


# VLSEC: A CONCEPT FOR SECURING VIRTUAL LINKS IN AVIONICS NETWORKS THROUGH CENTRALIZED GROUP KEY MANAGEMENT

Can André Dautel[1], Mario Werthwein[1], Mariann Homolya[1], Matthias Lehmann[1], Samuel Queiroz[1] & Bjoern Annighoefer[1]

[1] University of Stuttgart, Institute of Aircraft Systems (ILS), Stuttgart, Germany

## Abstract

Future connected aircraft increase the need for cybersecurity concepts that protect communication not only across external datalinks, but also within the internal avionics network. Existing aeronautical security mechanisms often focus on external communication links, specific endpoints, or architectural separation. However, internal avionics networks are typically deterministic, statically configured, connectionless, and multicast-oriented. These characteristics limit the direct applicability of conventional session-based security protocols that rely on bidirectional communication, dynamic negotiation, and pairwise security associations.

This paper introduces Virtual Link Security (VLSec), a conceptual architecture for securing data in transit in avionics networks through centralized key management. VLSec maps statically configured Virtual Link (VL)s to security-bounded communication groups and assigns dedicated group key material to each VLSec group. A Centralized Key Management System (CKMS) manages group membership and key lifecycle operations as a platform-level function, while local Security Agents enforce integrity, group-level authenticity, and optional confidentiality in the data path. By separating operational payload communication from key distribution and key lifecycle management, VLSec aligns security mechanisms with the deterministic and multicast-oriented nature of avionics communication. The resulting design trade-offs include group-key-based authentication, replay protection, CKMS criticality, key rotation, and safety-aware failure behavior.

**Keywords:** Avionics Cybersecurity, Virtual Links, Group Key Management, Data in Transit Protection, ARINC 664

## 1. Introduction and Motivation

Current press releases and recent research indicate a growing interest in increasing the interconnectivity between aircraft, between aircraft and ground infrastructure, and within the cabin [1, 2]. This development is driven by the demand of aircraft buyers, airline operators, regulatory authorities, and passengers for increasingly connected and “always-on” aircraft operations. To support this trend, new aeronautical communication concepts have been developed. For example, the L-band Digital Aeronautical Communications System (LDACS) [3] aims to increase the capacity and efficiency of external aircraft communication. As a result, future aircraft can no longer be considered as largely isolated avionics platforms, but are evolving into highly connected ecosystems in which safety-critical, operational, and passenger-related data may be exchanged within the aircraft, between the aircraft and ground infrastructure, and between aircraft [4].

This increasing connectivity creates new opportunities for operation, monitoring, maintenance, and enhancement of avionics systems. At the same time, however, it enlarges the attack surface of aircraft systems. New communication channels, interfaces, and access points increase architectural complexity and may introduce new vulnerabilities. Consequently, security requirements such as integrity, authenticity, confidentiality, freshness, and availability of transmitted data become more important.

In safety-critical avionics, inadequate communication security must not be considered in isolation, but has to be assessed in the context of the aircraft's functional safety [5]. Traditional avionics architectures have relied strongly on physical and logical isolation, domain separation, and controlled interfaces [6]. Future connectivity concepts may require controlled communication across traditional domain boundaries via cyber gateways, as the lines between them become blurred [7]. Cybersecurity must therefore not only be applied to external communication interfaces, but has to be considered throughout the entire communication chain.

Emerging aeronautical communication systems already include security mechanisms. Systems such as LDACS [8, 9] or Aeronautical mobile airport communications system datalink (AeroMACS) [10] address the protection of communication links between the aircraft and external communication partners. However, the scope of such protection is typically limited to a specific external communication link and often ends at a defined access point of the aircraft, such as a communication endpoint, gateway, or firewall. This does not automatically guarantee that data remains protected once it has entered the internal avionics network. In addition these future aeronautical communication systems have to consider threats arising from quantum-capable adversaries. One possible technology for protecting external aircraft communication links against such threats is Quantum Key Distribution (QKD) [11]. QKD enables two communication endpoints to establish shared secret key material by using quantum physical effects rather than relying solely on computational hardness assumptions. However the generated key material is initially available only at the corresponding communication endpoints. If this key material is required by avionics applications, gateways, or security functions inside the aircraft, it must be distributed through the internal avionics network in a secure and controlled manner.

This lack of protection for the avionics platform highlights a broader gap in future aircraft communications security. The scope of existing and emerging security concepts do not specify how protected data or key material should be managed and secured throughout the entire avionics network. At the same time, avionics networks differ significantly from conventional IT networks. They are deterministic, real-time capable, and in many cases organized according to connectionless and multicast-oriented communication principles [12]. In ARINC-664-based avionics networks, communication is organized using statically configured Virtual Link (VL)s [13]. In this context, a VL defines a unidirectional logical communication path from a sending end system to one or more receiving end systems.

To address this gap, this paper introduces Virtual Link Security (VLSec), a concept for securing internal VL-based communication in avionics networks through centralized key management. VLSec treats VLs as security-bounded communication groups and assigns dedicated group key material to these groups. The Centralized Key Management System (CKMS) maintains the mapping between VLs, group membership, and Group Key (GK), while local security components enforce the corresponding protection in the data path. In this way, key distribution and key lifecycle management are considered platform-level functions that complement external communication security mechanisms such as LDACS security, Aircraft Communications Addressing and Reporting System (ACARS) [14] Message Security, or future QKD-based key establishment.

The contribution of this paper is a conceptual architecture that maps VLs to security groups, manages the corresponding GK through a CKMS, and discusses the resulting design trade-offs for deterministic and multicast-oriented avionics networks. The concept does not replace existing architectural protection mechanisms such as domain separation, partitioning, or external communication security. Instead, it complements them by addressing data in transit protection and key lifecycle management inside the avionics network.

The remainder of this paper is structured as follows. Section 2 presents current architectural security mechanisms in avionics networks, introduces Integrated Modular Avionics (IMA) platforms and VL-based communication, and derives the resulting limitations for security mechanisms. Section 3

discusses related work on avionics platform security and key management. Section 4 introduces the proposed VLSec concept and its main architectural components. Section 5 discusses the concept, its benefits, and its limitations. Section 6 outlines future work, and Section 7 concludes the paper.

## 2. State of the Art

The current state of the art is discussed in three steps. First, network domains and established security mechanisms in modern aircraft are described. Next, the technologies used for avionics platforms, including IMA, are addressed. In the third step, key management and key lifecycle management addressing avionics networks are outlined.

### 2.1 Architectural Security Mechanisms in Avionics Networks

Modern avionics networks mitigate security risks through their architectural design. Rather than cryptographically securing every single internal data flow, they rely on segregating systems based on their criticality, limiting interaction between these systems, and controlling the interfaces through which data may be transmitted [15].

A central mechanism is the separation of the aircraft network into different domains, each representing distinct levels of protection, criticality, and trust [16, 6]. These domains include, in particular, the Aircraft Control Domain (ACD), the Airline Information Services Domain (AISD), and the Passenger Information and Entertainment Services Domain (PIESD). As shown in Figure **??**, this domain structure separates systems with different safety and security requirements and prevents less trustworthy areas from directly influencing more safety-critical areas [17].

The ACD contains safety-critical avionics functions that are directly related to the control and safe operation of the aircraft. This domain has the highest level of criticality and is generally isolated from direct external access during flight. The AISD serves as an intermediate domain between safety-critical avionics functions, airline operational communication, and less critical information services. The PIESD provides passenger-related services, such as in-flight entertainment and passenger information systems. Due to its proximity to passengers and external interfaces, this domain is assigned a lower trust level than the safety-critical domains.

Communication between domains is restricted to defined interfaces. For example, ACD and AISD are connected via a Network Interface Module. Such components ensure that data originating from domains with lower trust levels can affect domains of higher criticality only in a controlled and predefined manner, as is the case with data loading during maintenance . External communication links may also be assigned to different domains depending on their purpose and criticality. For example, ACARS may be connected to operational or safety-related domains, whereas maintenance access is typically assigned to less critical operational domains [15]. In addition to domain separation, communication in avionics networks relies on statically configured and deterministic structures. Routing paths and permitted data flows are defined before operation and are not dynamically negotiated at runtime, thereby limiting unauthorized communication [17].

### 2.2 IMA Platforms and Virtual Link Communication

Today's avionics platforms are based on the concept of Integrated Modular Avionics (IMA). The fundamental architectural principle of IMA is the combination of partitioned computing resources with a deterministic aircraft data network. Multiple avionics applications are consolidated on shared computing platforms, commonly referred to as IMA modules. These modules provide shared processing, memory, communication, and input/output resources to several applications or functions [18].
Within an IMA module, applications are executed in partitions. These partitions are isolated from each other by the underlying real-time operating system kernel and are assigned separate memory areas and time-defined execution windows. Communication between partitions and between an IMA module and other end systems is performed via statically configured communication ports, such as queuing and sampling ports. Access to these operating system services is provided through the

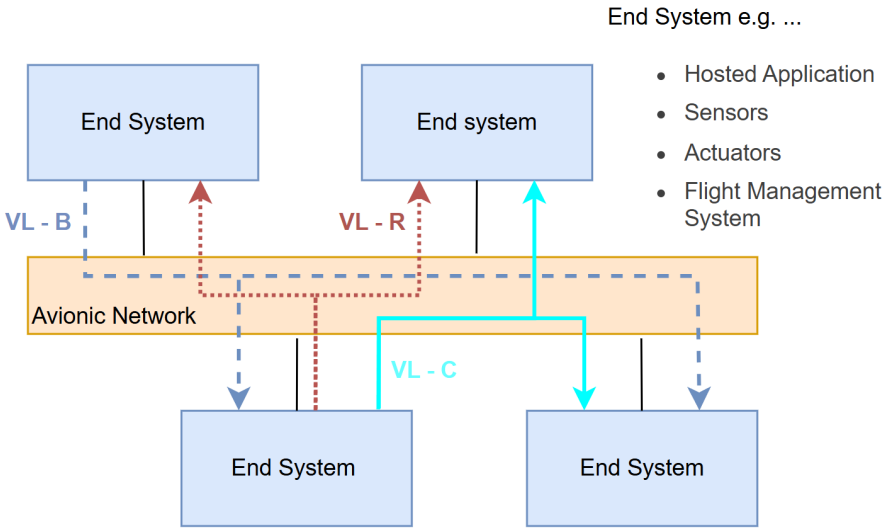



Figure 1 – Avionic network based on ARINC 664 [13] with focus on VL communication to provide a multicast communication environment. End systems can be any computing module connected to the avionic network e.g. hosted application on ARINC 653 IMA modules [18], sensors, actuators, or other computing resources. A VL is therefore a unidirectional 1-to-many connection.

ARINC 653 API, known as APEX. As a result, applications do not freely establish communication relationships at runtime. Instead, their communication capabilities are defined by the system configuration [18].

The IMA modules are connected to the aircraft data network, which interconnects computing modules, sensors, actuators, switches, and other end systems. In such networks, communication relationships are defined before operation. Both the IMA modules and the network switches are configured prior to commissioning. This configuration specifies which applications are allowed to exchange data, which communication ports are available, and how frames are forwarded through the avionics network [12].

In ARINC 664 networks, statically configured communication is organized using Virtual Links. A VL defines a unidirectional logical communication path from a single sending end system to one or more receiving end systems and is therefore considered multicast communication. Although the end systems are physically interconnected through network switches, the permitted logical communication paths are determined by the configured VLs.

From an application perspective, this communication is connectionless. An application publishes data to a port associated with a VL, while the receiving end systems are determined by the network configuration. Consequently, neither the sender nor the receivers have complete knowledge of the underlying communication structure associated with the VL. As the transport-layer protocol, User Datagram Protocol (UDP) enables efficient data transmission without requiring a connection to be established between the sender and the receivers [12].

## 2.3 Key Management

Key management, including key lifecycle management, is essential for reliable network security. The security of a cryptographic system does not depend solely on the strength of the cryptographic algorithms used, but also on the correct handling of the associated key material. Even strong and well-established cryptographic mechanisms are ineffective if the corresponding keys are compromised, distributed incorrectly, reused beyond their intended lifetime, or not revoked after a compromise [19].

Key lifecycle management therefore includes all security-relevant phases in which cryptographic key material is generated, stored, distributed, activated, used, rotated, revoked, and destroyed [19]. For communication systems, this lifecycle perspective is particularly important because an encrypted or authenticated message is only trustworthy if the key material used to protect it has been handled correctly throughout its lifetime. It must therefore be possible to detect expired, outdated, or compromised keys and to replace them in a controlled manner [20].

In aviation communication systems, key lifecycle operations are associated with defined operational or maintenance phases, such as commissioning, or scheduled maintenance [17]. While such procedures are manageable for specific communication systems or external datalinks , they become challenging at the avionics platform level. If a large number of internal communication relationships requires dedicated key material, manually provisioning and updating keys for each relationship would create considerable maintenance effort. Key management for internal avionics communication should therefore be treated as a platform-level function and must remain compatible with the deterministic and statically configured nature of avionics networks.

This requirement is particularly relevant for multicast-oriented communication. In group communication, a message is transmitted to several authorized receivers. To protect such communication efficiently, Symmetric-Key Encryption [21] is a strong security measure. All legitimate group members may share a GK, which is used as the Data Encryption Key (DEK) for protecting the operational data. A CKMS can manage such groups by maintaining the membership information, generating or updating the corresponding GK, and distributing them only to authorized participants.

In a typical centralized group key management approach, each group member has an individual Key Encryption Key (KEK) or another secure association with the central key management entity. The GK is then protected with the respective and individual KEK and distributed to the authorized members. If group membership changes—for example, because a participant is added, removed, or classified as compromised by the system—the CKMS generates a new GK and distributes it exclusively to the updated group of members, using the individually negotiated KEKs in each case. This prevents former members from accessing future communication and prevents new members from accessing communication that occurred before they joined the group [19].

## 3. Related Work

There have already been attempts to address certain aspects of avionics platform security. However, these cybersecurity concepts tend to focus primarily on securing the end systems themselves and only touch briefly on the proposed network restrictions. Meanwhile, key lifecycle management and key distribution pose a major challenge for avionics platforms.
Varner et al. [22] presented an architecture and implementation that establishes data in transit protection through partition-based end-to-end encryption. Their approach combines Hybrid Public Key Encryption (HPKE) with post-quantum-secure cryptographic methods. It does not describe a complete security concept, but rather a system component that can be integrated into existing ARINC-653-based software architectures. As shown in Figure 2a the cryptographic functionality is encapsulated in a dedicated crypto partition, instead of integrating cryptographic libraries directly into each application partition. This crypto partition provides cryptographic services to other partitions via a defined interface. Application partitions can request a seal operation, in which plaintext is encrypted and authenticated, or an open operation, in which ciphertext is decrypted and simultaneously verified for authenticity. The keys remain hidden in the crypto partition from all other partitions.

By encapsulating the cryptographic application in a separate partition, the cryptographic implementation is separated from the application logic. Since cryptographic methods can become insecure due to new attacks, an avionics system must be able to replace cryptographic components without having to completely re-qualify the functional software of the transmitters and users. While this approach demonstrates that cryptographic services can be integrated into ARINC-653-based avionics

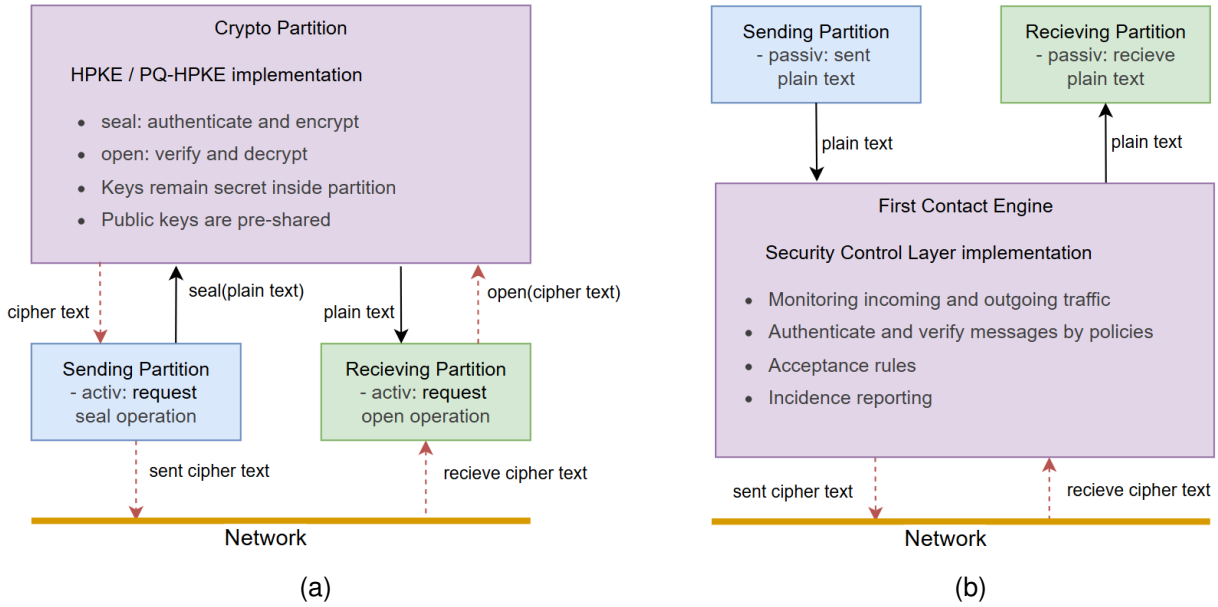



Figure 2 – (a) The crypto-partition [22] is placed inside an ARINC 653 partition. As a service other partitions can call the seal operation to authenticate and encrypt or the open operation to verify and decrypt their data. This is an active request from the application partitions. Public keys and secure keys are already known to the crypto-partition. (b) Software architecture of the First Contact Engine (FCE) based on the Approach of Werthwein et al. [23]. The FCE lays between the application layer of the IMA module and the network interface. Any data flow is directed through this FCE to be monitored and accepted.

platforms, it assumes that the required keys are already available within the crypto partition. Therefore, the distribution, update, and lifecycle management of cryptographic keys across the avionics network remain outside its scope. VLSec addresses this complementary problem by binding key material to Virtual Link security groups and managing it through a centralized platform function.

Werthwein et al. [23] developed a security concept for self-adaptive avionics platforms, specifically for the so-called Plug&Fly Avionics (PAFA) [24]. In contrast to the crypto-partition, which provides a service, Werthwein et al. position the security-relevant control entity as an additional layer of protection between the network interfaces and the application partitions. This component is referred to as the First Contact Engine (FCE). Its task is to inspect incoming and outgoing network traffic before it reaches other partitions or leaves the module. In this way, the FCE not only protects the partitions hosted on a PAFA module from unknown or unauthorized communication, but also protects the network from potentially compromised partitions that could send unauthorized messages. Furthermore, it abstracts the complexity and necessity of cryptographic mechanisms, such that an application must not use and must not be aware of them. Figure 2b illustrates a software architecture. This creates a control layer that monitors all external traffic of a module and forwards only those messages that comply with the specified communication rules. Conceptually, the FCE follows a zero-trust principle. Communication is not implicitly treated as trustworthy simply because it originates from a known end system.

The communication policies form the core of the security concept. They define which communications are permitted, under what conditions messages are accepted, and what response is triggered in the event of policy violations. Such policies can be formulated specifically for individual communication channels or messages. The policies contain several key characteristics. These include, first and foremost, temporal requirements, such as a generation time, the periodicity of a message, and an allowable jitter. This enables the FCE to verify whether a message arrives within the expected time window. The policies can specify whether and which cryptographic measures are required for a

message, such as signatures, hashes, or encryption. Acceptance rules define the conditions under which traffic is considered acceptable. Each FCE compiles detected rule violations and security-related events at the module level into an incident report.

The FCE concept is closely related to VLSec because it places security enforcement between the application layer and the network interface. However, its primary focus is the enforcement of communication policies and the detection of unauthorized traffic. VLSec builds on a similar placement of a local security layer, but focuses on how cryptographic key material can be assigned to Virtual Link-based communication groups and maintained throughout its lifecycle.

The reviewed approaches provide two important building blocks for cyber security approaches: local cryptographic processing and local policy enforcement. However, neither approach defines how key material is derived from the static VL configuration, or how keys are distributed and updated across the avionics network.

Key management has also been addressed in aeronautical communication systems. ARINC 823 defines ACARS Message Security for protecting ACARS messages exchanged over traditional air-ground datalinks [14]. Its protection mechanisms include message authentication, integrity, and confidentiality. More recent work on LDACS addresses authenticated key establishment, certificate management, and Public Key Infrastructure (PKI)-based mutual authentication between aircraft and ground infrastructure [25]. In addition, group key management has been considered for LDACS control-channel communication, since control messages may apply to all members of an LDACS cell and therefore require secure group communication [8].

These approaches demonstrate that key management is an established concern in aviation cybersecurity. However, their protection scope is primarily external air-ground communication, ground-ground datalink communication, or protocol-specific control-channel security. They do not address key lifecycle management for internal avionics communication based on statically configured Virtual Links.

Cryptographic protection under real-time constraints has also been investigated across domains such as automotive systems, railway signaling, and electrical power generation. Across these domains, cyclic sensor, control, and protection messages are commonly secured using symmetric message authentication, timestamps, sequence counters, or authenticated encryption, whereas public-key mechanisms are primarily used for initial authentication, certificate validation, key establishment, and software updates. The central implementation challenge is the bounded integration of cryptographic generation, verification, and communication overhead into the end-to-end timing analysis. Research on real-time security therefore models cryptographic operations as part of the schedulability problem and selects levels of protection, execution frequency, or authentication intervals subject to application time constraints [26].

In automotive control systems, this principle is reflected in schemes that integrate message authentication into periodic control transactions and, where resources are limited, authenticate selected messages or cumulatively protect several preceding messages. Such approaches explicitly balance authentication frequency against control performance and schedulability [27]. Railway signaling [28] and electrical substation [29] communication follow a similar architectural pattern: symmetric authentication is used for frequent, time-critical operational messages, while asymmetric cryptography supports less frequent trust-establishment and lifecycle operations. These examples show that cryptographic security in real-time networks is generally implemented as a cross-layer function involving key management, message freshness, network scheduling, and bounded cryptographic execution.

## 4. Virtual Link Security Concept

This section introduces Virtual Link Security (VLSec), a conceptual architecture for protecting data in transit in avionics networks. It addresses platform-specific communication constraints by using the existing Virtual Link structure for security grouping and key management.

## 4.1 Limitations on Cybersecurity Measures for Avionics Platforms

ARINC 664-based avionics networks use statically configured, connectionless, unidirectional, and multicast-oriented communication under strict real-time and determinism requirements. These properties do not prevent cryptographic protection, but they influence which mechanisms can be integrated directly and which require additional protocol functions, communication channels, or configuration effort.

Unidirectional VLs do not support acknowledgements, handshakes, key confirmation, or challenge-response authentication over the same path. Protocols based on dynamic session establishment, such as conventional uses of Transport Layer Security (TLS) [30] and Datagram Transport Layer Security (DTLS) [31], therefore require additional return VLs and corresponding configuration and verification effort. Mechanisms based on preconfigured keys or pre-established security associations are more compatible with the static communication model.

Multicast communication introduces a further constraint. A message is transmitted once to several receivers, whereas pairwise protection may require a separate security association and protected message for each receiver. For a VL with (n) receivers, up to (n) security associations may need to be configured and maintained. Although technically feasible, this increases processing, bandwidth, and key-management effort and does not preserve the original one-to-many communication model.

Real-time requirements do not exclude cryptography, provided that execution time and message overhead are bounded. Problems arise when security or key management operations depend on runtime interaction. Cryptographic processing, key negotiation, and key updates must therefore not introduce uncontrolled latency, jitter, or communication interruptions in the operational data path [12] For example, an interactive key update may be delayed if one receiver does not respond. Excluding key management from operational procedures can ensure stable communication. Preplanned key changes at defined activation times are therefore more suitable because they avoid negotiation in the time-critical payload path and provide predictable timing behavior.

Group-oriented protection is therefore a closer match for VL-based multicast communication than independent pairwise sessions. The sender and authorized receivers can be treated as a communication group using shared or group-specific cryptographic material, allowing a protected message to be transmitted to multiple receivers. This requires dedicated group key management, while key distribution, rotation, activation, and revocation should be handled separately from the operational payload path. Changes in group membership must likewise be managed without compromising predictable platform behavior.

## 4.2 Virtual Link Security Architecture

VLSec interprets each VL as a security-bounded communication group. All end systems that are authorized participants of a VL are treated as members of the corresponding VLSec group. Each VLSec group is assigned dedicated cryptographic key material. In particular, each VL is associated with a GK, which serves as the DEK for protecting the operational data transmitted over this VL. This key material is used by local Security Agents to provide integrity, group-level authenticity, and, where required, confidentiality. Membership is therefore a central element of the concept, because it determines which participants are authorized to receive and use the corresponding GK. The assignment between VLs, VLSec groups, group members, and cryptographic keys is managed by a CKMS. The CKMS maintains the knowledge base of configured VLs, derives the corresponding VLSec groups, tracks their membership, and distributes the associated GK only to authorized members. In this way, key distribution and key lifecycle management become platform-level functions rather than application-specific tasks.

The core idea of VLSec is therefore not to replace the existing avionics communication architecture, but to derive security boundaries from its static communication configuration. Operational payload

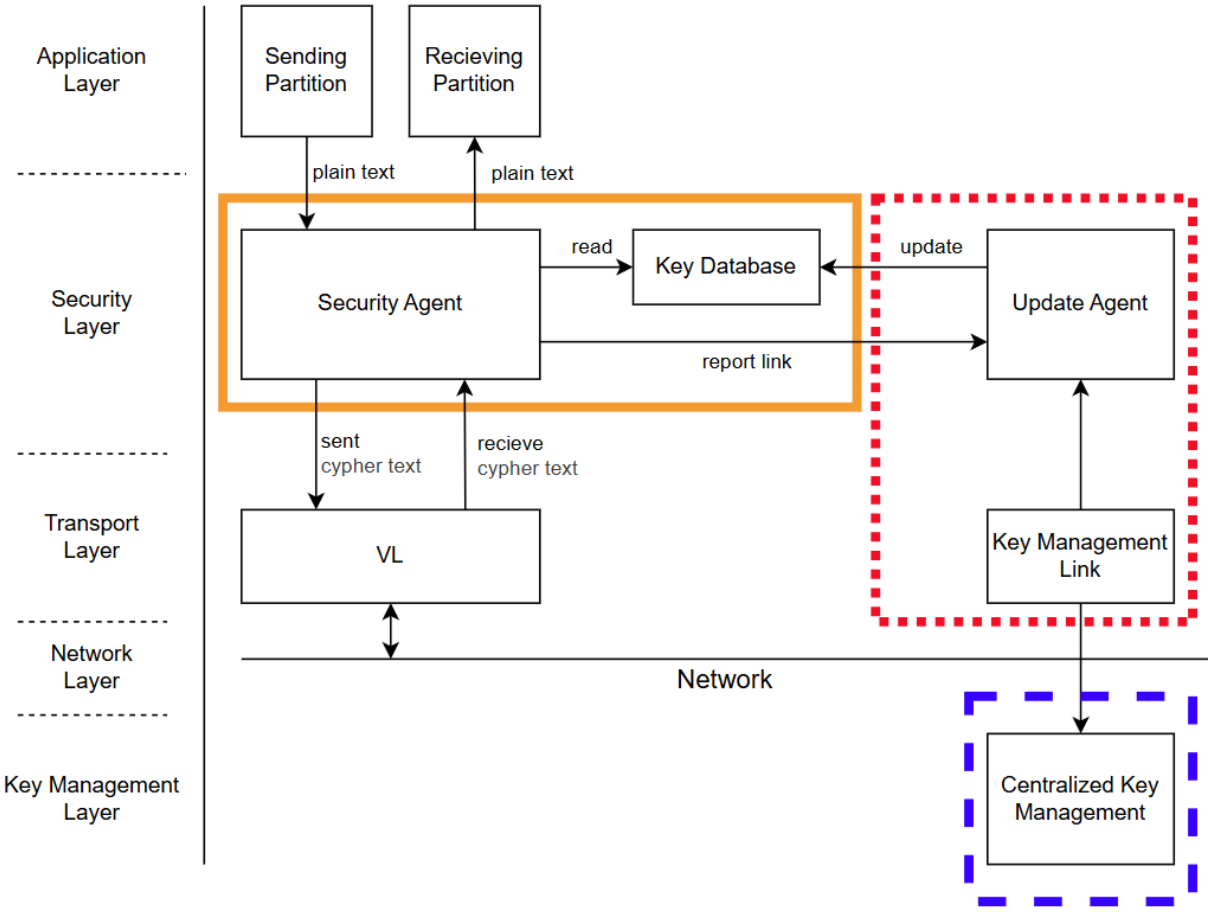



Figure 3 – VLSec architecture, integrated into the existing system communication of an avionics network. The diagram is divided into two sections: the internal security architecture of a network participant above the network boundary, and the network-side connection to the centralized key management system. Within the network participant, the Security Agent, together with the Key Database (orange, solid line), serves as a security layer to protect the application layer. The Update Agent and the Key Management Link (red, dotted line) form the interface between the module's security layer and the centralized Key Management Server (blue, dashed line). Key updates are imported from the Key Management Server into the Key Database via the Key Management Link using the Update Agent. Additionally, the Security Agent can report malfunctions or security-related events to the Update Agent via a Report Link, which forwards this information to the central Key Management system.

communication remains based on the configured VLs, while key distribution and key updates are handled separately through a dedicated key management path. The overall architecture, shown in Figure 3, is divided into three main components: the Security Agent with a Local Key Database, the Centralized Key Management (CKM) system, and the Key Update Agent.

**Security Agent with Local Key Database (solid, orange line):** The Security Agent forms the local security layer within a network node. Together with the Key Database, it implements cryptographic protection for application communication.

**Centralized Key Management System (dashed, blue line):** The CKMS manages the memberships of the VLSec groups and controls the lifecycle of the associated keys. It determines which participants are authorized to receive a specific GK and handles the controlled distribution and updating of the key material.

**Key Update Agent (dotted, red line):** The Key Update Agent acts as a local interface between the Security Agent or the Key Database and the central Key Manager. It is used to import new, updated, or revoked keys into the local Key Database.

Together, these three components form an architecture in which operational data communication, central key management, and local key updates are clearly separated from one another. This allows key lifecycle management to be implemented as a platform function without directly coupling the application logic to key management.

## 4.3 Security Agent and Local Key Database

The Security Agent and the local key database together form a security layer between the application layer and the network interface, similar to the FCE, and are thus directly integrated into a network participant's existing communication architecture. Applications continue to communicate via their designated communication ports without having to perform cryptographic operations themselves. Instead, the Security Agent transparently handles the security-related processing of incoming and outgoing messages. This relieves the application layer of cryptographic tasks, while security functions are still enforced directly within the module's communication path.

For outgoing communications, the Security Agent first determines the relevant security context of the message. This is defined by the message's assignment to a specific VLSec group. The Security Agent then selects the appropriate GK from the local key database. Depending on the security requirements, the message is authenticated, its integrity is protected, and, if confidentiality is required, the payload is encrypted. Only after this processing is the message passed to the network interface and transmitted over the network.

For incoming communication, the process occurs in reverse order. The Security Agent first receives the message from the network interface and assigns it to the corresponding VL or the appropriate VLSec group. The associated key is then selected from the local key database. Before the message is forwarded to the application layer, the Security Agent verifies its authenticity and integrity. Only if this check is successful is the message decrypted, if necessary, and then forwarded to the relevant application or partition. As a result, tampered, inauthentic, or improperly protected messages do not reach the application layer.

The local key database stores the DEK relevant to the respective VL and network participant. In addition to the actual key material, metadata is also stored. This may include, for example, a key identifier, the associated VL, the key's intended use, the activation time, the validity period, and the expiration time. The Security Agent uses this information to select the correct and valid key at runtime. However, the Security Agent itself does not have write access to the key database. This prevents operational software from unintentionally or unauthorized modifying key material.

## 4.4 Centralized Key Management

The second component of the concept is the CKMS. It has a system-wide view of the VL and the VLSec groups formed from them. Based on this, it determines which modules are authorized to receive the DEK. In addition, the CKMS handles key lifecycle management. It generates and manages the GK and associates them with the necessary lifecycle information. If a key expires or is nearing the end of its validity period, the CKMS ensures in a timely manner that a new GK is generated and distributed to all VLSec group members.

The CKMS plays a crucial role when changes are made to the membership or system configuration. In such cases, the previous GK must not be reused, as former participants might otherwise continue to have access to future communications. The CKMS therefore performs a re-keying. In this process, a new DEK is assigned to the affected VLSec group and distributed exclusively to the updated set of member. In this way, former participants are excluded from future communications.

### 4.5 Key Update Agent and Key Management Link

The Key Update Agent serves as the local bridge between the CKMS and the Key Database within an end system. Its task is to receive new DEKs, verify their authenticity and integrity, and then install them in the local Key Database. The Key Update Agent is thus responsible for updating the local key database, while the Security Agent uses the installed keys for operational data communication.

A direct key management link is established between the Key Update Agent and the CKMS. Unlike the normal operational avionics network, this Key Management Link is bidirectional and uses unicast. Therefore, end-to-end encryption such as DTLS can be performed via this link to negotiate KEK for encrypting and securely delivering DEK to network end systems.

In addition to the Key Management Link, there is also an Incident Report Link between the Security Agent and the Key Update Agent. The Security Agent uses this link to report security-related events that occur during the processing of operational communications. These may include, for example, failed authentication checks, invalid key states, expired keys, or other anomalies in the protected communications.
These incident reports are collected by the Key Update Agent and delivered via the Key Management Link to the CKMS. This enables centralized responses to security-related events, such as a key update or re-keying, and provides the ability to perform key lifecycle management when necessary. Since this Key Management Link is separate from the time-critical payload path, more complex post-quantum-secure cryptographic methods can be used to generate KEK to protect the key transfer.

## 5. Discussion

The VLSec concept treats cybersecurity in the avionics network as a system-wide platform task. Instead of protecting only individual communication links or individual applications, VLSec derives security groups from statically configured VLs and manages the corresponding GK through a CKMS. This provides a conceptual basis for protecting data in transit inside the avionics network and complements existing architectural protection mechanisms such as domain separation, partitioning, and controlled external interfaces.

A central benefit of the proposed architecture is the separation of operational payload communication from key lifecycle management. The Security Agent enforces cryptographic protection in the local data path, while key generation, distribution, rotation, and revocation are handled through a separate Key Management Link. This separation is important for avionics networks, since dynamic key negotiation should not interfere with deterministic and time-critical payload communication. At the same time, the separation of responsibilities between the Security Agent, the Local Key Database, the Key Update Agent, and the CKMS provides a clear architectural structure for implementing key lifecycle management as a platform-level function.

The concept is not limited to IMA modules. Since the Security Agent acts as a security layer between the application layer and the network interface, VLSec can in principle be applied to different end systems that communicate via VLs. The concept may also be applicable to adaptive avionics architectures, provided that communication relationships, group membership, and security rules can be derived from a trusted platform configuration. However, this applicability depends on the availability and trustworthiness of the configuration information from which VLSec groups are derived.

A first limitation results from the use of symmetric group keys. Group keys are well suited to multicast-oriented VL communication because they allow efficient protection of one-to-many data flows with limited computational overhead. They can provide integrity, group-level authenticity, and optional confidentiality. However, a shared group key only authenticates a message as originating from a member of the corresponding VLSec group. It does not provide unique sender authentication within the group. Any participant with access to the group key could, in principle, generate a cryptographically valid message. To achieve unique sender authentication or non-repudiation, additional mech-

anisms such as digital signatures or sender-specific authentication keys would be required. These mechanisms would have to be evaluated with respect to their computational overhead, memory requirements, and impact on real-time constraints.

A second limitation concerns the freshness and validity of messages. A cryptographically valid message must not automatically be accepted if it is outdated, delayed beyond its allowed time window, or replayed by an attacker. Replay protection and freshness checks are therefore necessary complements to message authentication. Possible mechanisms include sequence numbers, timestamps, sliding windows, permissible latencies, expected message periodicities, and acceptance rules. These mechanisms are closely related to policy-based approaches such as the FCE, but their concrete integration into VLSec remains an implementation decision.

The CKMS is another security-critical component. It enables consistent system-wide management of VL memberships and group keys, but it also represents a high-value target and a potential single point of failure. A compromise of this component could undermine the trustworthiness of the entire key distribution process. A failure of the component may prevent key updates or revocations from being performed in time. Therefore, practical implementations of VLSec require strong access control, protection of stored key material, monitoring, redundancy, and well-defined failure modes. It must also be specified how end systems behave when the Central Key Manager is temporarily unreachable but locally stored keys are still valid.

Reliable key rotation introduces further operational challenges. Although the Key Management Link is separated from the time-critical payload path, frequent membership changes, large VLSec groups, or compromised key material can cause significant management overhead. Therefore, key lifecycle policies should distinguish between planned key updates during commissioning, scheduled maintenance, or preflight checks and emergency revocation during operation or flight. In addition, key transition phases must be regarded as critical periods and taken into account, since not all end systems may receive new keys at exactly the same time. Fixed key state changes must ensure that missing or incorrect keys do not occur. The local key database and the security agent must therefore handle key validity periods, activation times, key identifiers, and any potential overlap between old and new keys in a controlled manner.

Finally, security decisions may have safety and availability implications. If the Security Agent rejects a message due to missing, expired, or inconsistent key material, the protected application may no longer receive operational data. Conversely, accepting messages with invalid or outdated security context may undermine the security objectives of VLSec. Practical implementations therefore have to define whether specific communication flows should fail closed, enter a degraded mode, or continue temporarily with locally available key material. These decisions cannot be made independently of the safety criticality and timing requirements of the affected avionics functions.

Overall, VLSec offers a promising conceptual architecture for applying centralized key management to VL-based avionics communications. Its greatest strength lies in deriving security groups from the existing static communication configuration and clearly separating the protection of operational data from key lifecycle management. Although questions remain open regarding sender authentication, replay protection, CKMS redundancy, key rotation policies, and safety-oriented failure behavior, these challenges primarily define the technical requirements that must be addressed when transferring the conceptual architecture into a certifiable implementation. Despite these challenges, the identified limitations do not invalidate the VLSec concept. The static and centrally configured nature of avionics communication provides favorable conditions for this approach, as group memberships, communication paths, timing requirements, and permissible system states can be predetermined. This enables key management mechanisms, transition procedures, and fault responses to be designed and verified in advance for specific operational scenarios. Consequently, VLSec offers a promising foundation for the gradually introduction of cryptographic security measures in real-time avionics networks without

requiring fundamental changes to the established communication architecture.

## 6. Future Work

Future work will focus on the refinement, implementation, and evaluation of the VLSec concept. A first step is the development of a prototype including the Security Agent, Local Key Database, Key Update Agent, and CKMS in a representative avionics test or simulation environment. This prototype will be used to evaluate the impact of VLSec on deterministic VL-based communication, including additional latency, jitter, CPU load, memory consumption, and key storage requirements.
Further work is required to specify the security mechanisms that complement group-key-based protection. This includes replay protection and freshness checks based on sequence numbers, timestamps, sliding windows, expected message periodicities, and acceptance rules. Since shared group keys only provide group-level authenticity, future work must also investigate mechanisms for unique sender authentication, such as sender-specific authentication keys or digital signatures, while considering their impact on real-time constraints.
Another important research direction is the robustness of the CKMS. Redundant CKMS architectures, secure synchronization between key managers, access control mechanisms, and defined failure modes have to be investigated. In addition, key lifecycle policies must be refined for different operational phases, including commissioning, scheduled maintenance, preflight preparation, normal operation, and emergency re-keying during flight.
Finally, future work should analyze how VLSec can be integrated with existing avionics security layers, such as crypto partitions and FCE-like policy enforcement mechanisms, as well as with future key sources such as QKD-based key establishment. The certification and assurance implications of VLSec should also be examined with respect to airworthiness security processes, IMA architectures, and ARINC-653-based development environments.

## 7. Conclusion

This paper presented VLSec as a conceptual architecture for extending data-in-transit protection into internal avionics networks. The central idea is to use the existing static VL configuration not only as a communication structure, but also as the basis for defining security-bounded communication groups. By assigning dedicated group key material to these groups and managing it through a centralized key management function, VLSec shifts key distribution and key lifecycle management from an application-specific concern to a platform-level task.

The proposed architecture separates operational payload communication from key management. Local Security Agents enforce integrity, group-level authenticity, and optional confidentiality in the data path, while the CKM system manages group membership and key lifecycle operations through a dedicated management path. This separation allows VLSec to align with the deterministic, statically configured, and multicast-oriented nature of avionics communication while complementing existing mechanisms such as domain separation, partitioning, and external communication security.

At the same time, VLSec introduces design trade-offs that must be addressed in future work. Shared group keys provide efficient protection for multicast-oriented communication, but do not provide unique sender authentication within a group. Replay protection, freshness checks, CKMS redundancy, key rotation policies, safety-aware failure behavior, and quantitative evaluation remain necessary steps toward a practical implementation. Overall, VLSec provides a foundation for treating internal VL-based communication security and key lifecycle management as integrated platform functions in future connected aircraft.

## 8. Contact Author Email Address

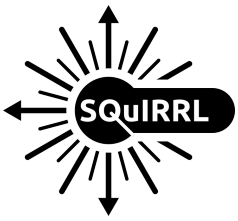
Can André Dautel
E-Mail: can.dautel@ils.uni-stuttgart.de
University of Stuttgart, Institute of Aircraft Systems (ILS)
Address: Pfaffenwaldring 27, 70569 Stuttgart, Germany

## 9. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

## Acknowledgement

The presented results are part of the work in the research project SQUIRRL (contract code: 16KIS2170), which was supported by the German Federal Ministry for Research, Technology and Space (BMFTR).







## References

[1] Airbus. Skywise core [X]: One data platform to address all your aircraft operations challenges. Product Webpage, 2026. Available at https://www.aircraft.airbus.com/en/services/enhance/skywise-data-platform/skywise-core-x (Accessed: 2026-05-27).

[2] Airbus. Airbus launches demonstrator to test global 5G connectivity in orbit. Press Release, 2026. Available at https://www.airbus.com/en/newsroom/press-releases/2026-01-airbus-launches-demonstrator-to-test-global-5g-connectivity-in (Accessed: 2026-05-27).

[3] Nils Mäurer, Thomas Gräupl, and Corinna Schmitt. RFC 9372: L-band digital aeronautical communications system (LDACS).

[4] Diana Lang. Connected aircraft. Presentation, 2025. Available at https://www.icao.int/sites/default/files/APAC/Meetings/2025/2025%20CNS%20SG29/5-Presentations/SP04_USA-AI.14-Connected-Aircraft.pdf (Accessed: 2026-06-14).

[5] EU/US task force. Future connectivity for aviation. White Paper, 2022. Available at https:https://www.easa.europa.eu/en/downloads/137252/en (Accessed: 2026-06-14).

[6] ARINC. ARINC 811 - commercial aircraft information security concepts of operation and process framework. Standard, Aeronautical Radio Incorporated, 2005.

[7] Seamless Air Alliance. Introduction to sr1. White Paper. Available at https://www.seamlessalliance.com/wp-content/uploads/SAA-White-Paper-Design-Layout-Final-Spreads.pdf (Accessed: 2026-06-14).

[8] Thomas Ewert, Nils Mäurer, and Thomas Gräupl. Concept of group key distribution procedures for the l-band digital aeronautical communications system (LDACS).

[9] Nils Mäurer, Thomas Gräupl, Christoph Gentsch, Tobias Guggemos, Marcel Tiepelt, Corinna Schmitt, and Gabi Dreo Rodosek. A secure cell-attachment procedure of LDACS. In *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 113–122.

[10] Behnam Kamali. *AeroMACS: An IEEE 802.16 Standard-Based Technology for the Next Generation of Air Transportation Systems*. Wiley, 1 edition.

[11] Can André Dautel, Matthias Lehmann, and Edwin Isidory. Introducing quantum key distribution for post-quantum secure aircraft maintenance and communication interfaces. In *2025 AIAA DATC/IEEE 44th Digital Avionics Systems Conference (DASC)*, pages 1–7, 2025.

[12] Christian Fuchs. The evolution of avionics networks from ARINC 429 to AFDX. 65.

[13] ARINC. ARINC664P1 - aircraft data network, part 1, systems concepts and overview. Standard, Aeronautical Radio Incorporated, 2019.

[14] ARINC. ARINC 823P1 - datalink security, part 1 - acars message security. Standard, Aeronautical Radio Incorporated, 2007.

[15] Edan Habler, Ron Bitton, and Asaf Shabtai. Assessing aircraft security: A comprehensive survey and methodology for evaluation. *ACM Comput. Surv.*, 56(4), November 2023.

[16] ARINC. ARINC664P5 - aircraft data network, part 5, network domain characteristics and interconnection. Standard, Aeronautical Radio Incorporated, 2005.

[17] Michael L. Olive, Roy T. Oishi, and Stephen Arentz. Commercial aircraft information security-an overview of ARINC report 811. In *2006 ieee/aiaa 25TH Digital Avionics Systems Conference*, pages 1–12.

[18] ARINC653P0-4 - avionics application software standard interface, part 0, overview of arinc 653. Standard, Aeronautical Radio Incorporated, 2025.

[19] Subhabrata Rana, Fatemeh Khoda Parast, Brett Kelly, Yang Wang, and Kenneth B. Kent. A comprehensive survey of cryptography key management systems. 78:103607.

[20] Elaine Barker. Recommendation for key management: Part 1 – general.

[21] Hans Delfs and Helmut Knebl. *Symmetric-Key Encryption*, pages 11–31. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[22] K Varner, W Zaeske, S Friedrich, A Kaiser, and A Bowman. Agile, post-quantum secure cryptography in avionics. *CEAS Aeronautical Journal*, 2026.

[23] Mario Werthwein, Matthias Brunner, and Bjoern Annighoefer. A concept enabling cybersecurity for a self-adaptive avionics platform with respect to rtca do-326 and rtca do-356. In *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*, pages 1–10, 2023.

[24] B Annighoefer, J Reinhart, M Brunner, and B Schulz. Requirements and concept for a self-organizing plug&fly avionics platform. In *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, 2021.

[25] Nils Mäurer, Thomas Gräupl, Corinna Schmitt, Gabi Dreo Rodosek, and Helmut Reiser. Advancing the security of ldacs. *IEEE Transactions on Network and Service Management*, 19(4):5237–5251, 2022.

[26] Monowar Hasan, Ashish Kashinath, Chien-Ying Chen, and Sibin Mohan. Sok: Security in real-time systems. *ACM Comput. Surv.*, 56(9), April 2024.

[27] Vuk Lesi, Ilija Jovanov, and Miroslav Pajic. Integrating security in resource-constrained cyber-physical systems. *ACM Trans. Cyber-Phys. Syst.*, 4(3), May 2020.

[28] Richard J. Thomas, Mihai Ordean, Tom Chothia, and Joeri de Ruiter. Traks: A universal key management scheme for ertms. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, ACSAC '17, page 327–338, New York, NY, USA, 2017. Association for Computing Machinery.

[29] Akila Herath, Chen-Ching Liu, Junho Hong, and Kuchan Park. Evaluation of real-time mitigation techniques for cyber security in iec 61850 / iec 62351 substations, 2025.

[30] Eric Rescorla. The transport layer security (TLS) protocol version 1.3. Num Pages: 160.

[31] Eric Rescorla, Hannes Tschofenig, and Nagendra Modadugu. RFC 9147: The datagram transport layer security (DTLS) protocol version 1.3.