Electronic Distribution of Airplane Software and the Impact of Information Security on Airplane Safety

Richard Robinson[1], Mingyan Li[1], Scott Lintelman[1], Krishna Sampigethaya[2], Radha Poovendran[2], David von Oheimb[3], Jens-Uwe Bußer[3], and Jorge Cuellar[3]

[1] Boeing Phantom Works, Box 3707, Seattle, WA 98124, USA
{richard.v.robinson,mingyan.li,scott.a.lintelman}@boeing.com
[2] Network Security Lab, University of Washington, Seattle, WA 98195, USA
{rkrishna,rp3}@u.washington.edu
[3] Siemens Corporate Technology, Otto-Hahn-Ring 6, 81730 München, Germany
{david.von.oheimb,jens-uwe.busser,jorge.cuellar}@siemens.com

Abstract. The general trend towards ubiquitous networking has reached the realm of airplanes. E-enabled airplanes with wired and wireless network interfaces offer a wide spectrum of network applications, in particular electronic distribution of software (EDS), and onboard collection and off-board retrieval of airplane health reports. On the other hand, airplane safety may be heavily dependent on the security of data transported in these applications. The FAA mandates safety regulations and policies for the design and development of airplane software to ensure continued airworthiness. However, data networks have well known security vulnerabilities that can be exploited by attackers to corrupt and/or inhibit the transmission of airplane assets, i.e. software and airplane generated data. The aviation community has recognized the need to address these security threats. This paper explores the role of information security in emerging information technology (IT) infrastructure for distribution of safety-critical and business-critical airplane software and data. We present our threat analysis with related security objectives and state functional and assurance requirements necessary to achieve the objectives, in the spirit of the well-established Common Criteria (CC) for IT security evaluation. The investigation leverages our involvement with FAA standardization efforts. We present security properties of a generic system for electronic distribution of airplane software, and show how the presence of those security properties enhances airplane safety.

1 Introduction

Safety concerns with airplane software have been extensively studied [10], [15], [17]. The FAA stresses the criticality of some of the software onboard airplanes through well established guidance assuring their proper design and development for continued airworthiness, e.g. RTCA/DO-178B [1] Level A safety-critical software. Yet the guidance does not even address the issue of software distribution and its security. Nowadays, airplane software is still distributed manually using disks and other storage media, and since security is not a primary objective, embedded systems onboard airplanes check (using CRCs) only for accidental modifications of software to be loaded. However, the proposed use of networks to distribute software electronically from ground to onboard systems raises unprecedented challenges to ensuring airworthiness [7], [11]. In particular, while the electronic distribution of

software (EDS) reduces overhead and improves efficiency and reliability of airplane manufacturing, operation and maintenance processes, these benefits come only at the cost of exposing the airplane to potential attacks, in particular via data networks. The FAA has recognized that current guidance and regulations for airplane software do not cover the requirements needed to address these vulnerabilities [4], [5].

1.1 Safety vs. Security

Although information security requirements are warranted, assessing their impact on airplane safety is non-trivial. It is clear from the established FAA guidance in [1] and elsewhere that the regulatory community is concerned about assuring the design and implementation of certain software components and that they consider that safety may be affected if such components were to become corrupted. Therefore, vulnerabilities in an EDS may present opportunities for attackers seeking to directly lower airplane safety, e.g. by corrupting safety-critical software distributed onboard, or to impede usability of onboard systems, e.g. by corrupting less critical software such as DO-178B [1] Level D. One must assume that international terrorists, as well as criminals pursuing economic damage, are capable today of employing advanced technologies for attacks. Thus it is now necessary to assess the impact of information security attacks against airplane safety and to develop strategies for mitigating the associated vulnerabilities. There is a body of literature that presents arguments for commonality among the safety and security disciplines [8], [9], [12], [16], but it remains an open question how to integrate the two fields. While indeed security affects safety, it is not clear how to express the relevant security considerations, and how to accommodate security risks and mitigations in the context of a safety analysis. There exist as yet no formal or agreed guidelines for certifying or assessing safety critical systems together with their security needs. In particular two questions remain open:

- How to integrate the mainly discrete methods applied in security analysis into the quantitative, probabilistic approaches typical of reliability analysis?
- How to combine the analysis of security, which refers to non-functional properties, with the functional SW correctness analysis in order to achieve a defined overall system safety level?

We believe that more research in this area is needed. Besides our necessarily limited contributions, we would like to benefit from any scientific advances there.

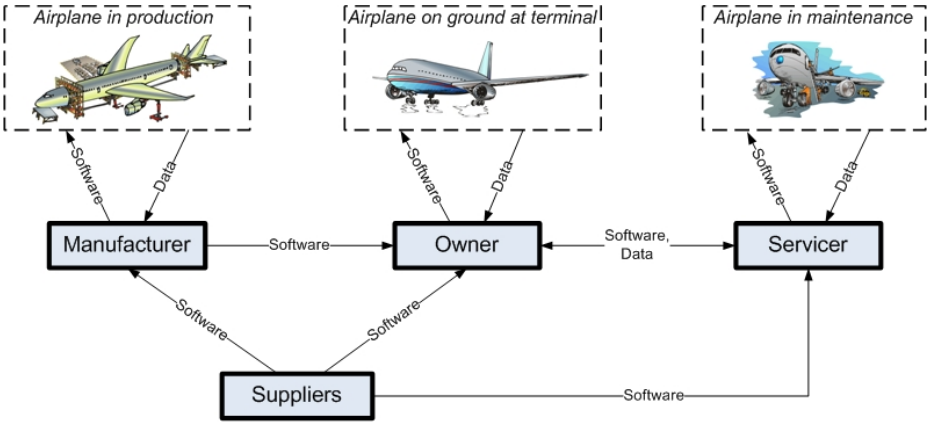
1.2 Our Contributions

The contributions of this paper are two-fold.

- We present security requirements for a generic EDS system, called Airplane Assets Distribution System (AADS). Our approach is based on the Common Criteria (CC) [3] methodology, identifying threats to AADS from an adversary attempting to lower airplane safety, deriving objectives to cover the threats, and stating functional requirements to cover the objectives.
- We assess the implications of information security threats on airplane safety. Our approach is based on the Information Assurance Technical Framework [2] to analyze the CC Evaluation Assurance Level (EAL) necessary and sufficient to address threats against the integrity of software of highest criticality.

2 Airplane Assets Distribution System (AADS)

The electronic distribution of airplane *information assets*, i.e. software and data, can be modeled by a generic system that we call the Airplane Assets Distribution System (AADS). Figure (a) illustrates the AADS model with the entities and flow of assets. Not all entities interact directly with all others. Note that functional overlaps are possible, with a single entity assuming roles of multiple entities, e.g. an airplane manufacturer can be a supplier for some software. The nature and content of interactions change depending on the lifecycle state of a specific airplane, which can be: in development, assembly, testing, use, resale, etc. The responsibility of the AADS for an asset begins when it takes over the asset from its producer, e.g. supplier or airplane, and ends when it delivers the asset at its destination, i.e., embedded systems such as a Line Replaceable Unit (LRU) in an airplane, or at the consumer of airplane-generated data. The path between the producer and the destination of the asset is referred to herein as the *end-to-end path*. Each of the links in this path must fulfill the security objectives given in Section 3.



(a) Airplane Assets Distribution System (AADS)

2.1 Assumptions

Processes in each entity in the AADS are assumed to be operating as designed and expected. In particular, the AADS is assumed to be administered in a proper way. Access privileges must be assigned and managed appropriately at each entity. Passwords and private keys are kept secret, and certificates are properly managed and protected. Each supplier is accountable to produce safety-assured software as per [1]. The networks used for asset distribution are assumed to be robust against well known denial of service attacks. It is worth noting that software distribution via physical media is generally adequate to meet requirements for timely software delivery to an aircraft. Finally, it is assumed that airplane owners are capable to manage the software configurations of airplanes reliably and correctly, and that airplanes produce status information accurately.

2.2 Adversary Model

An adversary in the AADS model is assumed to be capable of passive network traffic analysis as well as active traffic manipulation, node impersonation, and insider attacks. The objective of the adversary is to actually lower the safety margins of airplanes (as in the case of international terrorists) and/or to induce safety *concerns* and disturb business (as would be expected of sophisticated hackers or international criminal organizations).

For purposes of the present analysis, we consider the scope of adversarial attacks to be limited to security attacks over data networks. The process of loading software on LRUs within an airplane is assumed to be sufficiently protected with specific physical, logical, and organizational inhibitors, checks, and control. Loading is only performed at specified times, for example, when the airplane is in maintenance mode, and by authorized personnel using authorized equipment. Moreover, certain checks are in place to enable detection of corrupted software, e.g. checking the list of parts to be loaded with a configuration list provided by the airline, and if software is compatible with the destination LRU hardware and software environment.

Furthermore, it can be assumed that due to software and hardware redundancies (e.g. several code instances executing in parallel on different system platforms on the airplane), most unintentional or unsophisticated corruptions or misconfigurations in safety-critical software are detectable at least when loaded into an LRU. Therefore, to effectively cripple a safety-critical function in the airplane, the representation of software must be modified at several positions. This significantly increases the effort needed from the adversary.

Based on the motivation and impact of adversarial attacks over networks, we can classify security threats as described next.

2.3 Safety Threats

The adversary can attack the AADS to threaten airplane safety. We identify the following specific threats that could amount to sabotage of the airplane.

Asset Corruption. The contents of distributed software can be altered or replaced (in an undetectable manner) to provoke accidents. This type of corruption to airplane-loadable software is sometimes referred to as *coherent corruption*, emphasizing a distinction from arbitrary bit-substitutions, which generally would render a software component unloadable. Airplane-generated data can be also corrupted to threaten airplane safety, e.g. by altering safety-related reports.

Software Misconfiguration. In order to cause havoc, a mismatch between the airplane's intended and actual configuration can be provoked by preventing delivery of software, deleting software, or injecting inappropriate software during distribution.

Asset Diversion. Software can be diverted to an unsuitable recipient to provoke accidents, e.g. by disturbing the execution of other software at that destination.

Asset Staleness. The revocation and update of software that need to be changed for safety reasons can be blocked and delayed, thus impeding the distribution processes.

2.4 Business Threats

The adversary can attack the AADS to induce unjustified airplane safety concerns or to cause flight delays, and thereby present threats to business of airplane manufacturer and/or owner.

Asset Unavailability. Assets can be made inaccessible or unusable, for example by jamming asset distribution to disrupt airplane service.

Late Detection. Assets can be intentionally corrupted so that the tampering is detected late enough for the airplane to be put out of service. For example, when tampering of software is not detected during distribution from ground systems to airplane, but is detected only upon final load at the destination LRU in the receiving airplane. Software corruption that is detectable by an LRU, or whose installation renders the LRU non-functional, is distinct from that referred to above as *coherent corruption*.

False Alarm. Assets can be tampered to cause economic damage from misleading safety concerns. In particular,, corruption of configuration reports might cause an airplane to appear as if incorrectly configured, creating unwarranted flight delays from the misleading safety concerns.

Repudiation. Any entity in the AADS could deny having performed *security-relevant actions*, e.g. deny having distributed or received some software.

3 Securing AADS

The threats listed in the previous section must be countered and mitigated by appropriate security objectives, which in turn must be implemented using suitable mechanisms. This section presents the security objectives to counter the security threats listed above, followed by an overview of the mechanisms proposed to achieve them, as well as a brief rationale why they should be sufficient for this purpose.

3.1 Safety-relevant Security Objectives

1. *Integrity*. For every asset that is accepted at its destination, its identity and contents must not have been altered on the way—it must be exactly the same as at the source of the distribution. This includes protection against injection of viruses and other malicious code.
2. *Correct Destination*. An airplane must accept only those assets for which it is the true intended destination.
3. *Correct Version*. An airplane must accept assets only in the appropriate version.

4. *Authenticity*. For every security-relevant action, the identity of entities involved must be correct. This applies in particular to the alleged source of an asset.
5. *Authorization*. Whenever an entity performs a security-relevant action, it must have the authorization or privilege to do so. Otherwise the action must be denied.
6. *Timeliness*. Required software installations and updates must be capable of being performed and confirmed by appropriate status reports within a specified period of time. Note that otherwise the airline's configuration management (which is not strictly part of the AADS) must take a deliberate decision whether the respective airplane is still considered airworthy.

3.2 Business-relevant Security Objectives

7. *Availability*. All necessary assets must be available in a time window adequate to support regulatory requirements and business needs.
8. *Early Detection*. The fact that attackers have tampered with assets must be detected as early as possible; that is, by the next trusted entity handling it. In particular, a tampered part should be detected well before actually being loaded by its destination LRU.
9. *Correct Status Reporting*. Status information concerning asset disposition, in particular reports listing the contents of the current airplane on-board parts storage, must be kept intact during transport in order to avoid false claims about, for instance, missing parts.
10. *Traceability*. For every security-relevant action, as well as unsuccessful attempts to perform such actions, all relevant information must be kept for a period of time sufficient to support regulatory requirements and business needs, such as general short-term security audits. This information includes the identity of entity involved, the action type with essential parameters, and a timestamp.
11. *Nonrepudiation*. To support forensics, for instance after an airplane crash, entities must not be able to deny their security-relevant actions. Evidence for this must be verifiable by third parties and must be long-lived: at least 50 years.

Table 1 (see overleaf) shows which security objectives mitigate which threats. The mechanisms employed in the AADS to address the above objectives are described next.

Objectives \ Threats		Safety				Business			
		Cor-ruption	Misconfig-uration	Diver-sion	Stale-ness	Asset Un-availability	Late Detection	False Alarm	Repudi-ation
Safe-ty Rele-vant	Integrity	√							
	Correct Destination			√					
	Latest Version				√				
	Authenticity	√	√						√
	Authorization	√	√						
	Timeliness				√				
Busi-ness Rele-vant	Availability					√			
	Early Detection						√		
	Correct Status Reporting							√	
	Traceability	√	√						√
	Nonrepudiation								√

Table 1: Security threats and objectives to cover them.

3.3 Securing Distributed Assets Using Digital Signatures

Digital signatures constitute the main mechanism to secure distributed assets in the AADS. We note that the choice of using digital signatures, as opposed to other integrity protection solutions such as keyed hashes and virtual private networks (VPN), is made in order to additionally provide nonrepudiation of origin as well as data authenticity and data integrity across multiple AADS entities. The message sent from a source to destination appears as follows:

$$asset, metadata, sign_{source}(asset, metadata) \quad (1)$$

where $sign_X$ denotes a signature with the private key of entity *X*, and *metadata* denotes additional information associated with the asset or its handling. Common examples include the destination (or a class of destinations) constituting the intended delivery target for assets, and timestamps or similar tags that can be used to inhibit later replay. Using the public key of the source and the hash function, the receiver can check all the information received. In this way, the signature ensures the integrity of the asset and the metadata, thus covering also freshness and the correctness of the destination.

A major challenge remains with respect to authenticity (and the related authorization requirement): how does the receiver reliably know the public key of the source? The management of identities and associated keys and certificates is an important task [13], requiring implementation of key management facilities or availability of a PKI. A Public Key Infrastructure (PKI) [6] consists of a Registration Authority (RA) to authorize key/certificate requests from entities, a Certification Authority (CA) to generate and issue asymmetric key pairs and corresponding digital certificates for requesting entities and to determine validity of certificates, and a Certificate Repository to store and distribute certificates. An airline may assign the role of RA

and/or CA to a trusted third party, e.g. a government agency or commercial vendor. Alternatively, an airline can implement its own PKI and itself function as RA and CA.

Relying on a PKI, the source can simply append to its message a standard digital certificate, $cert_{source}$, provided by a CA trusted by the receiver:

$$cert_{source} = sign_{CA}(id_{source}, K_{source}, id_{CA}, validity\,period) \quad (2)$$

where id_X is an identifier for entity *X and* K_{source} is the public key of *source*. The receiver can check the certificate, needing to know only the public key of the CA, and thus obtain and verify the authenticity of K_{source}.

Yet a PKI is a complex system, which in turn needs to be certified, which is a major undertaking in itself. Driven by the considerations briefly shared in section 4.2, we are currently investigating light-weight alternatives to PKI.

The verification of asset signatures can be end-to-end or entity-to-entity, as follows.

Entity-to-entity integrity protection. For every signed asset, each intermediate entity verifies and re-signs it, and then forwards it to the next entity along the desired path for that asset. Depending upon business requirements, and state of an asset's life-cycle or workflow, re-signing may constitute replacement of an existing signature or addition of a new one. In an entity-to-entity arrangement, localized key management capabilities suffice to establish trust and authenticity.

End-to-end integrity protection. Each asset, signed by its producer, is verified at each intermediate entity as well as by the final destination. The end-to-end architecture may be argued to have stronger security properties than the entity-to-entity architecture.

Implementing an end-to-end architecture requires distributed entities to have access to information about more identities than merely those of their immediate neighbors. Generally, this means making use of a mature public key infrastructure. The main security advantage of end-to-end architecture is that the final receiver need not trust intermediate entities but just the first sender whom it can authenticate directly. Intermediate entities cannot undetectably tamper with data in transit.

As the life-cycle of AADS-distributed parts evolves, the practical lifetime of signatures must be considered. The cryptanalytic capabilities available to attackers improve over time, and the potential for compromise of secret keys increases. Signature lifetimes may be extended via periodic refreshment or replacement of signatures. New signatures can be based on longer keys and improved cryptographic algorithms, as they become available.

3.4 Other security mechanisms

Security-relevant actions like releasing, approving, ordering, receiving, and loading software, as well as issuing and revoking certificates must be authorized. This can be achieved, for instance, via role-based access control or certificate-based capabilities.

In order to support traceability, all security-relevant actions, as well as unsuccessful attempts to perform such actions, are timestamped and logged. Logs must be implemented with tamper-proof storage.

High availability can be achieved with host and network protection mechanisms, for instance efficient filtering, channel switching, and redundant storage and bandwidth.

3.5 Coverage analysis

The security mechanisms given in sections 3.3 and 3.4 suffice to cover required EDS security objectives given in sections 3.1 and 3.2, as described below. A more in-depth examination of the requirements coverage is contained in [14].

The *integrity* and *authenticity* of assets is guaranteed by the digital signature of the source and the corresponding public key or certificate(s), with the validity period of the signatures extended by refreshing them. Checking signatures as soon as possible during transmission (i.e. at each intermediate entity) contributes to *early detection* of improper contents. In the source signed asset, the timestamp together with version numbers ensures that an outdated asset is not accepted, satisfying *latest version*, in accordance with the principle that airlines must be responsible for managing the configurations of the aircrafts they own. Further, by including the intended destination among signed meta-data with distributed assets, diverted assets are not accepted, meeting *correct destination*. None of the above mechanisms can mitigate insider attacks, though appropriate access controls ensure that critical actions are initiated by *authorized* personnel only.

Signatures for integrity protection of status information and authorization of status-changing actions contribute to *correct status reporting* of information on assets. Signatures and audit logs are sufficient for achieving *nonrepudiation* and *traceability*.

Although *availability* cannot be fully guaranteed in AADS, existing techniques can be used to mitigate jamming attacks. Backup mechanisms, such as traditional physical transfer of storage media using bonded carriers, can be used to reduce impact of non-availability of assets or asset distribution. *Timeliness* relies on availability, timestamping and organizational measures: in case of asset uploads being due, the providers must notify the respective receivers in a timely way and specify the new version numbers as well as a deadline by which the assets must have been loaded. Moreover, they must make sure that the assets are available for being pulled by the receivers during the required period of time.

4 Assurance Levels and Impact on Safety

In this section we present our analysis of the implications of security threats to the safety of airplanes, and determine the minimum assurance levels that must be met by AADS. Moreover, we mention pragmatic considerations on achieving them.

4.1 Determination of Assurance Levels

The Threat Level for the expected threat source on airplane *safety*, according to [2], is that of international terrorists, i.e. T5 - sophisticated adversary with moderate resources who is willing to take significant risk. Some software is of ultimate criticality for flight safety and is assigned RTCA/DO-178B [1] Level A[1], and thus according to [2] have Information Value V5 - violation of the information protection policy would cause exceptionally grave damage to the security, safety, financial posture, or infrastructure of the organization. Since the failure of parts with software Level A is catastrophic, so too can be the effect of not achieving the integrity and authenticity protection that should be guaranteed by the AADS distributing such software. According to [2], the above assigned Threat Level T5 and Information Value V5 together imply selection of EAL 6.

To address security concerns emerging from *business threats*, an assurance level of EAL 4 is sufficient, as follows. The Threat Level according to [1] for the expected business Threat Source is that of organized crime, sophisticated hackers, and international corporations, i.e. T4 - sophisticated adversary with moderate resources who is willing to take little risk. Attacks against the availability of assets can cause major damage to airlines from the business perspective, by putting individual airplanes out of service. Moreover, one must be able to counter attacks against software that have a highly visible effect to passengers, in particular if they affect more than one airplane. For example, a hacker could corrupt Level D or Level E software e.g. controlling the cabin light or sound system, for which generally no strong defense may exist. In this way the attacker could create anomalies to provoke safety *concerns*. This can cause severe damage to the reputation of both the airline and the airplane manufacturer, in particular as it might appear that little confidence can be put on their ability to protect other, highly critical software in the airplane. This could cause the whole fleet to be grounded, even though mainly for psychological reasons. From scenarios like these, we propose that there are assets that have a business Information Value of V4 - violation of the information protection policy would cause serious damage to the security, safety, financial posture, or infrastructure of the organization. Given a Threat Level T4 and a Information Value V4 for parts, according to [2], EAL 4 is sufficient.

[1] For software of lower criticality level (B thru E according to [1]), some lower value would be sufficient, but since the AADS should uniformly handle software of all criticality levels, the desired EAL with respect to *safety threats* should be the one for Level A software.

4.2 Pragmatic issues

An assurance level of EAL 4 permits maximum assurance for the development of the AADS with the use of positive security engineering based on good commercial development practices [3]. Although these practices are rigorous, they do not require significant specialist knowledge, skills, and other economically taxing or time consuming resources.

As mentioned in Section 3.3, the state-of-the-art requires the AADS to make use of digital signatures which rely on some form of key management. Unfortunately, the maximum assurance level of current commercially available Public Key Infrastructure is EAL 4, and the practical value of evaluating the system to a level higher than its PKI environment can support is questionable. This could motivate specification of assurance for the AADS at the highest EAL available for PKI, which currently is EAL 4. Yet EAL 4 would be insufficient for the integrity protection needs of Level A software. Moreover, evaluating a whole system as complex as an AADS at an assurance level of EAL 6 would be extremely costly.

As a viable solution to the discrepancy just described, we suggest a two-level approach where the mechanisms covering the most critical safety-relevant objectives, namely those which counter the threat of corrupted software (i.e., integrity, authenticity, and authorization), reach EAL 6, while the remaining components are kept at EAL 4. Since the mechanisms requiring EAL 6 include key management, it is necessary to raise the certification of an existing PKI to that level, or to implement the necessary functionality within the highly-critical part of the AADS. Designing the AADS architecture such that the key management for the EAL 6 components is minimized should make the high-level certification effort bearable.

5 Conclusions and Future Work

In this paper, we have studied the safety and security aspects of electronic distribution of software (EDS) and data. We have identified information security threats to airplane safety emerging from attacks on safety-critical software. Additionally, we have found that attacks on less critical (and hence less protected) software controlling onboard utility systems can induce unwarranted and misleading safety concerns, impeding business of airplanes. We have proposed a secure EDS system, Airplane Assets Distribution System (AADS), which addresses the threats and serves as a guideline for design and evaluation of EDS systems implemented for use with airplanes. Further, we have evaluated the impact of security threats on safety, and suggested suitable assurance levels for enabling a Common Criteria security evaluation of EDS system implementations. Concerning the assurance assessment and certification effort for AADS, we have proposed a two-assurance-level approach that addresses integrity protection of safety-critical software while keeping evaluation cost manageable.

It is hoped that the security requirements described above and the analysis detailed in the AADS Protection Profile [14], will provide a permanently useful public reference, and that they may be adopted by the regulatory community in much the same way as existing RTCA and other guidance have been. Several difficult and interesting issues remain to be investigated and resolved. Future work should focus on advancing the knowledge on the relations of security and safety analysis of an EDS system, including quantifying vulnerabilities to evaluate and certify a safety critical system under security threats, and correlating security assessment methods with development assurance guidelines in RTCA/DO-178B [1] as well as using this mapping for insights into the interaction between information security and airplane safety.

Acknowledgements

We would like to thank Prof. Peter Hartmann from the Landshut University of Applied Sciences and several anonymous reviewers for their insightful and valuable comments that helped us to improve specific sections of this paper.

References

1. DO-178B: Software Considerations in Airborne Systems and Equipment Certification. Radio Technical Commission for Aeronautics (RTCA) (1992)
2. Information Assurance Technical Framework, Release 3.1. US National Security Agency. http://www.iatf.net/framework_docs/version-3_1/
3. Common Criteria. http://www.commoncriteriaportal.org/
4. Federal Aviation Administration, 14 CFR Part 25, Special Conditions: Boeing Model 787–8 Airplane; Systems and Data Networks Security—Isolation or Protection from Unauthorized Passenger Domain Systems Access, [Docket No. NM364 Special Conditions No. 25–07–01–SC], Federal Register, Vol. 72, No. 71 (2007). http://edocket.access.gpo.gov/2007/pdf/E7-7065.pdf
5. Federal Aviation Administration, 14 CFR Part 25, Special Conditions: Boeing Model 787–8 Airplane; Systems and Data Networks Security—Protection of Airplane Systems and Data Networks From Unauthorized External Access, [Docket No. NM365 Special Conditions No. 25–07–02–SC], Federal Register, Vol. 72, No. 72 (2007). http://edocket.access.gpo.gov/2007/pdf/07-1838.pdf
6. Adams, C., Lloyd, S.: Understanding PKI: Concepts, Standards, and Deployment Considerations. Addison-Wesley, 2nd edition (2003)
7. Bird, G., Christensen, M., Lutz, D., Scandura, P.: Use of integrated vehicle health management in the field of commercial aviation. NASA ISHEM Forum (2005). http://ase.arc.nasa.gov/projects/ishem/Papers/Scandura_Aviation.pdf
8. Brostoff, S., Sasse, M.: Safe and sound: a safety-critical approach to security. ACM workshop on new security paradigms (2001) 41-50
9. Ibrahim, L., Jarzombek, J., Ashford, M., Bate, R., Croll, P., Horn, M., LaBruyere, L., Wells, C.: Safety and Security Extensions for Integrated Capability Maturity Models, United States Federal Aviation Administration (2004). http://www.faa.gov/about/office_org/headquarters_offices/aio/ documents

10. Leveson, N.: Safeware: System Safety and Computers. Addison Wesley Longman, Reading, Massachusetts (1995)
11. Lintelman, S., Robinson, R., Li, M., von Oheimb, D., Sampigethaya, K., Poovendran, R.: Security Assurance for IT Infrastructure Supporting Airplane Production, Maintenance, and Operation. National Workshop on Aviation Software Systems (2006). http://chess.eecs.berkeley.edu/hcssas/papers/Lintelman-HCSS-Boeing-Position_092906_2.pdf
12. Pfitzmann, A.: Why Safety and Security should and will merge, Invited Talk. 23rd Intern. Conference SAFECOMP 2004, Potsdam, Springer-Verlag, LNCS 3219 (2004)
13. Robinson, R., Li, M., Lintelman, S., Sampigethaya, K., Poovendran, R., von Oheimb, D., Bußer, J.: Impact of Public Key Enabled Applications on the Operation and Maintenance of Commercial Airplanes. To appear in AIAA Aviation Technology, Integration and Operations (ATIO) conference (2007)
14. Robinson, R., von Oheimb, D., Li, M., Sampigethaya, K., Poovendran, R.: Security Specification for Distribution and Storage of Airplane-Loadable Software and Airplane-Generated Data, Protection Profile (2007). Available upon request
15. Rodriguez-Dapena, P.: Software safety certification: a multidomain problem, IEEE Software, Vol. 16, No. 4 (1999) 31-38
16. Stavridou, V., Dutertre, B.: From security to safety and back, Conference on Computer Security, Dependability and Assurance (1998) 182-195
17. Weaver, R.: The Safety of Software - Constructing and Assuring Arguments, DPhil Thesis, Department of Computer Science, University of York, UK (2003)