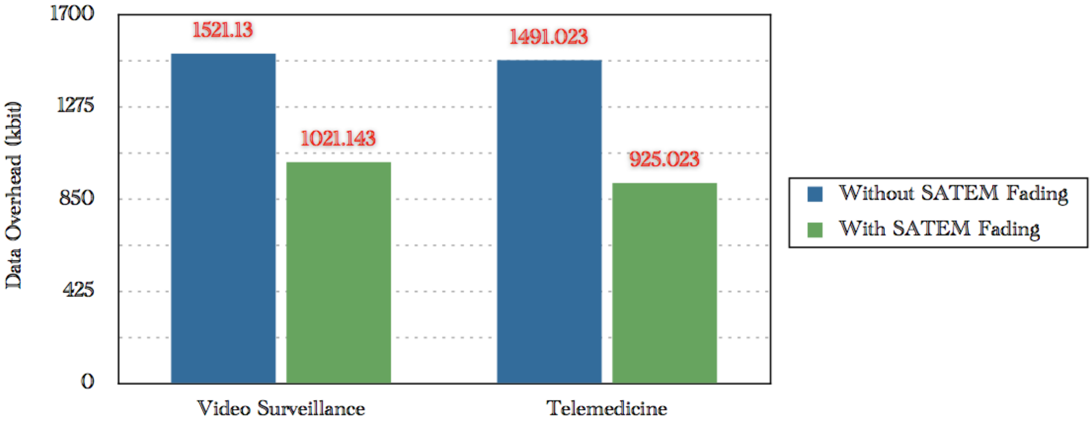


Figure 4.27: Video Surveillance and Telemedicine Overheads in the Intra-class Mode

Table 4.13 resumes the results obtained using the intra-class mode:

Table 4.13: Intra-class Mode Results

Traffic Class	**Average Data Overhead (Kbits)**	**Percentage of the Data in the Bypass Mode**
ATS	1.82	10.34%
AOC	1.543	21.79%
APC	52.466	34.37%
Telemedicine	1021.143	37.96%
Video Surveillance	925.023	32.86%

4.5.3.6 Inter-class Mode

In this mode, all the different traffic classes are used simultaneously and the SecMan framework selects a security policy for each traffic class. After the signal attenuation, the security policy is modified according to the traffic classes' priorities and the available system resources. Three simulation tests have been conducted, table 4.14 resumes the security policies selected in each of them:

Table 4.14: SecMan Security Policies (Inter-class Mode)

Test Number	**Traffic Class**	**SATEM Fading**	**Security Policies**
Test #1	ATS+AOC+APC	×	<SSH<AES(256), RSA, SHA> >
		✓	<TLS<RC4(128), RSA, MD5> >
	AOC NG	×	<IPSEC<3DES(192), RSA, SHA(96)> >
		✓	<IPSEC<IDEA(128), RSA, SHA(96)> >
Test #2	ATS+AOC+APC	×	<SSH<AES(256), RSA, SHA> >
		✓	<TLS<DES(40), RSA, SHA> >
	AOC NG	×	<IPSEC<3DES(192), RSA, SHA(96)> >
		✓	<IPSEC<BLOWFISH, RSA, RIPEMD> >
Test #3	All	×	<IPSEC<AES(256), RSA, SHA(96)> >
		✓	<IPSEC<CAST(40), DSA, RIPEMD> >

Unlike the intra-class mode, here SecMan was able to switch from one operational mode to another, except for the AOC NG traffic class which are UDP-based and thus can only be secured using the secure transparent mode. For instance, in the case of test (#1), we can see that SecMan has chosen a security policy using the secure application mode (<SSH<AES(256), RSA, SHA> >) for the ATS, AOC, and APC traffic classes. After the signal attenuation, the security policy that has been selected (*i.e.* <TLS<RC4(128), RSA, MD5> >) is relevant to the secure transport mode. Besides, using the inter-class mode, the system resource usage is optimized compared to operational modes where the applied security policies are static. This has the advantage to serve more data flows before the system reaches a saturation point.

Table 4.15 depicts a comparison between average throughputs measured using an adaptive security and a by default static security policy. The values shown in this table are averaged for all the tests performed in the inter-class mode and each of the operational modes (*i.e.* scenarios 4.5.3.2, 4.5.3.3, and 4.5.3.4). As we can see, the average throughputs for the adaptive security policies are lower than the static security policies: this difference is explained by the low cost security policies selected by SecMan after the SATEM fading in the inter-class mode.

Table 4.15: Adaptive vs. Static Security Policy Average throughputs

	Average Throughputs (kbits/sec)	
	Adaptive Security Policy	**Static Security Policy**
ATS	10.80	16.301
AOC	9.17	14.221
APC	92.78	131
Telemedicine	669.3	2260
Video Surveillance	1200.2	4600

In order to asset the benefits of the SecMan adaptive inter-class mode, figure 4.28 shows the percentage of the gain measured according the values shown in table 4.14. The measured benefits are compliant with the AHP scores of each security policy chosen by SecMan in the inter-class mode. As we can see, the gain is not the same for all traffic classes, which is due to the inter-class priorities taken into account in this mode. Indeed, the observed gain is more important for traffic classes with lower priorities: this can be explained by the fact that SecMan promotes operational traffic classes and keeps the highest average throughputs for them using high AHP score security policies in the adaptive mode, whereas APC and AOC NG traffic class average throughputs are lowered as much as possible.

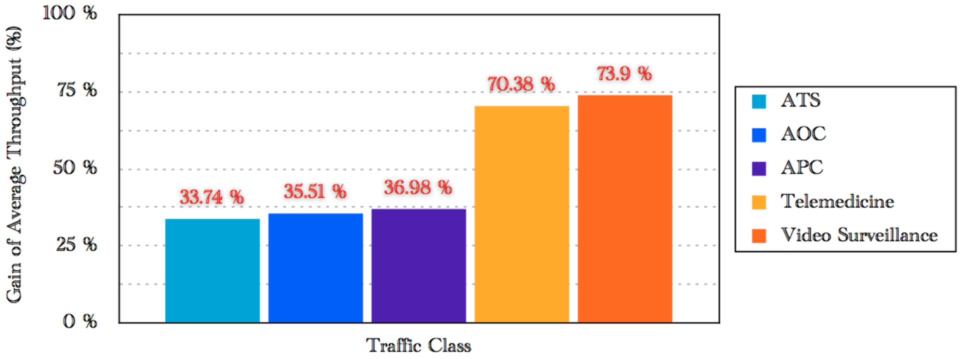


Figure 4.28: Gain of the Average Throughputs (Adaptive vs. Static Security Policy)

Figure 4.29 illustrates the overall AHP scores for 9 of the most commonly selected security policies by the SecMan framework:

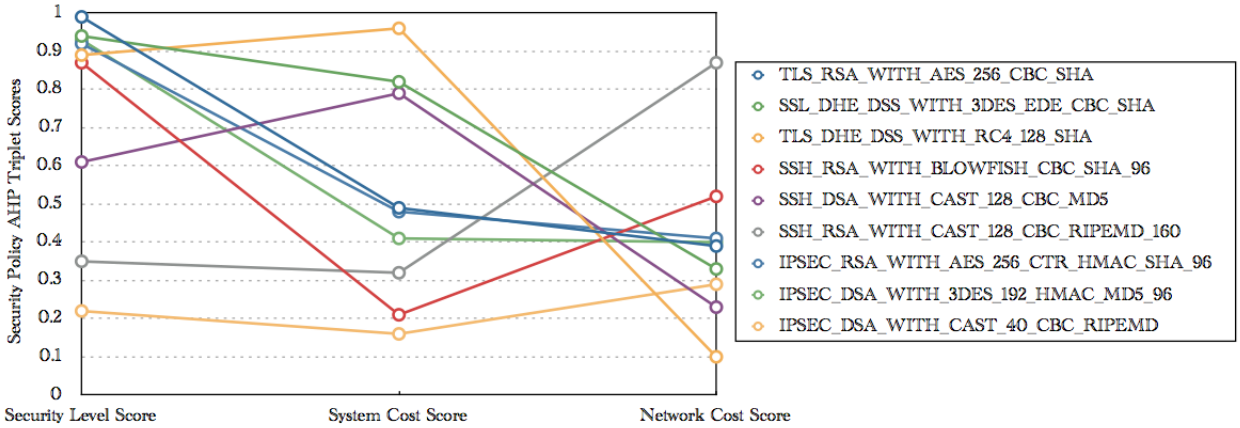


Figure 4.29: Non-Linearity between AHP Security and Cost Scores

As we can see, the security level and the network/system cost scores do not evolve similarly, meaning that is possible to get a security policy with a relatively low security score and high network/system cost scores. This non-linearity between the security level and the network/system costs highlights that a static security policy for all traffic classes (*c.f.* section 4.2.4) is not suitable when the objective is to optimize the resource management and reduce the security overhead.

Under the SecMan inter-class mode, a final case study has been added which corresponds to an extreme situation that consisted in an abrupt and rough decrease of the S/N ratio. Figure 4.30 shows the evolution of the data bit rates for each traffic class:

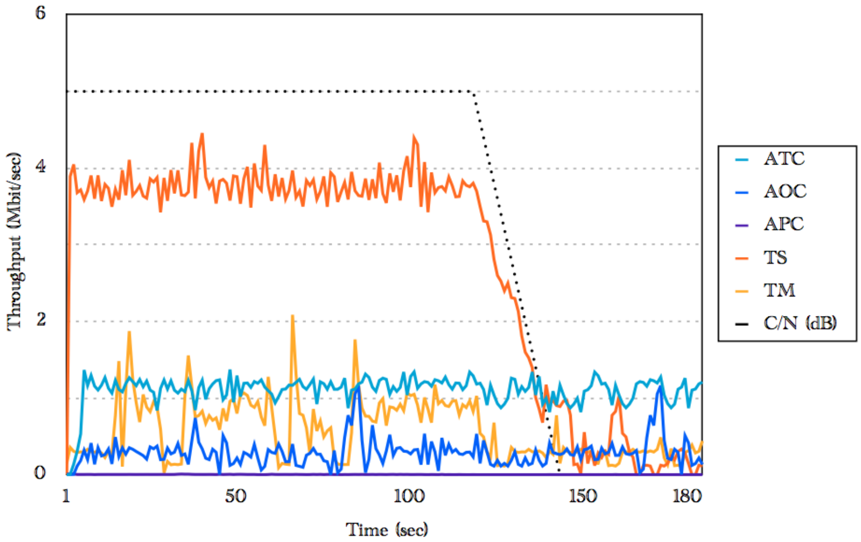


Figure 4.30: Throughput Evolution in Emergency Situations (Inter-class Mode)

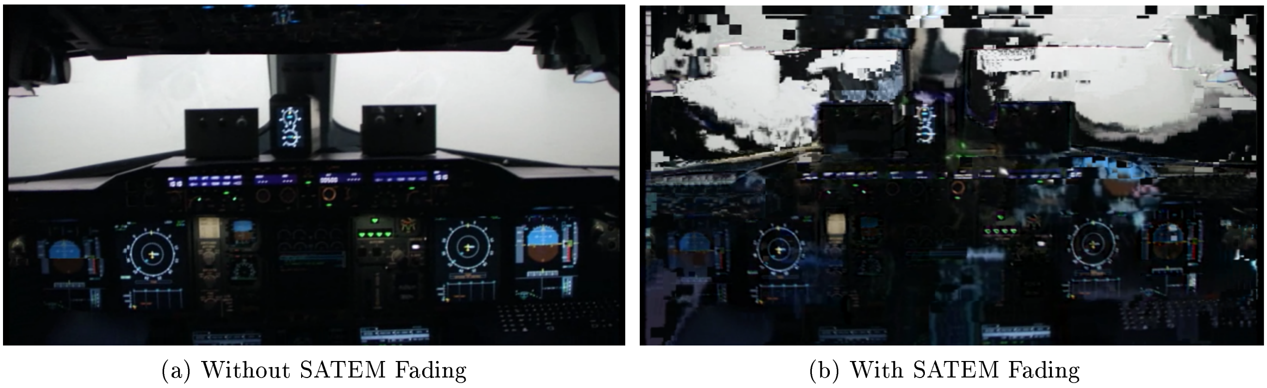
The decrease of the S/N ratio impacted more or less the throughputs for all the traffic classes, depending on the associated priority. The consequences of such an impact are noticed at several spots:

- the WGET network downloading tool was no more able to retrieve any other multimedia file from the APACHE web server repository. Having the lowest priority among all the traffic classes, it was quite predictable to observe such a throughput decrease for the APC traffic class, completely overflowed by higher traffic classes as depicted in figure 4.30;

- for the video surveillance and telemedicine applications, severe packet losses have been observed and the broadcasted video render has been altered (rough pixels, decrease of the frame rate). However, despite this video quality deterioration, SecMan provided a security policy for both applications with a low network cost score (<IPSEC<CAST(40), DSA, RIPEMD> > security policy with a network cost score equal to 0.29). Figure 4.31 shows a comparison between the video surveillance quality with and without the SATEM fading;

- the ATS and AOC traffic classes, given their high priorities, have been slightly deteriorated and continued to be served.

Moreover, the priorities have been respected also for the security level applied to each traffic class as depicted in figure 4.32. Besides, SecMan was able to react to a recovery of the satellite signal after the first SATEM fading:


(a) Without SATEM Fading

(b) With SATEM Fading

Figure 4.31: Video Surveillance Quality in Emergency Situations (Inter-class Mode)

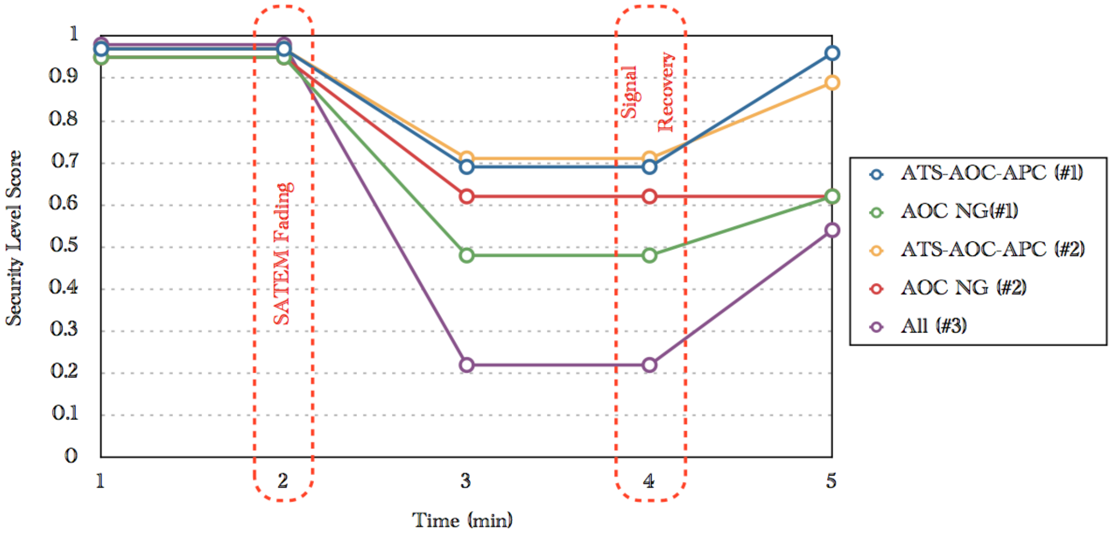


Figure 4.32: Security Level Score Evolution in the Inter-class Mode

4.6 Summary

In this chapter, we have introduced a novel adaptive security management approach for future aircraft communications. The QoS and resource assignment policies have been depicted and defined according to the priorities required between the operational and non-operational aeronautical services. The secure satellite-based network architecture presented in this chapter is fully compliant with the BSM reference architecture and ICAO SARPs. The SecMan framework and its five main algorithm components have been also introduced. The SecMan framework performances, evaluated in the scope of the final integration phase of the FAST project, have been discussed. According to those experiments, the adaptive security management brings several benefits such as an optimized resource management, a priority-aware security and an enhanced interoperability (technology-independent, reusable and compatible with any access layer).

At the end of chapter 2, a list of most critical security challenges in the future ATM has been provided. Scalability is the last but not the least issue that remains unsolved. In the next chapter, we propose a performance aware public key infrastructure for future connected aircraft, the aim is to reduce the air-ground messages induced by the security exchanges.

Chapter 5

A Performance-aware Public Key Infrastructure for Next Generation Connected Aircraft

Contents

In the previous chapter, we emphasized the need for a scalable solution to handle the different cryptographic credentials needed by the SecMan module and all it components. Indeed, the adaptive security policy presented in chapter 4 does not make sense if the global security framework is designed regardless of the signalization traffic induced by the PKI exchanges. Additionally, the scalability issues discussed in chapter 2 require an optimization of the so called PKI. Thus, in this chapter, original performance-aware PKI for future aeronautical data-based networks is introduced, the aim is to study the impact of the distribution/revocation/verification procedures on the system performances. Before going further into details, it is important to understand the main concepts of PKI.

Note: The Materials and results shown in this chapter have appeared in [Ben Mahmoud *et al.* 2010c, Ben Mahmoud *et al.* 2011a].

5.1 A Short PKI Overview

A PKI is a Public Key Infrastructure. In [Ferguson & Schneier 2003], Bruce Schneier[1] defines a PKI as *"...an infrastructure that allows you to recognize which public key belong to whom."*: sounds simple but it is a very relevant definition though. For an extended definition, a PKI can be seen as a set of practices, technologies, and security policies dedicated for the deployment, management, storage, and revocation of public key certificates.

5.1.1 PKI Components

Usually, a PKI is composed of the following entities:

- ◇ *The Certificate Authority* (CA): this is the central component of a PKI since it is the only entity that can publish public key certificates. Any digital certificate is signed by the issuing CA (assuming that everybody knows the CA's public key), which makes it the very foundation of a PKI security architecture;
- ◇ *The Registration Authority* (RA): this is an optional component that verifies the users' identity and requests the CA to issue an adequate digital certificate;
- ◇ *End Entities*: an end entity is a generic term used to denote a user, a device or a software module that needs a digital certificate in order to establish a secure communication. In the aeronautical context, an end entity could be a passenger, an aircraft, a crew member, or an airline operator, depending on the type of the communication to secure;
- ◇ *Repository*: this is also an optional component since it denotes a storage device for certificates so that they can be retrieved by end entities.

Among the different security operations performed within a public key infrastructure, the management of digital certificates is probably the most important function in a PKI.

5.1.2 Certificate Lifecycle Management

Usually, the certificate lifecycle management is composed of six steps:

- ◇ *Registration and public/private keys generation* (RK): the first step is the end entity registration and identity establishment. The registration procedure depends on which component has to generate the public/private keys. If the CA generates the key pair then the private key is securely passed to the registering end entity through an OOB (Out-Of-Band)[2] mechanism, otherwise (*i.e.* if the end entity generates its own key pair), the public key is passed to the CA which checks the private key using proof of possession mechanisms[3];
- ◇ *Certificate generation and distribution* (CGD): after the end entity registration and key pair generation, a certificate is issued and distributed respectively to the end entity and the certificate repository;

[1] B. Schneier is the founder and chief technical officer at Counterpane Internet Security, a managed security monitoring company. A world renowned scientist, security expert, and lecturer, he is the author of the *"Secrets and Lies: Digital Security in a Network World and Applied Cryptography"* best-seller.

[2] Some transactions between PKI components are performed through physical procedures (meaning in person) rather than implemented electronically.

[3] The digital signature, which is generated using the private key and verified using the corresponding public key, can be such a proof of possession mechanism.

- ◇ *Certificate regeneration* (CRG): when a certificate expires, the corresponding end entity informs the CA which has to renew the certificate;
- ◇ *Certificate revocation* (CRV): when a private key is compromised, the certificate is no longer valid and has to be revoked;
- ◇ *Certificate retrieval* (CRT): end entities retrieve certificates from the repository or may exchange certificates between each others, when PGP (Pretty Good Privacy)[4] protocol is used, for instance [Atkins *et al.* 1996];
- ◇ *Certificate validation* (CV): end entities may retrieve the lists of revoked certificates (*e.g.* CRLs) from a repository or may connect to an online server (*e.g.* OCSP) to validate a certificate when required (CRLs and OCSP are discussed further in section 5.2.3.1).

Considering the PKI components introduced in section 5.1.1 and the certificate management main steps mentioned above, figure 5.1 gives a general overview of these PKI basic concepts:

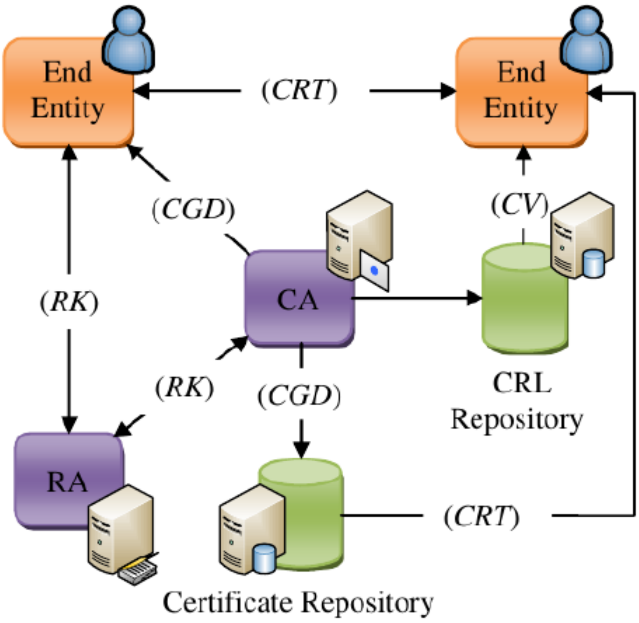


Figure 5.1: Basic PKI Environment

This is a simplified abstraction model of the basic PKIX[5] architecture model described in RFC 3280 [Housley *et al.* 2002]. PKI benefits can be expressed generally in terms of security services (data confidentiality, message integrity, user or device authentication, etc) and scalability (number of public keys to be handled, automatic distribution of digital certificates, etc). Consequently, these security infrastructures are considered as cost-effective security solutions for wide-range networks. Many applications had already leverage the power and usefulness of PKI: secure emails (*e.g.* PGP), secure web surfing (*e.g.* HTTPS), remote server access (*e.g.* SSH), banking transactions, or administrative signed electronic forms are among the existing applications relying on PKI.

In this context, several standardization activities relevant to PKI are underway in order to match the security requirements of current and future IT networks.

[4] PGP is a protocol used to enhance the security of e-mail communications by providing cryptographic privacy and authentication mechanisms for exchanged data based on the trust concept.

[5] http://datatracker.ietf.org/wg/pkix/charter/

5.1.3 PKI Standardization Activities

The aim behind these standardization activities is globally the same: define the main PKI components discussed above and point out the use of certificate formats depending on the specific environment or context of use. Some working groups deal with the certificate format definition such as the IETF SPKI (Simple Public Key Infrastructure)[6] group, the ITU-T X.509 certificate group [ISO 2008b], or the OpenPGP working group[7]. Other working groups establish the profiles of certificate formats for specific uses such as the IETF PKIX (Public Key Infrastructure X.509), IPSec and TLS working groups, the Open Mobile alliance forum[8], the S/MIME (Secure Multi-purpose Internet Mail Exchange) group[9], or PKCS (Public Key Cryptography Standards)[10] which is a series of specifications issued by the RSA Security Incorporation to deal with certificate structures and cryptographic algorithm usage for several applications. Some of these PKCS standards are standardized by the IETF into corresponding RFCs.

However, these standardization activities have no consideration for the specific context of civil aviation and ATM systems. Besides, as we discussed in chapter 2, the future datalink systems will rely on secure communications using either a combination of COTS technologies (*e.g.* the SecMan framework introduced in chapter 4) or security protocols specific to the aeronautical context (*e.g.* AMS). In both cases, as the number of aircraft increases and aeronautical networks grow in size, it is apparent that the scalability issues can not be maintained without the use of a dedicated PKI.

The civil aviations actors (airlines, aircraft manufacturers, civil aviation organizations) have already shown their interest for PKI-based security solutions, an overview of the related work is provided in the next subsection.

5.2 PKI in Future Datalink Systems

Many research work have been carried on PKI to enhance the security of next generation connected aircraft. These work can be categorized depending on the area of use of each specific PKI-based solution.

Areas of use for PKI-based security solutions in future datalink systems can be arranged as the following:

- Security of proprietary aeronautical protocols;
- Identity management;
- Security of maintenance and software data loading;
- Electronic form signature.

5.2.1 Areas of Use for PKI in Future Datalink systems

5.2.1.1 Security of Proprietary Aeronautical Protocols

Some work suggested the use of a PKI to provide the cryptographic credentials needed by a specific aeronautical communication protocol.

[6] http://datatracker.ietf.org/wg/spki/charter/

[7] http://www.openpgp.org/

[8] http://www.openmobilealliance.org/

[9] http://datatracker.ietf.org/wg/smime/charter/

[10] http://www.rsa.com/rsalabs/node.asp?id=2124

In this context, we can mention the work presented in [Getachew & Griner 2005] where the authors investigated an elliptic curve-based authentication protocol for the CPDLC application. As far as public keys and certificates are needed, the authors suggested the use of a PKI to create and distribute the credentials between the aircraft CPDLC client and the ground CPDLC server. The same way, AMS (which is as a reminder the secure version of the ACARS protocol) relies on a PKI for the key and certificate management lifecycle. Figure 5.2 illustrates the key lifecycle management as it appears in part 2 of the ARINC 823 standard [ARINC 2007a]:

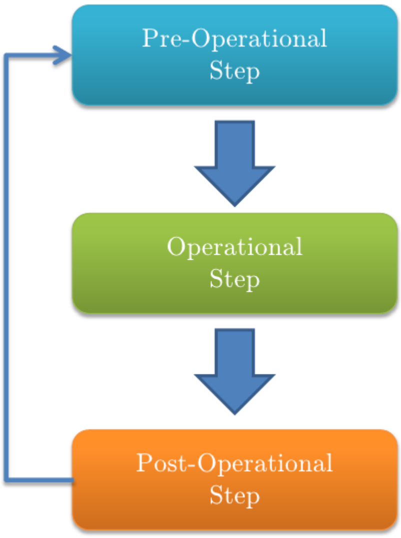


Figure 5.2: Key and Certificate Life Cycle Overview in Secure ACARS

This three steps certificate management lifecycle works as the following: in the pre-operational phase, entities are initialized within the PKI before using the security mechanisms implemented in AMS. At the operational phase, the security services are invoked using all the keys and certificates previously generated and distributed in the pre-operational phase. Finally, in the post-operational phase, keys and certificates are revoked when it is required by the PKI security policy.

The ICAO has dedicated an entire section to PKIs in the manual of technical provisions for the ATN [ICAO 2002]. Certificate policy, practices statements, certificate and CRLs formats, and CRL revocation schemes are specifically described. However, there is no indication on the PKI structure or the trust models to rely on in the last version of the document (*i.e.* DOC 9705 edition 3).

5.2.1.2 Identity Management

With the paradigm shift to digital data communications, the aeronautical industry needs an identity management infrastructure for all the actors involved in the ATM systems. Aircraft (ICAO unique number), avionic systems, passengers and airline/crew staff (X.509 certificates), devices (servers, routers, etc): all these entities need to be identified as legitimate users in the ATN network. [Patel & McParland 2001] tried to address some of these identity management issues in the specific FAA ATM environment. Figure 5.3 depicts an abstraction view of the PKI-based solution proposed by the authors as it appeared in [Patel & McParland 2001].

The authors provided this PKI-based solution in order to handle all the identities involved in the ATM environment with a particular emphasis on the ATN and FAA facilities (routers, end systems, LANs, etc). They suggested the use of cross certification to handle inter-domain identity management as an extension of the model presented in figure 5.3 (cross-certification will be detailed later in section

5.3.3). However, the authors have not provided technical details about the structure or the hierarchy of the extended PKI (how the CAs are linked? how certificates and keys are signed and exchanged between CAs? are there root CAs, bridge CAs, or delegated CAs? etc). Moreover, they assumed CRLs to be the by default certificate revocation scheme regardless of how much CRLs could be relevant compared to other alternatives such as online verification servers.

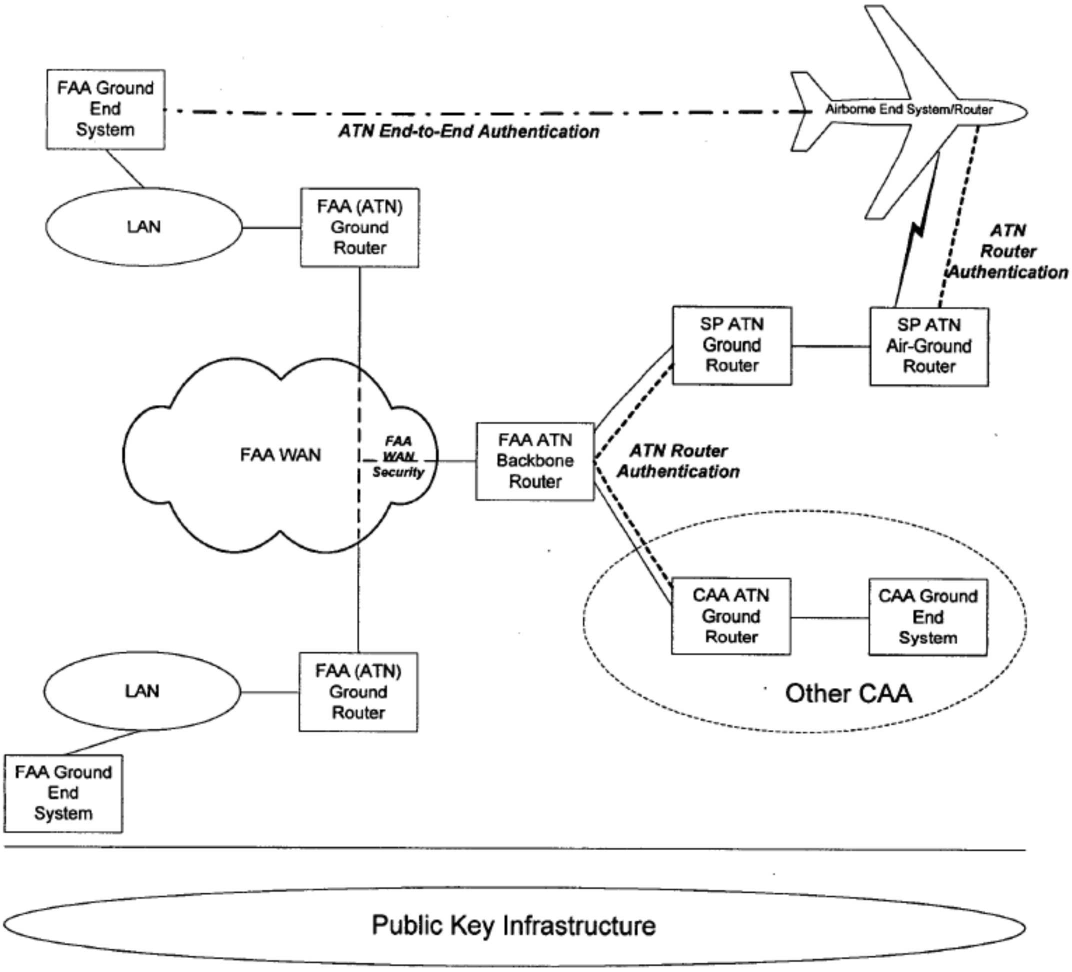


Figure 5.3: PKI Support for the FAA ATM [Patel & McParland 2001]

In [Nigringy & Phaltankar 2007], authors identified the main challenges of identity management in commercial aviation using a trusted third party PKI bridge called CertiPath[11]. However, the paper seemed more like a commercial advertisement than a technical description of the implemented PKI solution. The only deduction that can be made referring to this work is the interconnection between the trusted third party bridge and the CAs deployed by the commercial aviation actors (aircraft manufacturers, service suppliers, ground operators, airlines). Still, there is a lack of accuracy regarding the cross-certification between those CAs, key/certificates exchanges, or revocation schemes to be used.

5.2.1.3 Security of Maintenance and Software Data Loading

In the previous and current generations of aircraft (*e.g.* A320 by Airbus), data loading have been performed using limited and out-of-date technologies such as floppy disks or CDs physically loaded into the aircraft. In civil aviation, data loading is an operational procedure conducted usually when the aircraft is still at the gate in order to update, maintain, or assess the aircraft avionic softwares.

[11] http://www.certipath.com/

Nowadays, with the new generation of connected aircraft, the data loading procedure is achieved by data providers: the aircraft receives updated information throughout the flight and diagnostics are carried out even if the aircraft is not on the ground. Civil aviation organizations such as the FAA stressed the criticality of these operations through several security and safety guidance (*e.g.* RTCA DO-178B level A safety-critical software [RTCA 1992]). Thus, data loading has strict security requirements that would be likely fulfilled thanks to PKIs.

In this line, a generic EDS system called AADS has been presented in [Robinson *et al.* 2007a]. The system is compliant with the Common Criteria methodology and a PKI has been used to secure the data and software loading into the aircraft. The authors identified safety and business threats, then suggested the use of digital signatures and a PKI to secure the model. More details about this PKI-based solution have been provided by the same authors in [Robinson *et al.* 2007c] where the security challenges rising from the electronic distribution of software into the aircraft have been deeply discussed. An extended AADS model that uses a PKI for digital signatures has been presented as illustrated in figure 5.4:

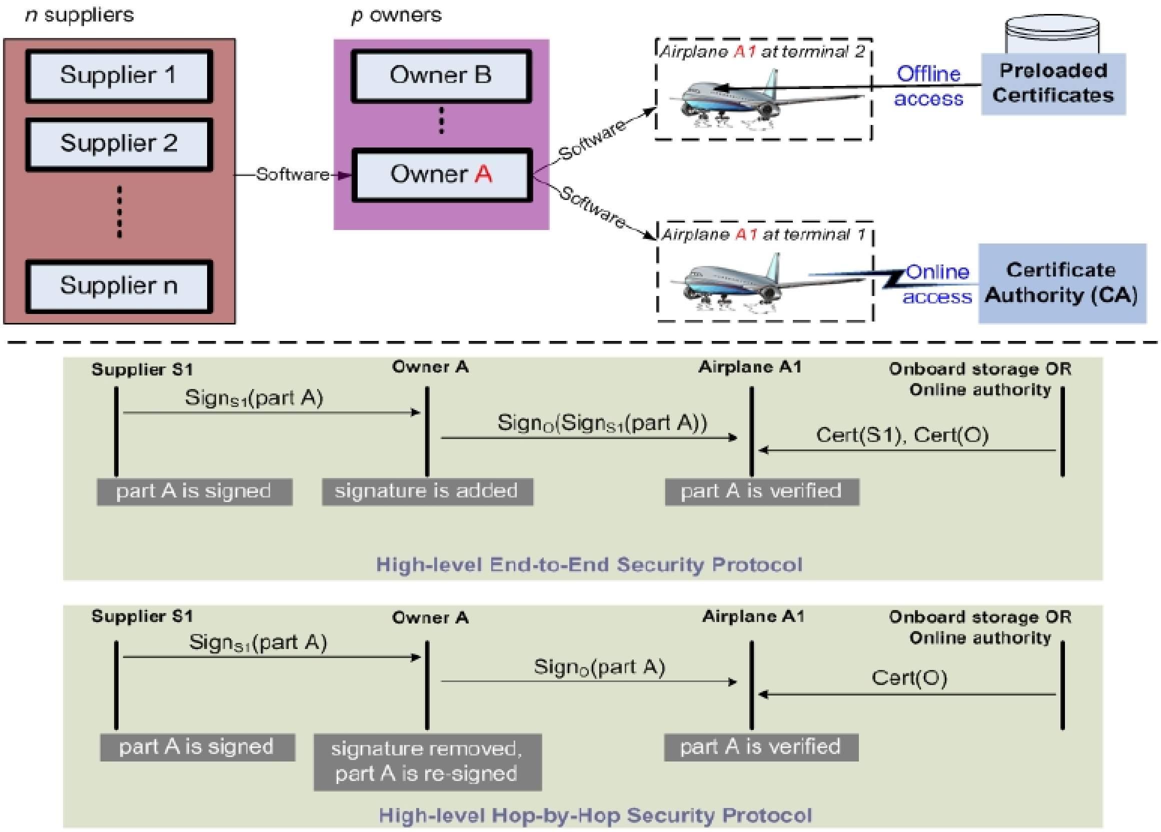


Figure 5.4: Secure Airplane Assets Distribution System [Robinson *et al.* 2007c]

The top half of figure 5.4 illustrates how the authors managed to securely distribute software data using either an offline preloaded certificates or an online dedicated CA. The bottom half of figure 5.4 illustrates high level protocols for secure AADS, with verification (end-to-end) or without verification (hop-by-hop) of supplier signature at the aircraft. In the end-to-end security protocol, each software supplier signs its own data parts, the owner receives this signature, verifies it, then adds its own signature and sends it to the aircraft. At the message receipt, the aircraft verifies both signatures using the adequate digital certificates retrieved beforehand from the preloaded certificate database or the issuing CA. This end-to-end approach is not scalable since the signature verification increases linearly with the number of certificates and signatures to check. In the second hop-by-hop approach, when the owner receives the software part signed by the supplier, it removes the signature and signs again the software part: in this case, the aircraft needs to verify only the owner signature.

This approach is more scalable than the first one but unfortunately it increases the computation overhead costs at the owner side. At the end of the paper, the authors made a comparison between all possible security schemes that satisfy AADS security properties. They concluded that an hybrid solution to secure the AADS application should rely on a PKI which provides online verification of certificate validity without any further details.

In order to provide answers to these questions, the same authors extended their work with a complete coverage of the security challenges and operational requirements needed by airlines to guarantee secure AADS transactions (chain of trust between AADS entities, AADS message integrity, certificate requirements for AADS, etc) [Robinson *et al.* 2007b]. They suggested the use of a single root CA that may be a third party vendor from which certificates are purchased, or an airline-based PKI. In order to do so, two approaches have been presented, respectively an ad-hoc technique and a structured approach employing a commercial PKI for the EDS application.

The ad-hoc approach consisted in pre-loading trusted certificates on aircraft via an OOB mechanism. In this ad-hoc approach, the authors suggested the use of external PKIs and self-signed certificates for digital signatures. The aircraft would have its own certificate that could be self-signed to protect data sent to other airline parties. For data loading operations, suppliers sign the aircraft software parts to ensure the data integrity. This signature could be removed by a manufacturer or an airline if needed and replaced with new signatures as it has been done in the hop-by-hop approach presented by the authors in [Robinson *et al.* 2007c]. The main advantage of the solution is its simplicity and reduced cost, the big drawback is that it does not consider the scaling issues we discussed earlier in chapter 2.

The structured PKI solution seems much more appropriate and offers long-term benefits in terms of scalability. But the authors considered it more expensive than the ad-hoc solution, specifically because of the PKI setting up and maintenance costs. They discussed also the certificate revocation main techniques and recommended the use of CRLs for checking certificate status at the aircraft in order to avoid the necessity of a direct connectivity with external networks (which is a pre-condition imposed by the use of an online verification server).

5.2.1.4 Electronic Form Signature

All the paper-based documents requiring a physical signature are progressively moving to the electronic form. The aviation industry has already moved in that way with the FAA Form 8130-3 [Engdahl 2011] or the EASA Form 1 [EASA 2007] for instance. These electronic forms are used thanks to new embedded device such as the EFB system that helps the crew, and more specifically the pilot, to carry on and perform flight management tasks efficiency with less papers (aircraft operating manual, passenger list, navigation maps, etc). Many benefits can be drawn from using such electronic devices instead of the current paper forms:

- electronic documents are easier to store and to analyze compared to paper-based documents;
- there is less uncertainty when authentication is based on digital signatures compared to handwritten signatures where traceability is not always obvious (*e.g.* human errors, paper deterioration, etc);
- it is more difficult to impersonate legitimate users or alter data when digital certificates are achieved using electronic form signatures;
- electronic forms are more convenient for flight crew than paper documents, mainly when several forms have to be handled by pilots or airline staff members.

As we have seen, there are many PKI-based security solutions that have been provided lately for the civil aviation industry. However, even if PKI is nowadays considered as a mature concept for future ATM systems, several issues need to be addressed. Despites the proposed solutions, specific requirements shall be fulfilled in order to effectively see an entirely dedicated PKI for ATM systems working as it should.

5.2.2 Requirements for ATM-dedicated PKI models

In a large heterogeneous network like the ATN, several aspects have to be considered for building up the PKI model meant to be used for the future connected aircraft applications. As mentioned in chapter 2, a PKI-based security platform should be a proper solution for key management with a design considering the following points:

◇ *Scalability*: should be more than ever examined when the aeronautical-dedicated PKI has to be imagined, modeled, then deployed. With the increasing air traffic and passenger load, the amount of cryptographic credentials becomes too heavy to manage by a single root CA. Besides, the heterogeneity of end entities requires different certificates with different formats to be handled. For instance, the aircraft certificate ID will be likely the ICAO unique number as it has been recommended in the ATA Spec 42 document whereas the passenger certificate will be likely the traditional X.509 based certificate as it is the case in the majority of Internet applications. Moreover, the designers should keep in mind that the PKI-exclusive data exchanges (certificates, keys, signatures, etc) induce a considerable overhead on the communication channel;

◇ *Interoperability*: the end entities are likely to be signed by different CAs, bringing many interoperability and roaming issues. For instance, if an airline CA does not have any type of agreement with other airlines in order to recognize the aircraft certificate signature, this aircraft could not be authenticated or identified properly as a valid communication party outside its own domain. That is to say, the PKI model must be able to deal with all tasks within the airline domain, while ensuring enough flexibility and interoperability with external end entities registered out of their scope;

◇ *Robustness*: the chain of trust between the PKI components should be also clearly stated and a single point of failure must be avoided, meaning that if one PKI component has been compromised, it should not have side effects on other parts of the communication systems.

The PKI has to integrate all these considerations to be fully operational and reliable. Regarding the requirements discussed above, table 5.1 summarizes the main contributions (*c.f.* section 5.2.1) made in the field of ATM systems using PKI models. As we can see, there is no solution that covers all the criteria. Furthermore, most of them are dedicated to a specific use (*i.e.* securing a proprietary protocol or application), meaning that these PKI solutions are not reusable for other purposes. Moreover, the scalability issues have rarely been addressed. Also, the CRL certificate revocation scheme has been considered as a *"by default solution"* since almost all these contributions relied on certificate revocation lists. It is true that, historically speaking, CRLs are older than online checking status protocols, but as we are about to see in section 5.2.3.1, every revocation scheme has its own advantages and drawbacks, and it is worthy to compare both approaches.

Our PKI model addresses the issues identified in these later work: the solution presented in section 5.3 takes into consideration the scalability issues through a thoughtful design of the trust model across three different certification levels based on entity location. Our solution is protocol-free, meaning it can be used for any aeronautical application that requires an operational PKI procedure in the background. The trust model allows a more cost-effective PKI transactions, meaning it decreases the air-ground signaling overhead required by keys and certificate exchanges.

Table 5.1: Aeronautical PKI-based Related Work Summary

Reference	**Scalability**	**Interoperability**	**Robustness**
[Getachew & Griner 2005]	×	×	×
[ARINC 2007a]	×	✓	×
[Nigringy & Phaltankar 2007]	×	✓	×
[Robinson *et al.* 2007a]	×	×	×
[Robinson *et al.* 2007c]	✓	×	×
[Robinson *et al.* 2007b]	×	✓	✓

Legend
✓: covered
×: not covered

Interoperability is also tackled since the model integrates the inter-airline communications. Besides, the design of the PKI model is driven by guidance provided by the ATA DSWG introduced in chapter 2. Indeed, the DSWG is the main working group dealing with PKI concepts in civil aviation. The working group aims to provide and update a document titled *"ATA Specification 42 - Aviation Industry Standards for Digital Information Security"* [ATA 2009], which includes a set of guidance and recommendations about the use of PKI in the aviation industry.

In order to understand the design choices we made in our PKI model, we provide a brief summary of the ATA Spec 42 document content in the next subsection.

5.2.3 ATA Spec 42 Document

The ATA Spec 42 document details the PKI requirements and specifications for the civil aviation industry. The document is structured as the following:

- ◇ the first chapter is a general introduction to the security challenges of future connected aircraft. The benefits of PKI are announced and fundamental security principles are recalled with regard to the civil aviation needs;
- ◇ the second chapter is dedicated to the identity and access management. Security considerations (*c.f.* chapter 2), PKI components and management procedures are also detailed (registration, key management/distribution, etc). CPs (Certificate policies)[12] are finally discussed for three purposes of use: end entities, software assurance (digitally signed software code, AADS), and ACARS message security;
- ◇ the third chapter introduces the PKI implementation guidances in civil aviation industry. These recommendations are mainly provided to help system designers whom aim to develop or implement a PKI-based security platform in the ATN network. The covered topics are: identity proof, key lifecycle management, certificate validity, and trust anchor management. This chapter is by far the most relevant to our work. Consequently, our contribution is driven by these guidances, such as the deployment of an airline dedicated CA, or the use of CRLS and OCSP certificate revocation schemes.

Thus, before analyzing how CRLs and OCSP behave in our performance-aware PKI model, we make a brief introduction for these two revocation approaches.

[12] These CPs are consistent with the IETF PKIX X.509 certificate described in the RFC 3647 [Chokhani *et al.* 2003]

5.2.3.1 Certificate Revocation Schemes

Certificate validation is the process of checking the validity of a given digital certificate. Two parameters are usually tested: the certificate validity period and the digital signature of the CA. Optionally, the certificate path and the trust anchor validations are performed. The final step is to check whether the certificate has been revoked or not. A certificate should be revoked before its expiration date if the CA suspects a public/private key comprise (the certificate becomes useless since the public key attached to it is mathematically related to the private key), a change of one or many certificate fields (*e.g.* subject name, algorithm used to generate the key pairs, etc), or a modification in the chain of trust between the CAs and the end entity.

In a safety-related context such as data link communications, the revocation scheme is an important process: any implemented PKI has to necessarily deploy a mechanism for revoking certificates and inform all involved entities about the certificate status. For instance, if the aircraft certificate has been corrupted and not revoked at time, an attacker could impersonate the identity of the aircraft and communicate with ground entities without being discovered. As mentioned before, the ATA Spec 42 document focused mainly on two certificate revocation schemes: the CRLs and the OCSP protocol.

Certificate Revocation Lists

The commonly used approach is to periodically publish a CRL which is a list of all revoked certificates within a given domain. A CRL is usually time-stamped and signed by the issuing CA. It contains the certificate serial numbers of all revoked certificate within the managed domain as shown in figure 5.5:

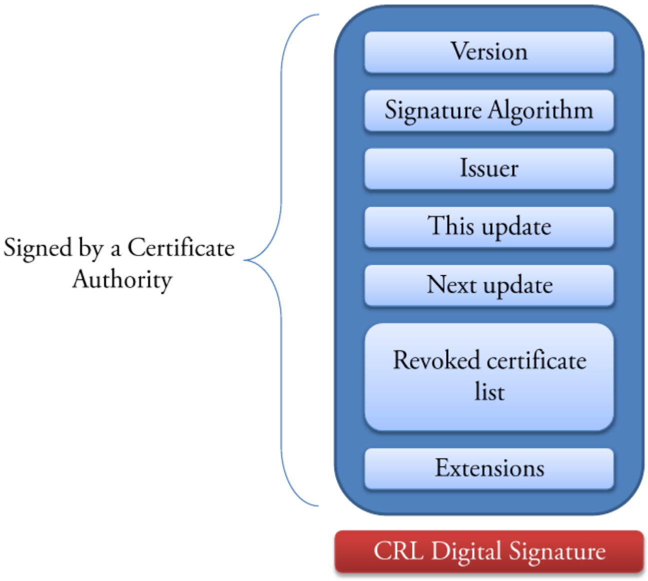


Figure 5.5: Basic CRL Structure

The following fields are included in the CRL structure:

- ◇ *Version*: indicates the version of the CRL;
- ◇ *Signature algorithm*: provides the algorithm OID used to in the CRL signature;
- ◇ *Issuer*: The CA name (*e.g.* VerySign);

- ◇ *This update*: CRL issuance date;
- ◇ *Next update*: Next CRL issuance date;
- ◇ *Revoked certificate list*: list of revoked certificates using their unique serial numbers and the corresponding revocation date;
- ◇ *Extensions*: if used, CRL extensions are specified in this field.

The main drawback of such an approach is that the CRL size grows very fast for large domains. Consequently, the induced network load becomes significant and in most cases unacceptable for remote clients downloading these lists. Cache techniques can be used at end entities, but it is still difficult to define the frequency of CRL updates and to get a fresh revocation list.

As a response to the overload and high update frequency issues induced by the CRL-based revocation schemes, a second approach consists in an online server that uses network protocols to check the certificate revocation status.

Online Check Status Protocol

Compared to CRLs, the main advantage of an online checking server is to request a targeted certificate status instead of a full revocation list where only one entry matters for the verifier. OCSP is a very simple request/reply based protocol designed to check the revocation status exclusively: an end entity requests the revocation information for one or more certificates using an OCSP request to the OCSP server. The OCSP responder checks the revocation status information and issues an OCSP response containing the certificate ID and the certificate status to the end entity. The OCSP protocol works as follows:

1. The OCSP client issues a status request to the OSCP responder:

   ◇ $OCSP_{clt} \rightarrow OCSP_{rsp} : version|serviceRequest|certificateID|extensions$

   The version field specifies the OCSP version used to issue the request, service request indicates which service is requested from the responder, certificate ID provides the unique serial numbers of the targeted certificates, and extensions may be used if needed by the responder;

2. The OCSP client suspends acceptance of the considered certificate until the responder provides the following answer:

   ◇ $OCSP_{rsp} \rightarrow OCSP_{clt} : version|responderName|certificateResponses|extensions|signature|$ $sign(hash(rsp))$;

   Before sending the message, the OCSP responder checks if the request is well formed, if the responder is configured to provide the requested service, and if all information needed are contained in the request. Then, it sends back a response composed of the OCSP syntax version, the name of the responder, as many responses as certificate requests, optional extensions, the object identifier of the used signature algorithm, and a signature computed on a hash of the response.

   The response for each certificate is composed of:

   - ▷ the target certificate ID;
   - ▷ the certificate status value which can be `GOOD`, `REVOKED`, or `UNKOWN`;
   - ▷ a response validity interval and optional extensions.

Despites its undeniable advantages over CRLs such as a lower validation processing time, a lighter overhead, and real-time checking responses, OCSP still suffers from several weaknesses. Firstly, the server response has to be signed in order to avoid server impersonation or forgery attacks. Secondly, more processing and signaling overheads have to be added to the network for each OCSP response: if there are many clients overflowing the same server with OCSP requests, the responder may take longtime to generate the signatures and answers to each OCSP request.

Also, from a robustness point of view, the main issue is that the OCSP server has to be always online, so that it remains reachable by OCSP clients, which makes it vulnerable to DoS attacks. In the simulation results presented further in this chapter, both CRLs and OCSP are analyzed regarding their performances (*i.e.* the induced signaling overhead on the air-ground data link channel) and scalability factor (*i.e.* depending on the network size, which certificate revocation scheme performs the best).

In order to fairly compare these two certificate revocation schemes, we introduce the PKI models we relied on in our analysis: a reference PKI model and our performance-aware PKI model.

5.3 A Scalable PKI for Future Datalink Systems

Several types of PKI have been defined for ground networks [Perlman 1999]. These PKI models can be indexed depending on the chain of trust between the CAs involved in the so called PKI. Indeed, it is usually necessary to have more than one CA when end users belong to different domains managed by different certificate authorities. In the specific aeronautical context, airlines should deploy their own CA. For the performance analysis performed further in this section, we have chosen a single CA model as a reference PKI model with one root certificate authority node deployed per airline. Our model relies on an enhanced multi-rooted CAs organized in a tree-based hierarchical way.

In the next sub-section, we provide an overview of the existing PKI models and justify why the multi-rooted hierarchical model should perform the best for the future ATM environment.

5.3.1 Overview of Existing PKI Models

According to the requirements mentioned in section 5.2.2, several PKI trust models can be considered for the future ATM system and categorized in the following groups:

1. *Bridge model*: [Nigringy & Phaltankar 2007] proposed a PKI model using a bridge CA as shown in figure 5.6. The idea is to establish trust relationships between CAs using cross certification: if a root CA needs to connect with other root CAs, it must go through the bridge CA. The main disadvantage of such a model is that the bridge CA is a centralized component: it will likely falls down because of a simple DoS attack, making all the communications and PKI procedures unavailable over the network. The solid line refers to an unidirectional certificate issuance, meaning a one-way trust relationship while dashed line refers to CAs cross-certification;

2. *Mesh model*: in this model, there is no hierarchical relationships between the CAs, meaning all the certificate authorities are equally arranged in the PKI as illustrated in figure 5.7. Consequently, bi-directional cross-certifications are needed in order to find a trusted relationship between CAs. The mesh model has good performances in small networks but it is not recommended for large scale domains since it is difficult to find rapidly a valid certificate path. Indeed, there is a high number of possible paths which can be tested between the certificate verifier and the CA trust anchor before finding the correct chain;

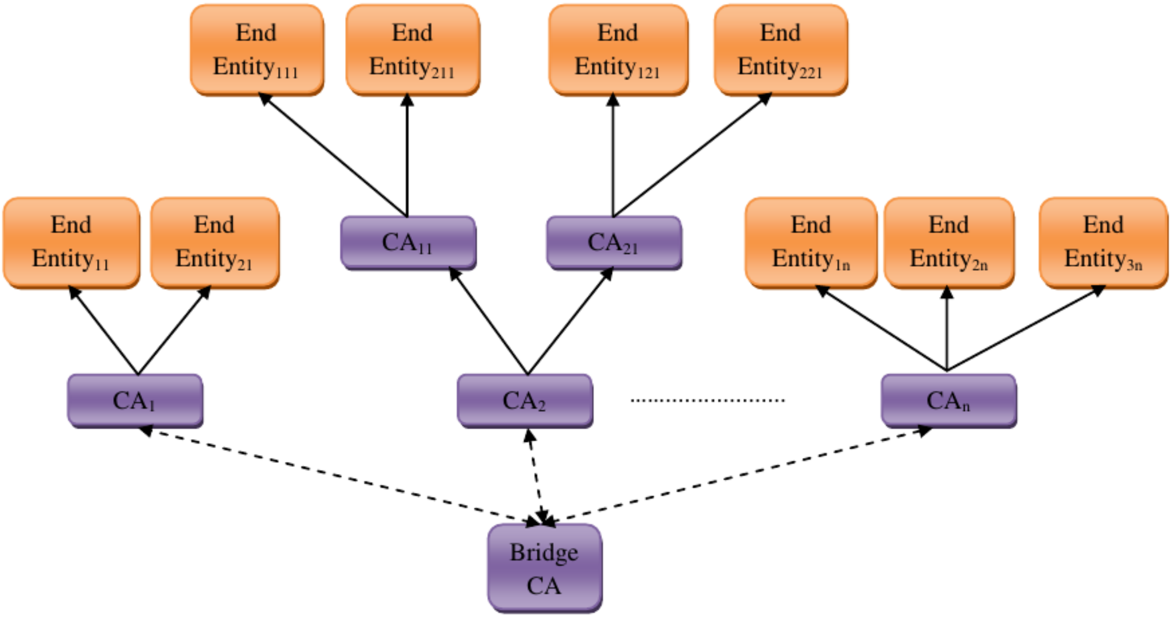


Figure 5.6: Bridge PKI Model

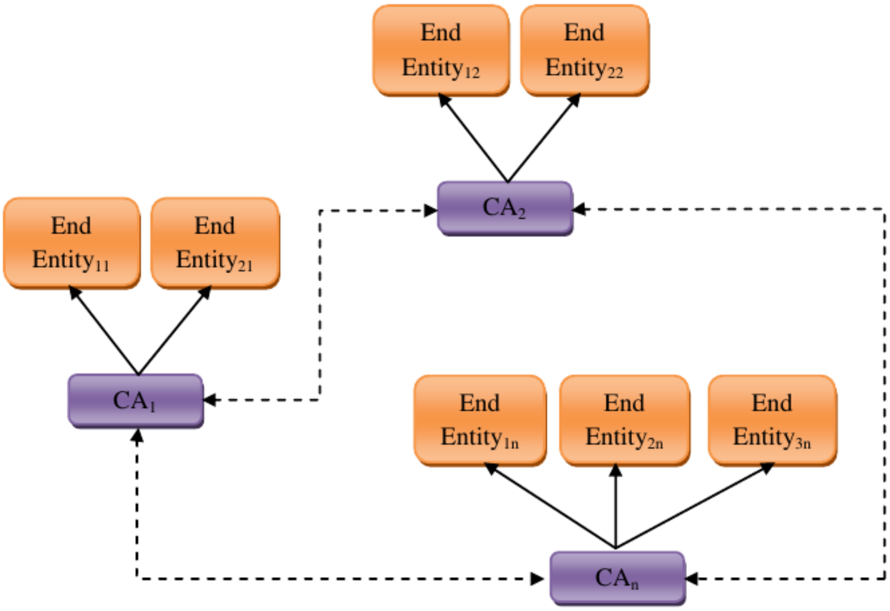


Figure 5.7: Mesh PKI Model

3. *Anarchy model*: this is the PKI model used in the PGP protocol for email signature verification. Every end entity can generate its own certificate and get others' certificate using emails or public databases such as the MIT (Massachusetts Institute of Technology) PGP key server[13]. In the anarchy model depicted in figure 5.8, there is no effective CA to sign the certificate, the idea is that if you know a person, you can sign his certificate with your own private key (relying on the concept of trusted community). Obviously, this model does not perform well beyond a small scale network where the concept of trusted community is hard to establish (*e.g.* Internet):

[13] http://pgp.mit.edu/

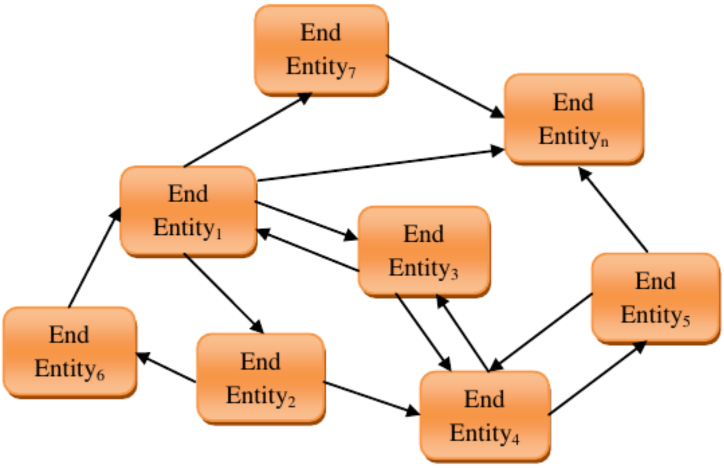


Figure 5.8: Anarchy PKI Model

4. *Single-rooted model*: in this model illustrated in figure 5.9, a single root CA is deployed in a hierarchical PKI. The model is more or less similar to the bridge model except that there is no need for cross-certification between local CAs and the root CA, which is considered as the only trusted node. As for the bridge CA, the single root CA remains a point of failure in the model;

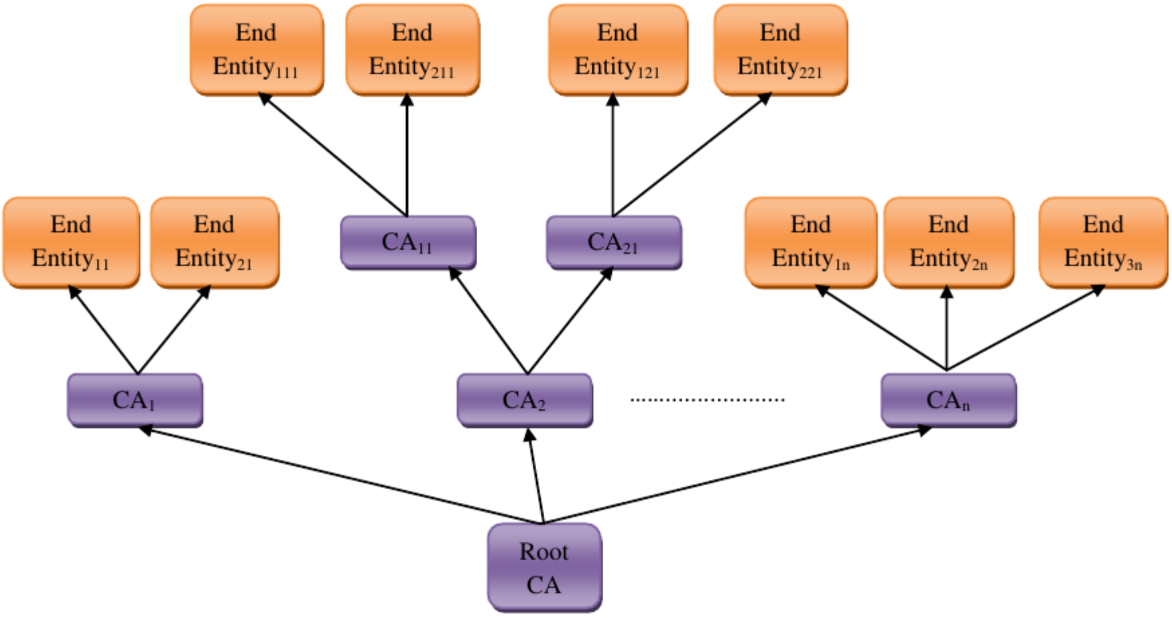


Figure 5.9: Single Rooted Hierarchical PKI Model

5. *Multi-rooted model*: this model is derived from the single rooted hierarchical PKI, but it includes several trusted root CAs arranged at the same level. The model scales well in large networks (*e.g.* Internet) but there is a lack of mutual authentications between the root CAs, which induces some security weaknesses. For instance, if a root CA is compromised, each user in the network should be informed in order to revoke the certificates signed by the attacked CA. If there is no agreements or communications between the trusted root CAs, a certificate signed by the attacked CA node is considered valid while it shouldn't. Figure 5.10 shows the principles of the multi-rooted model:

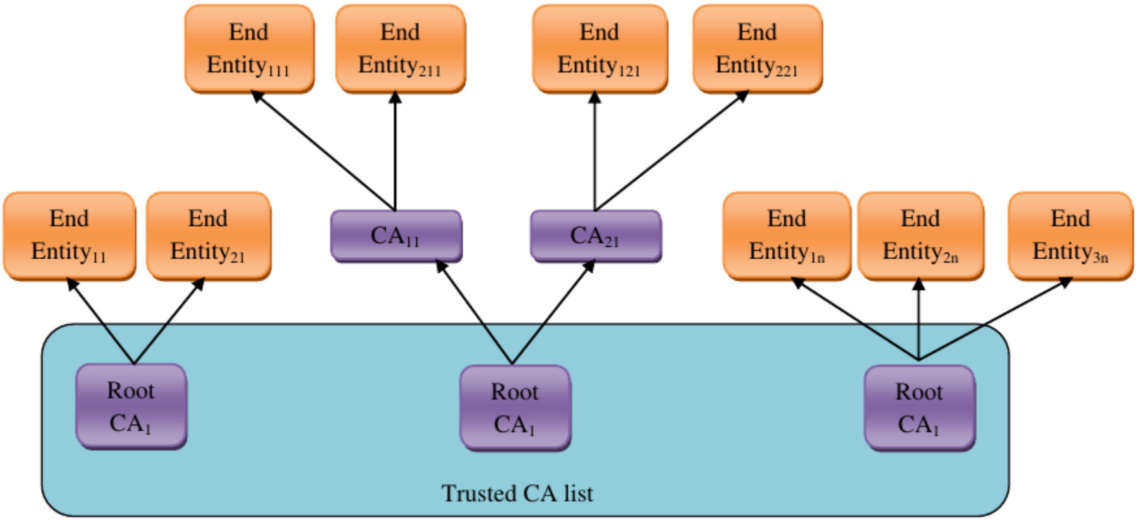


Figure 5.10: Multi Rooted Hierarchical PKI Model

Table 5.2 resumes the advantages and drawbacks of each approach relatively to the requirements for ATM-dedicated PKI models previously listed in section 5.2.2. The multi-rooted PKI model seems to be the best fitted for the ATM systems requirements. The medium interoperability and robustness can be easily enhanced using cross-certifications between the trusted list of root CAs. In the aeronautical context, this could help in providing flexible trust relationships between rooted airline CAs. In this way, if a root CA becomes unreachable for any reason, it will be revoked and the end entity registered to that CA will be informed with an update message issued from another root CA. Also, having delegated CAs at a second level of the hierarchy allows to save network and computation resources while minimizing the induced signaling overhead.

Table 5.2: Comparison of PKI Model Approaches

PKI Model	**Scalability**	**Interoperability**	**Robustness**
Bridge model	high	medium	low
Mesh model	low	high	medium
Single-rooted model	low	high	low
Multi-rooted model	high	medium	medium
Anarchy model	low	low	medium

Table 5.3 summarizes the notations used in the following sections.

5.3.2 Reference PKI Model

Figure 5.11 shows the entities involved in the reference PKI model, each root CA is deployed by the airlines on the ground:

- *The certificate verifier*: aims to verify the validity of a certificate;
- *The certificate owner*: possesses the digital certificate to be verified;
- *The Certificate Management Subordinate Entity (CMSE)*: this is the entity through which the verifier is able to check the certificate status and its validity (*e.g.* an OCSP server). In most case, the CMSE is merged with the CA.

Table 5.3: Notations

Notation	Description
K_i^+	The public key of an entity i
K_i^-	The private key of an entity i
$\{i, K_i^+\}_{K_{CA}^-}$	i certificate issued by CA
N_c	Total number of certificates
N_f	Total number of flights
$Size_c$	Average size of a certificate
t_c	Certificate validity period (in days)
t_s	SSP validity period (in days)
h_s	SSP digest using a hash function
$Nonce_i$	i^{th} randomly generated number
l_{sig}	digital signature length
l_{sn}	Certificate serial number length
C_{sig}	Signature generation time
C_v	Signature verification time
M	Exchanged data
O	Certificate owner
V	Certificate verifier
GCA	ground CA
R_c	% of revoked certificates
N_R	Frequency of certificate revocation check status messages per day
N_U	Frequency of revocation information update messages per day
$N_{c,CA}$	Certificate average number handled by CA
C_U^{Net}	Network cost to update a certificate between a CA and the CMSE
$C_{U,CA}^{CPU}$	Computation cost at CA to update a certificate
$C_{U,CMSE}^{CPU}$	Computation cost at CMSE to update a certificate
C_R^{Net}	Network cost to check a certificate between CMSE and a verifier
$C_{R,CA}^{CPU}$	Computation cost at CA to check a certificate
$C_{R,CMSE}^{CPU}$	Computation cost at CMSE to check a certificate
$C_{R,V}^{CPU}$	Computation cost at verifier to check a certificate

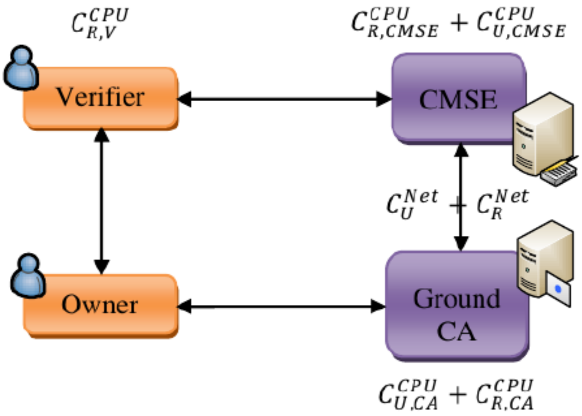

$C_{R,V}^{CPU}$
$C_{R,CMSE}^{CPU} + C_{U,CMSE}^{CPU}$
Verifier
CMSE
$C_U^{Net} + C_R^{Net}$
Owner
Ground CA
$C_{U,CA}^{CPU} + C_{R,CA}^{CPU}$

Figure 5.11: PKI Reference Model

It is important to note that both certificate verifier (noted V) and owner (noted O) can be either onboard or ground-located. For each entity, the relative computation and/or network cost is depicted in figure 5.11 as the following:

- ◇ $C_{R,V}^{CPU}$ is the computation cost at verifier to check a certificate validity;
- ◇ $C_{R,CMSE}^{CPU}$ and $C_{U,CMSE}^{CPU}$ are computation costs at CMSE respectively to check a certificate validity and to update a certificate information;
- ◇ $C_{R,CA}^{CPU}$ and $C_{U,CA}^{CPU}$ are computation costs at the ground CA (noted GCA) respectively to check a certificate validity and to update a certificate information;
- ◇ C_{R}^{Net} and C_{U}^{Net} are network costs between the CMSE and GCA respectively to check a certificate and to update a certificate information.

5.3.3 A Cross-Certified Multi-rooted Hierarchical PKI Model

In this section, we propose a multi-rooted hierarchical PKI model with cross-certification between trusted CAs. Figure 5.12 illustrates the model and the function of each entity:

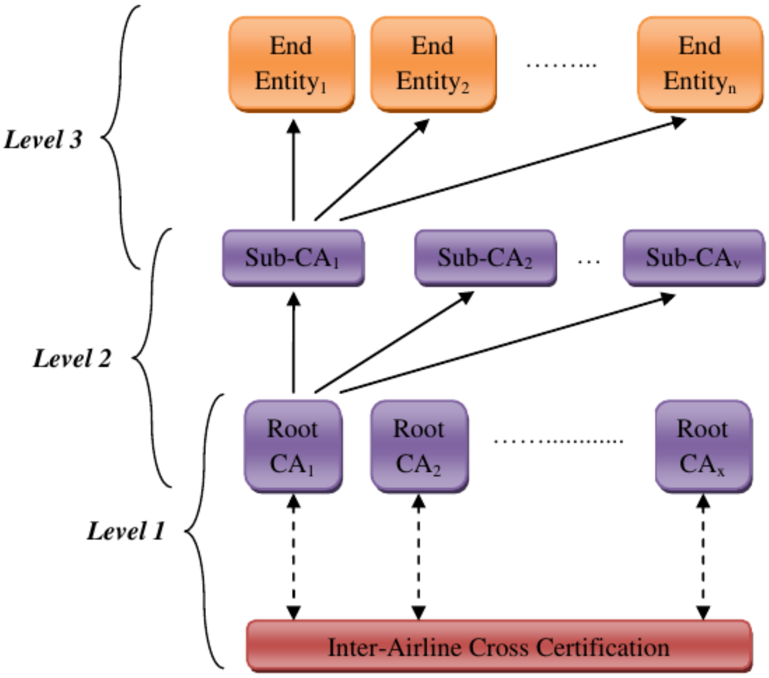


Figure 5.12: A Scalable PKI Model

In order to have a trusted relationship between third party authorities with end entities, and cost-effective communications in a large scale ATM system, it is suitable for CA to manage a limited user community. Therefore, the concept of trusted authorities is shared on different levels of the hierarchy as illustrated figure 5.12. This PKI model is divided in three levels:

1. *Level 1* is relevant to the inter-CA communications: a ground located root CA (denoted RCA later in the equations) is deployed for each airline and is responsible for all the end entities that belong to this airline. Indeed, in the ATA Spec 42 document, it is clearly stated that the CA should be deployed by a trusted agent like an airline. The end entity can be on the ground (*e.g.* an ATN

router) or on the air (*e.g.* an aircraft). Besides, as long as every root airline CA is independent of the others and has the authority on the aircraft labeled within its own domain, cross certification can be used between the ground trusted CAs. Cross-certification generally occurs between inter-domain CAs, meaning that these CAs do not belong to the same domain. Thus, the autonomy of local ground CAs and interaction between end entities belonging to different airlines can be always provided. Basically, when a certificate validation is needed between two end entities which do not belong to the same airline at a higher level of the PKI model, the root CAs have to cross certify, meaning a given airline CA has to sign the certificate of another airline CA;

2. *Level 2* is relevant to the communications between the root CA of an airline and aircraft managed in the airline domain: delegated (or subordinate) CAs (and denoted SCA later in the equations) are deployed onboard each aircraft and used to handle the onboard certificate entities. The basic idea is that each subordinate CA onboard each aircraft operates as a master and trusted certificate authority for all the end entities embedded in the same aircraft. In MANETs, the idea of trusted community is of common use, specially where end entities work in several clustered groups as illustrated in figure 5.13. we used this idea as a starting point to reduce the air-ground overhead induced by PKI operations. In our model, a community is an aircraft encompassing all the end entities onboard (devices, physical entities, software, etc) and a sub-CA (the locally trusted certificate authority);

3. *Level 3* is the last level of the hierarchy and deals with communications between end entities and sub-CAs onboard each aircraft: the sub-CA works as a trusted third party entity for the community onboard the aircraft, and is responsible of managing all the certificates of these entities. Therefore, end entities do not need to contact the ground root CAs when a certificate or a key is needed, the only trusted authority for them is the sub-CA onboard the aircraft. As shown in figure 5.12, with these three different levels of the PKI hierarchy, the signaling messages relevant to PKI operations are restricted between the airline ground CAs and the end entities.

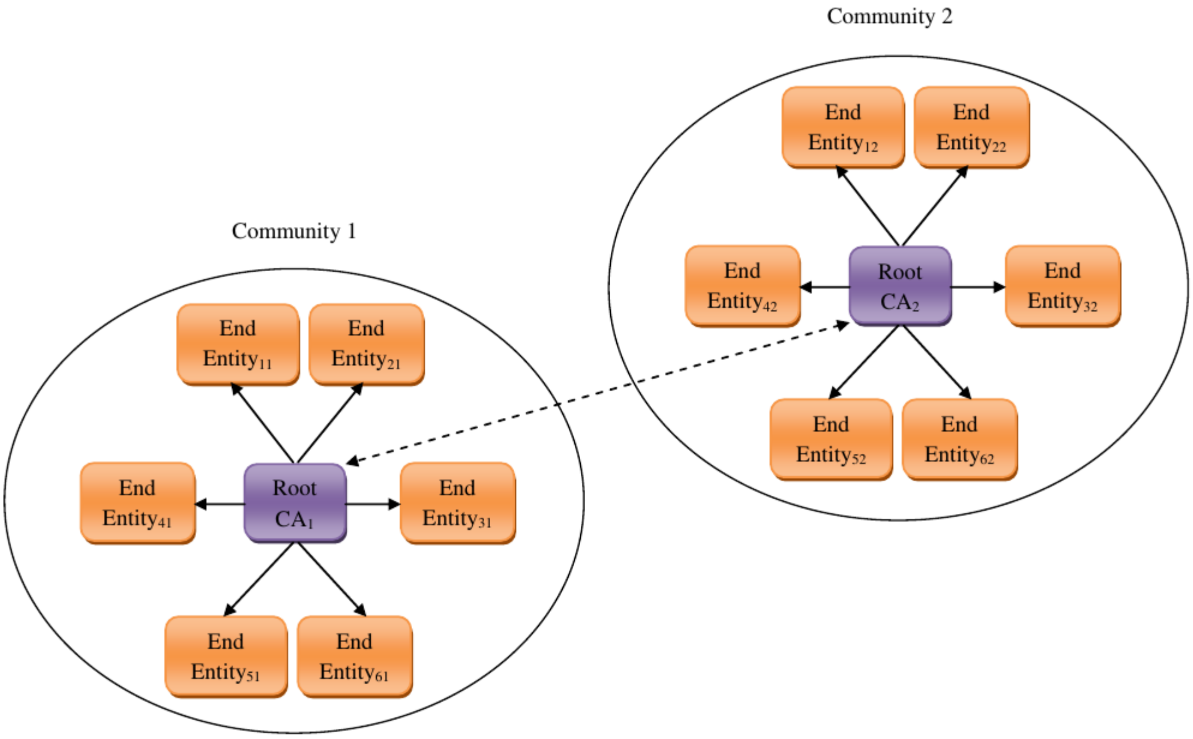


Figure 5.13: Concept of Trust Community in PKI

In order to clearly perceive the operations performed in this scalable PKI model, we provide a study case where two end entities at the higher level of the hierarchy want to communicate securely.

5.3.3.1 Illustrative Case Study

Figure 5.14 shows an illustrative case where two end entities aim to communicate securely in the cross-certified multi-rooted PKI model:

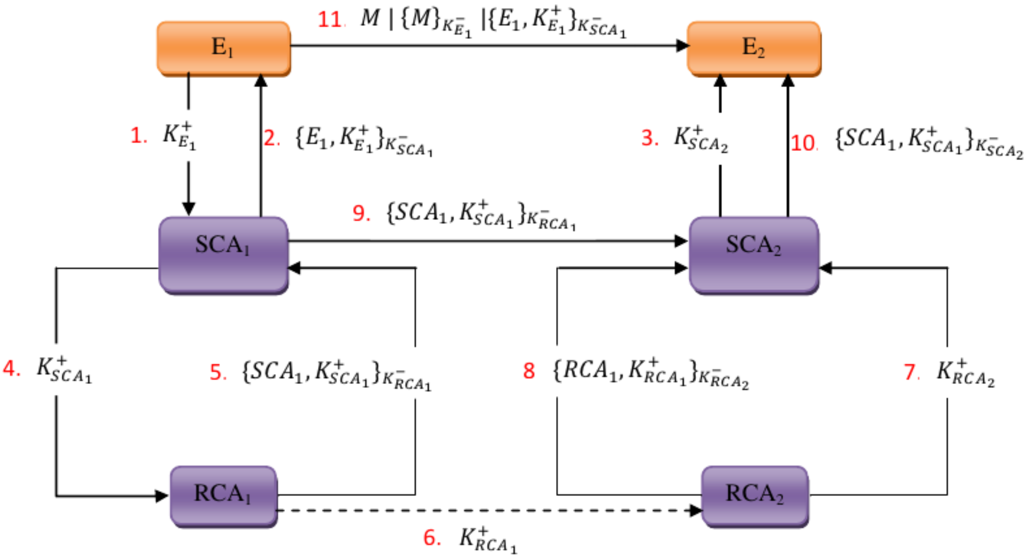

11. $M \mid \{M\}_{K^-_{E_1}} \mid \{E_1, K^+_{E_1}\}_{K^-_{SCA_1}}$
E1
E2
1. $K^+_{E_1}$
2. $\{E_1, K^+_{E_1}\}_{K^-_{SCA_1}}$
3. $K^+_{SCA_2}$
10. $\{SCA_1, K^+_{SCA_1}\}_{K^-_{SCA_2}}$
9. $\{SCA_1, K^+_{SCA_1}\}_{K^-_{RCA_1}}$
SCA1
SCA2
4. $K^+_{SCA_1}$
5. $\{SCA_1, K^+_{SCA_1}\}_{K^-_{RCA_1}}$
8 $\{RCA_1, K^+_{RCA_1}\}_{K^-_{RCA_2}}$
7. $K^+_{RCA_2}$
RCA1
RCA2
6. $K^+_{RCA_1}$

Figure 5.14: Inter-Airline Secure Communication Illustrative Case

In this example, we consider:

- Two airlines are involved in the communication, each one has it own ground-located root certificate authority (RCA_1 and RCA_2);
- Two end entities E_1 and E_2 are onboard different aircraft, each one registered to a given airline and managed by a different sub-CA (SCA_1 and SCA_2);
- E_1 wants to send a message M to E_2.

The illustrative case shown in figure 5.14 works as follows:

1. E_1 generates a public-private key pair and asks SCA_1 for a certificate by sending its public key $K^+_{E_1}$ using a trust mechanism. As far as E_1 and SCA_1 are onboard the same aircraft, an OOB mechanism can be used for such a communication. For instance, the public key could be sent using a symmetric key that has been embedded in a trusted software[14] before take-off (*i.e.* loaded when the aircraft is at the gate);
2. E_1 public key $K^+_{E_1}$ is binded to its identity information and a signed certificate $\{E_1, K^+_{E_1}\}_{K^-_{SCA_1}}$ is sent back to E_1 using the private key $K^-_{SCA_1}$ of SCA_1;

[14] This is the OOB mechanism used in current implementations of web browsers

3. E_2 obtains SCA_2 public key $K^+_{SCA_2}$ using a trust mechanism similar to the one used at the first message. However, airlines could have their own local PKI policies and the trust mechanisms used by two different airlines could be different;

4. at the same time, SCA_1 asks the trusted root CA RCA_1 for a certificate binded to its public key $K^+_{SCA_1}$;

5. RCA_1 signs the SCA_1 public key and sends back the $\{SCA_1, K^+_{SCA_1}\}_{K^-_{RCA_1}}$ certificate to the subordinate certificate authority;

6. at the lowest level of the hierarchy, a cross-certification between trusted airline CAs occurs between RCA_1 and RCA_2. The $K^+_{RCA_1}$ public key will be used later to verify the certificate of SCA_1 signed by RCA_1;

7. RCA_2 sends its public key $K^+_{RCA_2}$ to SCA_2 needed for the signature verification of the next message (message 8). This step can be avoided if the airline decides to locally store all trusted public keys onboard the aircraft before the take-off. However, such an embedded storage requires strong integrity and confidentiality protections. In order to avoid these storage issues and have a complete automated procedure, we decided in our model to send only required keys;

8. RCA_2 sends the RCA_1 public key signed with its own private key. After the reception of the $\{RCA_1, K^+_{RCA_1}\}_{K^-_{RCA_2}}$ certificate, SCA_2 will be able to verify the integrity of $K^+_{RCA_1}$ using the public key of its airline root CA RCA_2 and formally authenticate RCA_1 as a legitimate and trusted agent;

9. SCA_1 can send its public key to SCA_2 using the $\{SCA_1, K^+_{SCA_1}\}_{K^-_{RCA_1}}$ certificate received at message 5;

10. SCA_2 uses the public key $K^+_{RCA_1}$ deduced at message 8 to verify the signature of the certificate received at message 9. This allows SCA_2 to generate a new certificate $\{SCA_1, K^+_{SCA_1}\}_{K^-_{SCA_2}}$ for SCA_1 but this time signed with its own private key $K^-_{SCA_2}$. The certificate is sent to the end entity E_2 for the final signature verification;

11. finally, E_2 sends the message M signed with its own private key concatenated to an identity proof, meaning the certificate $\{E_1, K^+_{E_1}\}_{SCA^-_1}$ signed by SCA_1 and received at message 2. E_2 receives the $\{M\}|\{M\}_{K^-_{E_1}}|\{E_1, K^+_{E_1}\}_{SCA^-_1}$ message, verifies the validity of the E_1 public key using the public key of SCA_1 deduced at message 10, and checks the signature of $\{M\}_{K^-_{E_1}}$ to finally obtain the data M.

Now that the PKI model has been clearly defined, we proceed to the performance analysis of both PKI models. In the conducted simulations, passengers are assumed to be the only end entities holding a digital certificate onboard the aircraft. In fact, passenger statistics are more or less publicly available data, while it is relatively difficult to find accurate information on software data or embedded system devices using cryptographic credentials onboard the aircraft.

5.4 Performance Analysis

In this section, we compare the PKI models using three simulation scenarios, depending on the location of either the certificate verifier, the certificate owner or the root/subordinate CAs. The ground to ground case is useless here since there are no messages exchanged between the aircraft and ground entities, meaning no effect on the air-ground link resources. The comparison study have been conducted for two steps of the certificate lifecycle management: the certificate generation and distribution (CGD), and the certificate revocation (CRV) procedures (*c.f.* section 5.1.2). The primary goal was to evaluate the network and computation overheads generated by the different PKI models according to the physical locations of each entity. Our study has been based on a statistical approach of the number of flights in a specific airspace area and the relative number of passengers using cryptographic credentials.

5.4.1 Aircraft Source Data

For this purpose, we managed to use source traffic data[†] issued from the DSNA-DTI database. These are daily air traffic statistics for medium-range aircraft in the French airspace and are structured by hour of flight, aircraft family label (*e.g.* B738), and aircraft ICAO unique code as shown in table 5.4:

Table 5.4: a Sample from the DSNA-DTI Structured Source Data

Hour of Flight	Aircraft Label	ICAO Code
08:00:12	A320	GBL5MT
11:02:54	A319	AAF421
12:42:00	A321	MON954
14:00:01	C550	FYG583B
14:30:44	A320	JET327

In order to make this information more useful, we tried to estimate the maximum number of passengers that every aircraft carries, and then we extrapolated the results with the total number of aircraft. We relied on the EUROCONTROL performance database[15] and additional information about aircraft seats[16] to deduce the maximum capacity of each aircraft according to its ICAO code. Then, we synthesized the data and extracted the relevant information we needed. Moreover, as it has been suggested in a recent report of the French Civil Aviation Authority DGAC (Direction Générale de l'Aviation Civile) [DGAC 2010], we used an average aircraft filling percentage of 85% instead of the maximum aircraft capacity. Figure 5.15 shows the total number of flights and passengers per hour in the French sky:

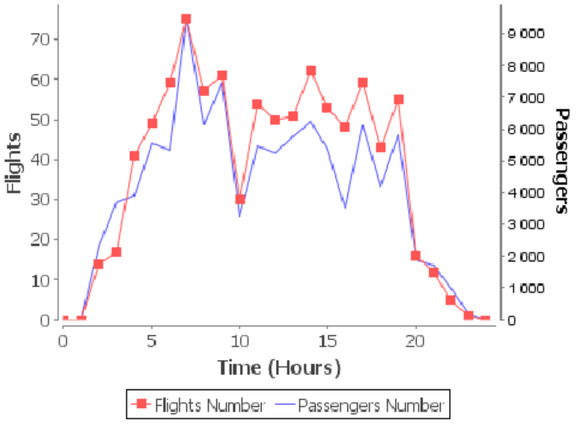


Figure 5.15: Daily Passenger and Aircraft Statistics (Air France Airline)

[†] Note that the aircraft source data have been provided by the DSNA-DTI, which is not a partner of the FAST project. It was our initiative to retreive these data from the French ANSP for the performance analysis provided in this section.

[15] http://www.elearning.ians.lu/aircraftperformance/

[16] http://www.seatguru.com/

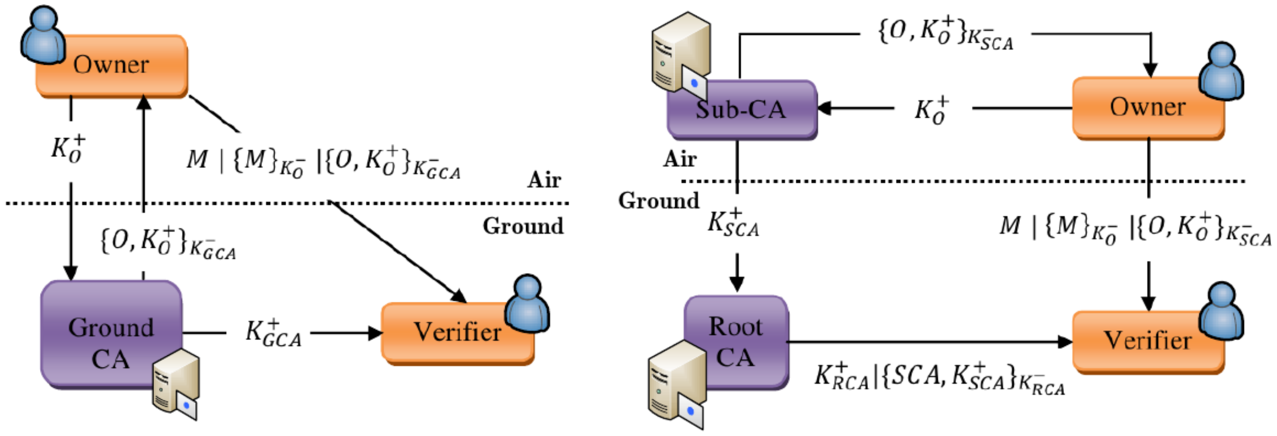
These statistics are relevant to the French *Air France* airline[17]: the observed average number of flights and passengers are respectively equal to 38 and 4200. These data are used to evaluate the certificate revocation overheads and the network/computational costs depending on the following experimental scenarios.

5.4.2 Experimental Scenarios

The pre-defined simulation scenarios have been established according the location of both the certificate owner and verifier. In all scenarios, the dashed line denotes the air-ground separation and M is the data message to send.

5.4.2.1 Scenario 1: Ground verifier and onboard owner

This is a typical case where a passenger sends a signed message to a ground entity which wants to proceed to the certificate verification. Figure 5.16 shows the exchanged data in both PKI models:


Owner
K_O^+
$M \mid \{M\}_{K_O^-} \mid \{O, K_O^+\}_{K_{GCA}^-}$
Air
Ground
$\{O, K_O^+\}_{K_{GCA}^-}$
Ground CA
K_{GCA}^+
Verifier
$\{O, K_O^+\}_{K_{SCA}^-}$
Sub-CA
K_O^+
Owner
Air
Ground
K_{SCA}^+
$M \mid \{M\}_{K_O^-} \mid \{O, K_O^+\}_{K_{SCA}^-}$
Root CA
$K_{RCA}^+ \mid \{SCA, K_{SCA}^+\}_{K_{RCA}^-}$
Verifier

(a) Reference Model

(b) Scalable Model

Figure 5.16: Scenario 1 PKI message exchanges

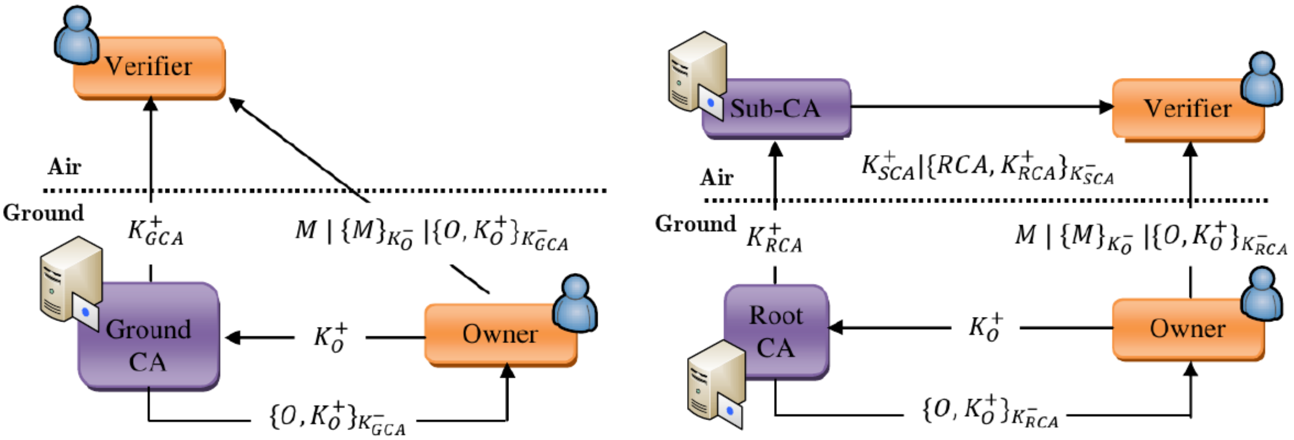
In the reference PKI model, the message sender asks for a certificate binded to its public key K_O^+. The airline ground CA generates the corresponding certificate $\{O, K_O^+\}_{K_{GCA}^-}$ signed with its own private key K_{GCA}^-. At the same time, the certificate verifier asks for the public key K_{GCA}^+ and use it to check the signature of the $\{O, K_O^+\}_{K_{GCA}^-}$ certificate. The procedure is obviously different in the hierarchical model because the sub-CA manages to generate the $\{O, K_O^+\}_{K_{SCA}^-}$ certificate of the message sender instead of the ground root CA. In order to proceed to the verification of the certificate signature, the verifier asks the root CA for the sub-CA public key K_{SCA}^+.

5.4.2.2 Scenario 2: Onboard verifier and ground owner

In this scenario, the certificate owner is on the ground and the verifier is onboard. The message exchanges relative to the reference model are quite similar to those presented in figure 5.16 except that there is no need to send the $\{O, K_O^+\}_{K_{GCA}^-}$ certificate to the aircraft this time as far as the owner is located at the ground. Only the ground airline CA public key K_{GCA}^+, needed for the signature verification of

[17] http://www.airfrance.com/indexCOM.html

the $\{O, K_O^+\}_{K_{GCA}^-}$ certificate, is sent to the onboard certificate verifier. In the scalable PKI model, we keep the same PKI overhead compared to those in figure 5.16: the public key K_{RCA}^+ and the message M binded to the adequate certificate $\{O, K_O^+\}_{K_{RCA}^-}$ are sent from the ground to the aircraft.


Verifier
Air
Ground
K_{GCA}^+
$M \mid \{M\}_{K_O^-} \mid \{O, K_O^+\}_{K_{GCA}^-}$
Ground CA
K_O^+
Owner
$\{O, K_O^+\}_{K_{GCA}^-}$
Sub-CA
Verifier
$K_{SCA}^+ \mid \{RCA, K_{RCA}^+\}_{K_{SCA}^-}$
Air
Ground
K_{RCA}^+
$M \mid \{M\}_{K_O^-} \mid \{O, K_O^+\}_{K_{RCA}^-}$
Root CA
K_O^+
Owner
$\{O, K_O^+\}_{K_{RCA}^-}$

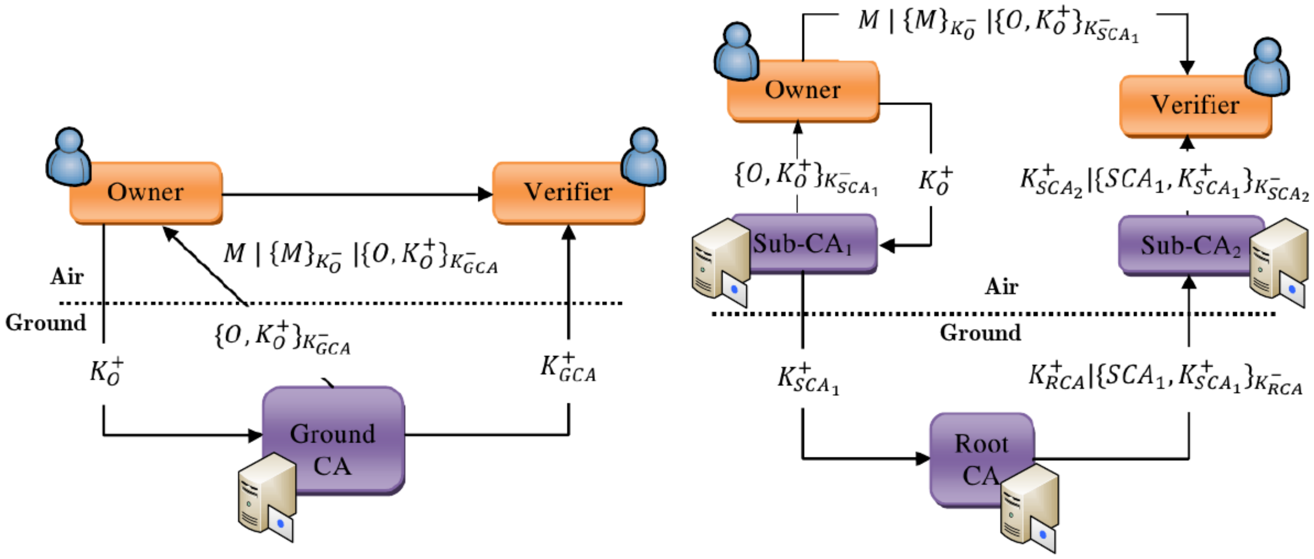
(a) Reference Model

(b) Scalable Model

Figure 5.17: Scenario 2 PKI message exchanges

5.4.2.3 Scenario 3: Onboard verifier and onboard owner

In the last scenario, the certificate verifier and owner are both onboard:


Owner
Verifier
$M \mid \{M\}_{K_O^-} \mid \{O, K_O^+\}_{K_{GCA}^-}$
Air
Ground
$\{O, K_O^+\}_{K_{GCA}^-}$
K_O^+
K_{GCA}^+
Ground CA
$M \mid \{M\}_{K_O^-} \mid \{O, K_O^+\}_{K_{SCA_1}^-}$
Owner
Verifier
$\{O, K_O^+\}_{K_{SCA_1}^-}$
K_O^+
$K_{SCA_2}^+ \mid \{SCA_1, K_{SCA_1}^+\}_{K_{SCA_2}^-}$
Sub-CA$_1$
Sub-CA$_2$
Air
Ground
$K_{SCA_1}^+$
$K_{RCA}^+ \mid \{SCA_1, K_{SCA_1}^+\}_{K_{RCA}^-}$
Root CA

(a) Reference Model

(b) Scalable Model

Figure 5.18: Scenario 3 PKI message exchanges

If they are on the same aircraft, there is no PKI signaling overhead since the delegated CA is a trusted entity by both of them and consequently, there is no need to call ground root CAs. However, if each one is in a different aircraft, there are some messages that need to be exchanged with the ground. Intra-airline AOC information exchange can be a direct application of this specific scenario, typically in adhoc aeronautical networks (see the conclusion chapter 6 for more details).

As far as both certificate owner and verifier are onboard, the air-ground messages are limited. In the reference model, the $M|\{M\}_{K_O^-}|\{O, K_O^+\}_{K_{GCA}^-}$ message is not concerned by the air-ground exchanges compared to scenarios 1 and 2. Still, the owner certificate $\{O, K_O^+\}_{K_{GCA}^-}$ has to be generated and sent by the ground CA to the certificate owner. In the hierarchical model, the air-ground exchanges are restricted compared to the two first scenarios.

In the scalable PKI model, the only air-ground exchanges are relevant to the sub-CA mutual authentication. Indeed, the only trusted third party here is the airline root CA located on the ground. More specifically, the certificate verifier asks the sub-CA to which he is registered to send him the public key $K_{SCA_1}^+$ needed to verify the signature of the $M|\{M\}_{K_O^-}|\{O, K_O^+\}_{K_{SCA_1}^-}$ message. Consequently, the airline root CA has to send the $K_{RCA}^+|\{SCA_1, K_{SCA_1}^+\}_{K_{RCA}^-}$ of sub-CA$_1$ to the sub-CA$_2$. Note that these exchanges are quite different from the illustrative case described in section 5.3.3.1 for the simple reason that here both sub-CAs are registered to the same airline root CA, therefore there is no need for cross-certification at *level 1* of the trust hierarchy.

5.4.3 Results and Discussion

In order to assess the performances of each PKI model in the three scenarios introduced above, some assumptions have been made for the parameters described in table 5.3:

- ◇ RSA has been used as the by default asymmetric algorithm for the key pair generation and the digital signature. The signature length is $l_{sig} = 256 Bytes$ (note that l_{sig} is used for both the signature length and the public key length later in the equations);
- ◇ The average certificate size $Size_c = 1KByte$ has been based on the average X.509 certificate length. Practically, we have used the OpenSSL[18] cryptography library to generate ten X.509 certificates and deduce an average certificate size;
- ◇ The exchanged data M has not been considered since the study aim was to measure only the additional overheads induced by PKI/cryptographic mechanisms. Consequently, when the air-ground network costs are computed, the message size is discarded.

5.4.3.1 Certificate Generation and Distribution Process

According to the PKI message exchanges described in figures 5.16, 5.17, and 5.18, here are the network cost equations respectively for the reference and the scalable PKI models. These network costs are driven mainly by the number of certificates N_c (for passengers) and the number of aircraft N_f involved in each scenario.

Scenario 1 Results

Equation 5.1 shows the network cost for the PKI reference model in the first scenario:

$$N_c * (K_O^+ + \{O, K_O^+\}_{K_{GCA}^-} + M|\{M\}_{K_O^-}|\{O, K_O^+\}_{K_{GCA}^-}) \cong 2 * N_c * (l_{sig} + Size_c) \tag{5.1}$$

In this case, there are two signatures (relevant to K_O^+ and $\{M\}_{K_O^-}$) and two certificates (both relevant to $\{O, K_O^+\}_{K_{GCA}^-}$) and the exchanges are driven by the number of certificates N_c handled by GCA.

[18] http://www.openssl.org/

Equation 5.2 shows the network cost of the hierarchical PKI model in the first scenario:

$$N_f * K^+_{SCA} + N_c * (M|\{M\}_{K^-_O}|\{O, K^+_O\}_{K^-_{SCA}}) \cong N_f * l_{sig} + N_c * (l_{sig} + Size_c) \tag{5.2}$$

In our PKI model, there is one less certificate to exchange between the root CA and the sub-CAs (*i.e.* $\{O, K^+_O\}_{K^-_{SCA}}$) and the exchanges are balanced between the number of aircraft N_f and the number of certificates N_c. We extrapolated these equations using the results obtained from the aircraft and passenger statistics in section 5.4.1 to deduce the network costs illustrated in figure 5.19:

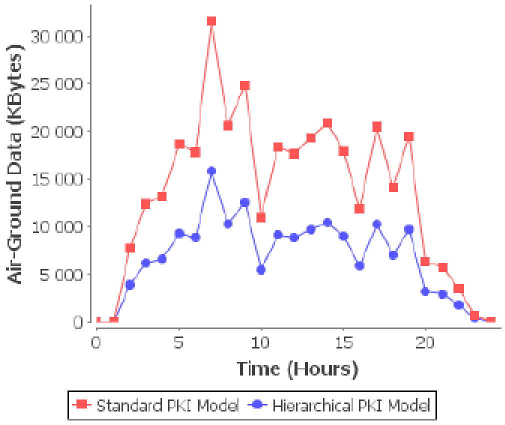


Figure 5.19: Scenario 1 - Network Costs

In this scenario, it is clear that the hierarchical PKI model is less expensive compared to the reference model. Indeed, an average difference between the two model costs has been assessed to 55%. This difference is mainly due to the fact that the network cost for the reference PKI model is affected by the number of total certificates N_c handled in the certificate generation and distribution procedure, whereas the network cost relative to our model rather involves the number of aircraft N_f. Indeed, as we have seen in the statistics deduced from the DSNA-DTI database, the number of passenger certificates is higher than the number of aircraft managed by an airline (in the case of the *Air France* airline, $N_c = 4200$ and $N_f = 38$). Moreover, an additional certificate (*i.e.* the owner certificate $\{O, K^+_O\}_{K^-_{GCA}}$) is exchanged in the reference model while there is no need for it in the hierarchical model.

Scenario 2 Results

The hierarchical model shows also better performances in the second scenario configuration. Equation 5.3 illustrates the network cost for the reference PKI model:

$$N_c * (K^+_{GCA} + M|\{M\}_{K^-_O}|\{O, K^+_O\}_{K^-_{GCA}}) \cong N_c * (2 * l_{sig} + Size_c) \tag{5.3}$$

In the reference PKI model, the exchanged messages are the owner certificate $\{O, K^+_O\}_{K^-_{GCA}}$, the key needed for the certificate verification K^+_{GCA}, and the signature $\{M\}_{K^-_O}$. As for the first scenario, the reference PKI model exchanges are driven by the number of certificates N_c.

Equation 5.4 shows the air-ground exchanges for the scalable PKI model:

$$N_f * K^+_{RCA} + N_c * (M|\{M\}_{K^-_O}|\{O, K^+_O\}_{K^-_{SCA}}) \cong N_f * l_{sig} + N_c * (l_{sig} + Size_c) \tag{5.4}$$

As for the first scenario, the network cost relevant to the hierarchical PKI model is balanced between N_c and N_f: N_f is used for the exchanges between RCA and SCA (*i.e.* K^+_{RCA}) whereas N_c is used for the exchanges relevant to the verifier (*i.e.* the message signature $\{M\}_{K^-_O}$ and the owner certificate $\{O, K^+_O\}_{K^-_{SCA}}$).

Figure 5.20 shows the relevant model performances after the extrapolation with the aircraft/passenger statistic data:

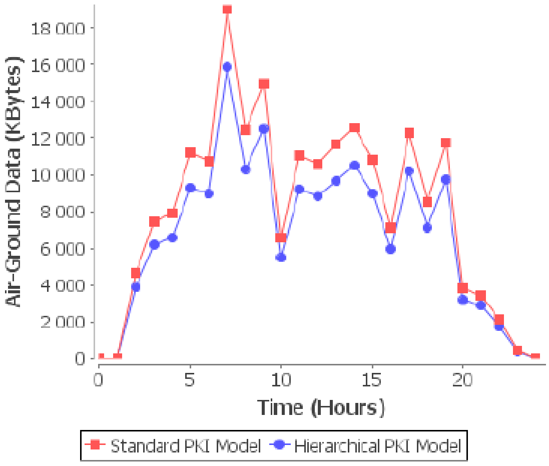


Figure 5.20: Scenario 2 - Network Costs

The difference between the standard and the hierarchical PKI model performances in the second scenario is less important compared to the first scenario (here the average gain is assessed to 20%). Indeed, in the second scenario, the owner is ground located, then when it asks for its certificate $\{O, K^+_O\}_{K^-_{GCA}}$, the exchange is performed on the ground (meaning there is no certificate to send to the aircraft). Though, a difference is still noticeable due to the higher impact of N_c in the reference model compared to the hierarchical one.

Scenario 3 Results

For the last scenario, the network cost relevant to the reference PKI model is depicted in equation 5.5:

$$\begin{gathered} N_c * (N_f - 1) * (K^+_O + \{O, K^+_O\}_{K^-_{GCA}} + K^+_{GCA} + M|\{M\}_{K^-_O}|\{O, K^+_O\}_{K^-_{GCA}}) \cong \\ N_c * (N_f - 1) * (3 * l_{sig} + 2 * Size_c) \end{gathered} \tag{5.5}$$

In this case, all the air-ground messages (*i.e.* two keys K^+_O and K^+_{GCA}, the certificate $\{O, K^+_O\}_{K^-_{GCA}}$ twice, and the signature $\{M\}_{K^-_O}$) are sent for each aircraft and each passenger (*i.e.* $N_c * (N_f - 1)$).

Equation 5.6 illustrates the air-ground exchanges relevant to the scalable PKI model:

$$\begin{gathered} (N_f - 1) * (K^+_{SCA_1} + K^+_{RCA}|\{SCA_1, K^+_{SCA_1}\}_{K^-_{RCA}})+ \\ N_c * (M|\{M\}_{K^-_O}|\{O, K^+_O\}_{K^-_{SCA_1}}) \cong \\ (N_f - 1) * (2 * l_{sig} + Size_c) + N_c * (l_{sig} + Size_c) \end{gathered} \tag{5.6}$$

Unlike the reference PKI model, the network cost for the scalable PKI model is balanced between the number of aircraft and the number of passengers: N_f is used for the exchanges between the airline root CA, SCA_1 and SCA_2, whereas N_c is used for the exchanged between the certificate owner and the certificate verifier.

As we can see in figure 5.21, the hierarchical model network cost remains always below the reference model network cost. A logarithmic scale has been used to effectively perceive the difference between these two models. As expected, the scalable PKI model shows better performances mainly because of the onboard location of the certificate owner and verifier: the average difference between the network costs is evaluated up to 92 % in this case. Our model is assisted by the lower number of certificates that the ground airline CA has to manage: the deployment of the sub-CA minimizes the air-ground exchanges for the PKI credentials of end entities. In the standard model, all these credentials have to be requested to the ground-located CA; and thus the air-ground amount of data is naturally more important.

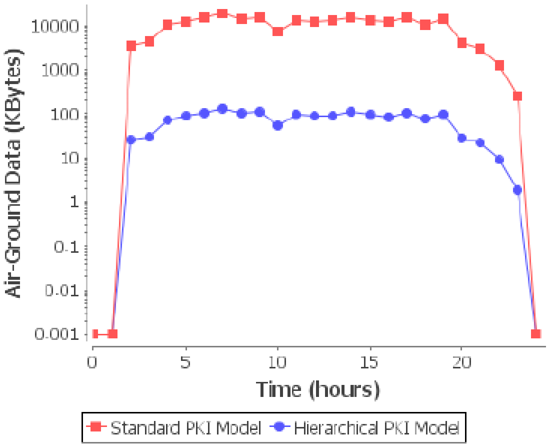


Figure 5.21: Scenario 3 - Network Costs

To resume, the hierarchical PKI model shows better performances compared to the standard reference PKI model in all the location-based scenarios used in the simulations. Our model takes advantage of several factors like a lower number of certificates to handle by the root ground-located CA, less air-ground PKI-related exchanges, and locally trusted sub-CAs which avoids additional transactions between end entities and the airline root CA.

As a last step of this performance study, we aim to find out how both PKI models behave with the certificate revocation process using either CRLs or the OCSP protocol.

5.4.3.2 Certificate Revocation Process

As for the certificate generation and distribution process, some assumptions have been made for the certificate revocation process:

- ◇ A passenger still holds one certificate, then the total number of certificates N_c is equal to the total number of passengers (per hour);
- ◇ RSA is always used as the asymmetric algorithm for the keys pairs and the digital signature ($l_{sig} = 256$ Bytes);
- ◇ The revocation information update periodicity is equal to one day: $N_U = 24$ (hours);

- ◇ The certificate serial number length is $l_{sn} = 20$ bits;
- ◇ The number of certificate check status messages generated per day N_R depends on the total number of certificates N_c and the percentage of daily revoked certificates ($R_c = 10\%$);
- ◇ The average number of certificates $N_{c,CA}$ handled by CA depends on the considered PKI model. In the reference model, its is equal to the total number of passengers per airline $N_{c,RCA} = N_c$ because the root airline CA is the only trusted authority. In the hierarchical PKI model, we take the average number of passengers for each sub-CA;
- ◇ The signature and verification times C_{sig} and C_V have been evaluated using the OpenSSL cryptographic library on a Pentium Core i7 CPU@2.67 Ghz. 50 simulations have been processed and an average has been calculated ($C_{sig} = 420$ msec and $C_V = 0.113$ msec).

Table 5.5 shows the formula of each network and processing cost per revocation mechanisms:

Table 5.5: Network and Processing Costs for the Certificate Revocation Procedure

Cost	**CRL**	**OCSP**
C_U^{net}	$N_U * (l_{sig} + N_R * t_c * l_{sn})$	0
$C_{U,CA}^{CPU}$	$N_U * C_{sig}$	0
$C_{U,CMSE}^{CPU}$	$N_U * C_V$	0
C_R^{net}	$N_R * (l_{sig} + N_{c,CA} * R_c * t_c * l_{sn})$	$N_R * l_{sig}$
$C_{R,CA}^{CPU}$	0	0
$C_{R,CMSE}^{CPU}$	0	$N_R * C_{sig}$
$C_{R,V}^{CPU}$	$N_R * C_V$	$N_R * C_V$

C_U^{net}, $C_{U,CA}^{CPU}$, and $C_{U,CMSE}^{CPU}$ are the network and processing costs relevant to the certificate revocation information update. As we can see, there is no update costs for the OCSP protocol because it works by co-locating the CMSE and the CA. Note that the CMSE is different depending on the used certificate revocation schemes. In the case of CRL-based approaches, the CMSE is the certificate repository mentioned in figure 5.1. In the OSCP scheme, the OSCP responder plays the role of the CMSE. The certificate revocation information update network cost relevant to the CRL-based scheme depends on the total number of revocation update messages N_U, and the size of the CRL sent between the CA and the CMSE. This cost is determined as the sum of the CRL signature l_{sig}, and the update network cost associated to each revoked certificate in the CRL. The later is equal to the product of the certificate serial number length l_{sn}, the number of certificate check status messages N_R, and the time for a revoked certificate being kept in a CRL t_c (*i.e.* equal to its lifetime validity).

The computation update cost at CA (resulting from the certificate revocation status update by the CRL) is equal to the product of the number of revocation information update messages N_U and the cost of signature C_{sig} because a new signature has to be performed every time for each new CRL. The network update cost at the CRL CMSE depends on N_U and the cost of signature verification: indeed after receiving the CRL, the CMSE has to check if the signature binded to the CRL is valid or not.

C_R^{net}, $C_{R,CA}^{CPU}$, $C_{R,CMSE}^{CPU}$, and $C_{R,V}^{CPU}$ are network and processing costs relevant to the certificate validity checking query. The revocation network cost between the verifier and the OCSP CMSE depends on the length of the signature l_{sig} (computed by the responder) and the number of revocation status messages N_R to send per day. For the CRL-based revocation scheme, the revocation network cost between the verifier and the CMSE depends on the number of revocation information query messages N_R, and the size of the CRL exchanged between the verifier and the CMSE. This cost is determined as the sum of the CRL signature l_{sig}, and the network cost associated to each revoked certificate in the CRL. The later is equal to the product of the certificate serial number length l_{sn}, the average number of certificates handled by one CA $N_{c,CA}$, the percentage of revoked certificates R_c and the time for a revoked certificate being kept in a CRL t_c.

The query computation costs $C_{R,CA}^{CPU}$ and $C_{R,CMSE}^{CPU}$ are null for CRLs because there is no need to make any kind of calculation whereas the OCSP responder has to sign every message before sending the response (see section 5.2.3.1). Thus it is equal to the product of the number of revocation status messages N_R on the signature cost C_{sig}. Note that there is no cost on the CA when it comes to OCSP because the CMSE (*i.e.* the OCSP responder) is responsible for the query responses. The cost of verification at the verifier is the same for both revocation schemes and is evaluated as the product of the number of revocation status messages N_R and the verification cost C_V.

Figure 5.22 illustrates the network costs C_U^{net} relevant to the update of certificate revocation information using a CRL-based revocation scheme. The OCSP approach is not presented because the OCSP responder is co-located with the CA, meaning the requested network capacity between the CA and the CMSE is null. The difference between the model network costs is not really significant as far as CRLs are heavy in both cases.

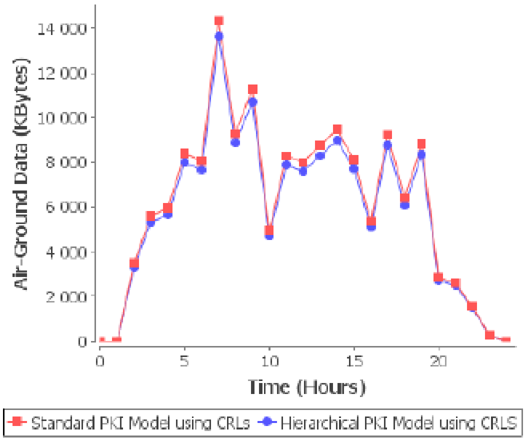


Figure 5.22: Revocation Update Network Costs between the CA and CMSE

Figure 5.23 shows the performances of both PKI models for the revocation request procedure. The benefits of the scalable PKI model are visible when the comparison is done for the revocation request messages: the standard model is clearly disadvantaged when it comes to the CRL approach because of the total number of certificates handled by the airline ground CA. The OCSP protocol has the same performances in both PKI models since the OCSP CMSE is co-located with the ground root CA and the percentage of revoked certificates N_R is the same in both cases. Still, it induces a lower revocation request network cost compared to the CRL-based approach.

Table 5.6 shows the numerical values for update and revocation query computation costs. Thes values are calculated using the formulas of table 5.5 and the input assumptions presented at the beginning of section 5.4.3.2. As expected, OCSP has the best computation performances: almost all computation costs related either to the revocation update (*i.e.* $C_{U,CA}^{CPU}$ and $C_{U,CMSE}^{CPU}$) or query messages (*i.e.* $C_{R,CA}^{CPU}$) are null. However, OCSP is disadvantaged due to the cost $C_{R,CMSE}^{CPU}$ related to the signature of OCSP responses at the CMSE:

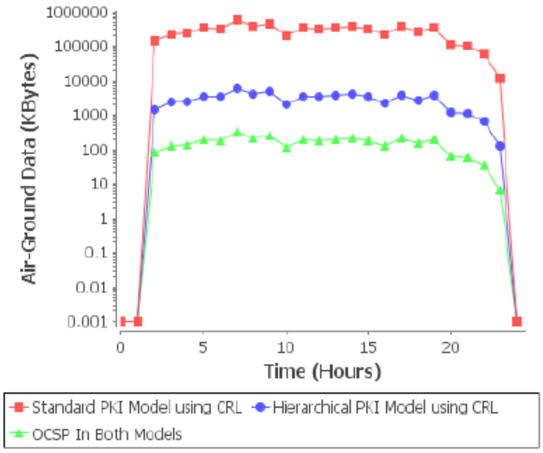


Figure 5.23: Revocation Request Network Costs between CA and Verifiers

Table 5.6: Network and Processing Costs for the Certificate Revocation Procedure

Cost (sec)	CRL	OCSP
$C^{CPU}_{U,CA}$	3.63E+004	0
$C^{CPU}_{U,CMSE}$	9.78	0
$C^{CPU}_{R,CA}$	0	0
$C^{CPU}_{R,CMSE}$	0	1.76E+002
$C^{CPU}_{R,V}$	4.75E-002	4.75E-002

Finally, we can resume the performances of both certificate revocation schemes as the following: each scheme has its own positive and negative points as we make it clear above: the CRL revocation method has many advantages such as its simplicity, an important amount of information binded to the revocation list, and a reduced risk again DoS attacks. However, as shown in the experiments, the big size of the CRLs is a major issue since the requested network capacity for updating and checking the status of a certificate is extremely high. Besides, in order to keep the revocation information as fresh as possible, every CRL embodies the next update date of the revocation information 5.5: since all the verifiers are going to send CRL requests at the same time to retrieve the new list, the network might be overloaded.

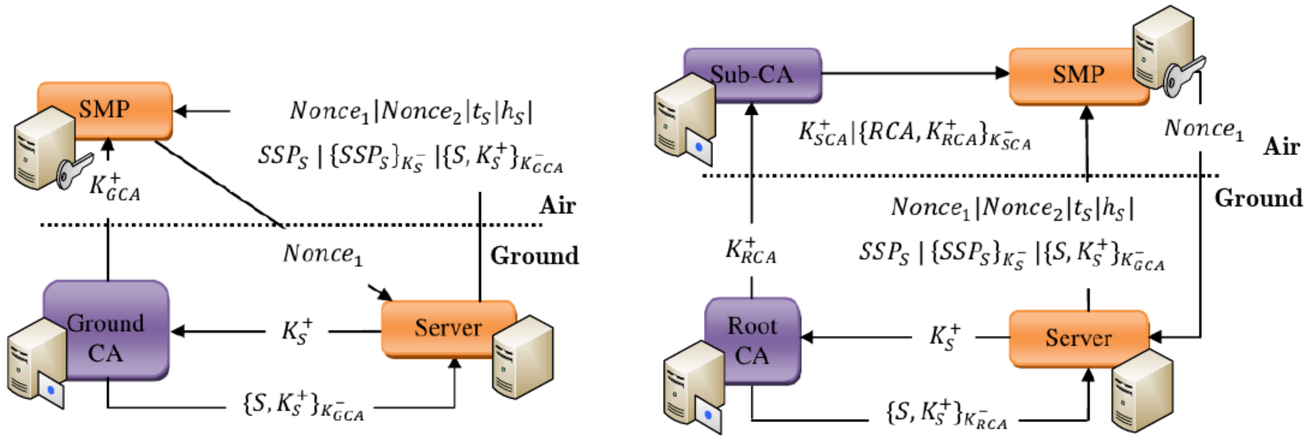
The OCSP is likely to be attacked since it deploys an online server, but in terms of induced overhead in a sensitive context where the air-ground network resources cannot be wasted, it stands as the recommended revocation method, mainly because of its performances and low signaling traffic. The main drawback of the OCSP protocol, as we already stated in 5.2.3.1 is that the CMSE (*i.e.* the OCSP responder) must digitally sign each response to provide message integrity and sender authentication. The computational load could be then very high, which may limits the scalability of the scheme, but this is optimized when OCSP is combined with the PKI scalable model introduced in this chapter.

In the previous chapter, we introduced the SNSSP negotiation protocol as a component of the SecMan framework. The protocol uses many cryptographic credentials (certificates, keys, nonces, hashes, etc) in order to establish an authenticated set of supported security mechanisms at both air and ground sides. The PKI model presented in this chapter allows us to efficiently distribute these security primitives before the establishment of the secure communications.

5.4.3.3 SNSSP Study Case

As an extension to the performance study previously discussed, we provide here the same comparison between the standard PKI model and the hierarchical PKI model. The protocol exchanges are detailed in section 4.4.1.2 of chapter 4. We focus mainly on the air-ground exchanges between the SMP and a ground server (denoted S) relevant to the initiation step of the SNSSP protocol as long as the security primitives are negotiated in this initial step.

Figure 5.24 depicts the exchanged messages relative to the initial negotiation protocol phase using respectively the reference and hierarchical PKI models. The air-ground exchanges are similar to the scenario 2 presented in 5.4.2.2 except few cryptographic primitives added to the messages as required by the SNSSP protocol (*c.f.* chapter 4). Note that messages exchanged between the certificate verifier and the sub-CA in the scalable PKI model are not taken into account because these both entities are onboard the same aircraft:


SMP
$Nonce_1|Nonce_2|t_S|h_S|$
$SSP_S \mid \{SSP_S\}_{K_S^-} \mid \{S, K_S^+\}_{K_{GCA}^-}$
K_{GCA}^+
Air
Ground
$Nonce_1$
Ground CA
K_S^+
Server
$\{S, K_S^+\}_{K_{GCA}^-}$
Sub-CA
SMP
$K_{SCA}^+|\{RCA, K_{RCA}^+\}_{K_{SCA}^-}$
$Nonce_1$
Air
Ground
$Nonce_1|Nonce_2|t_S|h_S|$
$SSP_S \mid \{SSP_S\}_{K_S^-} \mid \{S, K_S^+\}_{K_{GCA}^-}$
K_{RCA}^+
Root CA
K_S^+
Server
$\{S, K_S^+\}_{K_{RCA}^-}$

(a) Standard Model

(b) Hierarchical Model

Figure 5.24: SNSSP Exchanged Messages

The certificate revocation process is not addressed in this study case. Indeed, we have already recommended the use of OCSP as a primary revocation scheme. Besides, as discussed in 5.4.3.2, there is no difference between the use of OCSP in both PKI models 5.23. We rather focus on the certificate generation and distribution process network costs. According to the exchanges explained in figure 5.24, the network cost for the reference PKI model is:

$$\begin{gathered} N_c * (K_{GCA}^+ + SSP_S|\{SSP_S\}_{K_S^-}|\{S, K_S^+\}_{K_{GCA}^-}|t_S|h_S + 2 * Nonce_1 + Nonce_2) \cong \\ N_c * (2 * l_{sig} + Size_c + 3 * Nonce + SSP_S + h_S) \end{gathered} \tag{5.7}$$

As depicted in equation 5.3, the only message exchanged between the verifier (*i.e.* the SMP) and the ground CA is the key K_{GCA}^+. For the exchanges between the SMP and the server, three nonces (*i.e.* $Nonce_1$ twice and $Nonce_2$), the timestamp t_S, and the hash h_S are added to the initial message M (*i.e.* the SSP set of supported security protocols).

In the other hand, the network cost for the hierarchical PKI model is:

$$\begin{aligned} & N_f * K^+_{RCA} + N_c * (SSP_S|\{SSP_S\}_{K^-_S}|\{S, K^+_S\}_{K^-_{RCA}}|t_S|h_S + 2 * Nonce_1 + Nonce_2) \\ & \cong N_f * l_{sig} + N_c * (l_{sig} + Size_c + 3 * Nonce + SSP_S + h_S) \end{aligned} \tag{5.8}$$

The exchanges between the sub-CA and the SMP are not taken into account in the network cost equation because these two entities are onboard the same aircraft. As for the reference PKI model, three nonces, a timestamp, and a hash are added to the ground CA key K^+_{RCA} and the message SSP_S except that the network cost here is balanced between the number of aircraft N_f and the number of passengers N_c.

Figure 5.25 shows the network cost comparison between the two models:

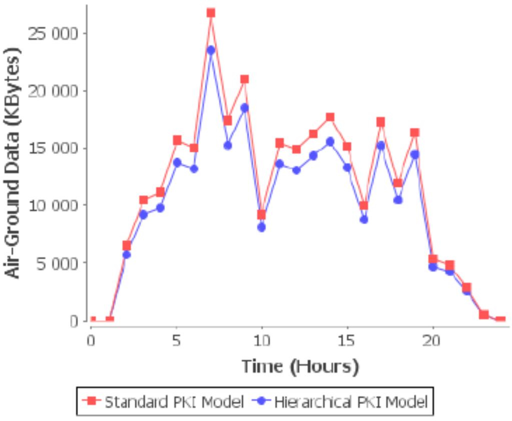


Figure 5.25: Network Costs to secure the SNSSP protocol

According to the values used in the test phase of the FAST project, the simulation inputs used in this study case are the following:

- The Supported Security Protocols (SSP) set is stored in an XML file with an average size equal to 400 Bytes;
- The hash is generated using SHA-1 and has a 20 Bytes length;
- The Nonce size is equal to 16 Bytes;
- RSA is used for the digital signature and $l_{sig} = 256$ Bytes;
- The certificate length is equal to 1 KByte.

As expected, the hierarchical PKI model is 20% less expensive than the reference model. These results are similar to those discussed in the second experimental scenario of the section 5.4.2.2. Indeed, the SMP plays here the role of the onboard certificate verifier and the ground server can be seen as the certificate ground-located owner. The main difference is that in this study case, other security primitives are added to the messages (*i.e.* nonces, timestamps, and the hashes). The data message M also is replaced by the SSP (*i.e.* the secure set of supported protocols at the server side).

5.5 Summary

In this chapter, we presented a new hierarchical PKI model for future ATM systems. First, we introduced the basic PKI concepts and main PKI-based contributions made in the scope of ATM systems. Then, we discussed the existing PKI models and introduced our performance-aware PKI model. The performance analysis relied on statistics issued from the French DSNA databases. The comparison with the PKI reference model has been performed regarding the CRL and OCSP revocation and key distribution schemes recommended by the DSWG in the ATA Spec 42 reference document. As the final results prove, it seems promising to deploy the multi-rooted hierarchical PKI using an online revocation checking status protocol like OCSP and a cross-certification level between airline ground CAs. In fact, this combination enhances considerably the network and system performances in an ATM environment. Finally, as a study case, we evaluated both PKI models using the SNSSP negotiation protocol introduced in chapter 4. Again, the hierarchical PKI model shows better performances than the standard model.

Chapter 6

Conclusions and Perspectives

This chapter summarizes the different contributions provided within the frame of this thesis and presents some issues that can be addressed as a continuity of this work.

6.1 Achieved Work

On the basis of the contributions presented in the foregoing chapters, several preliminary conclusions can be drawn.

A Propagation-based Network Security Risk Assessment Methodology

An original risk assessment approach based on risk propagation for network security has been provided in the scope of the SESAR project. The proposed methodology estimates the network risk level quantitatively based on several criteria such as the complexity of the conducted attack, or its impact within the network. Vulnerability statistics issued from the NVD public database are used within the methodology throughout the CVSS impact scores. Besides, the aeronautical network domain considerations (*i.e.* different traffic classes with different priorities) have been taken into account at an early phase of the risk assessment algorithm design. This methodology has been validated and tested through several simulations for both FAST and SESAR projects:

- in the case of the FAST project, a comparison between the nominal satellite system architecture (proposed at the very early step of the project) and the secure one has been provided. Experimental results emphasized the difference between the network risk associated to each scenario and highlighted the impact of the added security features on both the propagated and node risk values;
- in the case of the SESAR project, the main goal was to conduct a risk assessment of network security for the AeroMACS technology meant to be deployed in airport surface area communications. The experimental results highlighted several weaknesses of the AeroMACS system. A intermediate node (*i.e.* the ASN Gateway) has been identified as the point of failure of the network with a high risk level, mainly because of vulnerabilities associated to the connected IP COTS nodes (*e.g.* the DHCP server). An additional simulation scenario has been provided in order to compare the EAP and RSA authentication/authorization schemes supported by the AeroMACS privacy sublayer. Surprisingly, it has been shown that using EAP leads to a lower network risk than using RSA. Finally, several implementation, topology, and security guidance have been provided to enhance the end-to-end security of the overall AeroMACS airport system.

An adaptive Security Framework for Aeronautical Communications

The SecMan adaptive security framework has been provided in the scope of the FAST project. The proposed security solution takes into account the priority between different service classes, available resources, and the differentiated security requirements expressed for each service class. Five steps of the SecMan framework have been introduced:

1. the SNSSP negotiation protocol has been provided then formally verified using the AVISPA model checking tool. The SNSSP protocol allows the SecMan host (on the onboard side) and the server (on the ground side) to securely negotiate a set of supported security mechanisms and protocols for later use by the SecMan framework;
2. the classification of the previously negotiated set of security protocols has been provided. The classification approach uses the AHP multi-criteria hierarchical approach to estimate a score for the security level, the network, and the system costs of each security mechanism;
3. the network and system resource information have been collected then transmitted from the satellite emulator SATEM to the SecMan proxy using an adhoc client-server protocol;
4. the received network and system resource information have been processed and shared with respect to the traffic class priorities;
5. finally, the security policy selection process has been depicted according to the constraints established at the previous steps (*i.e.* security feature scores, resource available for each traffic class, security requirements for operational and non operational services, etc).

Finally, results obtained at the last integration phase of the FAST project have been discussed. Particularly, the benefits brought by the adaptive security framework have been underlined throughout several simulation scenarios.

A Scalable Public Key Infrastructure for Connected Aircraft

A performance-aware PKI for connected aircraft has been provided. The PKI model uses a cross layered multi-rooted hierarchy to handle the PKI exchanges and minimize the signaling overload. The PKI model is decomposed into three hierchical levels:

1. the first level is relevant to inter-CA communications (root CA deployed by each airline). As long as every root CA is independent of the others and has the authority on the aircraft labeled within its own domain, a cross certification scheme has been used on the ground between the root CAs;
2. the second level deals with the communications between the root airline CA and subordinate CAs onboard each aircraft. Each subordinate CA acts as a root and trusted CA within the same aircraft for all the embedded end entities. In this way, air-ground communications are minimized, only subordinate CAs have to contact ground-located airline CA;
3. the third level is relevant to communications between end entities and subordinate CAs onboard each aircraft. If a certificate needs to be retrieved or verified by an end entity, the operation is performed directly by contacting the subordinate CA inside the same aircraft. In this way, the ground root CA remains transparent to the end entities onboard the aircraft.

A performance comparison between our PKI model and a reference PKI model has been provided according to three location-based scenario (*i.e.* depending on the certificate verifier and owner position) and aircraft statistics, issued from the French DSNA-DTI database, have been used in the experiments. Besides, two certification revocation schemes have been compared, namely CRLs and OCSP, according to the ATA Spec 42 guidance and recommendations.

Experimental results showed that the OCSP protocol, despites some weaknesses, is the best choice to make for aeronautical communications when it is combined with the proposed scalable PKI model. As a study case, we turned our attention to the SNSSP protocol where the SecMan proxy and the ground server need several security materials (*i.e.* nonces, hashes, signatures) to secure the negotiation messages.

6.2 Future Work and Perspectives

The final section of this thesis opens new fields of research showing problems that remain still unsolved and which can be addressed in the foreseeable future.

6.2.1 Improving Achieved Work

In this section, we provide possible enhancements that could be made to extend the achieved work.

Enhancement of the Security Risk Assessment Methodology

Several improvements can be made in order to enhance the risk assessment methodology presented in chapter 3. Our current methodology considers the publicly known vulnerabilities stored in databases such as NVD, a step forward could be to foretell the occurrence of unknown attacks. The network security state could be predicted using IPS (Intrusion Prevention System), with stateful analysis of the protocols behaviors (based on predefined protocol states) or statistical anomaly analysis (based on network traffic normal conditions) for instance. Besides, it could be interesting to investigate how the accuracy of node correlation could be impacted using a different conditional relationship between nodes: the dependency relation model could be revisited and improved using existing approaches such as Bayesian networks, Petri or causal models. Furthermore, the network security risk assessment methodology could be adapted to node mobility: the presented model is only able to evaluate the risk level for a static network (*i.e.* deterministic number of nodes, exploitable vulnerabilities, security protections, etc).

Improving the SecMan Framework

Several improvements can be brought to the adaptive SecMan framework:

◇ *Operational and functional improvements*: the network and system information collecting process could be enhanced using an SNMP (Simple Network Management Protocol) approach. Several SNMP agents could be deployed and located on each onboard router. Information such as `ifSpeed` for the available bandwidth or `tcpCurrentEstab` for active TCP connexions can be retrieved from the MiBs (Management Information Bases) and sent to a NMS (Network Management System) which is co-located with the SMP's. The use of the third version of the SNMP protocol is also recommended as SNMP requests and responses will be protected. Another approach for the network and system information collecting process is to use the information retrieved by the satellite terminal using the DAMA protocol. In this frame, the cross-layered approach described in chapter 4 could be extended and a new reserved channel ID could be dedicated to the security signalization and control messages (PKI messages, cryptographic credentials, etc) and associated to one or several QoS DAMA capacity requests. In this way, the adaptation process will be made on-demand at the satellite terminal level beforehand the optimal security policy selection;

◇ *Design of highly certified embedded aeronautical networks*: the secure network architecture provided onboard the aircraft in chapter 4 should be complemented with a high-assurance system architecture for secure information sharing within the embedded network. A MILS (Multiple Independent Layers of Security) [Boettcher *et al.* 2008] approach could be used to support robust partitioning with a separation kernel for both safety-related IMA (Integrated Modular Avionics) and intermediate network nodes such as the NG router. MILS is a verifiable and secure architecture that allows to execute untrusted processes (*i.e.* having different security levels) on the same high-assurance system. In order to do this, the MILS architecture provides two kinds of separation:

1. *At the policy level*: the system is decomposed to a virtual architecture where the trusted systems components and communication channels are identified then formally proofed;
2. *At the resource sharing level*: the identified components are implemented, and shared physical resources are allocated to them.

Thus, MILS provides a high assurance security architecture that enables virtual components from different security domains to physically share network and system resources without compromising the integrity of the policy level. In critical embedded systems such as avionic systems, this prevents for instance a trusted ATS process from communicating with an untrusted APC process.

As several traffic classes with different security and safety requirements are passing through the NG router, the integration of MILS concepts into such a network device will firstly supply a logical separation between each partition dedicated to a specific class as a complement to the differentiated QoS scheme provided in chapter 4. This separation realized by the kernel allows the combination of trusted (*e.g.* ATS and AOC) and untrusted data (*e.g.* APC) on a single router. In this way, communications between processes of different security levels are strictly controlled.

Secondly, using a MILS compliant security architecture on the same hardware platform simplifies the software verification according to Common Criteria EAL's (Evaluation Assurance Level), and the safety certification requirements according to the DO-178B DAL's (Design Assurance Level). Indeed, MILS split kernel-level security functionalities into small modular components that can be evaluated using formal methods. At the end, a verifiable and secure NG router can be built from multiple, independently developed, and certified components.

Extending the Scalable PKI Performance Analysis

The performance analysis provided in chapter 5 could be extended to other PKI procedures and certificate revocation schemes:

◇ *CRLs-derived revocation schemes*: CRLs have been considered too expensive and inefficient in many ways mainly because of the high induced loads and large list sizes. Therefore, several modifications and extensions have been provided to address these issues. For instance, over-issued CRLs [Cooper 1999] allows to reduce the peak request rate: instead of having a single CRL and wait until the next update time given by this CRL, multiple CRLs are deployed such as they have overlapping validity times with different expiration date. In this way, not yet expired CRLs will be always available to end entities when a certificate revocation is needed. Another CRL extension that could be investigated is the ARLs (Authority Certification Lists) [Adams & Lloyd 1999], used to revoke CA certificates (when CA cross-certification is used). Using ARLs instead of full CRLs reduces the certificate path verification load: when CA certificate revocation occurs, a valid ARL is made available for each CA that has signed certificates in the path with the exception of self-signed trust anchors, making the size of each list significantly smaller;

- ◇ *Online revocation schemes*: in order to cope with OCSP limitations underlined in the last chapter (*i.e.* high processing loads at both responder and end entity, etc), several extensions and alternatives have been provided and could be integrated to the performance analysis. For instance, OCSP-X [Baker 1999] has been designed to provide additional options to the original OCSP (*e.g.* delegated path validation, evaluation of trust path). Another alternative is the SCVP (Simple Certificate Verification Protocol) [Freeman *et al.* 2007], which is a more general protocol compared to OCSP because it covers the entire certificate verification process (rather than the only certification revocation checking), avoiding in this way the certificate validation processing overhead induced locally at the end entities in OCSP. DCS (Data Certification Server) [Adams & Zuccherato 1998] is an another online revocation scheme which is capable of verifying the validity of digital signature, certification path, or certificate revocation status;
- ◇ *Other revocation schemes*: beside standardized online schemes and CRLs revocation schemes, early and recent suggestions have been provided as lightweight revocation schemes (*i.e.* reduced data-structures). CRSD (Certificate Revocation Status Directory) [Micali 2006], HCRS (Hierarchical Certificate Revocation Scheme) [Goyal 2004], or CRTs (Certificate Revocation Tree) [Kocher 1999] may provide interesting performances when combined with the scalable public key infrastructure proposed in chapter 5.

As a plethora of certificate revocation schemes are being developed and proposed, it will be interesting to investigate them in depth and analyze the network and computational cost of each of them. This may lead to new guidance and recommendations regarding the certificate revocation schemes for future aeronautical communications.

6.2.2 Addressing New Datalink Security Challenges

In this section, some datalink security challenges that have not been considered in this thesis are investigated. Even if the described security issues seem transversal to the ones addressed in our work, we provide several clues to reuse our contributions in those specific topics.

Aeronautical Adhoc Network Routing Security

Currently, several work are being conducted in the field of AANETs (Aeronautical Adhoc Networks). Their feasibility on both continental and transatlantic aeronautical areas has already been demonstrated [Besse *et al.* 2010, Tu & Shimamoto 2009, Medina *et al.* 2008]. AANETs are specific mobile ad hoc networks where an e-enabled aircraft acts as a self-aware node and communicates with other aircraft and ground entities. These networks can be used to provide a myriad of aeronautical or passenger applications such as Internet connection, operational services (*e.g.* black box data downloads) or more specific AOC services. As for mobile and vehicular ad hoc networks (MANETs, VANETs), these emerging network systems require specific routing protocols to cope with the aeronautical environment constraints.

In this line, several AANET routing protocols have been provided recently such as GRAA (Geographic Routing Protocol for Aircraft Ad Hoc Network) [Hyeon *et al.* 2010], AeroRP (Geographical Routing Protocol for Highly Dynamic Aeronautical Networks) [Peters *et al.* 2011], or ARPAM (Ad-hoc Routing Protocol for Aeronautical Mobile Ad-Hoc Networks) [Iordanakis & Dilintas 2007]. These protocols share the same characteristics:

- ◇ almost all of them are position-aided routing protocols (*i.e.* geographic routing protocols). This class of routing protocols seems to be an interesting options for AANETs: all modern commercial aircraft are equipped with a GPS (Global Positioning System), mainly for navigation and surveillance applications, making geographic information easy to share and to exploit by position-aided routing protocols. Moreover, this class of routing protocols offers good performances compared to proactive or reactive routing protocols: only 3D position information of one-hop node neighbors are needed, whereas routing/neighborhood tables must be maintained and packet routes shall be established beforehand in other classes of routing protocols;

◇ There is blatant lack of security in the proposed routing protocols making the AANETs defenseless against cyber attacks such as select forwarding, Sybil attacks, sinkhole or spoofing attacks. In the other hand, there is a strong need for confidentiality in inter-airline communications (*e.g.* Kerosene consumption, passenger lists, etc).

Considering these two points, we recently provided a secure geographic ah hoc routing protocol. The proposed solution is based on the integration of the GPSR (Greedy Perimeter Stateless Routing) routing protocol [Karp & Kung 2000] with the GPS-based aeronautical surveillance system ADS-B (Automatic dependent surveillance broadcast)[1]. It is worthy to underline that the proposed solution is generic in the sense it deals with several security issues common to all MANETs (*e.g.* sinkhole attacks, select forwarding attacks), while it addresses some specific AANET issues such as the confidentiality of airline data. However, the feasibility of using GPSR in the aeronautical context should be further investigated, mainly because of specific AANETs characteristics (*e.g.* aircraft radio range, MAC layer performances, aircraft connectivity, etc).

GPSR is a location-based routing protocol that uses two methods (*i.e.* greedy and perimeter forwarding mechanisms) to transmit data from source to destination. In order to avoid the beaconing scheme which increases the control packet overhead and collision probability, the ADS-B system is used to get the 3D aircraft positions and periodically broadcast a state vector (including the aircraft ICAO identifier, the 3D position, velocity, etc) to ground stations and other ADS-B IN equipped aircraft. Thus, the beaconing GPSR overhead is eliminated, which leads to better performances compared to an exclusive GPSR solution. Security is addressed at two different levels:

1. The integrity of the ADS-B geographic positions, broadcasted to the aircraft and used in the GPSR neighbor table, is provided using public key signatures and hash functions;

2. A hop-by-hop encryption is used to enhance the GPSR security and provide confidentiality for data packets between aircraft registered to different airlines through the use of distributed symmetric keys.

The security primitive exchanges needed by the security solution described above can be assimilated to the third PKI scenario (*i.e.* inter-aircraft exchanges) described in chapter 5. Hence, the ADS-B message integrity and the selective inter-airline encryption proposed here can be reinforced using the proposed scalable multi-rooted hierarchical PKI model. The aim is to optimize the distribution of keys and security primitives needed by the ADS-B/GPSR integrated solution.

LDACS Security

Under the EUROCONTROL and FAA Action Plan 17 activities, the LDACS (L-band Digital Aeronautical Communication System) [Jain *et al.* 2011] has been recently identified as the next best candidate to support future aeronautical requirements in continental areas (*e.g.* cellular communications). Two versions of the system have been provided: the FDD (Frequency Division Duplex) based LDACS-1 and

[1] http://www.faa.gov/nextgen/portfolio/trans_support_progs/adsb/

the TTD (Time Division Duplex) based LDACS-2 systems. These systems are meant to progressively replace aviation specific datalink systems such as VDL. However, both LDACS protocol stacks do not integrate security considerations as it has been the case for the AeroMACS system. This may lead to several security problems for the foreseeable LDACS usage. Enhancing the LDACS protocol stack with a security/privacy sublayer at MAC level should be then valuable for the security of datalink communications using the future LDACS system. Lessons learned from the AeroMACS experience (*i.e.* unauthenticated or unencrypted management messages, shared symmetric key groups, etc) shall be useful in order to avoid the same design mistakes and thus provide a robust security layer for the LDACS system. Besides, as long as a privacy sub-layer is added, it will be interesting to assess the network security risk related to the LDACS protocol using the methodology proposed in chapter 3 of this thesis. Hence, guidance and recommendations could be provided to improve the overall LDACS security as it has already been done for the AeroMACS system.

On Safety-Security Relationships in ATM Systems

Historically speaking, safety and security have always been considered as two distinguished and separated concepts. Nonetheless, there are strong dependencies between them, specially in high risk and critical situations within nuclear, electric power industry or ATM environments for instance. Moreover, the convergence of safety and security requirements brings new consequences and challenges which are still to be addressed. The ATM system is particularly prone to these ambiguities because it involves many entities from multiple disciplines with different purposes (communications, navigation, and surveillance). For instance, it will interesting to analyze the side effects of security features, added to preserve datalink communications from malicious attacks, on the occurrence of accidental events and highly requested availability of the operational services. Moreover, the safety-security relationship dependencies found after such an investigation could represent a new axis to improve the network security risk assessment methodology provided in this thesis: for example, a safety-security inference engine could be added to the methodology as a supplementary method to quantify the network security risk. In this case, the risk will be evaluated according to the propagation considerations mentioned in chapter 3 and the safety-security dependencies.

Appendix A

Nomenclature

All the notations used in the thesis manuscript are summarized in the following tables:

Table A.1: Chapter 3 Nomenclature

Notation	**Description**
$Risk_i$	Node risk evaluated on node i
$Risk_i^-$	Individual risk evaluated on node i
$Risk_i^+$	Propagated risk evaluated on node i
$Risk_{net}$	Network risk
$Value_i$	Value of node i
$FunctionValue_i$	Function value of node i
$ClassValue_i$	Class value of node i
t	An exploitable vulnerability
n	Total number of nodes in the network
n_i	Number of nodes connected with node i
T_i	Number of vulnerabilities exploitable on node i
S_i	Number of security features deployed to protect i
B_t	Number of information needed to exploit t
$P_t(i)$	Likelihood of occurrence of a threat exploiting t
$I_t(i)$	Impact of threat exploiting t
$P_t(i,j)$	Likelihood of propagation of a threat exploiting t
$I_t(i,j)$	Propagated impact of a threat exploiting t
f_{ij}	Number of flows detected between nodes i and j
F_{ij}	Total number of flows exchanged between i and j
$Motivation_t(i)$	Motivation of an attacker to exploit t
$TechnicalDifficulty_t(i)$	Technical difficulty level to exploit t
$\sigma(i,s)$	Scalar value as the product of SPV_i and SOV_s
SPV_i	Security protection vector for the node i
SOV_s	Security objective vector for the service s

Table A.2: Chapter 4 Nomenclature

Notation	Description
f_k	k^{th} data flow
SSP_X	set of supported security protocols by entity X
$SSP_{negotiated}$	set of negotiated supported security protocols
P_k	security policy able to secure the data flow f_k
P_k^*	optimal security policy chosen by SecMan to secure f_k
m_j	j^{th} security mechanism of a security policy P_k
a_{ij}^s	i^{th} algorithm providing security service s and supported by m_j
τ	lifetime parameter for SecMan and SATEM exchange periodicity
$\omega_{c,net}^{used}$	used network resource ratio for data flows of class c
$\omega_{c,syst}^{used}$	used system resource ratio for data flows of class c
$\omega_{c,net}^{free}$	available network resource ratio for data flows of class c
$\omega_{c,syst}^{free}$	available system resource ratio for data flows of class c
θ_c^{used}	used network resources for data flows of class c
θ_c^{free}	available network resources for data flows of class c
N_c	number of data flows associated with traffic class c
N_{total}	total number of data flows
$\%CPU$	percentage of used system resources on the SecMan proxy
R_k	security requirement vector for the data flow f_k
b_k^s	elementary security requirement for security service s associated to f_k
PSL	policy security level function
RBI	relative balance index function
$V_{sec}(P_k)$	security level score associated with P_k
$V_{sec}(m_j)$	security level score associated with m_j
$p_{k,j}$	binary indicator for the selection of m_j in P_k
$Cost_{net}(P_k)$	network cost of security polity P_k
$Cost_{syst}(P_k)$	system cost of security polity P_k
$V_{netCost}(m_j)$	network cost score of m_j
$V_{systCost}(m_j)$	system cost score of m_j

Table A.3: Chapter 5 Nomenclature

Notation	Description
K_i^+	The public key of an entity i
K_i^-	The private key of an entity i
$\{i, K_i^+\}_{K_{CA}^-}$	i certificate issued by CA
N_c	Total number of certificates
N_f	Total number of flights
$Size_c$	Average size of a certificate
t_c	Certificate validity period (in days)
t_s	SSP validity period (in days)
h_s	SSP digest using a hash function
$Nonce_i$	i^{th} randomly generated number
l_{sig}	digital signature length
l_{sn}	Certificate serial number length
C_{sig}	Signature generation time
C_v	Signature verification time
M	Exchanged data
R_c	% of revoked certificates
N_R	Frequency of certificate revocation check status messages per day
N_U	Frequency of revocation information update messages per day
$N_{c,CA}$	Certificate average number handled by CA
C_U^{Net}	Network cost to update a certificate between a CA and the CMSE
$C_{U,CA}^{CPU}$	Computation cost at CA to update a certificate
$C_{U,CMSE}^{CPU}$	Computation cost at CMSE to update a certificate
C_R^{Net}	Network cost to check a certificate between CMSE and a verifier
$C_{R,CA}^{CPU}$	Computation cost at CA to check a certificate
$C_{R,CMSE}^{CPU}$	Computation cost at CMSE to check a certificate
$C_{R,V}^{CPU}$	Computation cost at verifier to check a certificate

Appendix B

Operational and Non Operational Data Service Specifications

B.1 COCR Operational Service Traffic Specifications

Tables B.1 and B.2 provide informations on ATS and AOC data services used in the FRS as described in the COCR document. These specifications have been used to implement the cockpit source generator used in the FAST validation phase and described in chapter 2. For each service per airspace domain, we specify, the paquet size (in bytes), the quantity of packets sent per service instance, and the frequency of use in each airspace domain. The listed operational services have been filtered according to the assumptions discussed in chapter 2.

Service	**Frequency of Use per Airspace Domain**				**Quantity x**
	TMA Departure	*TMA Arrival*	*ENR*	*ORP*	**Size (bytes)**
ACL	4	4	5	2	2x93
ACM	2	2	8	6	1x88
ARMAND	0	0	1	0	1x88
C&P ACL	0	0	1	1	2x93
DLL	0	0	1	1	1x122
D-ALERT	1	1	1	1	1x1000
D-ORIS	0	0	1	1	3x93
D-OTIS	0	1	1	0	3x107
D-RVR	0	1	1	0	3x121
D-SIG	0	1	0	0	3x129
D-SIGMET	0	1	1	1	3x129
D-TAXI	0	1	0	0	1x98
FLIPINT	1	1	4	36	1x2763
PPD	1	1	1	1	1x277
SAP (Contract)	1	1	4	0	2x100
SAP (Report)	48	48	162	0	1x107

Table B.1: ATS Data Service Specifications (COCR phase 1)

B.2 COCR Operational Service Security Requirements

Tables B.3 and B.4 provide the security requirements for ATS and AOC services as defined in the COCR document. These information have been used in chapter 3 and mapped to a quantitative scale in order to estimate the network risk level in both FAST and SESAR case studies.

Service	**Frequency of Use per Airspace Domain**				**Quantity x**
	TMA Departure	*TMA Arrival*	*ENR*	*ORP*	**Size (bytes)**
AOCDLL	0	0	0	1	2x148
ENGINE	1	1	1	0	1x727
FLTPLAN	0	0	0	1	90
FLTSTAT	0	0	0	1	1x157
FREETXT	0	0	0	2	1x377
FUEL	1	1	1	2	3x127
LOADSHT	0	1	1	0	2x93
MAINTPR	0	0	0	1	4x133
MAINTRT	0	0	0	2	5x127
NOTAM	0	0	0	2	2x134
POSRPT	1	1	1	17	1x338
WXGRAPH	0	0	0	7	6x93
WXRT	8	14	14	88	1x103
WXTEXT	0	0	0	2	2x103

Table B.2: AOC Data Service Specifications (COCR phase 1)

The same security requirements have been used in chapter 4 in order to estimate the optimal security policy by the SecMan framework. Here is a brief description of the ranking scale as described in the COCR document:

(a) *None*: no need to provide security for the service;

(b) *Low*: limited impact on expected on the service;

(c) *Medium*: serious effects are expected on that service;

(d) *High*: high security levels are required for the service;

(e) *High-Severe*: severe impact on safety and business interests;

(f) *High-Catastrophic*: catastrophic effect on flight safety, regularity (ATS) or business interests(AOC) are expected.

B.3 Non Operational Service Security Requirements

Table B.5 provides the security requirements for non operational services defined in the FAST project. These information have been collected in the scope of WP3 using security questionnaires distributed to each partner in the project:

Service	**Confidentiality**	**Integrity**	**Availability**	**Authentication**
Telemedicine	Low	Medium	High-Severe	High
Video Surveillance	Low	High	High-Severe	High
Passenger Traffic	Low	Medium	Medium	High-Severe

Table B.5: Information Security Requirements for non Operational Services

Service	Confidentiality	Integrity	Availability
AOCDLL	None	High-Severe	High
CABINLOG	Low	Low	Low
ENGINE	Low	Medium	Medium
FLTLOG	Medium	Low	Low
FLTPLAN	Low	High-Severe	High
FLTSTAT	Medium	Low	Low
FREETXT	Medium	Low	Low
FUEL	Low	Low	Low
GATES	Low	Low	Low
LOADSHT	Medium	High-Severe	High
MAINTPR	Medium	Medium	Low
MAINTRT	Medium	Medium	Low
NOTAM	None	Medium	Medium
OOOI	Low	Low	Low
POSRPT	Low	Medium	Medium
SWLOAD	Low	Low	Low
TECHLOG	Medium	Medium	Medium
UPLIB	Medium	High-Severe	Medium
WXGRAPH	Low	Medium	Medium
WXRT	None	Medium	Medium
WXTEXT	Low	Medium	Low

Table B.3: Information Security Requirements for AOC Services

Service	Confidentiality	Integrity	Availability
ACL	Low	High-Severe	High-Severe
ACM	None	High-Severe	High-Severe
A-EXEC	Low	High-Catastrophic	High-Catastrophic
AIRSEP	Low	High-Severe	High-Severe
AIRSEP SURV	Low	High-Severe	High-Severe
AMC	None	Low	Medium
ARMAND	Low	Low	Low
C&P ACL	Low	High-Severe	High-Severe
C&P SURV	Low	High-Severe	Medium
COTRAC	Low	High-Severe	High-Severe
D-ALERT	Medium	High-Severe	High-Severe
D-ATIS	None	High-Severe	Medium
DCL	None	High-Severe	High-Severe
DFLUP	None	Medium	Low
DLL	Nonce	High-Severe	High-Severe
D-ORIS	None	Medium	Low
D-OTIS	None	High-Severe	Medium
D-RVR	None	High-Severe	Low
DSC	Low	High-Severe	Medium
D-SIG	None	Medium	Low
D-SIGMET	None	High-Severe	Medium
D-TAXI	Low	High-Severe	Medium
DYNAV	Low	High-Severe	Medium
FLIPCY	Low	High-Severe	Medium
FLIPINT	Low	High-Severe	High-Severe
ITP ACL	Low	High-Severe	High-Severe
ITP SURV	Low	High-Severe	Medium
M&S ACL	Low	High-Severe	High-Severe
M&S SURV	Low	High-Severe	Medium
PAIRAPP ACL	Low	High-Severe	High-Severe
PAIRAPP SURV	Low	High-Severe	Medium
PPD	Low	Low	Low
SAP	Low	Medium	Low
SURV	Low	High-Severe	Medium
TIS-B	Low	High-Severe	Medium
URCO	None	Medium	Medium
WAKE	None	High-Severe	High-Severe

Table B.4: Information Security Requirements for ATS Services

Appendix C

Software and Simulations Tools

C.1 Opnet Modeler

Opnet Modeler is the main tool provided by OPNET (Optimum Network Engineering Tool) Technologies Inc., a leading provider of solutions for application and network performance management. Opnet Modeler is a well-known network simulation tool available for academic and industrial R&D teams, it allows analyzing and designing communication networks, devices, protocols, and IT applications. The tool is based on an object oriented design approach and discrete event simulations. It uses an intuitive graphical user interface and provides a broad suite of protocols and technologies (TCP, IP, MPLS, and many others). Opnet Modeler relies on a hierarchical modeling environment as shown in figure C.1[1]:

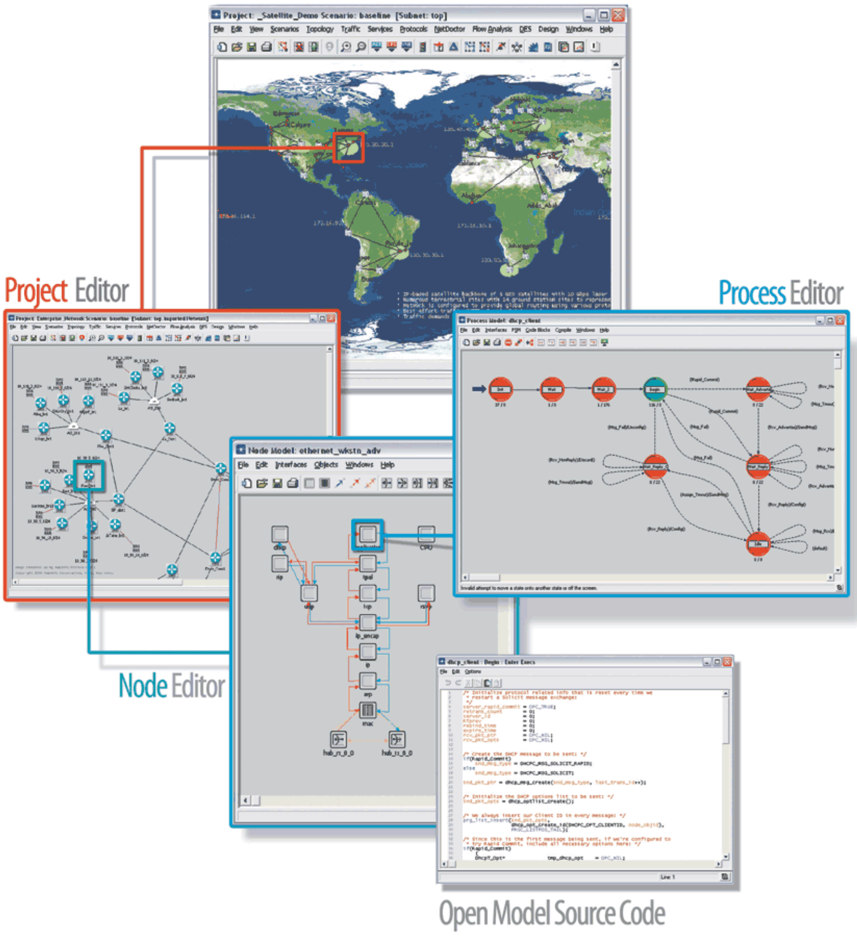


Figure C.1: Opnet Hierarchical Modeling Environment

[1] http://www.opnet.com/solutions/network_rd/modeler.html

The hierarchy levels shown in figure C.1 are:

- ◇ *Project editor level*: corresponds to the top level of the Opnet hierarchical modeling environment. It allows the users to establish a network topology by interconnecting devices using a set of nodes already afforded by the tool (*e.g.* routers, servers, clients, proxies, etc). Every node has it own set of parameters to configure (*e.g.* used protocol, network address, physical address, etc);
- ◇ *Node editor level*: is the second level of the hierarchy, it allows the users to design the node architecture using functional block modules and data flows. A Module is used to represent for instance an application, a specific protocol, buffers, control managers, etc. Modules communicate between each others using information flows (*e.g.* data/control packets, buffer sizes, transmission delay, etc);
- ◇ *Process editor level*: the final level of the hierarchy is used to design the functional block modules (upper level) using finite state machines. Each state corresponds to a given process activity and is specified using the C/C++ code language. The transition between state machine depends on execution conditions (*e.g.* interruption event).

Figure C.2 shows the Opnet model developed in order to simulate the operational ATS traffic in the ORP airspace domain, according to the COCR reference document:

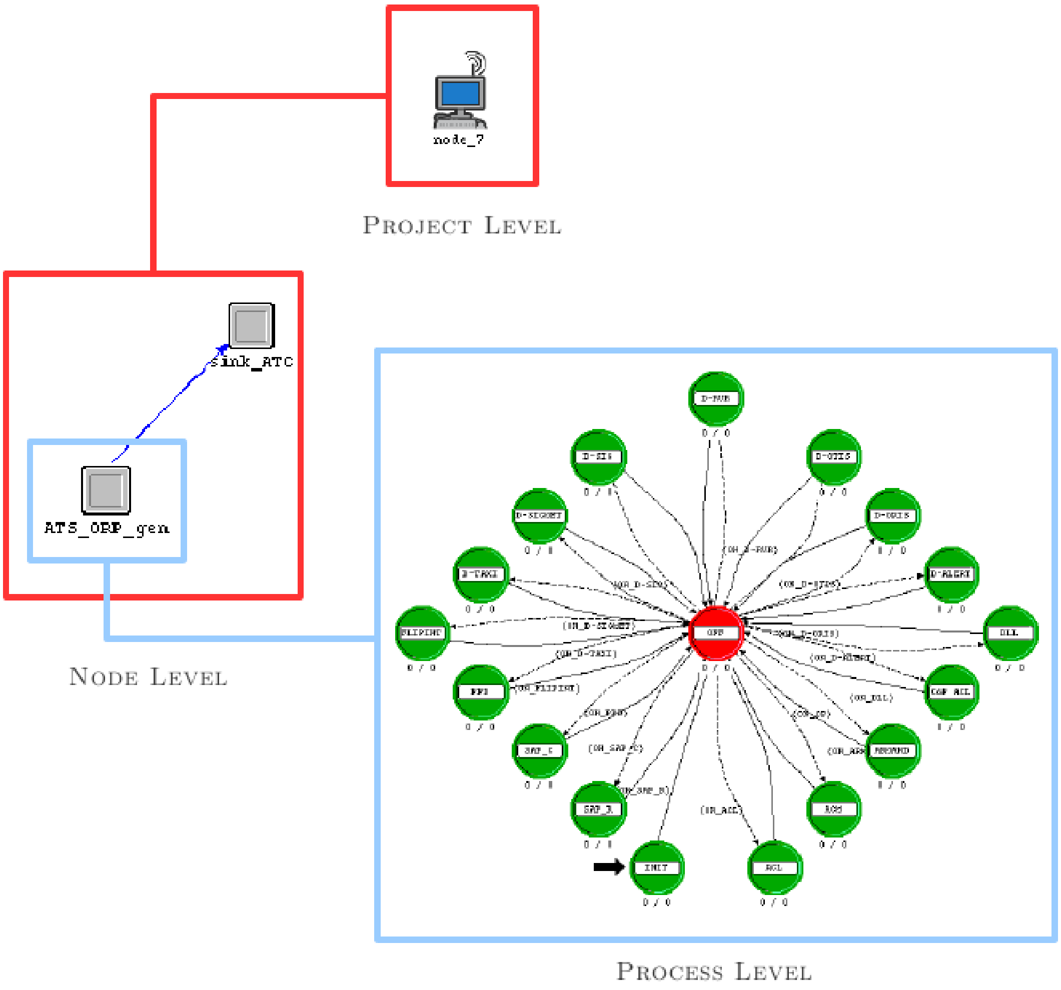


Figure C.2: ATS Traffic Model in the ORP Airspace Domain

At the project level, a single network node is created. At the node level, two functional block modules are created: the `ATS_ORP_gen` module is in charge of the operational traffic generation according to the COCR inputs, and the `sink_ATC` module is responsable of collecting statistics and destroying packets that are no longer needed in order to free the memory allocated to them. The blue arrow allows the packet streams to be transfered from the `ATS_ORP_gen` traffic generator module to the `sink_ATC` process module. At the lowest level of the Opnet environment hierarchy, the `ATS_ORP_gen` module is described using the ON-OFF stochastic processes described in chapter 2. The model shown in figure C.2 has been duplicated and adapted for each airspace domain (*i.e.* ENR and ORP) and each operational traffic class (*i.e.* ATS and AOC).

C.2 MARIONNET Emulation Tool

MARIONNET is a virtual network tool allowing to emulate a physical network with many network devices (*e.g.* terminal hosts, servers, routers, hubs, switches, ethernet cables, gateway, cloud, etc). When there are too many network nodes in the physical network, MARIONNET can be used to emulate a part (or even all) of these nodes in order to make easier the configuration and the test phases. Figure C.3 shows the graphical interface provided by the MARIONNET environment:

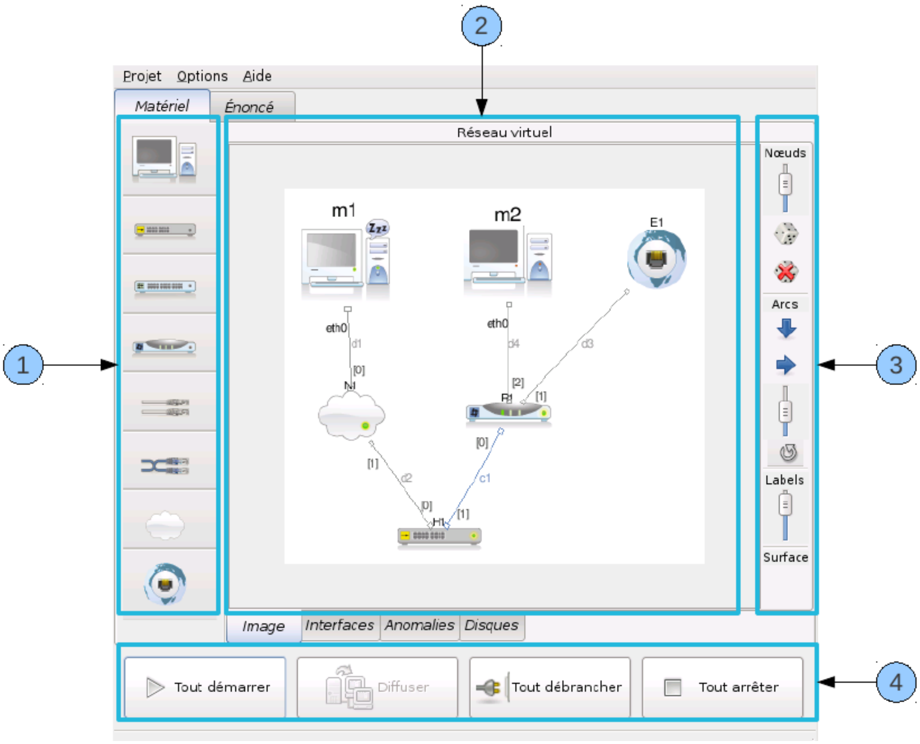


Figure C.3: Marionnet Graphical Main Interface

Palette 1 of figure C.3 shows the emulated network components which can be configured at the very fine level of details, depending on the nature of the node:

◇ *computer*: for each host terminal, the user can specify the kernel/OS Linux distribution running on that node, the amount of RAM, and the number of Ethernet interfaces. For each computer nodes, the user can launch and configure a terminal like on a physical machine (*e.g.* routing table, firwall rules, etc);

◇ *hubs and switches*: for each hub or switch, the user can configure the number of ports, and supported protocols. A LED grid is associated to every hub or switch representing the port activities as for physical nodes and allowing to observe the network traffic;

- *IP routers*: replicate the behavior of CISCO routers. The user can also configure the number of supported ports and the routing firmware (*e.g.* Quagga);

- *cables*: both straight, crossover, and serial cables are supported for interconnections between all types of nodes;

- *clouds*: represent a random IP network on which the user has no control (random delay, random PER, etc);

- *sockets*: represent a network gateway which interconnect the emulated network to the physical host network (routing traffic from/to emulated nodes from/to physical nodes, provide Internet connection for virtual nodes, etc).

Part 2 of figure C.3 shows the virtual network with all the interconnected nodes. For each network components, the user has access to an X-term terminal in order to configure and control the node as he does for a physical node. Part 3 allows to control and tweak the virtual network (node orientation, label edition, arcs, nodes and network surface size, etc). Finally, part 4 is a control panel in order to start, pause, or stop the network emulation. Marionnet has been used for chapter 3 simulations (a part of the physical nodes has been emulated).

C.3 SATEM Satellite Emulator

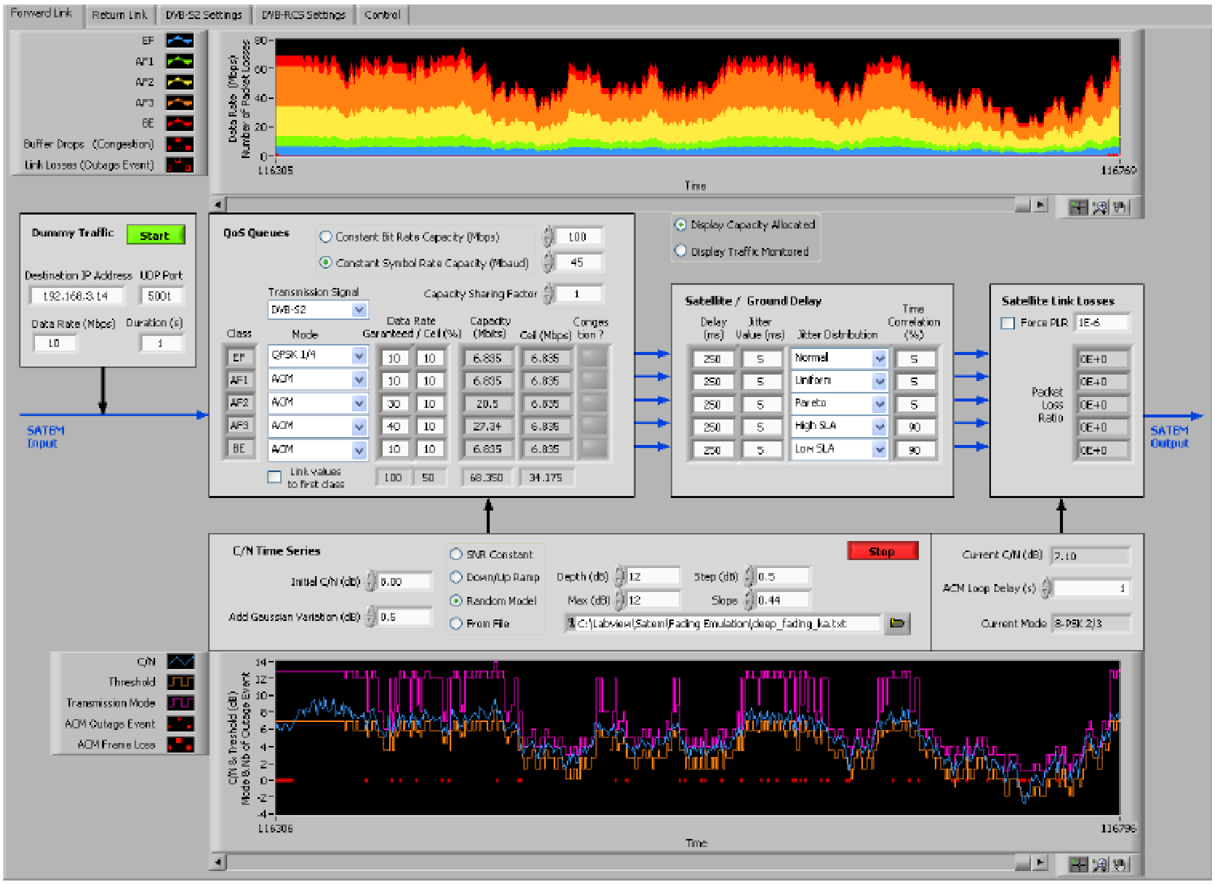
SATEM is an emulator tool for IP-based satellite link. It allows the user to replicate the behavior of a real satellite network (both uplink and downlink) and to configure a differentiated QoS, capacity variation (*i.e.* link fading), bandwidth sharing, packet error rates, delay and jitter. The system has been implemented on two different supports:

1. *on a router*: the product is thereby transportable and easy to deploy with a low cost. However, this kind of implementation fits with light networks where system performances are not the top priority;

2. *on a compact computer*: the product is more flexible with higher RAM and CPU resources. This implementation allows the user to emulate several satellite links on several interfaces, and to handle heavy applications and complex traffic management. This is the implementation used in the FAST integration and testbed phase.

The system has also an built-in traffic generator able to simulate a satellite traffic independently of end entities. The SATEM satellite emulator is generally deployed between the satellite access gateway and the satellite terminal using Ethernet cables.

The satellite link emulation and statistic edition are managed using a LabView GUI as shown in figure C.4:



Dummy Traffic Start

Destination IP Address: 192.168.3.14
UDP Port: 5001
Data Rate (Mbps): 10
Duration (s): 1

QoS Queues

○ Constant Bit Rate Capacity (Mbps) 100
● Constant Symbol Rate Capacity (Mbaud) 45

Transmission Signal: DVB-S2
Capacity Sharing Factor: 1

Class	Mode	Data Rate Garanteed / Ceil (%)		Capacity (Mbits)	Ceil (Mbps)	Congestion ?
EF	QPSK 1/4	10	10	6.835	6.835	
AF1	ACM	10	10	6.835	6.835	
AF2	ACM	30	10	20.5	6.835	
AF3	ACM	40	10	27.34	6.835	
BE	ACM	10	10	6.835	6.835	
	□ Link values to first class	100	50	68.350	34.175	

● Display Capacity Allocated
○ Display Traffic Monitored

Satellite / Ground Delay

Delay (ms)	Jitter Value (ms)	Jitter Distribution	Time Correlation (%)
250	5	Normal	5
250	5	Uniform	5
250	5	Pareto	5
250	5	High SLA	90
250	5	Low SLA	90

Satellite Link Losses

□ Force PLR 1E-6

Packet Loss Ratio: 0E+0, 0E+0, 0E+0, 0E+0, 0E+0

C/N Time Series

Initial C/N (dB): 8.00
Add Gaussian Variation (dB): 0.5

○ SNR Constant
○ Down/Up Ramp
● Random Model
○ From File

Depth (dB): 12
Max (dB): 12
Step (dB): 0.5
Slope: 0.44
C:\Labview\Satem\Fading Emulation\deep_fading_ka.txt

Stop

Current C/N (dB): 7.10
ACM Loop Delay (s): 1
Current Mode: 8-PSK 2/3

Figure C.4: SATEM LabView GUI

Several functions are implemented on SATEM:

- generation of a CBR traffic (SATEM input);
- generation of the S/N ratio using four methods: constant, linear, random, or based on a configuration file;
- differentiated QoS using five FIFOs (EF, AF1, AF2, AF3, and BE);
- a delay can be applied on each FIFO which corresponds to the propagation delay of a satellite link;
- the jitter can also be managed on each FIFO (return link) using one of these statistical distribution laws: uniform, PARETO, BoD (with high or low priority);
- generation of a random PER.

All these functions allow the user to obtain a realistic behavior as he could expect from an IP satellite link. SATEM has been used in the FAST project to test and validate the adaptive capabilities of the SecMan framework using specifically the variation of the S/N ratio and satellite link fading.

C.4 AVISPA/SPAN Security Protocol Checking Tool

AVIPSA is a security protocol analyzer that uses the HLPSL to specify which security goal has to be achieved by the security protocol. The HLPSL language allows the user to specify the data structures and intruder capabilities using a control flow pattern-oriented specification.

Figure C.5 illustrates the AVISPA system architecture:

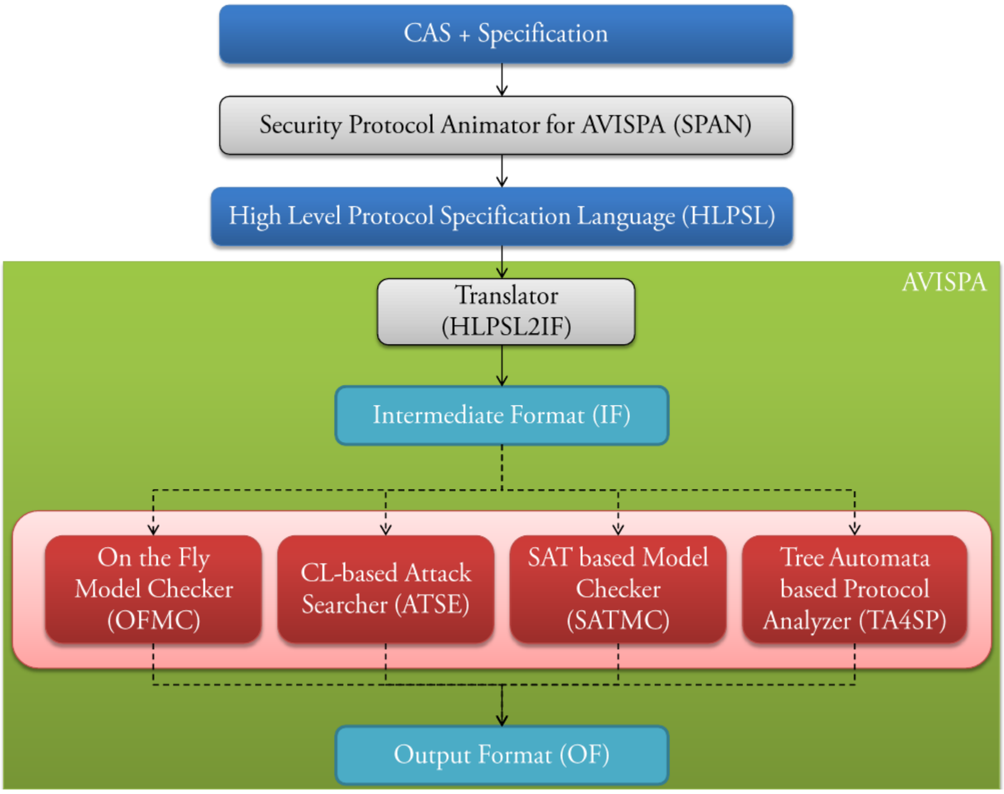
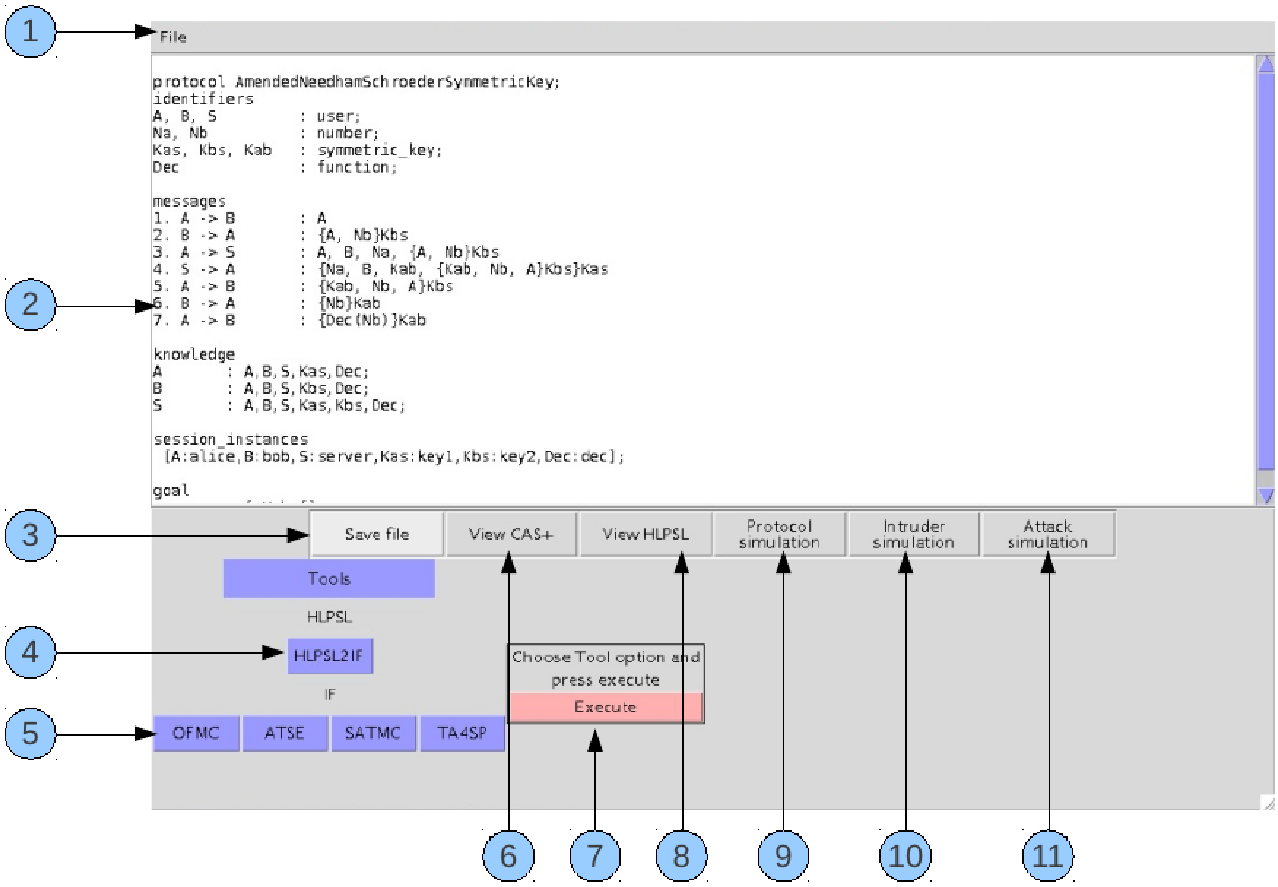

CAS + Specification
Security Protocol Animator for AVISPA (SPAN)
High Level Protocol Specification Language (HLPSL)
AVISPA
Translator (HLPSL2IF)
Intermediate Format (IF)
On the Fly Model Checker (OFMC)
CL-based Attack Searcher (ATSE)
SAT based Model Checker (SATMC)
Tree Automata based Protocol Analyzer (TA4SP)
Output Format (OF)

Figure C.5: AVISPA System Architecture

With the introduction of the freely available SPAN GUI tool for AVISPA, a new specification language has been made available, namely CAS+. The CAS+ language has been derived from the CASRUL language [Jacquemard *et al.* 2000] and leads to protocol specifications as precise as HLPSL. The role of SPAN is to symbolically animate the protocol in order to have a better look and understanding of its specifications. Figure C.6 shows the main graphical interface of the SPAN tool. MSCs (Message Sequence Charts) [Harel & Thiagarajan 2003] are animated and represent one or several sessions of the protocol according to the information given by the protocol specification. It is possible to check the values of variables sent by each principal user at every moment. SPAN includes the same features as those in the web graphical interface developed beforehand for AVISPA: edition of protocol specifications in both CAS+ and HLPSL syntax (Labels 2, 6, and 8), selection/configuration of one or several backends (Labels 4, 5, and 7), saving and loading protocol specifications (Labels 1 and 3), and adds the following new features:

- *protocol simulation*: allows the user to build a step by step MSC-based specification using trusted agents (Label 9). At every moment, SPAN allows the user to choose between all possible transitions between principals: this approach helps the protocol designers to resolve interactively all the choices that may arise during the construction of MSC (*e.g.* non-deterministic protocols);
- *intruder simulation*: allows the user to build a step by step MSC-based specification using trusted agents in the presence of an active/passive intruder (Label 10). After each sent message, SPAN shows how the intruder gained knowledge thanks to that message, and proposes to construct and send a malicious message to legitimate users;
- *attack simulation*: automatically builds an MSC-based attack trace from the outputs of OFMC and CL-Atse backends (Label 11).



```
protocol AmendedNeedhamSchroederSymmetricKey;
identifiers
A, B, S           : user;
Na, Nb            : number;
Kas, Kbs, Kab     : symmetric_key;
Dec               : function;

messages
1. A -> B         : A
2. B -> A         : {A, Nb}Kbs
3. A -> S         : A, B, Na, {A, Nb}Kbs
4. S -> A         : {Na, B, Kab, {Kab, Nb, A}Kbs}Kas
5. A -> B         : {Kab, Nb, A}Kbs
6. B -> A         : {Nb}Kab
7. A -> B         : {Dec(Nb)}Kab

knowledge
A       : A,B,S,Kas,Dec;
B       : A,B,S,Kbs,Dec;
S       : A,B,S,Kas,Kbs,Dec;

session_instances
 [A:alice,B:bob,S:server,Kas:key1,Kbs:key2,Dec:dec];

goal
```

Figure C.6: SPAN Main Graphical Interface

C.4.1 Dolev-Yao Intruder Model

The verification procedure and intruder simulations described in chapter 4 have been conducted under a Dolev and Yao's intruder environment. The Dolev and Yao's Method [Dolev & Yao 1981] mainly focuses on a formal model of the intruder capabilities. In this model, the intruder has all means to control the network and capture exchanged messages as much as he wants for later analysis. Indeed, the network is assumed to be under the full control of the intruder who can read, intercept, modify, create, split, re-generate or even block messages exchanged by legitimate users. Also, the intruder has unlimited time to attack the network and its capacities in terms of available computation resources are also unlimited. Moreover, it is assumed that the cryptographic properties of the security protocols are perfect, meaning that cryptanalysis attacks are not possible (*i.e.* brute force or dictionary attacks). Formally, these are the Dolev & Yao intruder capabilities:

- ◇ if the intruder knows messages M_1 and M_2, he can constitute a new message $M_1|M_2$ where $|$ is the concatenation symbol;
- ◇ if the intruder knows M and the key K, he can forge and send $\{M\}_K$;
- ◇ the intruder is able to compute and use *Hash(M)* if he knows *Hash* and M;
- ◇ the intruder has the ability to generate *Nonce* numbers as he wants;
- ◇ the intruder is able to deduce M from $\{M\}_K$ if he has knowledge of K;

- the intruder is able to deduce M from $\{M\}_{K_i^-}$ if he has knowledge of K_i^+ (and vice versa);
- the intruder is able to deduce M_1 from a message $M_1|M_2$ or $M_2|M_1$.

C.4.2 Structure of CAS+ Protocol Specification

The CAS+ specification syntax is composed of 6 different sections: identifiers declarations, messages, session instances, intruder knowledge, and security goals.

C.4.2.1 Identifiers Declarations

First, the identifiers are declared using one of the following types:

- `user`: principal name;
- `public_key` (the postfix ' symbol is used to represent the private key associated to a given public key);
- `symmetric_key`;
- `function`: one way hash function;
- `number`: abstraction of any kind of data (*e.g.* text, numeric value, record, etc).

C.4.2.2 Structure of Messages

The messages exchanged between the users are listed in this section and have the following structure:

$$i.S_i \longrightarrow R_i : M_i$$

where for each i^{th} M_i message, S_i and R_i denote respectively the sender and the receiver users. The arrow $\longrightarrow$ can be written in ASCII symbols as the following:

- –> represents a Dolev-Yao channel;
- => represents a read and write protected channel;
- ∼> represents a write protected channel.

The message M_i can be expressed as the following using the declared identifiers:

- _′ for private keys;
- _(_) for hash functions;
- _, _ for pairing and concatenations;
- {_}_ for encryption;
- _ˆ_ for exponentials;
- _#_ for XOR operations.

C.4.2.3 Knowledge

In the knowledge section, a set of known identifiers is specified for each user. The CAS+ syntax supposes that the own name of every user is implicitly included in his initial knowledge.

C.4.2.4 Session Instances

In this section, it is possible to instantiate several sessions of the same security protocol before the protocol verification. The sessions instances section includes the intruder capabilities and arrangement of sessions (concurrent or sequential).

C.4.2.5 Intruder Knowledge

The same syntax as the one used in the knowledge field is adopted here: a set of known values introduced in the session instance field is specified for the intruder.

C.4.2.6 Security Goals

The final section is dedicated to the security properties that must be checked by the AVISPA backends. Three different type of goals can be expressed:

- `secrecy` for the confidentiality of an identifier for a set of users;
- `authentication` for authentication and integrity security services;
- `weak authentication`: an identifier and two users are specified. The first user authenticates the second one on the given identifier (*e.g.* a secret shared between them).

CAS+ specifications for the SNSSP protocol have been introduced and commented in chapter 4.

C.4.3 Structure of HLPSL Protocol Specification

The HLPSL syntax is decomposed of three main parts: role descriptions, role sessions, and role environment.

C.4.3.1 Role Descriptions

First, the basic role of each user is described using finite state automata, where transitions are triggered when a message is sent or received. With regards to "Alice & Bob" nomenclature, the HLPSL syntax makes internal state of roles, nonce generation, message sending and reception explicit. Later, a role can be instantiated by one or more agents playing the associated role. Each basic role describes what information the participant knows initially (parameters), its initial state, and ways in which this state can change (transitions). The parameters appearing in the `played_by` section, mean that any agent given behaves as specified by this role. The transition section contains a set of transition, each one refers to the receipt and the sending of a message. Every transition is triggered according to some preconditions, and has an action to be performed. The `'` symbol means that a variable value has changed after the current transition is completed. `RCV` and `SND` are variables of type channel, which is usually a Dolev Yao (denoted `dy`) channel (*c.f.* chapter 4).

Here is the role declaration relevant to the initial negotiation and re-negotiation phases of the SNSSP protocol specification. Three roles are declared: SecMan proxy (denoted `P`), ground server (denoted `S`), and certification authority (denote `Ca`):

Basic Role Description C.1 SNSSP Initial Negotiation

```
role role_P(P:agent,S:agent,F:function,Ca:agent,SND,RCV:channel(dy))
played_by P def=
local State:nat,CertRequest:text,Kca:public_key,Ks:public_key,Nonce1:text,SSPs:text,
H:function,IdS:text,Lifetime:text
init State := 0
transition
1.  State=0 ∧ RCV(start) =|> State':=1 ∧ CertRequest':=new() ∧ SND(CertRequest')
2.  State=1 ∧ RCV({IdS'}_inv(Kca')) =|> State':=2 ∧ Nonce1':=new() ∧ SND(Nonce1')
4.  State=2 ∧ RCV(SSPs'.H(SSPs').IdS.Nonce1.Lifetime'.{F(H(SSPs').IdS.Nonce1.
Lifetime')}_inv(Ks')) =|> State':=3
end role

role role_S(S:agent,P:agent,F:function,H:function,Ca:agent,SND,RCV:channel(dy))
played_by S def=
local State:nat,Ks:public_key,Nonce1:text,SSPs:text,IdS:text,Lifetime:text
init State := 0
transition
3.  State=0 ∧ RCV(Nonce1') =|> State':=1 ∧ Ks':=new() ∧ Lifetime':=new() ∧
IdS':=new() ∧ SSPs':=new() ∧ SND(SSPs'.H(SSPs').IdS'.Nonce1'.Lifetime'.{F(H(SSPs').
IdS'.Nonce1'.Lifetime')}_inv(Ks'))
end role

role role_Ca(Ca:agent,S:agent,P:agent,SND,RCV:channel(dy))
played_by Ca def=
local State:nat,CertRequest:text,Kca:public_key,IdS:text
init State := 0
transition
1.  State=0 ∧ RCV(CertRequest') =|> State':=1 ∧ Kca':=new() ∧ IdS':=new() ∧
SND(IdS'_inv(Kca'))
end role
```

Basic Role Description C.2 SNSSP Re-negotiation

```
role role_P(P:agent,S:agent,F:function,SND,RCV:channel(dy))
played_by P def=
local State:nat,Nonce2:text,Ks:public_key,Nonce3:text,SSPs:text,H:function,
IdS:text,Nonce4:text
init State := 0
transition
1.  State=0 ∧ RCV(start) =|> State':=1 ∧ Nonce3':=new() ∧ Nonce2':=new() ∧
SND(Nonce2'.Nonce3')
2.  State=1 ∧ RCV(H(SSPs').IdS'.Nonce3.Nonce4'.{F(H(SSPs').IdS'.Nonce3.Nonce4')}_inv
(Ks')) =|> State':=2
end role
```

```
role role_S(S:agent,P:agent,F:function,H:function,SND,RCV:channel(dy))
played_by S def=
local State:nat,Nonce2:text,Ks:public_key,Nonce3:text,SSPs:text,IdS:text,Nonce4:text
init State := 0
transition
1.  State=0 ∧ RCV(Nonce2’.Nonce3’) =|> State’:=1 ∧ Ks’:=new() ∧ Nonce4’:=new() ∧
IdS’:=new() ∧ SSPs’:=new() ∧ SND(H(SSPs’).IdS’.Nonce3’.Nonce4’.{F(H(SSPs’).IdS’.
Nonce3’.Nonce4’)}_inv(Ks’))
end role
```

C.4.3.2 Role Sessions

After role declaration, these roles are combined together in order to form one or several protocol sessions where commonly shared knowledge is made explicit in the HLPSL specification. In role sessions, there is no `transition` section as for the basic role declaration, but rather a `composition` section where the basic roles are instantiated. The ∧ symbol means that basic roles should be executed in parallel to form the session. Here are the role sessions for the SNSSP protocol:

Role Session C.3 SNSSP Initial Negotiation

```
role session1(S:agent,P:agent,F:function,H:function,Ca:agent) def=
local SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy)
composition
role_S(S,P,F,H,Ca,SND3,RCV3) ∧ role_Ca(Ca,S,P,SND2,RCV2) ∧ role_P(P,S,F,Ca,SND1,RCV1)
end role
```

Role Session C.4 SNSSP Initial Negotiation

```
role session1(S:agent,P:agent,F:function,H:function) def=
local SND2,RCV2,SND1,RCV1:channel(dy)
composition
role_S(S,P,F,H,SND2,RCV2) ∧ role_P(P,S,F,SND1,RCV1)
end role
```

C.4.3.3 Role Environment

At the end, the environment used for the protocol verification, the intruder (denoted `i`), his initial knowledge (name of agents, public keys, own private key, functions, and shared keys), global constants, and number of protocol sessions are explicitly defined, which corresponds to a top level role. Here are the role environments both the initial negotiation and re-negotiation phases relevant to the SNSSP protocol:

Role Environment C.5 SNSSP Initial Negotiation

```
role environment() def=
const hash_0:function,const_1:function,secman:agent,server:agent, authority:agent,
auth_1:protocol_id
intruder_knowledge = {secman,authority,server}
composition
session1(server,secman,const_1,const_1,authority)
end role
```

```
goal
authentication_on auth_1
end goal
environment()
```

Role environment C.6 SNSSP Re-negotiation

```
role environment() def=
const hash_0:function,server:agent,secman:agent,const_1:function,const_1:function,
auth_1:protocol_id
intruder_knowledge = {secman,server}
composition
session1(server,secman,const_1,const_1)
end role
goal
authentication_on auth_1
end goal
environment()
```

C.5 Description of the Analytic Hierarchical Process Method

AHP is an analytical multi-criteria technique used to resolve decision making problems for complex and heterogeneous systems. The idea is to find out what are the factors involved in the decision process and then constitute a hierarchical model according to the dependencies between these factors. The advantage of the AHP method is to combine qualitative and quantitative factors (*e.g.* factor *A* is *low* and factor *B* is equal to *5*), and reduce the subjectivity of the results when the decider has to choose one of the proposed final alternatives. Figure C.7 shows the AHP hierarchy approach:

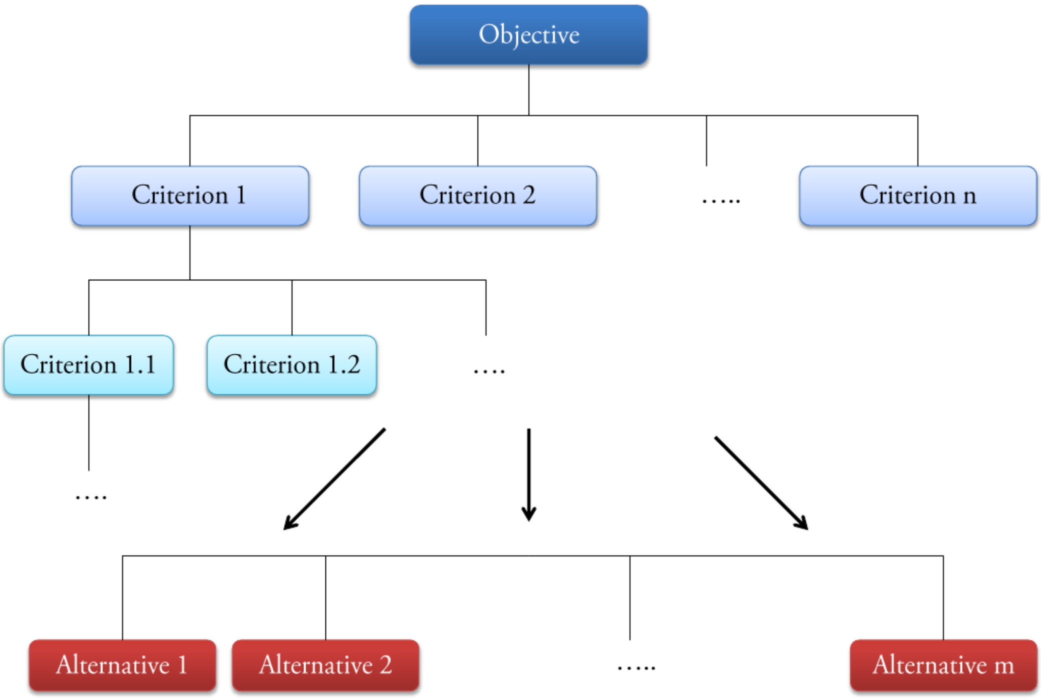


Figure C.7: The AHP Hierarchy Approach

The AHP process is mainly composed of the following steps:

1. *Establishment of the hierarchical structure*: the first step is to identify the main objective, a set of criteria and sub-criteria, and the set of alternative choices used for the comparison. The alternatives are the leafs of the tree whereas the objective corresponds to its root as shown in figure C.7. For instance, if the objective is to choose the best car among a selection of three different vehicles (which are the alternatives), the criteria could be the price, the style, the comfort, and the fuel consumption;

2. *Pair-wise comparison*: pair-wise comparisons are performed at each level of the hierarchy, starting from the bottom of the tree, with respect to the upper level objectives. Once all the criteria of the same level are weighted, the evaluation is moved to the upper level. AHP uses a [1..9] ratio scale to weight the criteria and sub-criteria at the same level of the hierarchy, depending on the relative importance they may represent for each others. A square comparison matrix $D = (d_{ij})$ of order n is then established under the constraints that:

   ◇ $d_{ij} = 1/d_{ji}, \quad \forall i \neq j$;

   ◇ $d_{ii} = 1, \quad \forall i \in \mathbb{N}^*$.

   The comparison matrix D is said to be reciprocal;

3. *Consistency evaluation*: the comparisons provided in the previous step need to be consistent, then the consistency of the matrix D has to be tested. The consistency is the logical coherence among the weights (*i.e.* the judgments). For example, if an object A has greater value than an object B (we write $A > B$), and B has greater value than an object C $(B > C)$, then logically A has greater value than C $(A > C)$. This logic of preference is called transitive property. Thus consistency is closely related to the transitive property of the matrix D. Consequently, the weights d_{ij} are consistent if they are transitive, that is :

$$d_{ik} = d_{ij} * d_{jk}, \quad \forall i, j, k \in \mathbb{N}^* \tag{C.1}$$

   The next step is to find a vector ω of order n such that $D * \omega = \lambda * \omega$. For such as matrix, ω is said to be an eigenvector and λ is an eigenvalue. For a consistent matrix, λ is equal to n. For matrices involving human judgment, the condition $d_{ik} = d_{ij} * d_{jk}$ shown in equation C.1 does not hold as human judgments are more or less consistent. In such a case, the ω vector should satisfy the equation:

$$D * \omega = \lambda_{max} * \omega \tag{C.2}$$

   where $\lambda_{max} \geq n$. The difference, if any, between λ_{max} and n is an indication of the inconsistency of the judgments. If $\lambda_{max} = n$ then the judgments have turned out to be consistent;

4. *Weights synthesis and score computation*: finally, a Consistency Index (CI) is calculated as:

$$CI = \frac{\lambda_{max} - n}{n - 1} \tag{C.3}$$

   CI needs to be assessed against judgments made completely at random and a Consistency Ratio (CR) is calculated by dividing CI for the set of judgments on the Index CI for the corresponding random matrix:

   ◇ if $CR \geq 0.1$, the judgment may be too inconsistent to be reliable and the process must be repeated;

   ◇ if $CR = 0$, it means that the judgments are perfectly consistent, but practically a $CR < 0.1$ is sufficient.

C.6 Risk Assessment Algorithm Diagram Class

The risk assessment algorithm presented in chapter 3 has been implemented in Java. Figure C.8 shows the diagram class relative to the risk assessment algorithm source code:

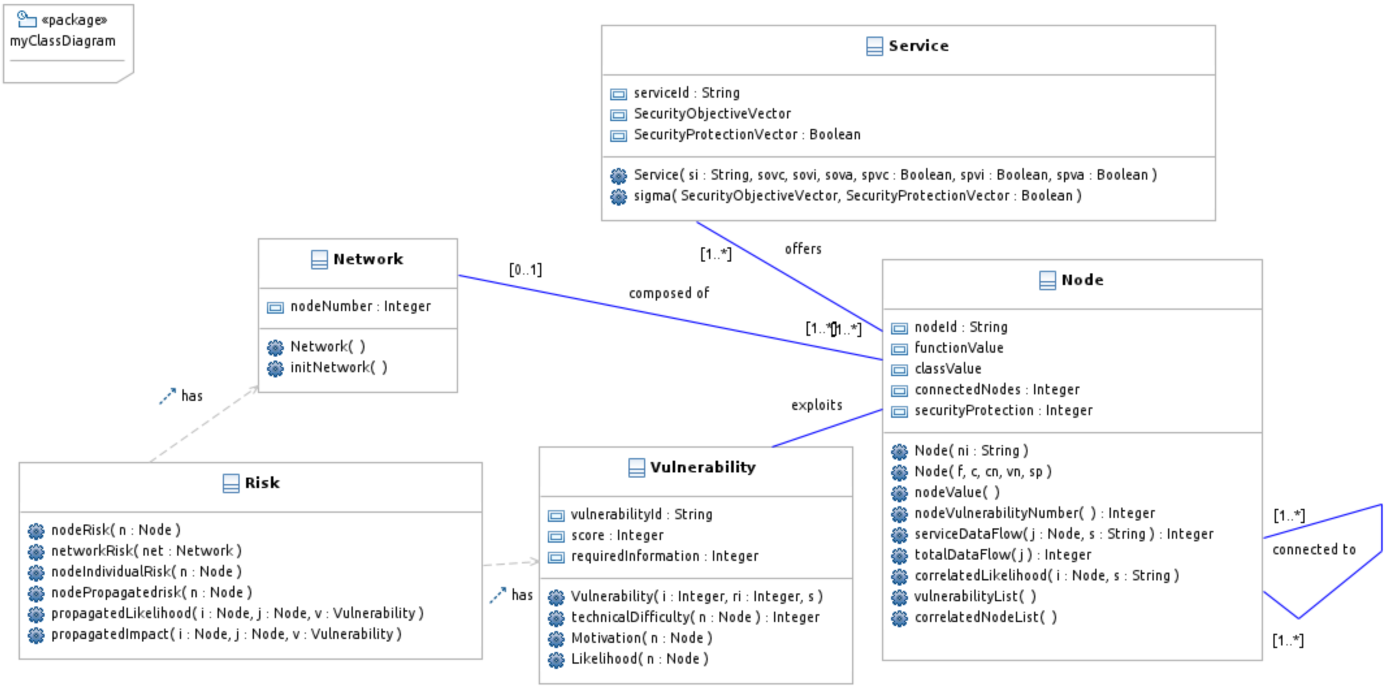

«package»
myClassDiagram
Service
serviceId : String
SecurityObjectiveVector
SecurityProtectionVector : Boolean
Service(si : String, sovc, sovi, sova, spvc : Boolean, spvi : Boolean, spva : Boolean)
sigma(SecurityObjectiveVector, SecurityProtectionVector : Boolean)
Network
nodeNumber : Integer
Network( )
initNetwork( )
[0..1]
composed of
[1..*]
offers
[1..*][1..*]
exploits
has
Node
nodeId : String
functionValue
classValue
connectedNodes : Integer
securityProtection : Integer
Node(ni : String)
Node(f, c, cn, vn, sp)
nodeValue( )
nodeVulnerabilityNumber( ) : Integer
serviceDataFlow(j : Node, s : String) : Integer
totalDataFlow(j) : Integer
correlatedLikelihood(i : Node, s : String)
vulnerabilityList( )
correlatedNodeList( )
[1..*]
connected to
[1..*]
Risk
nodeRisk(n : Node)
networkRisk(net : Network)
nodeIndividualRisk(n : Node)
nodePropagatedrisk(n : Node)
propagatedLikelihood(i : Node, j : Node, v : Vulnerability)
propagatedImpact(i : Node, j : Node, v : Vulnerability)
has
Vulnerability
vulnerabilityId : String
score : Integer
requiredInformation : Integer
Vulnerability(i : Integer, ri : Integer, s)
technicalDifficulty(n : Node) : Integer
Motivation(n : Node)
Likelihood(n : Node)

Figure C.8: Diagram Class of the Risk Assessment Algorithm

5 classes have been created, namely:

- ◇ *Node*: regroups all the information associated with a node (i.e. identifier, function value, class value, number of connected nodes, number of security protections). Each node is associated with a set of services and has one or many specific vulnerabilities. Class methods allow to estimate the node value, node vulnerability number, number of data flows associated with a service, total number of data flows (all associated services), the correlated likelihood, and to create two lists of objects (vulnerability list and correlated node list);

- ◇ *Network*: is composed of a set of several nodes (a method allows to create and initiate the network) and associated with one overall risk;

- ◇ *Vulnerability*: is identified using the NVD nomenclature (*i.e.* CVE followed by the vulnerability publication year and its ID) and has a impact score (which is the CVSS score in the NVD database) and a number of information required to exploit the vulnerability. Methods allows to estimate the motivation and technical difficulty to exploit that vulnerability, and to deduce the associated likelihood of occurrence of threat;

- ◇ *Risk*: the risk class allows to estimate all the risk parameter associated with the other classes, namely the individual risk, the propagated risk, the node risk, and the network risk. The propagated parameters (*i.e.* propagated likelihood and propagated impact) are also calculated in this class;

- ◇ *Service*: is composed by an identifier using the name of the service (*e.g.* technical_log_updates) and both security objective and security protection vectors. A method allows to estimate the sigma value (*c.f.* chapter 3) resulting from the product of both vectors.

C.7 SecMan Integration into the Linux Kernel

In order to handle the traffic coming on the input network interface using the SecMan framework, we used the Netfilter [2] packet framework implemented on Linux kernels: every packet is intercepted, sent to the SecMan components for the adequate treatments, and then re-injected into the OS kernel. In order to communicate with the Netfilter kernel module, a classifier sub-module gets the coming packets, classify them using the priority scale provided in chapter 4, then sends them to the next SecMan components.

We used the Iptables [3] Linux command to configure the Netfilter module and route the packets to the corresponding table and chain. As shown in figure C.9, packets going through the Mangle table of the Forward chain are intercepted, sent to the SecMan framework, then re-injected into the following Netfilter table using a Iptables ruleset:

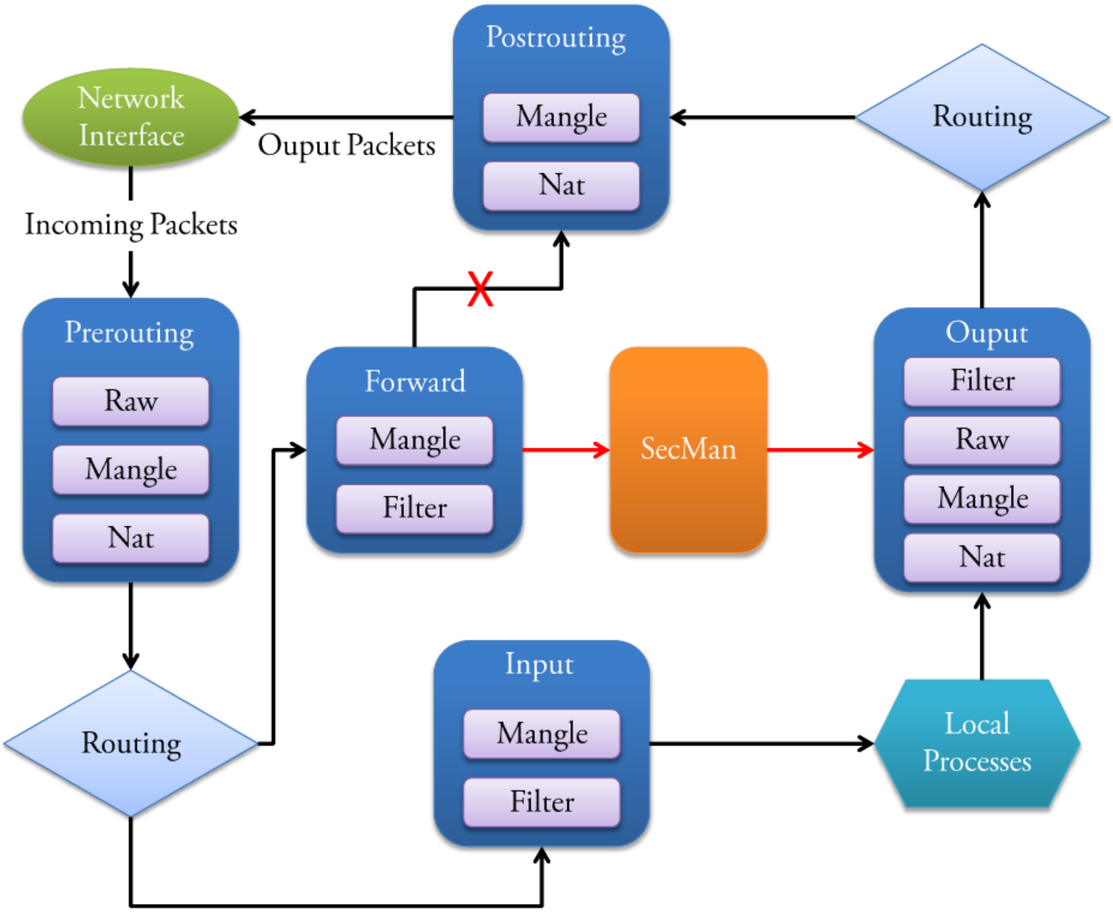


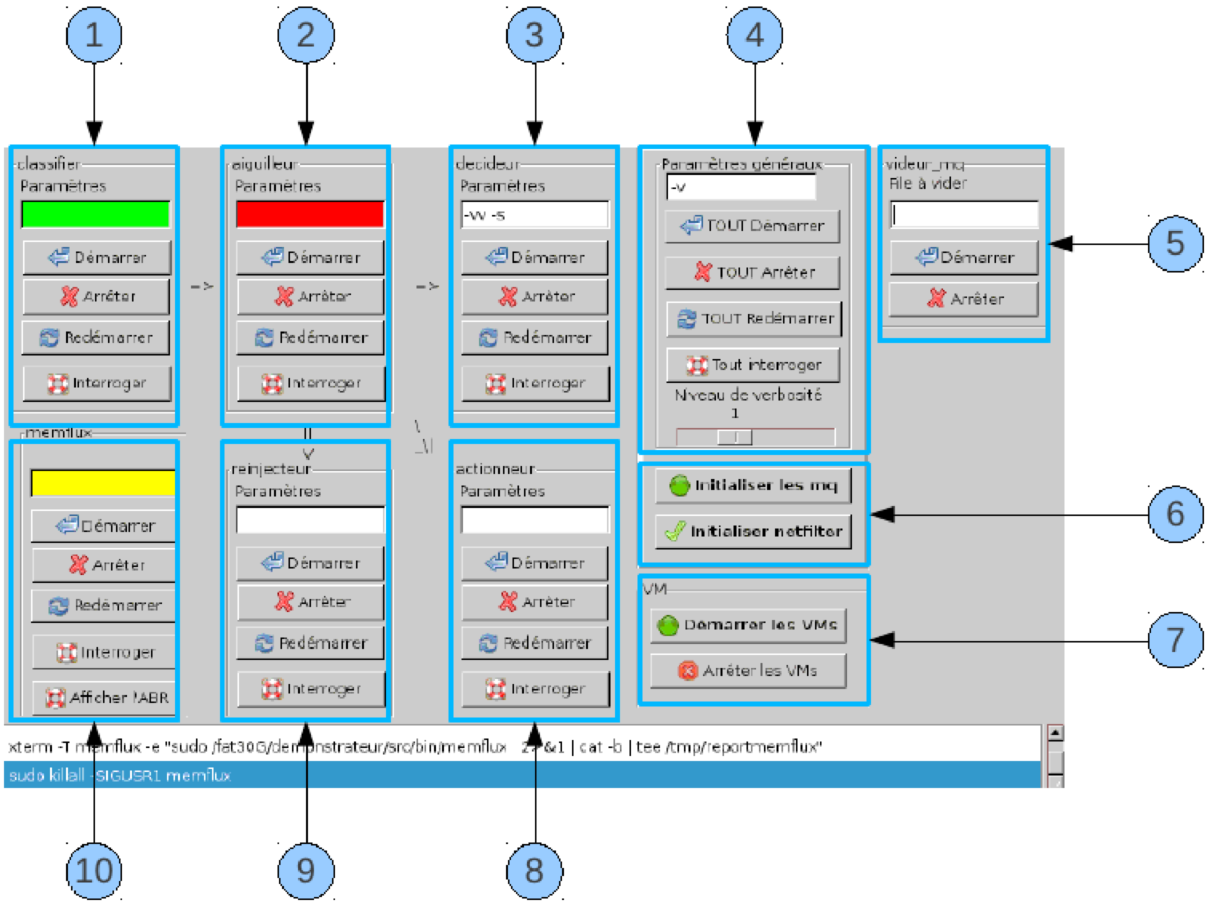
Figure C.9: SecMan Integration with Netfilter Kernel Module

Figure C.10 shows the SecMan management GUI used mainly for debugging issues at early steps of the framework development. Several features have been implemented for configuration matters:

- ◇ *Label 1*: configuration of the classifier interfacing the Netfilter kernel module and the SecMan framework;
- ◇ *Label 2*: configuration of routing rules inside the SecMan module (to which functional block the packet should be sent to?);

[2] http://www.netfilter.org/

[3]



decideur Paramètres: -vv -s

Paramètres généraux: -v

Niveau de verbosité: 1

```
xterm -T memflux -e "sudo /fat30G/demonstrateur/src/bin/memflux 2>&1 | cat -b | tee /tmp/reportmemflux"
sudo killall -SIGUSR1 memflux
```

Figure C.10: SecMan Management GUI

- ◇ *Label 3*: configuration of the static security policies using the operational modes with the adequate option:

  - ▷ `-t` for the secure transport mode;
  - ▷ `-s` for the secure application mode;
  - ▷ `-i` for the secure transparent mode;
  - ▷ `-b` for the insecure `Bypass` mode.

  For each of these options, it is possible to force the static policy using an uppercase option letter (*e.g.* `-B` to force the Bypass mode).

- ◇ *Label 4*: start all the remaining processes using the same options. It is also possible to adjust the verbosity level of the SecMan output display;

- ◇ *Label 5*: clean up and remove the contents of a message queue if required;

- ◇ *Label 6*: initialize and create the FIFOs and the Iptables setrule for redirecting the packets from the input network interface to the SecMan framework, then send them to the output network interface;

- ◇ *Label 7*: start the virtual machines dedicated to the FAST testbed (*c.f.* chapter 4) and execute in parallel the `htop` Linux program for system resource monitoring;

- ◇ *Label 8*: configuration of the static security policy. For instance, using the insecure transport mode, it is possible to indicate the security features to use (*e.g.* `stunnel -C DES-CBC-MD5 -d 1002 -r 1003 -c`);
- ◇ *Label 9*: if needed, this window is used to force the re-injection of packets into the Output Netfilter chain (*c.f.* figure C.9);
- ◇ *Label 10*: control of the SecMan message queues.

For each of the labels shown above, it is possible to start, stop (using the `SIGTERM` signal), or re-start the software component individually. For each program, a button ("intérroger" in figure C.10) allows to display statistics and the state of the program (*e.g.* number of input packets, dropped packets) on the shell terminal associated to them (practically, when the button is pushed on, a `SIGUSR1` signal is sent to the process). For each configuration window, a text zone is associated and allows the user to insert the adequate starting options for a given process. The text zone color can be green if the process is under operation, red if stopped, or yellow if the `SIGUSR1` signal has been triggered. The bottom part of the SecMan management GUI is a non-interactive output console showing the history of all the Linux-based commands that correspond to the processes launched by the user.

C.8 Overview of the Transport Layer Security Protocol

This section aims to provide more details about the TLS security protocol used in the thesis work. Specifically, a particular attention is given to the sub-security layers and the signalization handshake phase as a support to the results provided in the last section of chapter 4.

Transport Layer Security (TLS), originally known as Secure Socket Layer (SSL), is a security protocol developed by Netscape[4] and renamed by the IETF when it purchased the patent in 2001. TLS is mostly used to protect Internet exchanges, mainly HTTP transactions or secure emails (*i.e.* IMAPS and POPS), but the protocol is design to work with most TCP-based applications. Broadly, the TLS operations can be divided into two major parts. The first one refers to the establishment of a secure connection, which goes from key exchanges (*e.g.* shared session keys) to the agreement on the used algorithms. The other part is the effective transfer of application data while ensuring the security services (namely confidentiality, user authentication, and data integrity). Figure C.11 shows the TLS protocol architecture:

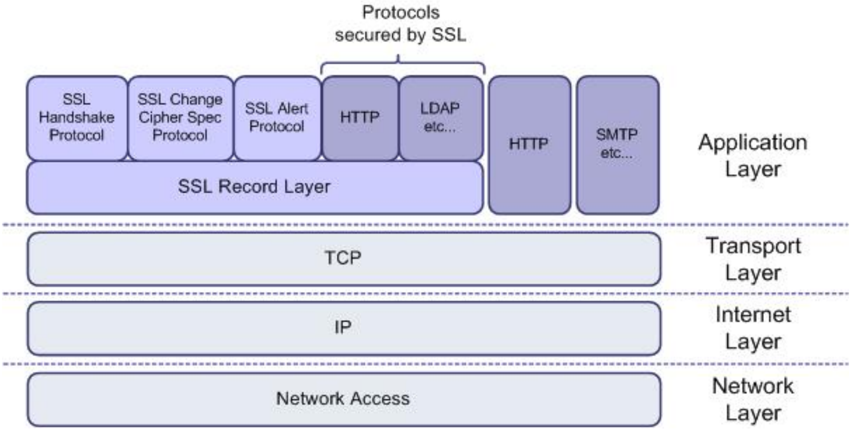


Figure C.11: TLS Protocol Architecture

[4] http://isp.netscape.com/

As we can see, TLS is a layered protocol and consists of four sub-protocols:

1. *Handshake protocol*: is used to perform authentication and key exchanges. The handshake protocol is an essential component of TLS, the main idea is to establish and negotiate all the security credentials beforehand the effective transfer of application data. Besides the negotiation phase (supported version of the TLS/SSL protocol, algorithms, hash functions), the Handshake protocol phase provides a strong authentication between end entities using digital certificates. Once the handshake is complete, the communicating entities share a secret which is used to built a secure channel over which application data can be exchanged. Besides, TLS is an asymmetric protocol, meaning that it differentiates between a client and a server. Therefore, the TLS handshake sequence may vary, depending on whether RSA or Diffie-Hellman key exchange is used. Even if the TLS handshake protocol allows the client and the server to be authenticated to each other, in most cases, it is only the server that is authenticated. Figure C.12 shows the message flow exchange required to establish a new TLS session using the TLS handshake protocol:

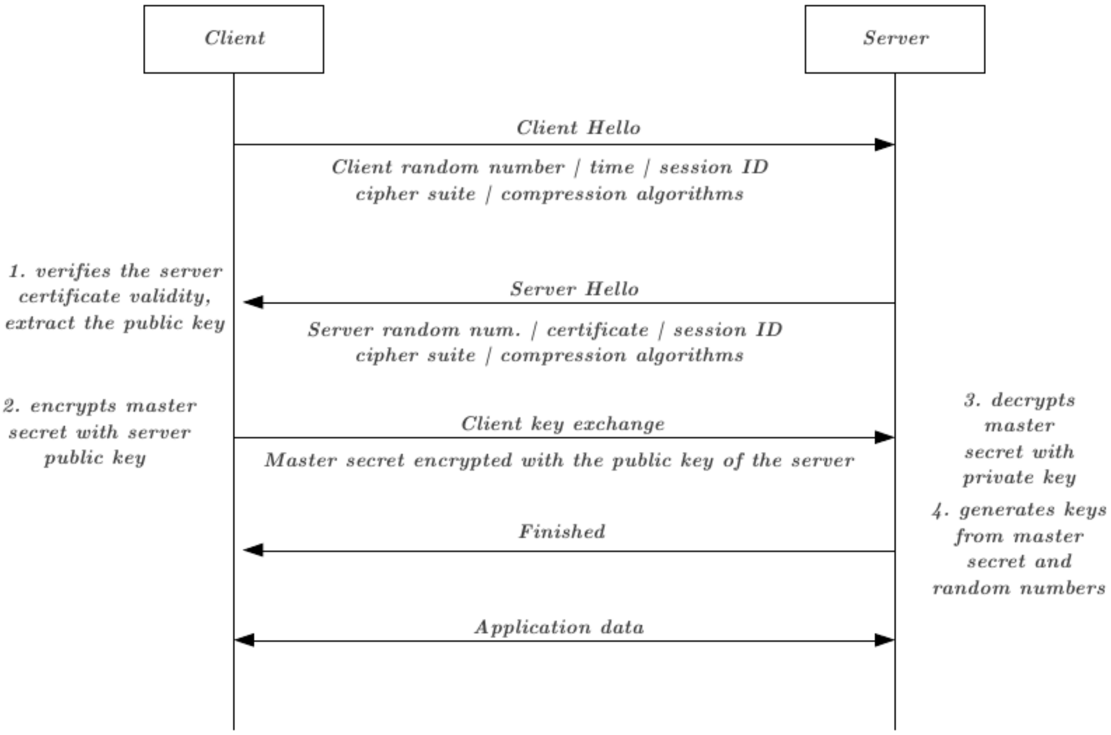

Client
Server
Client Hello
Client random number / time / session ID
cipher suite / compression algorithms
1. verifies the server certificate validity, extract the public key
Server Hello
Server random num. / certificate / session ID
cipher suite / compression algorithms
2. encrypts master secret with server public key
Client key exchange
Master secret encrypted with the public key of the server
3. decrypts master secret with private key
4. generates keys from master secret and random numbers
Finished
Application data

Figure C.12: Message Flow in TLS Handshake Protocol - Initial Handshake

The client sends a Hello message to the server which includes a random number used to prevent replay attacks. In response to the client, the server replies with a Hello message followed by a digital certificate containing its own public key. If needed, the server can send a chain of certificates belonging to the CAs in the certification hierarchy. After verifying the server's certificate, the client generates a pre-master secret, encrypts it using the public key binded to the server's certificate, then sends it to the server which decrypts it using its own private key. On each side, the pre-master secret is used to generate a master secret key that is shared between the two parties. This key is used to generate symmetric keys for both message encryption and message authentication. The established TLS session is identified at the end by a session ID that was initially included in the server Hello message. In order to do not restart the handshake phase from the beginning, it is possible to re-establish a cached TLS session as shown in figure C.13:

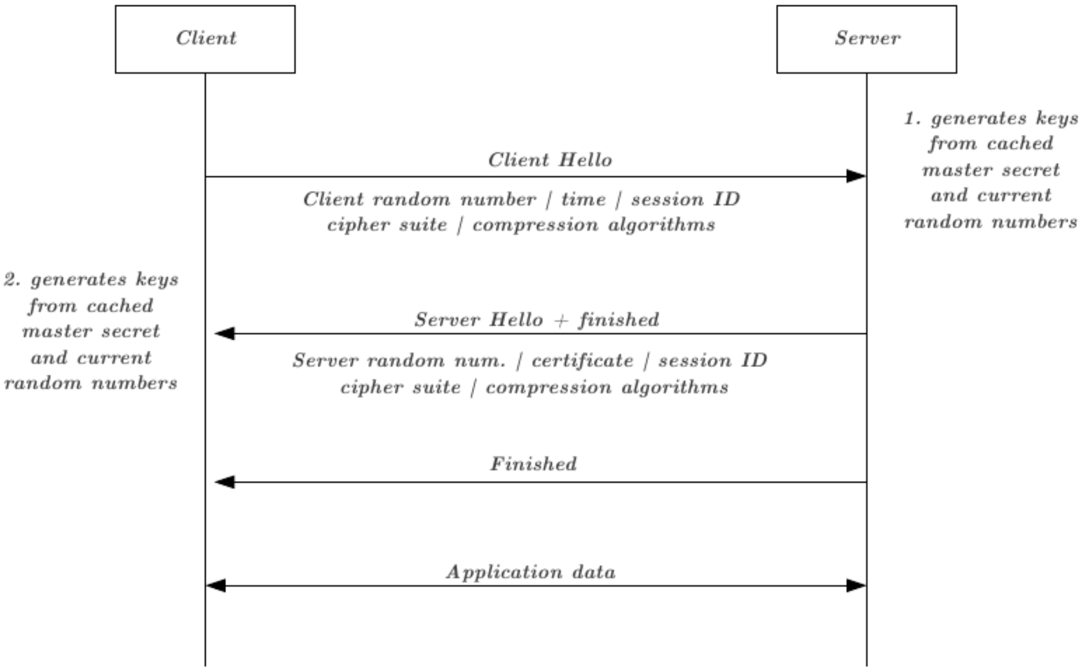

Client
Server
1. generates keys from cached master secret and current random numbers
Client Hello
Client random number / time / session ID cipher suite / compression algorithms
2. generates keys from cached master secret and current random numbers
Server Hello + finished
Server random num. / certificate / session ID cipher suite / compression algorithms
Finished
Application data

Figure C.13: Message Flow in TLS Handshake Protocol - Session Reuse

When needed, the client simply specifies the session ID of the previous TLS session it wishes to reuse when sending the Hello message. The server verifies in its cache if it has state associated with this session ID: if it exists, the server reuses the stored master secret to create keys for the secure channel. The client repeats then the same process and generates an identical set of keys. It is even possible to set multiple secure channels between the same pair of end entities by reusing a single session state. This is an important feature of the TLS handshake protocol for TCP-based applications. For instance, a single secure web page may be composed of multiple HTTP links and being able to reuse an existing TLS session to obtain the multiple links reduces the latency and processing overhead involved in setting up the secure channel;

2. *Change Cipher Spec protocol*: is used by the client or the server to indicate the receiving party that the data will be protected using the negotiated *CipherSpec* and keys. It is used to update the cipher suite to be used in the current TLS session without having to renegotiate the connection;

3. *Alert protocol*: is used for signaling errors and TLS session closure. This layer is formed with two fields:

   - ◇ *the severity level*: field contains a “1” value for a cautionary or warning message, suggesting the session closure and re-initialization of the handshake phase. The severity level field contains sends a “2” value for a fatal alert message that requires both parties to discontinue their TLS session;
   - ◇ *the alert description*: indicates the specific error that caused the Alert Message to be sent from a party (*e.g.* handshake failure, unsupported certificate, revoked certificate);

4. *TLS Record protocol*: is the lower sub-layer of the overall TLS protocol architecture, it allows to:

   - ◇ fragment the application data from upper layers;
   - ◇ apply compression techniques on the resulting packets;
   - ◇ deduce and add MAC authentication;
   - ◇ cipher the data packets then pass it to the TCP layer.

Appendix D

Publications

International Conferences

◇ [Ben Mahmoud *et al.* 2011c] **Ben Mahmoud Mohamed Slim**, N. Larrieu, and A. Pirovano. "Risk Assessment Results of the 15.2.7 WP in SESAR".

In Proceedings of the SAE 2011 AeroTech Congress and Exhibition. October 2011. Toulouse, France.

Oral Excellence Distinction

◇ [Ben Mahmoud *et al.* 2011d] **Ben Mahmoud Mohamed Slim**, N. Larrieu, and A. Pirovano. "A Risk Propagation Based Quantitative Assessment Methodology for Network Security - Aeronautical Network Case Study".

In Proceedings of the 6th International Conference on Network Architectures and Information Systems Security IEEE SAR-SSI 2011. May 2011. La Rochelle, France.

◇ [Ben Mahmoud *et al.* 2010c] **Ben Mahmoud Mohamed Slim**, N. Larrieu, and A. Pirovano. "A Performance-aware Public Key Infrastructure for Next Generation Connected Aircraft".

In Proceedings of the IEEE/AIAA 29th Digital Avionics Systems Conference, DASC 2010. October 2010. Salt Lake City, Utah, USA.

Best of Track "Datalink for Air Traffic Management" award

◇ [Ben Mahmoud *et al.* 2010e] **Ben Mahmoud Mohamed Slim**, N. Larrieu, and A. Pirovano. "Security Architecture Design for Satellite Aeronautical Data Link Communications".

In Proceedings of the AIAA 28th International Communications Satellite Systems Conference, ICSSC 2010. September 2010. Anaheim, California, USA.

◇ [Ben Mahmoud *et al.* 2010f] **Ben Mahmoud Mohamed Slim**, N. Larrieu, A. Pirovano, and A. Varet. "An Adaptive Security Architecture for Future Aircraft Communications".

In Proceedings of the IEEE/AIAA 29th Digital Avionics Systems Conference, DASC 2010. October 2010. Salt Lake City, Utah, USA.

◇ [Ben Mahmoud *et al.* 2009a] **Ben Mahmoud Mohamed Slim**, N. Larrieu, and A. Pirovano. "An Aeronautical Data Link Security Architecture Overview".

In Proceedings of the IEEE/AIAA 28th Digital Avionics Systems Conference, DASCâ09. October 2009. Orlando, Florida, USA.

National Conferences

◇ [Ben Mahmoud *et al.* 2010d] **Ben Mahmoud Mohamed Slim**, N. Larrieu, and A. Pirovano. "Sécurisation des Communications Aéronautiques de Données".

2ème Journée des doctorants de l'Ecole Nationale de l'Aviation Civile. June 2010. Toulouse, France.

- ◇ [Ben Mahmoud *et al.* 2010a] **Ben Mahmoud Mohamed Slim**, N. Larrieu, and A. Pirovano. "Définition d'une Architecture de Sécurité Adaptative pour les Communications Aéronautiques".

  11ème Congrès des doctorants EDSYS 2010. May 2010. Toulouse, France.

Technical Reports and Project Deliverables

- ◇ [Ben Mahmoud *et al.* 2011f] **Ben Mahmoud Mohamed Slim** and N. Larrieu. "SESAR WP 15.2.7 Preliminary Results". September 2011.
- ◇ [Ben Mahmoud *et al.* 2011a] **Ben Mahmoud Mohamed Slim**, N. Larrieu and A. Pirovano. "Rapport de la Revue Finale du Projet FAST". June 2011.
- ◇ [Ben Mahmoud *et al.* 2011e] **Ben Mahmoud Mohamed Slim** and N. Larrieu. "Security Risk Analysis of AeroMACS System in SESAR". May 2011.
- ◇ [Ben Mahmoud *et al.* 2011b] **Ben Mahmoud Mohamed Slim**, N. Larrieu, and A. Pirovano. "Rapport de Validation, Lot de Travail 3.1 du Projet FAST".

  Technical Committee "Integration and Validation". January 2011.
- ◇ [Ben Mahmoud *et al.* 2010b] **Ben Mahmoud Mohamed Slim**, N. Larrieu, and A. Pirovano. "La Gestion de la Sécurité des Communications dans le Projet FAST".

  Technical Committee "System Architecture". February 2010.
- ◇ [Ben Mahmoud *et al.* 2009b] **Ben Mahmoud Mohamed Slim**, N. Larrieu, and A. Pirovano. "Campagne de Simulation Permettant un Dimensionnement du Lien Satellite du Projet FAST".

  Technical Committee "Applications". December 2009.
- ◇ [Ben Mahmoud *et al.* 2009d] **Ben Mahmoud Mohamed Slim**, N. Larrieu, and A. Pirovano. "Introduction aux Communications Aéronautiques".

  Technical Committee "Applications". November 2009.
- ◇ [Ben Mahmoud *et al.* 2009e] **Ben Mahmoud Mohamed Slim**, N. Larrieu, and A. Pirovano. "Recensement et Caractérisation des Flux Applicatifs du Projet FAST".

  Technical Committee "Applications". April 2009.
- ◇ [Ben Mahmoud *et al.* 2009c] **Ben Mahmoud Mohamed Slim**, N. Larrieu, and A. Pirovano. "Définition des Besoins de Sécurité Relatifs aux Flux Applicatifs du Projet FAST".

  Technical Committee "Applications". March 2009.

Bibliography

[Aboba *et al.* 2004] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson and H. Levkowetz. *Extensible Authentication Protocol (EAP).* RFC 3748 (Proposed Standard), June 2004. Updated by RFC 5247. (Cited on page 76.)

[Adams & Lloyd 1999] Carlisle Adams and Steve Lloyd. Understanding the public-key infrastructure: Concepts, standards, and deployment considerations. Macmillan Technical Publishing, 1999. (Cited on page 180.)

[Adams & Zuccherato 1998] C. Adams and R. Zuccherato. *Data Certification Server Protocols.* IETF Internet Draft, June 1998. (Cited on page 181.)

[Ahmed *et al.* 2008] M. S. Ahmed, E. Al-Shaer and L. Khan. *A Novel Quantitative Approach For Measuring Network Security.* In Proc. INFOCOM 2008. The 27th Conf. Computer Communications. IEEE, pages 1957–1965, 2008. (Cited on pages 56 and 57.)

[Ali *et al.* 2004] M.S. Ali, R. Bhagavathula and R. Pendse. *Airplane data networks and security issues.* volume 2, pages 8.E.1–81–12 Vol.2, Oct. 2004. (Cited on page 36.)

[ARINC 2005] ARINC. *Aircraft Data Network, Part 5, Network Domain Characteristics and Interconnection, ARINC Specification 664.*, April 2005. (Cited on page 29.)

[ARINC 2006] ARINC. *ARINC Report 618-6 Air/Ground Character-Oriented Protocol Specification*, June 2006. (Cited on page 1.)

[ARINC 2007a] ARINC. *Draft 04 (strawman) Of Aeec Project Paper 823 Datalink Security, Part 2: Key Management*, September 2007. (Cited on pages 34, 147 and 152.)

[ARINC 2007b] ARINC. *Draft 1 Of Arinc Project Paper 823 Datalink Security, Part 1: Acars Message Security*, August 2007. (Cited on page 33.)

[Armando & Compagna 2004] Alessandro Armando and Luca Compagna. *SATMC: A SAT-Based Model Checker for Security Protocols.* In Logics in Artificial Intelligence, pages 730–733. 2004. (Cited on page 112.)

[Armando *et al.* 2005] Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, PC Heam, J. Mantovani, S. Modersheim, D. von Oheimb, M. Rusinowitch, J. Santos Santiago, M. Turuani and L. Vigano. *The AVISPA Tool for the automated validation of internet security protocols and applications.* In Proceedings of CAV 2005: 17th International Conference on Computer Aided Verification, 2005. (Cited on page 110.)

[ATA 2009] ATA. *ATA Spec 42, Aviation Industry Standards for Digitial Information Security, Revision 2009.1*, 2009. (Cited on pages 33 and 152.)

[Atkins *et al.* 1996] D. Atkins, W. Stallings and P. Zimmermann. *PGP Message Exchange Formats.* RFC 1991 (Informational), August 1996. Obsoleted by RFC 4880. (Cited on page 145.)

[Babiarz *et al.* 2006] J. Babiarz, K. Chan and F. Baker. *Configuration Guidelines for DiffServ Service Classes*, August 2006. (Cited on page 95.)

[Baker 1999] Phillip Hallam Baker. *OCSP Extensions, Draft IETF PKIX OCSPX*, 1999. (Cited on page 181.)

[Barbeau 2005] Michel Barbeau. *WiMax/802.16 threat analysis.* In Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks, Q2SWinet '05, pages 8–15, New York, NY, USA, 2005. ACM. (Cited on pages 37 and 78.)

[Barth *et al.* 2009] Adam Barth, Benjamin I. P. Rubinstein, Mukund Sundararajan, John C. Mitchell, Dawn Xiaodong Song and Peter L. Bartlett. *A Learning-Based Approach to Reactive Security.* CoRR, vol. abs/0912.1155, 2009. (Cited on page 48.)

[Basin *et al.* 2004] David Basin, Sebastian Modersheim and Luca Vigano. *Technical Report No. 450 OFMC: A Symbolic Model-Checker for Security Protocols*, 2004. (Cited on page 112.)

[Basin 1999] David A. Basin. *Lazy Infinite-State Analysis of Security Protocols.* In Proceedings of the International Exhibition and Congress on Secure Networking - CQRE (Secure) '99, pages 30–42, London, UK, 1999. Springer-Verlag. (Cited on page 112.)

[Ben Mahmoud *et al.* 2009a] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *An aeronautical data link security overview.* In Proc. IEEE/AIAA 28th Digital Avionics Systems Conf. DASC '09, 2009. (Cited on pages 13 and 213.)

[Ben Mahmoud *et al.* 2009b] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *Campagne de Simulation permettant un Dimensionnement du Lien Satellite du Projet FAST*, December 2009. (Cited on pages 13 and 214.)

[Ben Mahmoud *et al.* 2009c] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *Définition des Besoins de Sécurité Relatifs aux flux Applicatifs du Projet FAST*, March 2009. (Cited on pages 92 and 214.)

[Ben Mahmoud *et al.* 2009d] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *Introduction aux Communications Aéronautiques*, November 2009. (Cited on pages 13 and 214.)

[Ben Mahmoud *et al.* 2009e] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *Recensement et Caractérisation des Flux Applicatifs du Projet FAST*, April 2009. (Cited on pages 13 and 214.)

[Ben Mahmoud *et al.* 2010a] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *Définition d'une architecture de Sécurité Adaptative pour les Communications Aéronautiques.* In 11'eme Congr'es des Doctorants EDSYS 2010, Toulouse, France, May 2010. (Cited on pages 92 and 214.)

[Ben Mahmoud *et al.* 2010b] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *La Gestion de la Sécurité des Communications dans le Projet FAST*, February 2010. (Cited on pages 92 and 214.)

[Ben Mahmoud *et al.* 2010c] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *A performance-aware Public Key Infrastructure for next generation connected aircrafts.* In Proc. IEEE/AIAA 29th Digital Avionics Systems Conf. (DASC), 2010. Best Paper of Session "Data Link for Future Air Traffic Management" Award. (Cited on pages 143 and 213.)

[Ben Mahmoud *et al.* 2010d] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *Sécurisation des Communications Aéronautiques de Données.* In 2ème Journée des doctorants de l'Ecole Nationale de l'Aviation Civile, Toulouse, France, June 2010. (Cited on pages 92 and 213.)

[Ben Mahmoud *et al.* 2010e] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *Security Architecture Design for Satellite Aeronautical Data Link Communications.* In AIAA International Communications Satellite Systems Conference, 2010. (Cited on pages 92 and 213.)

[Ben Mahmoud *et al.* 2010f] M. S. Ben Mahmoud, N. Larrieu, A. Pirovano and A. Varet. *An adaptive security architecture for future aircraft communications.* In Proc. IEEE/AIAA 29th Digital Avionics Systems Conf. (DASC), 2010. (Cited on pages 92 and 213.)

[Ben Mahmoud *et al.* 2011a] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *Rapport de la Revue Finale du Projet FAST*, June 2011. (Cited on pages 92, 143 and 214.)

[Ben Mahmoud *et al.* 2011b] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *Rapport de Validation, Lot de Travail 3.1 du Projet FAST*, January 2011. (Cited on pages 92 and 214.)

[Ben Mahmoud *et al.* 2011c] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *Risk Assessment Results of the 15.2.7 WP in SESAR.* In SAE 2011 AeroTech Congress and Exhibition, 2011. Oral Excellence Distinction. (Cited on pages 45 and 213.)

[Ben Mahmoud *et al.* 2011d] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *A Risk Propagation Based Quantitative Assessment Methodology for Network Security - Aeronautical Network Case Study.* In Proc. Conf. Network and Information Systems Security (SAR-SSI), pages 1–9, 2011. (Cited on pages 45 and 213.)

[Ben Mahmoud *et al.* 2011e] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *Security Risk Analysis of the AeroMACS System in SESAR*, May 2011. (Cited on pages 45 and 214.)

[Ben Mahmoud *et al.* 2011f] M. S. Ben Mahmoud, N. Larrieu and A. Pirovano. *SESAR WP 15.2.7 Preliminary Results*, September 2011. (Cited on pages 45 and 214.)

[Bennett & Kailay 1992] S. P. Bennett and M. P. Kailay. *An application of qualitative risk analysis to computer security for the commercial sector.* In Proc. Eighth Annual Computer Security Applications Conf., pages 64–73, 1992. (Cited on pages 47 and 53.)

[Besse *et al.* 2010] F. Besse, F. Garcia, A. Pirovano and J. Radzik. *Wireless Adhoc Network Access for Aeronautical Communications.* 2010. (Cited on page 181.)

[Boettcher *et al.* 2008] C. Boettcher, R. DeLong, J. Rushby and W. Sifre. *The MILS component integration approach to secure information sharing.* In Proc. IEEE/AIAA 27th Digital Avionics Systems Conference DASC 2008, pages 1.C.2–1–1.C.2–14, 26–30 Oct. 2008. (Cited on page 180.)

[Boichut & Oehl 2004] O. Kouchnarenko Boichut Y. P.-C. Heam and F. Oehl. *Improvements on the Genet and Klay Technique to Automatically Verify Security Protocols.* In In Proceedings of Int. Ws. on Automated Verification of Infinite-State Systems (AVIS'2004), joint to ETAPS'04, Barcelona (Spain), 2004. (Cited on page 112.)

[Burrows *et al.* 1990] Michael Burrows, Martin Abadi and Roger Needham. *A logic of authentication.* ACM TRANSACTIONS ON COMPUTER SYSTEMS, vol. 8, pages 18–36, 1990. (Cited on page 110.)

[Chen & Zheng 2006] Xiuzhen Chen and Qinghua Zheng. *Quantitative Hierarchical Threat Evaluation Model for Network Security.* Journal of Software, vol. 17, no. 4, pages pp.885–897, April 2006. (Cited on pages 55 and 57.)

[Chen *et al.* 2009] Jianyong Chen, Cunying Hu, Huawang Zeng and Jun Zhang. *Impact of Security on QoS in Communication Network.* In Proc. Int. Conf. Networks Security, Wireless Communications and Trusted Computing NSWCTC '09, volume 2, pages 40–43, 2009. (Cited on page 9.)

[Chevalier *et al.* 2004] Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, J. Mantovani, S. Modersheim and L. Vigneron. *A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols.* In Austrian Computer Society, 2004. (Cited on page 111.)

[Chokhani *et al.* 2003] S. Chokhani, W. Ford, R. Sabett, C. Merrill and S. Wu. *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework.* RFC 3647 (Informational), November 2003. (Cited on page 152.)

[Cisco 2006] Cisco. *InterSwitch Link and IEEE 802.1Q Frame Format. Document ID: 17056*, August 2006. (Cited on page 128.)

[CNES 2009] CNES. *SATCOM for ATM : Estimation of Capacity Required for AMS(R)S Communications Around 2020 Over European Area*, July 2009. (Cited on pages 22 and 26.)

[Cooper *et al.* 2008] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley and W. Polk. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.* RFC 5280 (Proposed Standard), may 2008. (Cited on pages 11 and 32.)

[Cooper 1999] D. A. Cooper. *A model of certificate revocation.* In Proc. 15th Annual Computer Security Applications Conf. (ACSAC '99), pages 256–264, 1999. (Cited on page 180.)

[Cremers 2006] C. Cremers. *Scyther - Semantics and Verification of Security Protocols.* PhD thesis, Eindhoven University of Technology, 2006. (Cited on page 112.)

[Danfeng *et al.* 2009] Yan Danfeng, Yang Fangchun and Lu Yu. *Service-based quantitative calculation of risk for NGN.* In Proc. 2nd IEEE Int. Conf. Broadband Network & Multimedia Technology IC-BNMT '09, pages 306–310, 2009. (Cited on pages 55 and 57.)

[DCSSI 2004] DCSSI. *EBIOS : Introduction, démarche, techniques, outillage*, 2004. (Cited on page 53.)

[DGAC 2010] DGAC. *Observatoire de l'aviation Civile : Tendance et Derniers Résultats du Transport Aérien International.*, 2010. (Cited on page 164.)

[Dierks & Rescorla 2008] T. Dierks and E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.2.* RFC 5246 (Proposed Standard), aug 2008. Updated by RFC 5746. (Cited on page 36.)

[Diffie & Hellman 1976] W. Diffie and M. Hellman. *New directions in cryptography.* IEEE Transactions on Information Theory, vol. 22, no. 6, pages 644–654, 1976. (Cited on page 31.)

[Dolev & Yao 1981] D. Dolev and A. C. Yao. *On the security of public key protocols.* In Proc. 22nd Annual Symp. Foundations of Computer Science SFCS '81, pages 350–357, 1981. (Cited on page 199.)

[Douligeris & Serpanos 2007] C. Douligeris and D. Serpanos. *PKI Systems.* In Network Security:Current Status and Future Directions, pages 409–418. IEEE, 2007. (Cited on page 5.)

[Duquerroy *et al.* 2004] Laurence Duquerroy, Sébastien JOSSET, Olivier Alphand, Pascal Berthou and Thierry Gayraud. *SatIPSec: an optimized solution for securing multicast and unicast satellite transmissions.* In 22nd AIAA International Communications Satellite Systems Conference (ICSSC), Monterey (USA), 2004. 11 pages. (Cited on page 36.)

[EASA 2007] EASA. *Authorized Release Certificate EASA Form 1*, 2007. (Cited on page 150.)

[ECAC 2006] ECAC. *Manual On Air Passenger Health Issues, Recommendation ECAC/28-1*, 2006. (Cited on page 16.)

[Ehammer *et al.* 2008] M. Ehammer, T. Graupl, C. H. Rokitansky and T. Brikey. *Security consideration for IP based aeronautical networks.* In Proc. IEEE/AIAA 27th Digital Avionics Systems Conference DASC 2008, pages 2.E.1–1–2.E.1–13, 2008. (Cited on page 36.)

[Elkeelany *et al.* 2002] O. Elkeelany, M. M. Matalgah, K. P. Sheikh, M. Thaker, G. Chaudhry, D. Medhi and J. Qaddour. *Performance analysis of IPSec protocol: encryption and authentication.* In Proc. IEEE Int. Conf. Communications ICC 2002, volume 2, pages 1164–1168, 2002. (Cited on page 39.)

[Engdahl 2011] Stan Engdahl. *Implementing FAA Form 8130-3 Electronically - Replacing Paper-based Forms with e-Forms*, 2011. (Cited on page 150.)

[ERAA 2006] ERAA. *Unifying European air transport Air traffic communication plans Avanti Air moves forward*. Journal of The European Regions Airline Association, June 2006. (Cited on page 3.)

[Eren & Detken 2008] E. Eren and K.-O. Detken. *WiMAX Security - Assessment of the security Mechanisms in IEEE 8O2.16d/e*. The 12th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2008, Orlando, Florida, USA., 2008. (Cited on pages 37 and 78.)

[ETSI 2003] ETSI. *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Protocol Framework Definition; Methods and Protocols for Security; Part 1: Threat Analysis*, 2003. (Cited on page 61.)

[ETSI 2005] ETSI. *Digital Video Broadcasting (DVB): Second Generation Framing Structure, Channel Coding, and Modulation Systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications, EN 302 307, V 1.1.1*, March 2005. (Cited on page 92.)

[ETSI 2006] ETSI. *Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM) Services and Architectures: QoS Functional Architecture, Draft TS 102 462, V0.4.2*, January 2006. (Cited on pages 92, 95 and 96.)

[ETSI 2009] ETSI. *Digital Video Broadcasting (DVB); Interaction Channel for Satellite Distribution Systems, EN 301 790*, May 2009. (Cited on pages 36, 92 and 97.)

[EUROCONTROL & FAA 2007] EUROCONTROL and FAA. *Action Plan 17 FCS, Final conclusions and recommendations Report*, November 2007. (Cited on page 37.)

[EUROCONTROL 2002] EUROCONTROL. *Communications Operating Concept and Requirements for the Future Radio System*, 2002. (Cited on pages 4 and 17.)

[EUROCONTROL 2010a] EUROCONTROL. *Link 2000+ Programme: ATC Data Link Operational Guidance for LINK 2000+ Services*, March 2010. (Cited on page 1.)

[EUROCONTROL 2010b] EUROCONTROL. *Long-Term Forecast - IFR Flights Movements 2008-2030*. Technical report, 2010. (Cited on pages ix and 2.)

[EUROCONTROL 2010c] EUROCONTROL. *Network Operations Report 2009 - Indicators and Analysis of the ATM Network Operations Performance*. Technical report, May 2010. (Cited on page 2.)

[EUROCONTROL 2011] European Commission EUROCONTROL. *First SESAR Release 2011*, 2011. (Cited on page 7.)

[Evan Darby 1998] et al Evan Darby. *Design Review Of The Controller-pilot Data Link Communications - Build I (cpdlc-1) Functionality And Computer-human Interface For The Display System Replacement*. Technical Report DOT/FAA/CT-98/16, August 1998. (Cited on page 3.)

[FAA 2011] FAA. *FAA's NextGen Implementation Plan*, March 2011. (Cited on page 8.)

[Ferguson & Schneier 2003] Niels Ferguson and Bruce Schneier. *Practical Cryptography*, 2003. (Cited on page 144.)

[Franck *et al.* 2010] Laurent Franck, Mohamed Mrabet and Bernard Comet. *Dimensioning of an aeronautical telemedicine service via satellite.* In AIAA International Communications Satellite Systems Conference. MO - Dépt. Micro-Ondes (Institut Télécom-Télécom Bretagne), MEDES - MEDES IMPS (CHU de Toulouse), 2010. (Cited on page 16.)

[Fredriksen *et al.* 2002] Rune Fredriksen, Monica Kristiansen, Bjørn Gran, Ketil Stølen, Tom Opperud and Theo Dimitrakos. *The CORAS Framework for a Model-Based Risk Management Process.* In Stuart Anderson, Massimo Felici and Sandro Bologna, editeurs, Computer Safety, Reliability and Security, volume 2434 of *Lecture Notes in Computer Science*, chapitre 11, pages 39–53. Springer Berlin / Heidelberg, Berlin, Heidelberg, September 2002. (Cited on page 53.)

[Freeman *et al.* 2007] T. Freeman, R. Housley, A. Malpani, D. Cooper and W. Polk. *Server-Based Certificate Validation Protocol (SCVP).* RFC 5055 (Proposed Standard), December 2007. (Cited on page 181.)

[Frigault & Wang 2008] M. Frigault and Lingyu Wang. *Measuring Network Security Using Bayesian Network-Based Attack Graphs.* In Proc. 32nd Annual IEEE Int. Computer Software and Applications COMPSAC '08, pages 698–703, 2008. (Cited on pages 56 and 57.)

[Getachew & Griner 2005] Dawit Getachew and James H. Griner. *An Elliptic Curve Based Authentication Protocol For Controller-Pilot Data Link Communications*, 2005. (Cited on pages 35, 147 and 152.)

[Goyal 2004] Vipul Goyal. *Fast Digital Certificate Revocation.* In Security and Protection in Information Processing Systems, volume 147 of *IFIP International Federation for Information Processing*, pages 488–500. Springer Boston, 2004. (Cited on page 181.)

[Gupta *et al.* 2005] Vipul Gupta, M. Millard, S. Fung, Yu. Zhu, N. Gura, H. Eberle and S. C. Shantz. *Sizzle: a standards-based end-to-end security architecture for the embedded Internet.* In Proc. Third IEEE Int. Conf. Pervasive Computing and Communications PerCom 2005, pages 247–256, 2005. (Cited on page 39.)

[Hall *et al.* 2010] E. Hall, J. Budinger, R. Dimond, J. Wilson and R. Apaza. *Aeronautical mobile airport communications system development status.* In Proc. Integrated Communications Navigation and Surveillance Conf. (ICNS), 2010. (Cited on pages x, 74, 75 and 78.)

[Harel & Thiagarajan 2003] David Harel and P. S. Thiagarajan. *Message Sequence Charts.* In In UML for Real: Design of Embedded Real-Time Systems, pages 77–105. Kluwer Academic Publishers, 2003. (Cited on page 198.)

[Harkins & Carrel 1998] D. Harkins and D. Carrel. *The Internet Key Exchange (IKE).* RFC 2409 (Proposed Standard), november 1998. Obsoleted by RFC 4306, updated by RFC 4109. (Cited on pages 34 and 134.)

[Housley *et al.* 2002] R. Housley, W. Polk, W. Ford and D. Solo. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.* RFC 3280 (Proposed Standard), April 2002. Obsoleted by RFC 5280, updated by RFCs 4325, 4630. (Cited on page 145.)

[Huang *et al.* 2009] Rui Huang, Danfeng Yan and Fangchun Yang. *Research of security metric architecture for Next Generation Network.* In Network Infrastructure and Digital Content, 2009. IC-NIDC 2009. IEEE International Conference on, pages 207 –212, nov. 2009. (Cited on pages 55 and 57.)

[Hyeon *et al.* 2010] SeUng Hyeon, Ki-Il Kim and SangWoo Yang. *A new geographic routing protocol for aircraft ad hoc networks.* In Proc. IEEE/AIAA 29th Digital Avionics Systems Conf. (DASC), 2010. (Cited on page 181.)

[IATA 2009] IATA. *World Air Transport Statistics (WATS), 53th edition*, 2009. (Cited on page 5.)

[ICAO 2002] ICAO. *Manual of Technical Provisions for The ATN, Doc 9705, Ed 3*, 2002. (Cited on pages 3, 8, 28, 34, 95 and 147.)

[ICAO 2005] ICAO. *ICAO Manual on Required Communications Performance, Appendix N (Manual on RCP) to the Report.* Technical Report Version 5.1, September 2005. (Cited on pages 4 and 38.)

[ICAO 2006a] ICAO. *Aeronautical Communications Panel (ACP) WG F, Need for Spectrum for Future Aeronautical Air/Ground Communication Systems*, June 2006. (Cited on page 2.)

[ICAO 2006b] ICAO. *Annex 10 To the Convention on International Civil Aviation: Aeronautical telecommunications.* Technical report, 2006. (Cited on pages 14 and 92.)

[ICAO 2008] ICAO. *Manual for the ATN using IPS Standards and Protocols (Doc 9896)*, 2008. (Cited on pages 4, 28 and 34.)

[Ingham & Forrest 2002] Kenneth Ingham and Stephanie Forrest. *A History and Survey of Network Firewalls.* 2002. (Cited on page 102.)

[Iordanakis & Dilintas 2007] M. Iordanakis and G. Dilintas. *ARPAM Routing Protocol Vulnerabilities in Aeronautical Mobile Ad Hoc Networks.* In 2nd International Scientific Conference eRA, September 2007. (Cited on page 181.)

[ISO 1999] ISO. *ISO/IEC 15408: Common Criteria for Information Technology Security Evaluation*, 1999. (Cited on pages 35 and 52.)

[ISO 2004] ISO. *Information technology - Security techniques - Management of information and communications technology security - Part 1: Concepts and models for information and communications technology security management*, 2004. (Cited on page 52.)

[ISO 2005a] ISO. *ISO/IEC 27001: Information Technology, security techniques, information security management systems requirements*, 2005. (Cited on page 52.)

[ISO 2005b] ISO. *ISO/IEC 27002: Information technology - Security techniques - Code of practice for information security management*, 2005. (Cited on page 55.)

[ISO 2007] ISO. *ISO/IEC 13818-1 2007: Information technology – Generic coding of moving pictures and associated audio information: Systems*, 2007. (Cited on page 17.)

[ISO 2008a] ISO. *ISO/IEC 27005: Information Technology, Information Security Techniques, Information Security Risk Management*, 2008. (Cited on pages 52 and 53.)

[ISO 2008b] ISO. *ISO/IEC 9594-8:2008 Information technology, Open systems interconnection, The Directory: Public-key and attribute certificate frameworks*, 2008. (Cited on page 146.)

[ITU-T 2010] ITU-T. *Advanced video coding for generic audiovisual services*, 2010. (Cited on page 17.)

[Iyengar *et al.* 2007] S. Iyengar, H. Cruickshank, P. Pillai, G. Fairhurst and L. Duquerroy. *Security requirements for IP over satellite DVB networks.* In Proc. 16th IST Mobile and Wireless Communications Summit, pages 1–6, 2007. (Cited on page 36.)

[Jacquemard *et al.* 2000] Florent Jacquemard, Michaël Rusinowitch and Laurent Vigneron. *Compiling and verifying security protocols.* In Proceedings of the 7th international conference on Logic for programming and automated reasoning, LPAR'00, pages 131–160, Berlin, Heidelberg, 2000. Springer-Verlag. (Cited on page 198.)

[Jagoda & Pietrzak 1997] Andy Jagoda and Michael Pietrzak. *Medical Emergencies In Commercial Air Travel.* Emergency Medicine Clinics of North America, vol. 15, no. 1, pages 251 – 260, 1997. (Cited on page 16.)

[Jain *et al.* 2011] R. Jain, F. Templin and Kwong-Sang Yin. *Analysis of L-Band Digital Aeronautical Communication Systems: L-DACS1 and L-DACS2.* In Proc. IEEE Aerospace Conf, pages 1–10, 2011. (Cited on page 182.)

[Jing 2009] Liu Jing. *Risk Evaluation Process Model of Information Security.* In Proc. Int. Conf. Measuring Technology and Mechatronics Automation ICMTMA '09, volume 2, pages 321–324, 2009. (Cited on page 47.)

[Johnston & Walker 2004] D. Johnston and J. Walker. *Overview of IEEE 802.16 security*, 2004. Security & Privacy, IEEE. (Cited on page 78.)

[Jones & de La Chapelle 2001] W. H. Jones and M. de La Chapelle. *Connexion by Boeing-broadband satellite communication system for mobile platforms.* In Proc. Communications for Network-Centric Operations: Creating the Information Force. IEEE Military Communications Conf. MILCOM 2001, volume 2, pages 755–758, 2001. (Cited on page 15.)

[Kantor 1991] B. Kantor. *BSD Rlogin.* RFC 1282 (Informational), December 1991. (Cited on page 133.)

[Karp & Kung 2000] Brad Karp and H. T. Kung. *GPSR: greedy perimeter stateless routing for wireless networks.* In Proceedings of the 6th annual international conference on Mobile computing and networking, MobiCom '00, pages 243–254, New York, NY, USA, 2000. ACM. (Cited on page 182.)

[Kent & Seo 2005] S. Kent and K. Seo. *Security Architecture for the Internet Protocol.* RFC 4301 (Proposed Standard), December 2005. (Cited on page 34.)

[Kent 2005] S. Kent. *IP Authentication Header.* RFC 4302 (Proposed Standard), December 2005. (Cited on page 38.)

[Kocher 1999] Paul Kocher. *A Quick Introduction to Certificate Revocation Trees (CRTs).* Technical report, ValiCert, 1999. (Cited on page 181.)

[Kondakci 2010] S. Kondakci. *A causal model for information security risk assessment.* In Proc. Sixth Int Information Assurance and Security (IAS) Conf, pages 143–148, 2010. (Cited on pages 56 and 57.)

[Krawczyk *et al.* 1997] H. Krawczyk, M. Bellare and R. Canetti. *HMAC: Keyed-Hashing for Message Authentication.* RFC 2104 (Informational), February 1997. (Cited on page 78.)

[Lacamera 2007] Daniele Lacamera. *PEPsal: A Performance Enhancing Proxy for TCP Satellite Connections for Internetworking and Resource Management in Satellite Systems.* IEEE Aerospace and Electronic Systems Magazine, vol. 22, no. 8, 2007. (Cited on page 36.)

[Lao & Wang 2008] Guoling Lao and Liping Wang. *The Quantification Management of Information Security Risk.* In Proc. 4th Int. Conf. Wireless Communications, Networking and Mobile Computing WiCOM '08, pages 1–4, 2008. (Cited on page 48.)

[Liang & Wang 2005] W. Liang and W. Wang. *A quantitative study of authentication and QoS in wireless IP networks*. In Proc. IEEE 24th Annual Joint Conf. of the IEEE Computer and Communications Societies INFOCOM 2005, volume 2, pages 1478–1489, 2005. (Cited on page 39.)

[Loddo & L.Saiu 2008] J.V. Loddo and L.Saiu. *MARIONNET: a Virtual Network Laboratory and Simulation Tool*. Simulation Works, 2008. (Cited on page 67.)

[Lowe 1995] Gavin Lowe. *An attack on the Needham-Schroeder public-key authentication protocol*. Information Processing Letters, vol. 56, no. 3, pages 131 – 133, 1995. (Cited on page 110.)

[Lv 2009] Huiying Lv. *Research on Network Risk Assessment Based on Attack Probability*. In Proc. Second Int. Workshop Computer Science and Engineering WCSE '09, volume 2, pages 376–381, 2009. (Cited on pages 56 and 57.)

[Maughan *et al.* 1998] D. Maughan, M. Schertler, M. Schneider and J. Turner. *Internet Security Association and Key Management Protocol (ISAKMP)*. RFC 2408 (Proposed Standard), November 1998. Obsoleted by RFC 4306. (Cited on page 106.)

[McParland *et al.* 2001] T. McParland, V. Patel and W.J. Hughes. *Securing air-ground communications*. volume 2, pages 7A7/1–7A7/9 vol.2, Oct 2001. (Cited on page 35.)

[Medina *et al.* 2008] D. Medina, F. Hoffmann, S. Ayaz and C.-H. Rokitansky. *Feasibility of an Aeronautical Mobile Ad Hoc Network Over the North Atlantic Corridor*. In Proc. 5th Annual IEEE Communications Society Conf. Sensor, Mesh and Ad Hoc Communications and Networks SECON '08, pages 109–116, 2008. (Cited on page 181.)

[Meng *et al.* 2009] Q. Meng, M.H. Dong, Y. Li and Z.W. Ming. *Network Security Analysis Model based on Logic Exploitation Graph*. In Computer Engineering, volume 9, pages 147–149, 2009. (Cited on pages 56 and 57.)

[Micali 2006] Silvio Micali. *Efficient Certificate Revocation*. Technical report, Massachusetts Institute of Technology, 2006. (Cited on page 181.)

[Microsoft 2004] Microsoft. *The Security Risk Management Guide*, 2004. (Cited on page 48.)

[Murawski *et al.* 2004] R. W. Murawski, S. C. Bretmersky and V. K. Konangi. *Evaluation of VDL modes in the en-route domain*. In Proc. 23rd Digital Avionics Systems Conf. DASC 04, volume 1, 2004. (Cited on page 4.)

[Myers *et al.* 1999] M. Myers, R. Ankney, A. Malpani, S. Galperin and C. Adams. *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*. RFC 2560 (Proposed Standard), June 1999. (Cited on page 11.)

[NASA 2009] NASA. *Initial Assessment Of Security Provisions Of The IEEE 802.16e-2005 Standard For Airport Surface Communications*. Technical report, 2009. (Cited on page 78.)

[Needham & Schroeder 1978] Roger M. Needham and Michael D. Schroeder. *Using encryption for authentication in large networks of computers*. Commun. ACM, vol. 21, pages 993–999, December 1978. (Cited on page 110.)

[Nigringy & Phaltankar 2007] Jeff Nigringy and Kaus Phaltankar. Identity assurance in commercial aviation faciliated through a trusted third party hub. 2007. (Cited on pages 148, 152 and 155.)

[NIST 1995] NIST. *Federal Information Processing Standards Publication (FIPS- PUBS) 180-1, Announcing the Standard for SECURE HASH STANDARD*, April 17 1995. (Cited on page 32.)

[NIST 1999] NIST. *Federal Information Processing Standards Publication (FIPS PUBS) 46-3, Data Encryption Standard (DES)*, October 2005 1999. Reaffirmed. (Cited on page 31.)

[NIST 2001] NIST. *Federal Information Processing Standards Publication (FIPS PUBS) 197, Announcing the Advanced Encryption Standard (AES)*, November 26 2001. (Cited on page 31.)

[NIST 2002] NIST. *Federal Information Processing Standards Publication (FIPS- PUBS) 800-30, Risk Management Guide for Information Technology Systems*, July 2002. (Cited on pages 52 and 53.)

[NIST 2005] NIST. *Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication, NIST Special Publication 800-38B*, 2005. (Cited on page 78.)

[Olive *et al.* 2006] M.L. Olive, R.T. Oishi and S. Arentz. *Commercial Aircraft Information Security : an Overview of Arinc Report 811.* pages 1–12, Oct. 2006. (Cited on page 34.)

[Olive 2001] M. L. Olive. *Efficient datalink security in a bandwidth-limited mobile environment - an overview of the Aeronautical Telecommunications Network (ATN) security concept.* In Proc. DASC Digital Avionics Systems The 20th Conf, volume 2, 2001. (Cited on page 35.)

[Patel & McParland 2001] V. Patel and T. McParland. *Public key infrastructure for air traffic management systems.* volume 2, pages 7A5/1–7A5/7 vol.2, Oct 2001. (Cited on pages xi, 147 and 148.)

[Patel & McParland 2008a] Vic Patel and Tom McParland. *Aeronautical Communication Panel Working Group I â Internet Protocol Suite (IPS), Enhancements to Security Guidance*, August 25-29 2008. (Cited on page 34.)

[Patel & McParland 2008b] Vic Patel and Tom McParland. *Application of Mobile IP and Proxy Mobile IP Security*, March 2008. (Cited on page 34.)

[Patel & McParland 2008c] Vic Patel and Tom McParland. *Updated Security Requirements for the âManual for the ATN using IPS Standards and Protocolsâ*, June 2008. (Cited on page 34.)

[Patel *et al.* 2010] R. Patel, Bhavesh B., A. Patel, D. Patel, M. Rajarajan and A. Zisman. *Comparative Analysis of Formal Model Checking Tools for Security Protocol Verification.* In Recent Trends in Network Security and Applications, volume 89 of *Communications in Computer and Information Science*, pages 152–163. 2010. (Cited on pages xiii and 111.)

[Perlman 1999] R. Perlman. *An overview of PKI trust models.* vol. 13, no. 6, pages 38–43, 1999. (Cited on page 155.)

[Peters *et al.* 2011] K. Peters, A. Jabbar, E. K. Cetinkaya and J. P. G. Sterbenz. *A geographical routing protocol for highly-dynamic aeronautical networks.* In Proc. IEEE Wireless Communications and Networking Conf. (WCNC), pages 492–497, 2011. (Cited on page 181.)

[Postel & Reynolds 1983] J. Postel and J.K. Reynolds. *Telnet Protocol Specification.* RFC 854 (Standard), May 1983. Updated by RFC 5198. (Cited on page 133.)

[Postel 1980] J. Postel. *User Datagram Protocol.* RFC 768 (Standard), August 1980. (Cited on page 36.)

[Preneel *et al.* 1998] Bart Preneel, Vincent Rijmen and Antoon Bosselaers. *Recent Developments in the Design of Conventional Cryptographic Algorithms.* In State of the Art in Applied Cryptography, Course on Computer Security and Industrial Cryptography - Revised Lectures, pages 105–130, London, UK, UK, 1998. Springer-Verlag. (Cited on page 120.)

[Radzik *et al.* 2007] José Radzik, Alain Pirovano, Na Tao and Michel Bousquet. *Satellite system performance assessment for In-Flight Entertainment and Air Traffic Control.* Space Comms., vol. 21, pages 69–82, January 2007. (Cited on page 4.)

[Rivest *et al.* 1978] R. L. Rivest, A. Shamir and L. Adleman. *A method for obtaining digital signatures and public-key cryptosystems.* Communications of the ACM, vol. 21, pages 120–126, February 1978. (Cited on page 31.)

[Robinson *et al.* 2007a] Richard Robinson, Mingyan Li, Scott Lintelman, Krishna Sampigethaya, Radha Poovendran, David von Oheimb, Jens-Uwe Buauer and Jorge Cuellar. *Electronic Distribution of Airplane Software and the Impact of Information Security on Airplane Safety.* vol. 4680, pages 28–39, 2007. (Cited on pages 35, 149 and 152.)

[Robinson *et al.* 2007b] Richard Robinson, Mingyan Li, Scott A. Lintelman, Krishna Sampigethaya, Radha Poovendran, David von Oheimb and Jens-Uwe Buauer. *Impact of Public Key Enabled Applications on the Operation and Maintenance of Commercial Airplanes.* 2007. (Cited on pages 35, 150 and 152.)

[Robinson *et al.* 2007c] Richard Robinson, Krishna Sampigethaya, Mingyan Li, Scott Lintelman, Radha Poovendran and David von Oheimb. *Secure Network-Enabled Commercial Airplane Operations: IT Support Infrastructure Challenges.* 2007. (Cited on pages xi, 149, 150 and 152.)

[Roy 2001] A. Roy. *Secure aircraft communications addressing and reporting system (ACARS).* In Proc. DASC Digital Avionics Systems The 20th Conference, volume 2, pages 7A2/1–7A2/11 vol.2, 2001. (Cited on page 5.)

[RTCA 1992] RTCA. *DO-178B, Software Considerations in Airborne Systems and Equipment Certification.* Technical report, RTCA SC-167 EUROCAE WG-12, 1992. (Cited on page 149.)

[RTCA 2010] RTCA. *ED-202, Airworthiness Security Process Specification.* Technical report, RTCA SC-216, August 2010. (Cited on page 33.)

[Saaty 2000] Thomas.L Saaty. Fundamentals of the analytic hierarchy process. 2000. (Cited on page 104.)

[Saillard & Genet 2011] Ronan Saillard and Thomas Genet. *CAS + Manual*, March 2011. (Cited on page 111.)

[Sampigethaya *et al.* 2007] K. Sampigethaya, Mingyan Li, R. Poovendran, R. Robinson, L. Bushnell and S. Lintelman. *Secure wireless collection and distribution of commercial airplane health data.* pages 4.E.6–1–4.E.6–8, Oct. 2007. (Cited on page 35.)

[Schiffman 2005] M. Schiffman. *A Complete Guide to the Common Vulnerability Scoring System (CVSS).* In press, June 2005. (Cited on page 52.)

[Schuba *et al.* 1997] Christoph L. Schuba, Ivan V. Krsul, Markus G. Kuhn, Eugene H. spafford, Aurobindo Sundaram and Diego Zamboni. *Analysis of a Denial of Service Attack on TCP.* In Proceedings of the 1997 IEEE Symposium on Security and Privacy, SP '97, pages 208–, Washington, DC, USA, 1997. IEEE Computer Society. (Cited on page 102.)

[Sher & Magedanz 2006] Muhammad Sher and Thomas Magedanz. *Developing Network Domain Security (NDS) Model for IP Multimedia Subsystem (IMS).* Journal of Networks, vol. 1, no. 6, 2006. (Cited on page 57.)

[Shingledecker *et al.* 2005] C. Shingledecker, S. Giles, Jr. Darby E. R., J. Pino and T. R. Hancock. *Projecting the effect of CPDLC on NAS capacity.* In Proc. 24th Digital Avionics Systems Conf. DASC 2005, volume 1, 2005. (Cited on page 3.)

[SITA 2004] SITA. *AIRCOM Transition From ACARS to ATN*, 2004. (Cited on page 5.)

[Smith *et al.* 2001] Nancy Smith, John Moses, Stephan Romahn, Peter Polson, John Brown, Melisa Dunbar, Everett Palmer and Sandra Lozito. An assessment of flight crew experiences with fans-1 atc datalink. 4th USA/Europe Air Traffic Management R&D Seminar, December 2001. (Cited on page 1.)

[Song *et al.* 2001] Dawn Xiaodong Song, Sergey Berezin and Adrian Perrig. *Athena: a novel approach to efficient automatic security protocol analysis.* J. Comput. Secur., vol. 9, pages 47–74, January 2001. (Cited on page 112.)

[Srivastava & Motani 2005] V. Srivastava and M. Motani. *Cross-layer design: a survey and the road ahead.* vol. 43, no. 12, pages 112–119, 2005. (Cited on pages 97 and 103.)

[Stephens 2004] B. Stephens. *Security architecture for aeronautical networks.* volume 2, pages 8.E.2–81–19 Vol.2, Oct. 2004. (Cited on page 36.)

[Thanthry *et al.* 2006] N. Thanthry, M. S. Ali and R. Pendse. *Security, Internet Connectivity and Aircraft Data Networks.* IEEE Aerospace and Electronic Systems Magazine, vol. 21, no. 11, pages 3–7, 2006. (Cited on page 36.)

[Thanthry *et al.* 2007] N. Thanthry, I. P. Emmadi, A. Srikumar, K. Namuduri and R. Pendse. *SVSS: an intelligent video surveillance system for aircraft.* In Proc. IEEE/AIAA 26th Digital Avionics Systems Conf. DASC '07, 2007. (Cited on page 17.)

[Tu & Shimamoto 2009] H. D. Tu and S. Shimamoto. *A Proposal of Relaying Data in Aeronautical Communication for Oceanic Flight Routes Employing Mobile Ad-Hoc Network.* In Proc. First Asian Conf. Intelligent Information and Database Systems ACIIDS 2009, pages 436–441, 2009. (Cited on page 181.)

[Wargo & Dhas 2003] C. A. Wargo and C. Dhas. *Security consideratiolis for the e-enabled aircraft.* In Proc. IEEE Aerospace Conf, volume 4, 2003. (Cited on page 36.)

[Wilson 2011] Stuart Wilson. *The Network Security Architecture and Possible Safety Benefits of the AEROMACS Network.* In Proceedings of 2011 Integrated Communications Navigation and Surveillance (ICNS) Conference, 2011. (Cited on page 78.)

[Wing 2008] Jeannette M. Wing. *Scenario Graphs Applied to Network Security.* In Information Assurance. Morgan Kaufmann, Burlington, 2008. (Cited on page 48.)

[Xu & Huang 2006] Sen Xu and Chin-Tser Huang. *Attacks on PKM Protocols of IEEE 802.16 and Its Later Versions*, 2006. Wireless Communication Systems, 2006. ISWCS '06. (Cited on page 78.)

[Yau & Zhang 1999] S. S. Yau and Xinyu Zhang. *Computer network intrusion detection, assessment and prevention based on security dependency relation.* In Proc. Twenty-Third Annual Int. Computer Software and Applications Conf. COMPSAC '99, pages 86–91, 1999. (Cited on pages 56 and 57.)

[Ylonen & Lonvick 2006] T. Ylonen and C. Lonvick. *The Secure Shell (SSH) Protocol Architecture.* RFC 4251 (Proposed Standard), January 2006. (Cited on page 50.)

[Yongli *et al.* 2008] T. Yongli, X. Guoai and Y. Yixian. *Information Security Management Measurement Model based on AHP.* Journal of Liaoning Technical University, vol. 27, 2008. (Cited on pages 55 and 57.)

[Yu *et al.* 2009] Wang Yu, Chen Jianhua and He Debiao. *A New Collision Attack on MD5.* In Proc. Int. Conf. Networks Security, Wireless Communications and Trusted Computing NSWCTC '09, volume 2, pages 767–770, 2009. (Cited on page 108.)

[Zhang *et al.* 2004] Yong-Zheng Zhang, Bin-Xing Fang and Xiao-Chun Yun. *A risk assessment approach for network information system.* In Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on, volume 5, pages 2949 – 2952 vol.5, aug. 2004. (Cited on pages 56 and 57.)

Glossary

(A)VBDC	(Absolute) Volume Based Dynamic Capacity
AAA	Authorization Authentication Accounting
AAC	Aeronautical Administrative Communications
AADS	Airline Assets Distribution System
AANETs	Aeronautical Adhoc Networks
ABR	Available Bit Rate
ACARS	Aircraft Communication Addressing and Reporting System
ACL	ATC Clearance
ACM	ATC Communication Management
ACPs	Aeronautical Communication Panels
ADS-B	Automatic Dependent Surveillance - Broadcast
AEEC	Airlines Electronic Engineering Committee
AeroMACS	Aeronautical Mobile Airport Communication System
AeroRP	Geographical Routing Protocol for Highly Dynamic Aeronautical Networks
AES	Advanced Encryption Standard
AES	Aeronautical Earth Station
AFDC	Airline Flight Dispatch
AFDX	Avionics Full-Duplex Switched Ethernet
AH	Authentication Header
AHMMS	Airplane Health Monitoring and Management System
AHP	Analytical Hierarchical Process
AISS	Aeronautical Information System Security
ALE	Annual Loss Expectancy
AMC	ATC Microphone Check
AMS	ACARS Message Security
AMS(R)S	Aeronautical Mobile Satellite (Route) Service
AMSS	Aeronautical Mobile Satellite Service
ANASTASIA	Airborne New Advanced Satellite Techniques and Technologies in a System Integrated Approach
ANSP	Aeronautical Network Service Provider
AOA	Autonomous Operations Area
AOC	Airline Operational Communication
AOCDLL	AOC Data Link Logon
APC	Airline Passenger Communications
APT	Airport
ARL	Authority Certification List
ARMAND	Arrival Manager Information Delivery
ARPAM	Ad-hoc Routing Protocol for Aeronautical Mobile Ad-Hoc Networks

ASN	Access Service Network
ATA	Air Transportation Association
ATC	Air Traffic Control
ATM	Air Traffic Management
ATM	Asynchronous Transfer Mode
ATN	Aeronautical Telecommunication Network
ATS	Air Traffic Service
AVISPA	Automated Validation of Internet Security Protocols and Applications
BAN	Burrows Abadi Needham
BoD	Bandwidth on Demand
BS	Base Station
BSM	Broadband Satellite Multimedia
C&P ACL	Crossing and Passing ACL
CA	Certificate Authority
CBC	Cipher block Chaining
CBR	Constant Bit Rate
CC	Common Criteria
CCB	Connexion By Boeing
CCTA	Central Computer and Telecommunications Agency
CDF	Cumulative Distribution Function
CGD	Certificate Generation and Distribution
ChID	IDentifiers
CL-Atse	Constraint Logic based Attack Searcher
CLTP	Connectionless Transport Protocol
CLUSIF	Club de la Sécurité de l'Information Français
CMAC	Cipher based Message Authentication Code
CMSE	Certificate Management Subordinate Entity
CNES	Centre National d'Etudes Spatiales
CNS	Communications Navigation and Surveillance
COCR	Communications Operating Concept and Requirements
CORAS	Risk Assessment of Security Critical Systems
COTP	Connection Oriented Transport Protocol
COTS	Commercial Off The Shelf
CP	Certificate Policy
CPDLC	Controller to Pilot Data Link Communication
CPS	Common Part Sublayer
CPU	Central Processing Unit
CR	Capacity Requests
CRA	Continuous Rate Assignment
CRAMM	CCTA Risk Analysis and Management Method
CRG	Certificate Regeneration
CRLs	Certifcate Revocation Lists
CRSD	Certificate Revocation Status Directory

CRT	Certificate Retrieval
CRT	Certificate Revocation Tree
CRV	Certificate Revocation
CS	Convergence Sublayer
CSN	Connectivity Service Network
CV	Certificate validation
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
D-ALERT	Data Link Alert
D-ORIS	Data Link Operational En Route Information Service
D-OTIS	Data Link Operational Terminal Information Service
D-RVR	Data Link Runway Visual Range
D-SIG	Data Link Surface Information and Guidance
D-SIGMET	Data Link Significant Meteorological Information
D-TAXI	Data Link Taxi Clearance
DAL	Design Assurance Level
DAMA	Demand Assigned Multiple Access
DBSy	QinetQ's Domain Based Security
DCDCU	Datalink Control and Display Unit
DCL	Departure Clearance
DCS	Data Certification Server
DCSSI	Direction Centrale de la Sécurité des Systèmes d'Information
DDoS	Distributed Denial of Service
DES	Data Encryption Standard
DGAC	Direction Générale de l'Aviation Civile
DGCIS	Direction Générale de la Compétitivité de l'Industrie et des Services
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services
DLL	Data Link Logon
DMZ	Demilitarized Zone
DSCP	Differentiated Services Code Point
DSEC	Datalink Security
DSNA-DTI	Direction des Services de la Naviguation Aérienne-Direction de la Technique et de l'Innovation
DSWG	Digital Security Working Group
DVB-RCS	Digital Video Broadcasting - Return Channel via Satellite
DYNAV	Dynamic Route Availability
EAL	Evaluation Assurance Level
EAP	Extensible Authentication Protocol
EASA	European Aviation Safety Agency

EBIOS	Expression des Besoins et Identification des Objectifs de Sécurité
ECAC	European Civil Aviation Commission
EDS	Electronic Distribution of Software
EFB	Electronic Flight Bag
ENGINE	Engine Performance Reports
ENR	En Route
ES	End Systems
ESA	European Space Agency
ESP	Encapsulating Security Payload
ETSI	
EUROCAE	European Organization for Civil Aviation Equipment
EUROCONTROL	European Organization for the Safety of Air Navigation
FAA	Federal Aviation Administration
FANS	Future Air Navigation System
FAST	Fiber-like Aircraft Satellite Telecommunications
FCA	Free Capacity Assignment
FCC	Federal Communications Commission
FDD	Frequency Division Duplex
FIPS	Federal Information Processing Standards
FL	Flight Level
FLINPINT	Flight Path Intent
FLIPCY	Flight Plan Consistency
FLTPLAN	Flight Plan Data
FLTSTAT	Flight Status
FMS	Flight Management system
FNISA	French Network and Information Security Agency
FREETXT	Free Text
FRS	Future Radio System
ft	feet
FUEL	Fuel Status
FUI	Fonds Unique Interministériel
GEO	Geostationary
GES	Ground Earth Station
GKEK	Group Key Encryption Key
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing
GRAA	Geographic Routing Protocol for Aircraft Ad Hoc Network
GTEK	Group Traffic Encryption Key
GUI	Graphical User Interface
H-NSP	Home Network Service Provider

HCRS	Hierarchical Certificate Revocation Scheme
HD	High Definition
HF	High Frequency
HLPSL	High Level Protocol Specification Language
HMAC	Hash based Message Authentication Code
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IAT	Inter-Arrival Time
IATA	International Air Transport Association
ICAO	International Civil Aviation Organization
IETF	Internet Engineering Task Force
IF	Intermediate Format
IFC	In Flight Connectivity
IFE	In Flight Entertainment
IFMEs	In-flight Medical Events
IFR	Instrumental Flight Rules
IKE	Internet Key Exchange
IMA	Integrated Modular Avionics
IP	Internet Protocol
IPS	Internet Protocol Suite
IPS	Intrusion Prevention System
IPSec	IP Security
IS	Intermediate Systems
ISAE	Institut Supérieur de l'Aéronautique et de l'Espace
ISAKMP	Internet Security Association and Key Management Protocol
ISO	International Organization for Standardization
ISSRM	Information System Security Risk Management
IT	Information Technology
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
JCG	Joint Coordination Group
JPEG	Joint Photographic Expert Group
KHz	Kilo Hertz
LAAS	Laboratoire d'Analyse et d'Architecture Systèmes
LANs	Local Area Networks
LCC	Logical Link Control
LDACS	L-band Digital Aeronautical Communication System
LED	Light Emitting Diode
LOADSHT	Load Sheet Request Transfer

MAC	Medium Access Control
MAINTPR	Maintenance Problem Resolution
MAINTRT	Real Time Maintenance Information
MANETs	Mobile Adhoc Networks
MARION	Méthodologie d'Analyse de Risques Informatiques Orientée par Niveaux
MASPS	Minimum Aviation System Performance Standards
MBRA	Multicast and Broadcast Rekeying Algorithm
MCDMA	Multi-Criteria Decision Making Algorithm
MEHARI	Méthode Harmonisée d'Analyse du Risque Informatique
MELISA	Méthode d'Evaluation de la Vulnérabilité Résiduelle des Sytèmes d'Armement
MHz	Mega Hertz
MiBs	Management Information Bases
MILS	Multiple Independent Layers of Security
MIT	Massachusetts Institute of Technology
MITM	Man In The Middle
MODCOD	Modulation Code
MPE	Multi Protocol Encapsulation
MPEG	Moving Picture Experts Group 2
MPLS	Multi Protocol Label Switching
MS	Mobile Station
MSC	Message Sequence Chart
NAS	National Airspace System
NCC	Network Control Center
NEWSKY	Networking the sky
NEXTGEN	Next Generation Air Transportation System
NGN	Next Generation Network
NIDS	Network Intrusion Detection System
NIS	Network Infrastructure and Security
NIST	National Institute of Standards and Technology
NM	Nautical Mile
NMS	Network Management System
NNC	Network Node Correlation
NONCE	Number used Once
NOP	Network Operations Report
NOTAM	Notice to Airmen
NVD	National Vulnerability Database
OCSP	Online Certificate Status Protocol
OCTAVE	Operationally Critical Threat Asset and Vulnerability Evaluation
OFMC	On the Fly Model Checker
OI	On-board Internet

OID	Object Identifier
OOB	Out-Of-Band
OPNET	Optimum Network Engineering Tool
ORP	Oceanic Remote and Polar
OSA	Operational Safety Assessment
OSI	Open Systems Interconnection
OSVDB	Open Source Vulnerability Database
PDU	Protocol Data Unit
PEP	Performance Enhancing Proxy
PGP	Pretty Good Privacy
PIAC	Peak Instantaneous Aircraft Count
PID	Priority IDentifier
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure
PKM	Privacy Key Management
POSRPT	Position Report
PPD	Pilot Preferences Downlink
PSL	Policy Security Level
QID	Queuing IDentifier
QoS	Quality of Service
R&D	Research and Development
RA	Registration Authority
RAM	Random Access Memory
RBDC	Rate Based Dynamic Capacity
RBI	Relative Balance Index
RCA	Root Certificate Authority
RCP	Required Communication Performance
RCTPs	Required Communication Technical Performances
RK	Registration and keys generation
RP	Reference Point
RSA	Rivest Shamir Adleman
RSS	RDF Site Summary
RTCA	Radio Technical Commission for Aeronautics
S/MIME	Secure Multi-purpose Internet Mail Exchange
S/N	Signal to Noise
SA	Security Association
SAC	Satellite Access Control
SANDRA	Seamless Aeronautical Networking through integration of Data links Radios and Antennas
SAP	Service Access Point
SAP	System Access Parameters
SARPs	Standards and Recommended Practices
SATCOM	Satellite Communication

SATEM	Satellite Emulator
SatIPSec	Satellite IPSec
SATMC	SAT based Model Checker
SATSIX	Satellite-based Communications Systems within IPv6 Networks
SBB	Swift Broad Band
SC-216	Special Committee 216
SCA	Subordinate Certificate Authority
SCVP	Simple Certificate Verification Protocol
SD	Satellite Dependent
SDR	Security Dependency Relation
SDU	Segment Data Unit
SecMan	Security Manager
SESAR	Single European Sky ATM Research
SHA-1	Secure Hash Algorithm
SI	Satellite Independent
SIP	Session Initiation Protocol
SITA	Société Internationale de Télécommunications Aéronautiques
SITS	Security Integrated Tool Set
SLA	Service Level Agreement
SLC	Satellite Link Control
SMAC	Satellite Medium Access
SMP	Security Manager Proxy
SNMP	Simple Network Management Protocol
SNSSP	Secure Negotiation of Supported Security Protocols
SOV	Security Objective Vector
SPAN	Security Protocol ANimator for AVISPA
SPD	Security Policy Database
SPKI	Simple Public Key Infrastructure
SPV	Security Protection Vector
SRQC	Service-based Risk Quantitative Calculation
SS	Subscriber Station
SSH	Secure SHell
SSID	Service Set IDentifier
SSL	Secure Socket Layer
SSP	Supported Security Protocols
SSPD	Supported Security Protocol Database
TA4SP	Tree Automata based on Automatic Approximations for the Analysis of Security Protocols
TBTP	Terminal Burst Time Plan
TCP	Transmission Communication Protocol
TELNET	Terminal Network
TEN-T EA	Trans-European transport Network Executive Agency
TLS	Transport Layer Security
TMA	Terminal Maneuvering Area

TTD	Time Division Duplex
TYPHON	Telecommunications and Internet Protocol Harmonization Over Networks
UBR	Unspecified Bit Rate
UDP	User Datagram Protocol
V-NSP	Visited Network service Provider
VDL	Very High Frequency DataLink
VHF	Very High Frequency
VLAN	Virtual Local Area Network
WAPs	Wireless Access Points
WATS	World Air Transport Statistics
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless LAN
WSNs	Wireless Sensor Networks
WXGRAPH	Graphical Weather Information
WXRT	Real Time Weather Reports for Met Office
WXTEXT	Textual Weather Reports
XML	Extensible Markup Language

Résumé de Thèse

1 Motivation et Problématique

Les communications aéronautiques sont appelées à évoluer dans les années à venir. Actuellement, la voix analogique reste le moyen principal pour communiquer entre l'avion et le sol dans des bandes de fréquences réservées. Néanmoins, plusieurs organismes internationaux prédisent une migration imminente vers des communications numériques dans un futur proche. Par exemple, EUROCONTROL (Organisation européenne pour la sécurité de la navigation aérienne) et la FAA (Federal Aviation Administration - USA) ont rédigé en partenariat le COCR (Communications Operating Concept and Requirements for the Future Radio System) qui est un document technique recensant, entre autres, les nouveaux services cockpit (ATS - Air Traffic Services) et compagnie (AOC - Aeronautical Operational Control Services).

Ces services basés sur des communications de données devraient remplacer les communications voix progressivement dans les années à venir. De plus, les compagnies aériennes seront certainement amenées à déployer de nouveaux services passagers (APC - Aeronautical Passenger Communication Services), comme l'Internet cabine par exemple, mais aussi des services AOC "nouvelle génération" comme la télé médecine [1] ou la télé-surveillance [2].

Avec une telle diversité des flux, le trafic air-sol devient de plus en plus dense et hétérogène. Il apparaît donc opportun de mixer ces flux afin d'offrir une meilleure utilisation des ressources. Dans la perspective de faciliter l'interopérabilité entre les différents réseaux concernés, le déploiement d'un réseau de communications aéronautiques tout-IP (Internet Protocol) devient une évidence pour le futur de l'ATM (Air Traffic Management). Afin de faciliter cette interconnexion entre le réseau ATN (Aeronautical Telecommunications Network) et les autres réseaux terrestres basés sur IP, l'OACI (Organisation de l'Aviation Civile Internationale) a permis l'utilisation d'ATN avec la suite protocolaire IPS (Internet Protocol Suite). L'implémentation de l'ATN/IPS permettra non seulement une meilleure adéquation entre les réseaux de communications terrestres et aéronautiques, mais aussi l'utilisation de protocoles et de standards matures et bien connus de la communauté industrielle (COTS - Commercial Off The Shelf).

A long terme, les communications aéronautiques numériques présentent donc un potentiel considérable. Leur utilisation systématique permet d'envisager des gains importants en fiabilité, en coût, et en qualité de transmission. Elles permettent également des échanges plus riches et plus fréquents entres les systèmes

[1] Système permettant de prodiguer des soins médicaux à distance et l'échange d'informations médicales.

[2] Système de surveillance à distance.

informatiques sol et les systèmes avioniques embarqués.

2 Considérations de Sécurité

Inévitablement, l'industrie devra tôt ou tard faire face aux conséquences directes d'une telle mutation. La sécurité des communications devient une priorité à gérer dans un contexte aussi critique : une attaque pourra non seulement compromettre les données échangées entre l'avion et le sol, mais aussi mettre en danger la vie des personnes à bord (suite à une attaque de déni de service qui empêcherait le pilote de communiquer avec le contrôleur aérien par exemple). Une architecture de sécurité devra donc être rigoureusement définie afin d'assurer confidentialité, intégrité et authentification des données échangées.

Cette architecture devra aussi tenir compte d'autres contraintes propres aux communications aéronautiques. Dans un environnement aussi imprévisible, l'état du réseau reste très fluctuant et dépend de plusieurs paramètres exogènes tels que les performances intrinsèques de certaines technologies utilisées pour communiquer de l'avion vers le sol (comme le débit par exemple). De plus, la politique de sécurité appliquée devra aussi gérer les priorités qui existent entre des flux relatifs à la sûreté du vol (ATS) et le reste des communications. Dans un système aux conditions aussi variables, une politique de sécurité statique sera coûteuse et inadéquate au contexte et aux contraintes qu'impose cet environnement complexe.

Par ailleurs, l'augmentation du trafic aérien ainsi que le nombre de passager à bord aura aussi un impact sur l'architecture de gestion de clé à déployer. En effet, les communications ATS, AOC, et APC requièrent l'utilisation d'outils cryptographiques qui devront être négociés au préalable (*e.g.* certificats, clés publiques/privées, clés de session, etc). D'une manière générale, la gestion des clés dans les réseaux à grande échelle nécessite la mise en place d'une Infrastructure à Clés Publiques (PKI pour Public Key Infrastructure) qui offre toutes les fonctionnalités de base nécessaires comme la distribution des clés, la révocation des certificats expirés, ou la vérification de la validité des entités cryptographiques. En environnement ATM, le déploiement d'une PKI doit tenir compte de plusieurs facteurs pour peuvent influencer de près ou de loin les performances : réseau à large échelle, hétérogénéité des entités impliquées, trafic de signalisation à minimiser, ressources réseaux limitées.

Aussi, le déploiement de technologies sur étagère nécessitent une évaluation constante et rigoureuse des vulnérabilités qui pourront être exploitées sur le réseau. Ainsi, la méthodologie d'analyse et d'estimation du risque à utiliser pour cette fin devra s'adapter au contexte des réseaux aéronautiques de données (corrélation entre les nœuds, estimation quantitative du risque, impact des vulnérabilités sur les nœuds et le réseau dans sa totalité).

Auparavant, quelques solutions pour sécuriser les échanges air-sol ont été proposées, comme l'AMS (ACARS Message Security) pour l'ACARS[3] (Aicraft Communications Addressing and Reporting System). Néanmoins, ces solutions ne prennent pas en compte les contraintes liées à l'interopérabilité entre réseaux hétérogènes, la priorité entre domaines, ou les ressources réseaux limitées par exemple.

[3] Système de communication entre l'avion et le sol par liaison radio de très haute fréquence ou satellite.

Cette thèse présente trois contributions dans le domaine de la sécurité des futures communications aéronautiques de données, à savoir :

1. **Une méthodologie quantitative d'analyse du risque lié à la sécurité réseau :** cette méthodologie repose sur deux idées fondamentales. La première est la quantification des paramètres au sein de la méthodologie comme l'impact que peut avoir une attaque sur un nœud (les données sont récupérées à partir d'une base des vulnérabilités publique appelé NVD (National Vulnerability Database)) ou la probabilité occurrence d'une attaque en fonction des possibilités offertes à l'intrus. La deuxième est la propagation du risque entre les différents nœuds qui constituent le réseau. Cette propriété est très importante dans la mesure ou elle permet de faire ressortir les liens qui permettent à l'attaquant de profiter de la corrélation qui existent entre les nœuds les plus vulnérables du réseau. Les paramètres utilisés dans la méthodologie peuvent être groupés en trois catégories :
   (a) *Les paramètres liés au risque* : dans cette famille, on distingue 4 type de risque, à savoir :
      - Le risque global par nœud : produit de la valeur du nœud avec la somme du risque individuel et du risque propagé par nœud. La valeur d'un nœud prend en considération le degré de corrélation, la valeur fonctionnelle et la priorité de la classe du trafic véhiculé ;
      - le risque individuel par nœud : prend en considération les vulnérabilités intrinsèques de chaque nœud. Les vulnérabilités sont extraites de la base des vulnérabilités NVD. La formule du risque utilisée est celle classique, à savoir produit de la probabilité d'occurrence d'une attaque et de l'impact qu'elle peut causer ;
      - le risque propagé par nœud : prend en considération la corrélation entre les nœuds (exemple : nombre de flux échangées). La formule est similaire à celle utilisée pour le risque individuel, sauf qu'ici on utilise la probabilité de corrélation ainsi que l'impact due à la propagation d'une attaque d'un nœud à un autre ;
      - le risque réseau : qui est quantifié tout simplement en tant que la somme de tous les risques globaux des nœuds.

   (b) Les paramètres liés à l'impact d'occurrence d'une attaque : dans cette famille, on distingue 2 types d'impacts différents, à savoir :
      - l'impact causé par une vulnérabilité intrinsèque au nœud : ce paramètre correspond au score CVSS associé dans la base NVD ;
      - l'impact de corrélation : qui prend en considération la valeur du nœud affecté ainsi l'impact de la vulnérabilité propagée depuis le nœud source.

   (c) *Les paramètres liés à la probabilité d'occurrence d'une attaque* : dans cette famille, on distingue 2 type de probabilités :
      - la probabilité d'occurrence d'une vulnérabilité intrinsèque au nœud : ce paramètre est quantifié en tant que rapport entre la motivation que possède un attaquant pour exploiter l'attaque et les difficultés techniques qu'il peut rencontrer ;
      - la probabilité de corrélation :égal au rapport entre le nombre de flux échangés entre les deux nœuds sur le nombre de flux total.

   L'algorithme d'estimation du risque que nous proposons est le suivant :

```
$V \leftarrow \{\varnothing\}$; //initiate a set of vulnerable nodes
$NV \leftarrow \{\varnothing\}$; //initiate a set of processed nodes
for all $i \in network$ do
    $Risk_i^- \leftarrow 0$;
    $Risk_i^+ \leftarrow 0$;
    $Risk_i \leftarrow 0$;
    $C_i \leftarrow \{\varnothing\}$; //initiate a set of correlated nodes with node i
end for
for all $i \in network$ do
    identify vulnerabilities;
    $Value_i \leftarrow n_i * FunctionValue_i * ClassValue_i$;
    if any vulnerability is detected then
        add node $i$ to $V$;
    end if
    for all vulnerability t do
        store $t$ and associated $CVSS$ score;
    end for
end for
for all node $i \in V$ do
    store correlated nodes with node $i$ in $C_i$;
    for all vulnerability $t$ do
        $TechnicalDifficulty_t(i) \leftarrow S_i + B_t$;
        $Motivation_t(i) \leftarrow Value_i * T_i$;
        $P_t(i) \leftarrow Motivation_t(i)/TechnicalDifficulty_t(i)$;
        $Risk_i^- \leftarrow Risk_i^- + (P_t(i) * I_t(i))$;
    end for
end for
while $V \neq \{\varnothing\}$ do
    for all $j \in V$ do
        for all $i \in C_j$ do
            for all vulnerability $t$ do
                $\sigma(i,s) \leftarrow SPV_i * (SOV_s)^{tr}$;
                $I_t(i,j) \leftarrow Value_i * \sigma(i,s) * I_t(j)$; //s is the targeted service by t
                $P(i,j) \leftarrow f_{ij}/F_{ij}$;
                $TechnicalDifficulty_t(j) \leftarrow S_j + B_t$;
                $Motivation_t(j) \leftarrow Value_j * T_j$;
                $P_t(i) \leftarrow Motivation_t(i)/TechnicalDifficulty_t(i)$;
                $P_t(i,j) \leftarrow P_t(j) * P(i,j)$;
                $Risk_i^+ \leftarrow Risk_i^+ + (P_t(i,j) * I_t(i,j))$;
                $P_t(i) \leftarrow P_t(i) + P_t(i,j)$; //update the likelihood of threat
                if $P_t(i) > 1$ then
                    $P_t(i) \leftarrow 1$; //the likelihood of threat should not exceed 1
                end if
            end for
            if node $i \notin Vand \notin NV$ then
                store node $i$ in $V$; //the node is now vulnerable
            end if
        end for
        copy node $j$ to $NV$ and remove it from $V$; //this node has been processed
    end for
end while
for all node $i \in network$ do
    $Risk_i \leftarrow Value_i * (Risk_i^- + Risk_i^+)$;
end for
for all node $i \in network$ do
    $Risk_{net} \leftarrow Risk_{net} + Risk_i$;
end for
```

La méthodologie est validée puis testée dans le cadre du projet Européen SESAR qui vise à moderniser le ciel Européen ainsi que les technologies CNS utilisées dans les futures infrastructures aéronautiques. Le réseau testé utilise la technologie AeroMACS, qui s'inspire fortement du WiMAX IEEE 802.16, technologie sans fil similaire au WIFI mais utilisée pour les réseaux métropolitains à plus large échelle (WMAN). L'architecture protocolaire est similaire à celle du WiMAX avec une sous-couche sécurité au niveau MAC :

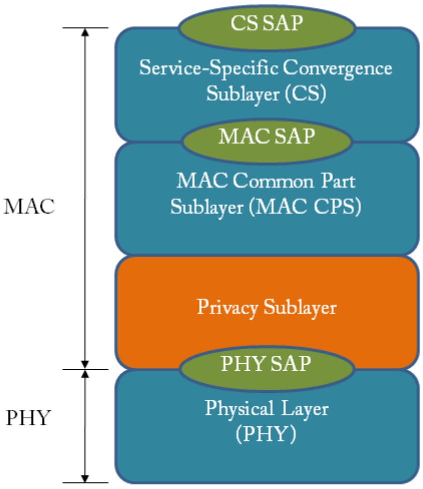

CS SAP
Service-Specific Convergence Sublayer (CS)
MAC SAP
MAC Common Part Sublayer (MAC CPS)
Privacy Sublayer
PHY SAP
Physical Layer (PHY)
MAC
PHY

Figure 1 – Architecture Protocolaire de l'AeroMACS

Cette sous-couche offre plusieurs services de sécurité comme la gestion de clés à travers PKMv2. Néanmoins, plusieurs vulnérabilités ont été relevées dans des études récentes : le risque lié à la sécurité réseau est donc plus au moins élevé selon la configuration faite à ce niveau au préalable par l'administrateur réseau. Ainsi, la méthodologie d'estimation de risque proposée permet d'analyser le risque inhérent à plusieurs scénarios et servira à long terme comme un moyen d'aide à la décision pour les administrateurs sécurité/réseau. L'architecture retenue pour la partie validation se compose de deux réseaux logiques :

(a) CSN (Connectivity Service Network) qui est le réseau de cœur de l'AeroMACS comportant les services et assurant l'interface avec les clients. Les différents serveurs déployés sont :
   - un serveur pour les communications ATS ;
   - un serveur pour les communications AOC ;
   - un serveur pour les communications portuaires ;
   - un serveur d'authentification AAA ;
   - un serveur d'attribution d'adresse DHCP.

(b) ASN (Access Service Network) qui est le réseau d'accès composé d'une passerelle AeroMACS, les stations de bases connectées aux stations mobiles (avions et véhicules de surface).

Les résultats ont été interprétés comme des recommandations pour le futur

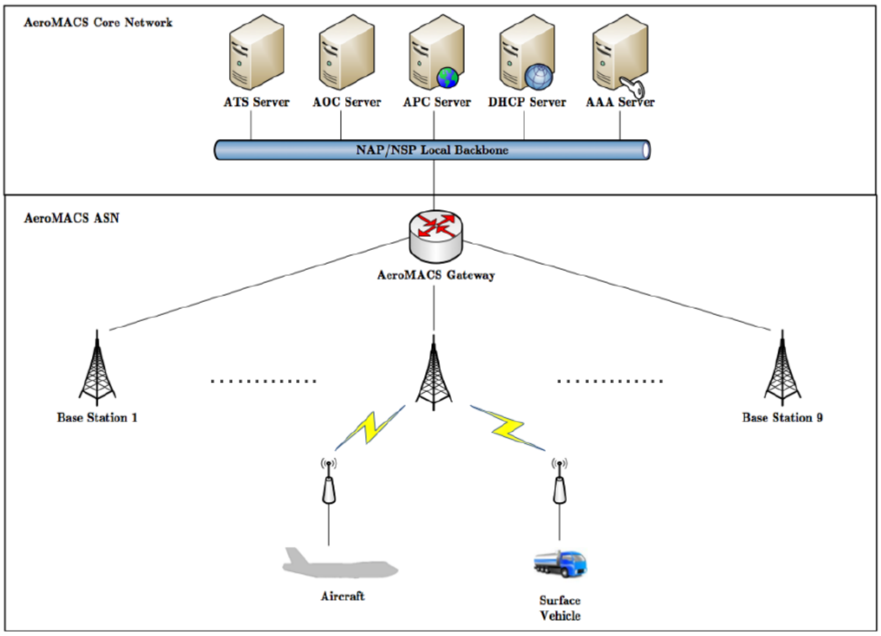


FIGURE 2 – Architecture du Système AeroMACS

déploiement de l'AeroMACS. Ces recommandations ont été regroupées ont trois catégories :

(a) Les recommandations relatives à la topologie du réseau : L'observation majeure résultant de la campagne de simulation est le fait que la passerelle AeroMACS est le talon d'Achille de la topologie à cause d'un degré de corrélation élevé. La solution serait de déployer plus d'une passerelle ASN (deux ou plus) et de distribuer les stations de base ainsi que les stations mobiles afin d'obtenir un risque propagé moins élevé, et par conséquent un risque réseau faible ;

(b) Les recommandations relatives à la sécurité du réseau : Un sous scénario de simulation consistait à comparer deux techniques d'authentification d'AeroMACS, à savoir RSA et EAP. Les résultats obtenus ont montré qu'utiliser EAP permet d'avoir un risque beaucoup moins élevé qu'en utilisant RSA ;

(c) Les recommandations relatives à l'implémentation de l'AeroMACS : Les simulations ont montré que les choix d'implémentation faits principalement sur les produits COTS utilisés (comme les types de serveurs AAA ou DHCP) ont un impact considérable sur les résultats obtenus. Ainsi, les serveurs à privilégier sont ceux qui ont un minimum de vulnérabilités exploitables mais aussi un score CVSS bas.

2. **Une architecture de gestion adaptative de la sécurité :** dans le cadre du projet Aerospace Valley FAST (Fiber-like Aircraft Satellite Telecommunications), une architecture de gestion adaptive de la sécurité a été proposée. Le composant central de cette architecture appelée SecMan (Se-

curity Manager) est une passerelle de sécurité déployée à bord de l'avion et qui permet d'adapter le niveau de sécurité en fonction de la classe du trafic ainsi que des ressources réseaux disponibles. L'architecture a été testée puis validée lors de la phase finale d'intégration du projet. L'architecture globale du système est composée d'un segment bord, un segment sol, et un segment air-sol.

Au niveau du sol, une Gateway (GW) est connectée à deux routeurs : un routeur ATN pour le réseau aéronautique, et un routeur Internet pour les services APC destinés aux passagers. Pour le réseau à bord, deux routeurs sont connectés au terminal satellite : un routeur ATN/IPS pour l'ATS et un routeur NG (Next Generation) pour les services AOC et APC. Les normes DVB (DVB-S2 pour le lien aller et DVB-RCS pour le lien retour) ont été retenues comme méthodes d'accès pour le lien satellitaire du projet FAST.

Au niveau du domaine APC, plusieurs points d'accès WIFI seront dispatchés à travers l'avion afin d'assurer une disponibilité continue du service Internet cabine et télé-médecine, qui requiert une mobilité absolue. Une connexion multi-SSID (Service Set IDentifier) permet de séparer les deux types de trafics pour des raisons évidente de sécurité (chaque application aura sa propre clé pour se connecter) et de disponibilité des ressources, surtout quand une application aussi critique que la télé-médecine veut se connecter à un des points d'accès de l'avion.

Au niveau du routeur NG, un premier niveau de QoS IP est mise en oeuvre avec Diffserv[4] (Differentiated services) en associant chaque domaine connecté au routeur NG à une file d'attente donnée (télé-médecine, télé-surveillance, services AOC "standards" et Internet cabine).

Au niveau du terminal satellite, deux ports physiques sont considérés pour connecter le routeur ATN/IPS (flux ATS) et le routeur NG. Ainsi, la séparation entre les services cockpit ATS et le reste du trafic est assurée conformément aux normes et pratiques recommandées par l'OACI. L'architecture retenue est basée sur la norme DVB-RCS comme décrit dans l'architecture de référence BSM (Broadband System Multimedia) de l'ETSI (European Telecommunications Standards Institute).

A l'entrée du terminal satellite, un classifier IP est mis en place afin de différencier les flux IP provenants des deux routeurs. Une phase de "mapping" est ainsi réalisée afin de gérer les priorités entre les flux grâce à plusieurs niveaux de priorité PID (Priority ID), assignés à chaque classe de service et envoyés vers la file d'attente correspondante identifiée grâce à un QID (Queuing ID). De cette façon, la séparation entre les couches dites "satellite dépendant" (SD) et "satellite indépendant" (SI) est assurée conformément au modèle BSM de référence.

Une seconde version de l'architecture système à bord est ensuite présentée, cette fois-ci en environnement sécurisé grâce au module de gestion de la sécurité (SecMan). L'interconnexion des différents composants sera décrite et les principes de fonctionnement y sont introduits. Deux relais sécurité (SMP - SecMan Proxy) sont considérés et connectés respectivement au routeur ATN et au routeur NG. Chaque proxy est isolé dans une zone

[4] Architecture réseau qui spécifie un mécanisme pour classer et contrôler le trafic tout en fournissant un qualité de service.

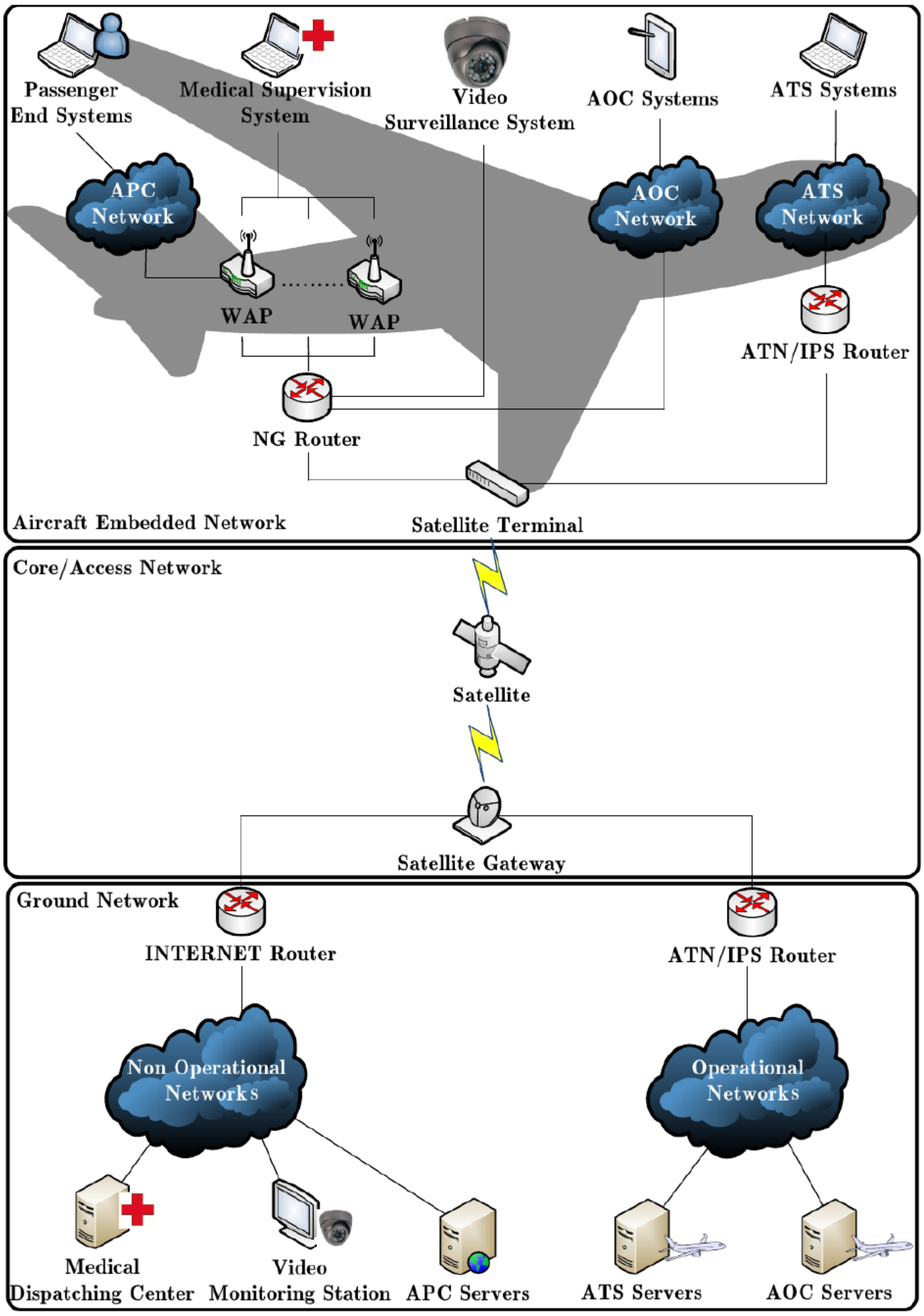
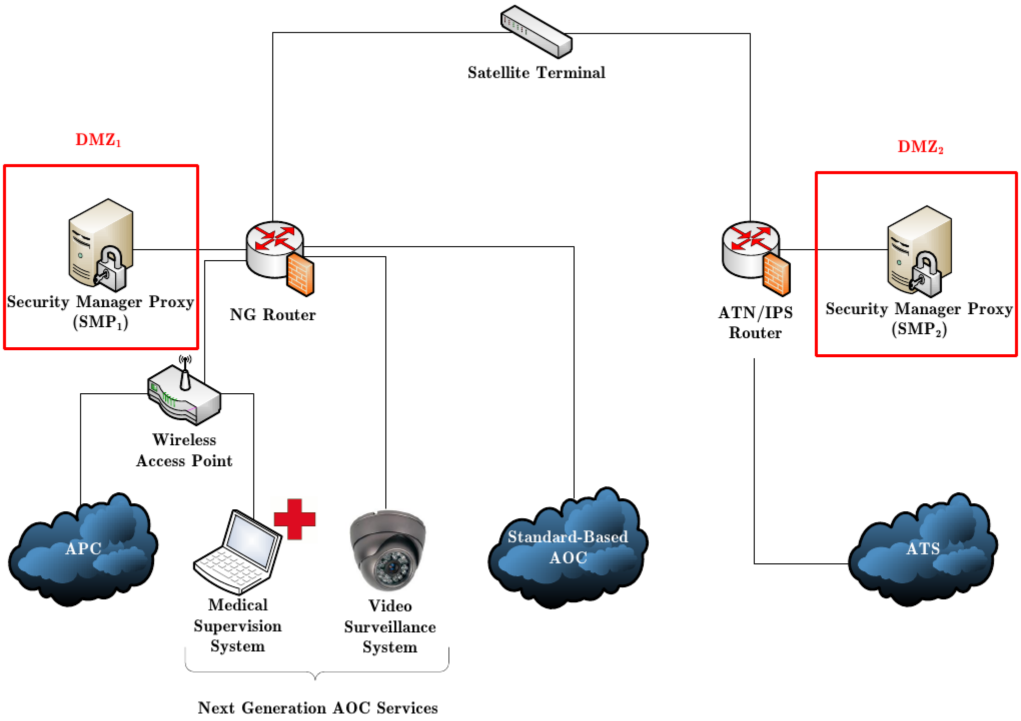


FIGURE 3 – Architecture Système

démilitarisée DMZ (Demilitarized Zone) grâce à des fonctions de pare-feu (Firewalling) implantées au niveau des routeurs.
Les pare-feux retenus sont de type Stateful Inspection, réputés assez ro-

bustes et efficaces contre les attaques de déni de services (DoS - Denial of Service), contrairement aux pare-feux Stateless, reposants sur des règles et des listes d'accès trop simples pour éviter des tentatives d'intrusions évoluées. Les pare-feux applicatifs (niveau 7) peuvent être une alternative intéressante, mais le traitement et le calcul qu'ils engendrent ralentissent considérablement les échanges, ce qui va à l'encontre d'une politique de sécurité adaptative et visant justement à améliorer les performances réseaux.


Satellite Terminal
DMZ1
DMZ2
Security Manager Proxy (SMP1)
NG Router
ATN/IPS Router
Security Manager Proxy (SMP2)
Wireless Access Point
APC
Medical Supervision System
Video Surveillance System
Standard-Based AOC
ATS
Next Generation AOC Services

Afin d'avoir une politique de sécurité flexible, SecMan est capable de fonctionner en deux modes :

1)*Mode Intra-classe* : ce mode est utilisé quand les flux appartiennent à la même classe. Par conséquent, la priorité entre les flux n'est pas prise en compte et la politique de sécurité s'adapte en fonction des ressources réseaux disponibles. SecMan fonctionne en mode Intra-classe sur le proxy sécurité connecté au routeur ATN/IPS car seul le trafic ATS est considéré (SMP en DMZ_1),

2)*Mode Inter-classe* : ce mode est utilisé quand les flux appartiennent à plusieurs classes de trafic. La politique de sécurité s'adapte donc en fonction des ressources réseaux et des priorités entre flux issus de domaines différents. SecMan fonctionne en mode Inter-classe sur le proxy sécurité connecté au routeur NG (SMP en DMZ_2).

De plus, plusieurs modes de sécurisation peuvent être appliqués en adaptant la politique de sécurité au besoin, quelque soit le mode de fonctionnement de SecMan(Inter-classe ou Intra-classe) :

1) *Mode transparent non sécurisé* : les paquets qui transitent sur les rou-

teurs sont simplement routés sans aucun mécanisme de sécurité (tout en respectant les contraintes de QoS),

2) *Mode transparent sécurisé* : il sécurise les paquets échangés à l'aide d'un mécanisme tel que IPSec (IP Security) et est transparent pour l'utilisateur,

3) *Relais au niveau transport* : SMP sécurise les connexions de bout en bout (HTTPS par exemple),

4) *Relais applicatif* : SMP se comporte dans ce mode comme un proxy applicatif "classique" (proxy http ou ftp par exemple).

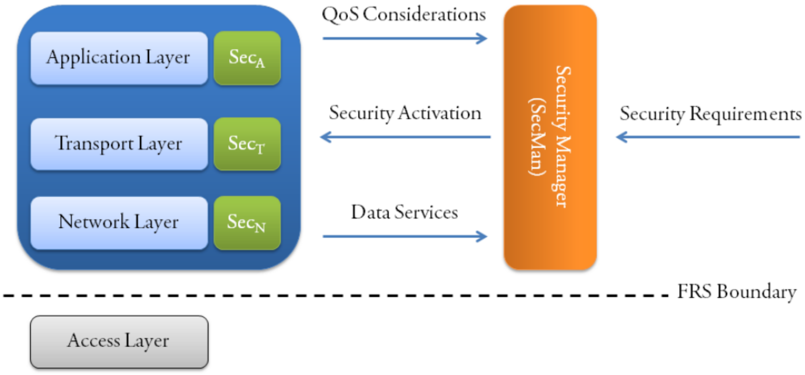


Figure 4 – Architecture Protocolaire de SecMan

Comme il a été décrit précédemment, compte tenu des services à sécuriser et de l'intérêt de converger vers un réseau aéronautique tout-IP, il apparaît pertinent de porter l'étude sur un réseau IP. Il est à noter que le module gère les mécanismes de sécurité des couches hautes de la pile TCP/IP, au-dessus de la limite FRS (Future Radio System) définie dans le COCR. Le module est informé de l'évolution constante des ressources réseaux disponibles à travers un mécanisme cross-layer [5]. Il prend également en compte les besoins de sécurité exprimés en amont à travers une phase d'évaluation des risques pour les différents flux en entrée de SMP. Plusieurs routines d'activation des mécanismes de sécurité sont aussi créées au travers de primitives telles que *ChangeCipher* pour modifier l'algorithme de chiffrement ou *ByPass* pour ne pas activer un mécanisme de sécurité donné. Un des composants les plus importants de SecMan est sans doute le décideur. Le décideur est l'algorithme qui va permettre la prise de décision vis à vis de la politique de sécurité à appliquer. A cette fin, les algorithmes d'aide à la décision multi-critères (MCDMA - Multi Criteria Decision Making Algorithm) sont adaptés surtout dans un système aussi riche et complexe. Il s'agit de méthodes et de calculs permettant de choisir la solution optimale parmi tout un ensemble de solutions en essayant de prendre en compte les exigences des acteurs et leur comportement à la fois dans un cadre de processus de décision "humain" et dans un cadre de processus de décision "automatique". AHP (Analytic Hierarchical Process) fait partie des méthodes MCDMA les plus efficaces et les plus appréciées notam-

[5] Mécanisme d'optimisation des échanges inter-couches afin d'améliorer le débit et de réduire la latence.

ment grâce à sa manière simple et structurée de résoudre le problème (un but, des critères de décision, et des alternatives). Une étape de comparaison par paires, permettant de pondérer les poids d'importances entre les critères, précède une suite de calcul matriciel qui aboutit finalement à un classement des alternatives listées.

Afin d'adapter la méthode au contexte particulier des mécanismes de sécurité, une liste non exhaustive de métriques pouvant servir à la comparaison a été établie. Ces critères sont des caractéristiques inhérentes aux différents algorithmes constituants les protocoles de sécurité (taille de la clé, taille de l'empreinte, nombre de rondes). Ces protocoles ont été au préalable négocié entre l'avion et l'entité sol grâce à une phase de négociation sécurisée. Au terme de la négociation, une table des mécanismes supportés est stockée dans une base de donnée embarquée appelée SSPD (Supported Security Protocol Database). Ainsi, les protocoles de sécurité sont évalués selon leur robustesse et leur impact en terme de ressources systèmes et réseaux.

L'architecture de sécurité présentée possède plusieurs avantages :

1) *Consommation des ressources réseaux optimisée* : l'optimisation du niveau de sécurité permet de diminuer le surplus induit par les mécanismes de sécurité et de fait, la consommation des ressources réseaux est mieux gérée,

2) *Gestion des priorités* : la priorité entre les différents services relatifs à l'ATS, AOC et APC est gérée,

3) *Politique de sécurité sélective et multi-couches* : SecMan est capable d'activer plusieurs mécanismes de sécurité opérants sur une ou plusieurs couches de la pile protocolaire TCP/IP en fonction des besoins identifiés afin de maximiser la robustesse offerte,

4) *Compatibilité et interopérabilité accrues* : SecMan est tout à fait compatible avec n'importe quel réseau IP utilisant une technologie autre que le satellite au niveau des couches d'accès réseaux car les mécanismes de sécurité sont gérés au-dessus de la limite FRS du COCR. Il est aussi notable que les concepts d'interconnexion du système présenté peuvent être appliqués hors contexte aéronautique.

3. **Une infrastructure à clé publique adaptée au contexte aéronautique :** la PKI développée dans cette troisième contribution repose sur trois niveaux différents. Le premier concerne la certification croisée entre les autorités de certification (CA) mères déployées par les compagnies aériennes. Le deuxième niveau concerne les échanges entre les CA mères au sol et les CA subordonnées déployées à bord de chaque avion. Le troisième niveau concerne les échanges entre les CA subordonnées et les entités finales (passager, système à bord, etc). La PKI a été implémentée puis testée en utilisant des statistiques de vol issues de la base de donnée de la DSNA-DTI. La PKI hiérarchique proposée permet de diminuer le trafic de signalisation entre l'avion et le sol, et permet par conséquent d'économiser les ressources réseaux du lien air-sol. Les scénarios testés prennent en considération la position géographique du possesseur du certificat et de l'entité qui vérifie la validité du dit certificat, à savoir à bord de l'avion ou au sol. Les coûts réseau et système correspondants à chaque scénario ont été quantifié d'une manière analytique puis extrapolés aux données

des avions de la DSNA-DTI.

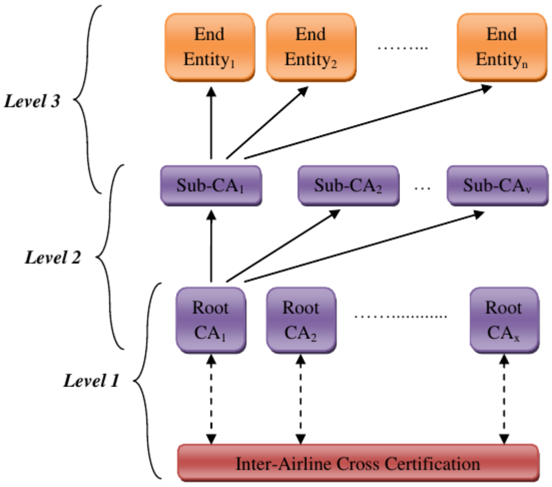


Figure 5 – PKI Hiérarchique pour les futures communications aéronautiques