A Secure Cell-Attachment Procedure of LDACS

Nils Mäurer, Thomas Gräupl
Institute of Communication and Navigation
German Aerospace Center (DLR)
Wessling, Germany
{*nils.maeurer, thomas.graeupl*}*@dlr.de*

Christoph Gentsch
Institute of Data Science
German Aerospace Center (DLR)
Jena, Germany
{*christoph.gentsch*}*@dlr.de*

Tobias Guggemos
Institute of Earth Observation
German Aerospace Center (DLR)
Wessling, Germany
{*tobias.guggemos*}*@dlr.de*

Marcel Tiepelt
KASTEL: Cryptography and Security Group
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
{*marcel.tiepelt*}*@kit.edu*

Corinna Schmitt and Gabi Dreo Rodosek
Research Institute CODE
Universität der Bundeswehr München
Neubiberg, Germany
{*corinna.schmitt, gabi.dreo*}*@unibw.de*

***Abstract*—In Europe the Single European Sky air traffic management master plan foresees the introduction of several modern digital data links for aeronautical communications. The candidate for long-range continental communications is LDACS. LDACS is a cellular, ground-based digital communications system for flight guidance and communications related to safety and regularity of flight. Hence, the aeronautical standards for cybersecurity of the link layer and the network layer apply. In previous works, threat- and risk analyses of LDACS were conducted, a draft for an LDACS cybersecurity architecture was introduced, algorithms proposed, and the security of the Station-to-Station (STS)-based Mutual Authentication and Key Establishment (MAKE) procedure of LDACS formally verified. However, options for cipher-suites and certificate management for LDACS are still missing. This paper proposes a cell-attachment procedure, which establishes a secure LDACS communication channel between an aircraft and corresponding ground-station upon cell-entry of the aircraft, that addresses these shortcomings. It introduces a full cell-attachment protocol including cipher-suites and certificate revocation for LDACS.**

***Index Terms*—Cybersecurity, Authentication, Key Establishment, SIGMA, LDACS, Tamarin, Control Channel Protection, Communication Performance**

1. Introduction

EUROCONTROL estimates European air traffic to recover from the COVID-19 pandemic by 2024 to 74% to up to 105% compared to the air traffic level of 2019 [1]. With increasing recovery, shortcomings of the current Air Traffic Management (ATM) become pressing once again. Increasing saturation of the Very High Frequency (VHF) band in some regions of the world, such as Europe [2], lacking digitalization, bandwidth and cybersecurity [3] are all obstacles for civil air traffic growth. In Europe, the SESAR ATM Master Plan [4] foresees the introduction of several modern digital data links for ATM communications. The candidate for long-range terrestrial communications, covering the En-Route (ENR) phase of flight, is L-band Digital Aeronautical Communication System (LDACS). LDACS is a cellular, ground-based digital communications system for flight guidance and communications related to the safety and regularity of flight [5]. Internationally, LDACS is reflected in the Global Air Navigation Plan (GANP) of International Civil Aviation Organization (ICAO), and is currently under standardization [6].

LDACS will be one link layer technology transporting data in ICAO's Air Traffic Network/IP-Protocol Suite (ATN/IPS) network [7]. Hence relevant aeronautical standards for the cybersecurity of the link layer technology itself, the network infrastructure it is deployed in and relevant applications enabled by LDACS apply. Those are ICAO Doc 9896 [7] and Radio Technical Commission for Aeronautics DO-379 [8].

All these documents define access control, options to protect user data in transit on link layer, and protection of the control plane of the radio access technology, as a requirement to be incorporated into the Air Traffic Network (ATN)/IP-Protocol Suite (IPS) network. In previous works, threat- and risk analysis of LDACS were conducted [9], [10], a draft for an LDACS cybersecurity architecture introduced [11], [12], algorithms proposed [13], [14] and the security of the Station to Station (STS)-based Mutual Authentication and Key Establishment (MAKE) procedure of LDACS formally verified [15]. Thereby, the low data rates of aeronautical systems, which originates from low limited dedicated spectrum for civil aviation and resulting 500 kHz channel sizes for LDACS [16], is respected. Hence, the rationale for the cell-attachment procedure is to reduce the amount of security message exchanges between GS and AS and the overall amount of the LDACS security overhead [11], [13].

Previous works were missing options for cipher suites for LDACS, negotiation of security algorithms, enabling security goals such as authentication or message integrity protection, as well as the possibility to check for the validity of LDACS certificates. As previous MAKE procedures did not take these requirements into account, a remodelled, more efficient cell-attachment procedure with as few messages and security data as possible on the link,

is missing.

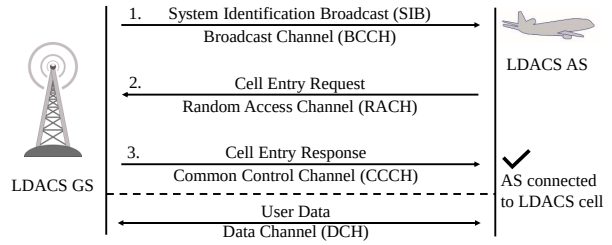

Figure 1: LDACS cell entry procedure [16]

The objective of this paper is to propose a cell-attachment procedure establishing a secure LDACS communication channel between an aircraft and corresponding ground-station upon cell-entry of that aircraft. Thus, Section 2 includes all required technical background knowledge and is completed with related work presented in Section 3. Section 4 pins down the design goals on the envisioned protocols leading to the protocol specification (cf. Section 5). The prove of reaching the design goals and envisioned functionality is depicted in Section 6 before concluding the paper.

2. Background on LDACS

LDACS is a ground-based digital bidirectional communications system for flight guidance and communications related to the safety and regularity of flight [5]. It has been developed in Europe and is currently under standardization in ICAO [6], [16]. It covers current Air Traffic Services (ATS), Aeronautical Operational Control (AOC) data and also foresees future applications, enabling new concepts such as sectorless ATM. A single LDACS cell can serve up to 512 Aircraft Station (AS) that communicate to an LDACS Ground Station (GS) in the Reverse Link (RL). The GS communicates to the AS in the Forward Link (FL). LDACS offers dynamic Coding and Modulation Scheme (CMS) depending on channel quality and enables 230.53 to 1428.27 kbps in the FL and 235.30 to 1390.40 kbps in the RL per LDACS cell, which is up to 90 times the net capacity than the currently used terrestrial links like the VHF Digital Link Mode 2 (VDLm2) system [16]. For the cell-attachment procedure of LDACS, a basic understanding of the LDACS cell entry procedure, as well as logical channels and the difference between user and control plane is required. To enter an LDACS cell, served by a GS, an AS has to undergo an initial cell entry procedure shown in Figure 1.

Every GS sends a continuous stream of data in the FL consisting of the Broadcast (BC), where the GS announces the existence of that LDACS cell, and Common Control (CC) control channel, where resources are allocated to AS, and the user Data Channel (DCH). Every AS sends in data bursts using resources allocated by the GS. Hence, the RL consists of the Random Access (RA), where AS requests access to an LDACS cell, the Dedicated Control (DC) control channel, where AS request resources, that allow them to send user data, and the user DCH. LDACS data is transported in Physical Layer Service Data Unit (PHY-SDU), e.g., with CC PHY-SDU referring to a total possible amount of 728 b CC control data that can be transported within this specific PHY-SDU. [16]

TABLE 1: AMS Security Algorithms

Scheme	Implementation
Signature	Elliptic Curve Digital Signature Algorithm (ECDSA) with (1) SHA-256 (256 b) or (2) SHA-1 (160 b)
Key Establishment	Elliptic Curve Diffie-Hellman (ECDH) unified static model per ANSI X9.63
Key Derivation	Key Derivation Function (KDF) per ANSI X9.63 with SHA-256 as underlying hash-algorithm using as input: (1) the shared secret calculated from combining the static public key $Q_{U/V}$ and static private key $q_{V/U}$ (2) an initialization time *InitTime = Time in UTC* measured by the AS and (3) a random value $Rand_V$ chosen by the GS as input values.
Message Authentication Code	Hash-based Message Authentication Code (HMAC) with SHA-256 truncated to 128, 64, 32 b most significant bits (32 b by default defined in ICAO Doc 9705 [20])
Confidentiality	Either NULL encryption algorithm per RFC 2410 [21] or AES with 128-bit block size in Cipher-Feedback (CFB) mode (AES-128-CFB128)

3. Related Work

Here relevant security details of related terrestrial aeronautical communications systems Aeronautical Mobile Airport Communication System (AeroMACS) and Aircraft Communications Addressing and Reporting System (ACARS) are introduced with a special focus the at requirements for LDACS security. The section closes with comparing different MAKE protocols, cipher suite options, and certificate revocation schemes with each other to pave the ground for the design goals and envisioned protocol presented in this paper.

3.1. ACARS Message Security (AMS)

AMS offers two secure session establishment protocols: (1) based on public/private keys [17] and (2) based on a pre-shared secret key [18]. The first assumes an AMS specific Public Key Infrastructure (PKI) to be in place with corresponding Certificate Revocation List (CRL) and Certificate Distribution Center (CDS). The second assumes that a shared secret key has been agreed prior to secure session establishment attempt. This variation of the protocol behaves similarly like 4G, where a permanent key K is shared between User Equipment (UE), more specifically within the Universal Subscriber Identity Module (USIM) in the Universal Integrated Circuit Card (UICC), and Home Subscriber Service (HSS), more specifically within the Authentication Center (AuC) [19]. As LDACS is very likely to have its own PKI, the analysis is focused on the first AMS secure session establishment protocol.

Looking at the proposed protocols in [17], [18], especially at the two-way authentication, certificate exchange and key establishment procedure, some problems become apparent. For instance Blanchet et. al [22] identified a *key compromise impersonation attack* if the long-term key of an aircraft is compromised, due to session keys being derived taking the long-term keys as input. One possible solution is using ephemeral public/private Diffie-Hellman key pairs for that purpose.

Furthermore, the cipher suites are also limited: (1) key establishment only foresees ECDH and no other alternative additionally; (2) user data confidentiality is either not ensured in any way or the only other option is AES128-CFB128; (3) message integrity and data origin-authenticity is not provided or only by HMAC-256 with truncated to 32 b. Finally certificate revocation is handled poorly: while the GS requests signed CRL records to check the validity of the aircraft's certificate, the AS never receives proof of the validity of the GS certificate. These shortcomings of AMS should be avoided in the security design of LDACS.

3.2. AeroMACS

AeroMACS Minimum Operational Performance Standards [23] mainly refer to IEEE 802.16-2009 standard [24] for the implementation of security: The key management protocol of AeroMACS uses Extensible Authentication Protocol (EAP) [25] or a PKI with X.509 digital certificates [26] together with PKCS#1 v2.1 [27], or a sequence of RSA authentication first, followed by EAP. The used key management protocol by AeroMACS is PKMv2 [23]. Here, the focus is on the certificate-based part. Every AS carries a unique X.509 certificate issued by the AS manufacturer binding the AS Message Authentication Code (MAC) address to the RSA encryption key. The certificates are based on certificate profiles, which are in turn based on X.509 v03 certificates. The information contained in the AS certificate is explicitly listed for completeness and comparison with LDACS:

```
countryName = <Country Manufacturer>
organizationName = <Company Name>
organizationalUnitName = <City Manufacturer>
commonName = <Serial Number>
commonName = <MAC Address>
```

The GS certificates have the same fields with two adoptions:

```
organizationalUnitName = WirelessMAN
commonName = <BSID>
```

AeroMACS uses Security Association (SA), which are sets of security information a GS and one or more AS share to support secure communications across a network. Three SAs (primary, static and dynamic) are defined [24]. The primary SA is obligatory and set up during the initial connection establishment between AS and GS during the authorization process. The used PKMv02 has two phases for AeroMACS:

Phase 1 deals with Authentication and Authorization. The AS first presents the certificate of its Certificate Authority (CA). In a second message a 64 b random value, its certificate, a Security Association ID (SAID) and an RSA signature over all fields are sent [24], [28]. The GS replies with the third message, including the previous and a new random value, the pre-Primary Authorization Key (pre-PAK) encrypted with the public key of the AS, the lifetime and sequence number of the PAK, its certificate and an RSA-based signature over all attributes in the message. At this point, AS and GS are mutually authenticated and use the *Dot16KDF* to derive the PAK. Optionally, the EAP procedure may follow, before the Authorization Key (AK) is derived via *Dot16KDF*, PAK and previously exchanged input parameters.

TABLE 2: AeroMACS Security Algorithms [23]

Scheme	Implementation
Signature	RSA signature algorithm defined in PKCS #1 [27] with SHA-1 [29]
Key Establishment	GS chosen *pre-PAK* encapsulation via RSA and public key of AS $\rightarrow AK$ key derivation via *Dot16KDF* based on *EAP* or *RSA* or both $\rightarrow KEK$s and $H/CMAC$ keys are derived from $AK \rightarrow TEK$ is generated by the GS and transmitted encrypted via AES and the KEK: Hence the TEK results from AES key wrap with 128-bit key.
Key Derivation	Dot16KDF
Message Authentication Code	AES-128-CCM
Confidentiality	AES-128-CCM

Phase 2 represents the PKMv2 SA TEK 3-Way handshake. Cryptographic capabilities are exchanged between AS and GS: a cipher-suite is agreed upon, keys for either cipher-based MAC (CMAC) or HMAC via *Dot16KDF* and AK, the Key Encryption Key (KEK). The latter is used by the GS to encrypt the Traffic Encryption Key (TEK) used for user data protection. Cryptographic suites are encoded in 3 Bytes: (1) the encryption algorithm and key length, (2) data authentication algorithm, (3) TEK Encryption Algorithm. Although IEEE 802.16-2009 offers different cipher suite options, AeroMACS defines only one set (cf. Table 2). All three PKMv2 SA TEK 3-Way messages are protected by either CMAC or HMAC generated Message Authentication Codes (MACs).

Summarizing, AeroMACS foresees no ephemeral keys (i.e., *pre-PAK* and TEK are both chosen by GS only). It only supports one cipher suite option and has no immediate proof, if the used certificates are still valid (i.e., no CA signed CRL sent with the response).

4. Secure LDACS Cell-Attachment: Design Goals

The design of a security mechanism in LDACS cell-attachment copes with requirements established in previous work. Further, standard protocols and well-established approaches for embedding cryptography are re-used wherever possible.

Previous security analysis of LDACS identified **requirements of the cell-attachment procedure** [10], [11], [13], [15]: *Mutual Authentication*, *Perfect Forward Secrecy* and *Secure Key Establishment* in event of an adversary or compromise of long-term keys. This requires the key establishment method to be based on ephemeral keys, where both parties contribute to the final shared secrets and the inputs are chosen freshly for every protocol run. LDACS can be used for multiple purposes and depending on its use, different security levels are desired. This requires multiple options for cipher suites, authentication and key establishment. Additionally, validity of certificates based on a mutually trusted entity, such as a CA, has to be proven during the cell-attachment procedure. As

explained in Section 2, the LDACS data plane is split into user- and control-data. The user-data data plane must provide integrity and message origin-authentication and can optionally provide confidentiality [7], [30]. To protect the control plane of LDACS, specifically the CC channel, where resource allocations take place, a concept for Group Key Management was introduced [12], [31].

To ensure **robustness**, mechanisms of existing protocols are re-used to integrate security into the LDACS cell-attachment process. In particular, the SIGMA protocol for mutual authenticated key agreement using four messages is used [32]. Hence, the protocol introduced below and in Figure 2 will bear some similarities to the SIGMA [32] and the Internet Key Exchange version 2 (IKEv2) [33] protocol.

It is foreseen that LDACS has different **security levels** with **different cipher-suites**. One reason for defining higher, post-quantum security levels, is the lifetime of digital aeronautical communications systems, with legacy systems such as VDLm2 existing since the 90s [34]. Another reason is the variety of traffic transported by LDACS: AOC data requires higher data protection, as it contains more personal or company-related information. ATS data on the other hand has to be integer, while being available and readable by all aircraft in the vicinity, as well as multiple international air traffic control offices.

First, LDACS needs to provide options for different Diffie-Hellman Key Exchange (DHKE) together with public Diffie-Hellman (DH) parameter, following the work done in [13] and looking at TLS 1.3 [35], these can be ephemeral DH based on the discrete logarithm problem in elliptic curves (e.g. ECDH) but also quantum-resistant candidates, such as Supersingular Isogeny Key Encapsulation (SIKE) [36]. Secondly, different security levels must be reflected in the used cryptography of the accepted certificate, hence the certificate type and version must be communicated, as well as the signature- and hash-algorithm. Thirdly, using the idea of Authenticated Encryption with Associated Data (AEAD), introduced in TLS 1.3 [35], respective algorithms protecting user data must be agreed upon, as well as underlying hash-algorithms required by AEAD.

Certificate revocations can be tracked by an entity within a PKI via a CRL [26] or the Online Certificate Status Protocol (OCSP) [37]. Here, OCSP is chosen over CRL due to three reasons: (i) bandwidth limitations on the air gap are a major concern for LDACS and OCSP requires less network bandwidth, (ii) the ground-connection from a GS to a CDS has several magnitudes more throughput than the wireless LDACS connection, hence regular updates enable near real-time status checks via OCSP, and (iii) since AS and GS both rely on trust derived from a CA higher up in their chain of trust, they both can trust a CA signed OCSP message, guaranteeing the validity of a certificate of a communication partner at a certain point in time.

5. Secure LDACS Cell-Attachment: The Protocol

With the design goals at hand, security mechanisms are embedded into the existing LDACS cell-attachment.

TABLE 3: Content of LDACS GS certificates

Field	Value
Version	Positive integer
Serial Number	Positive integer generated by issuing CA
Issuer Signature Algorithm	LDACS Security Level 1: ECDSA, SHA-256 [38] LDACS Security Level 2: ECDSA, SHA-384 [38] LDACS Security Level 3: Falcon512 [39] LDACS Security Level 4: Falcon1024 [39]
Issuer Signature Value	Bit string calculated by Issuing CA on ASN.1 DER-encoded tbsCertificate [26]
Issuer Distinguished Name	ID = <UA> C = <COUNTRY> O = <PKI Operating Organization> CN = <PKI Operating Organization>
Validity Period	notBefore: set by Issuing CA, time of certificate creation notAfter: set by Issuing CA, notBefore + 1 year
Subject Distinguished Name	ID = <UA> C = <COUNTRY> O = <PKI Operating Organization> CN = <GS Operating Organization>
Subject Public Key Information	key for every security level: bit string extracted from certificate signing request LDACS Security Level 1: algorithmIdentifier: ID-EC256PublicKey parameter: P-256/brainpoolP256r1 [38], [40] LDACS Security Level 2: algorithmIdentifier: ID-EC384PublicKey parameter: P-384/brainpoolP384r1 [38], [40] LDACS Security Level 3: algorithmIdentifier: ID-Falcon512PublicKey [39] LDACS Security Level 4: algorithmIdentifier: ID-Falcon1024PublicKey [39]
Issuer's Signature	LDACS Security Level 1: ECDSA, SHA-256 [38] LDACS Security Level 2: ECDSA, SHA-384 [38] LDACS Security Level 3: Falcon512 [39] LDACS Security Level 4: Falcon1024 [39]

TABLE 4: LDACS AS specific certificate content differing from GS certificate content

Field	Value
Validity Period	notBefore: set by Issuing CA, time of certificate creation notAfter: set by Issuing CA, notBefore + 3 year
Subject Distinguished Name	ID = <UA> C = <COUNTRY> O = <PKI Operating Organization> OU = < *"ICAO airline three-letter designator"* > CN = < *"air device subject CN"* >

5.1. LDACS Certificates and Certificate Handling

Following the recommendations for network nodes in the ATN/IPS from ICAO Doc 9896 [7] and from DO-379 [8] the content of LDACS GS certificates is depicted in Table 3. The differences of an AS certificate to the GS certificate are depicted in Table 4. Since LDACS supports different levels of security, quantum-resistant signature schemes are only necessary from level two upwards. From this point, however, the LDACS PKI must be entirely based on quantum resistant schemes.

Two strategies are combined to save the bandwidth to transmit certificate data during during flight: (1) end-entity certificates, together with the certificate chain up to its root of trust shall be stored securely in local storage [7] and (2) all relevant GS certificates shall

be installed upon relevant AS prior the flight during maintenance, i.e. for the flight's geographical region.

LDACS AS and GS will be part of the same LDACS PKI with the requirement of a CA higher up in the trust chain, that both entities trust. As an end-entity certificate compromise is much more likely to occur during the relatively short relevant time of flight compared to a sub-CA takeover, due to the CAs using stronger cryptography than end-nodes, it is assumed that CAs in the chain of trust remain trustworthy during the time of flight. Hence a trusted CA signed OCSP confirmation of the validity of one end-entity certificate, is assumed to be trustworthy during flight. ICAO Doc 9896 [7] defines validity check update rates for offline CAs every 45 days and for online CAs every 48h. The update rates for validity checks of respective end-entity certificates are defined at every new LDACS cell-attachment attempt for both, the AS and GS certificate. With that, the GS requests the update status of AS and GS certificate from a CDS via a secure ground connection every time, an AS attempts to join an LDACS cell. Finally, the validity proof of the GS must also be part of the LDACS cell-attachment procedure depicted in Figure 2.

5.2. LDACS Cipher Suites

Four different security levels are defined, two per- and two post-quantum, with additional ones to be defined in the future if necessary. LDACS supports two kinds of algorithm lists: The first `EPLDACS` represents choices regarding MAKE and either integrity and authenticity protection of user data only or AEAD for user data protection, the second `CCLDACS` choices for Group Key Management (GKM) and MAC for control data protection.

AEAD is applied onto LDACS user-data Sub-Network (SN) Packet-Data Units (PDUs) [11], which are 128 B to 1536 B long [16] and may therefore limit the choice of cryptographic algorithms. Overall, security level 1-2 defines pre-quantum algorithms, with 1 using a 128 b key and 2 using a 256 b key. Security Level 3-4 defines post-quantum algorithms, with 3 using a 128 b key and 4 using a 256 b key again. As AES-CMAC for MAC generation and verification, hence integrity and authenticity protection of messages, and AES-CCM for AEAD remains secure, also with the possible threat of quantum computers [41], on security Level 3-4 only the key establishment and signature algorithms are updated to post-quantum cryptography [41]. Hence, the use of efficient Elliptic Curve Cryptography (ECC) based signature schemes (e.g. ECDH) for security level 1-2 with *P-256/brainpoolP256r1* at security level 1 and *P-384/brainpoolP384r1* curves at security level 2 are specified. Security Level 3 and above requires quantum-resistant key-lengths and schemes. Isogeny based ephemeral key establishment such as SIKE and the post-quantum signature scheme *Falcon* with corresponding public parameters are used for this purpose. Among the post-quantum signature candidates, those were selected that have the smallest signature plus public key size, as it is possible that certificates are transmitted via LDACS during the cell-attachment procedure: At security Level 3, *Falcon512* is selected among the three current National Institute of Standards and Technology (NIST) Post-Quantum Cryptography competition round 3 candidates, as its signature is with 666 B[1] almost four times smaller as the other final-round candidate Dilithium with 2420 B[2]. Also public key sizes of *Falcon512*, with 897 B, are considerably smaller than the only other post-quantum candidate Rainbow-I, that offers a smaller signature size of 528 B but at the cost of a public key size of 157.8 kB [42]. At security Level 4 *Falcon1024* is used. Finally, *SIKEp434_c* and *SIKEp751_c* are chosen as post-quantum key establishment algorithms for their relatively small public key sizes of 197 B, 315 B respectively [36]. All suggested security algorithms within `EPLDACS` and `CCLDACS` are listed in Table 5 and Table 6, for the four specified security levels.

TABLE 5: LDACS `EPLDACS` Security Algorithms (*SL = Security Level)

Scheme	Implementation
Signature	SL 1: ECDSA, SHA-256, P-256/brainpoolP256r1 [38], [40] SL 2: ECDSA, SHA-384, P-384/brainpoolP384r1 [38], [40] SL 3: Falcon512 [39] SL 4: Falcon1024 [39]
Key Establishment	SL 1: ECDH, P-256/brainpoolP256r1 [38], [40] SL 2: ECDH, P-384/brainpoolP384r1 [38], [40] SL 3: SIKEp434_c [36] SL 4: SIKEp751_c [36]
Key Derivation	SL 0-3: HMAC Key Derivation Function (HKDF) [43]
Message Authentication (only)	SL 1: 96 b tag, 128 b key, AES-128-CMAC [44] SL 2: 128 b tag, 256 b key, AES-256-CMAC [44] SL 3: 96 b tag, 128 b key, AES-128-CMAC [44] SL 4: 128 b tag, 256 b key, AES-256-CMAC [44]
Please note: Choice of *Message Authentication (only)* or *AEAD* are mutually exclusive.	
Authenticated Encryption with Associated Data (AEAD)	SL 1: 96 b tag, 128 b key, AES-128-CCM [45] SL 2: 128 b tag, 256 b key, AES-256-CCM [45] SL 3: 96 b tag, 128 b key, AES-128-CCM [45] SL 4: 128 b tag, 256 b key, AES-256-CCM [45]

Following the notations of TLS 1.3 [35], the cryptographic methods of `EPLDACS` are negotiated as `LDACS_ECDHE_ECDSA_WITH_AES_128_CCM_SHA256_P256` as an example for LDACS security level 1 with AEAD in place. Where no encryption is required `LDACS_ECDHE_ECDSA_WITH_NULL_AES_128_CMAC_SHA256_P256` offers an alternative, providing message integrity and authenticity protection only.

GKM allows protecting the CC channel of LDACS [31]. Thus, relevant algorithms to protect the control data plane of LDACS are also specified, as well as corresponding Traffic Encryption Key (TEK) and Key Encryption Key (KEK) within `CCLDACS`. A detailed description of the protection of LDACS the control plane is not in scope of this work, and only the preliminary algorithms are presented here.

5.3. The LDACS Cell-Attachment Procedure

With the cipher suite options for LDACS being defined, the LDACS cell-attachment procedure is formulated

1. https://falcon-sign.info/, accessed 07/16/2021
2. https://pq-crystals.org/dilithium/index.shtml, accessed 07/16/2021

TABLE 6: LDACS CCLDACS Security Algorithms (*SL = Security Level)

Scheme	Implementation
GKM	SL 0-3: One-way Function Tree (OFT) [46]
Key Establishment (TEK and KEK encapsulation)	SL 1: AES-128-CCM [45] SL 2: AES-256-CCM [45] SL 3: AES-128-CCM [45] SL 4: AES-256-CCM [45]
CC PHY-SDU protection	SL 1: 96 b tag, 128 b key, AES-128-CMAC [44] SL 2: 128 b tag, 256 b key, AES-256-CMAC [44] SL 3: 96 b tag, 128 b key, AES-128-CMAC [44] SL 4: 128 b tag, 256 b key, AES-256-CMAC [44]
DC PHY-SDU protection	SL 1: 75 b tag, 128 b key, AES-128-CMAC [44] SL 2: 75 b tag, 256 b key, AES-256-CMAC [44] SL 3: 75 b tag, 128 b key, AES-128-CMAC [44] SL 4: 75 b tag, 256 b key, AES-256-CMAC [44]
Key Derivation	SL 1-4: HKDF [43]

as illustrated in Figure 2. Basic entities of communication are the CDS together with OCSP server, the GS and the AS. Moreover a generic public key agreement scheme is considered: If x is an ephemeral secret key, then $\mathrm{P} \leftarrow f(\mathrm{x})$ is the ephemeral public key. $\mathrm{z} \leftarrow g(\mathrm{P}', \mathrm{x})$ is a shared secret computed from public key P' and a secret x. ENC_K denotes the encryption under key K.

Steps 1-3: The GS regularly checks the validity of its locally stored certificates $Cert_{GS}$ and $Cert_{AS_i}$ with the CDS and also receives signed OCSP batches. This additionally enables the AS to verify its own certificate.

Steps 4-9: In a BC message, the GS announces its existence together with its unique identifier UA_{GS}, LDACS specific address SAC_{GS} and current certificate fingerprint. The approaching AS receives this beacon, stores the GS identifiers and checks whether or not the GS certificat is stored it locally or not. The $Status_{Cert_{GS}}$ flag is set to "1" else to "0", respectively. The AS replies in the RA revealing its identifier UA_{AS}. The GS checks whether the AS certificate is stored locally and valid or not and assigns the LDACS specific address SAC_{AS} to the AS. The SAC_{AS}, all supported EPLDACS and CCLDACS cryptographic options and a nonce N_{GS} is then sent to the GS. With that the cell entry procedure of LDACS is done and the MAKE protocol begins via the opened DCH channel.

Steps 10-11: The AS chooses respective DHKE and AEAD algorithms from the provided options and stores that choice in the *algo* parameter. It also stores the received stores SAC_{AS}, and the CCLDACS algorithms. Depending on the chosen algorithms a private key x_{AS} is chosen, the public key P_{AS} is computed, and a nonce N_{AS} generated. All this is sent to the AS in the first MAKE message.

Steps 12-13: The GS verifies N_{GS}, chooses its own private key x_{GS}, calculates P_{GS} and responds with the $Cert_{GS}$ (if requested), N_{AS}, P_{GS}, its OCSP validity proof and respective the CA signature.

Steps 14-15: The AS verifies the validity of the $Cert_{GS}$. It further calculates the shared secret z, from which it derives three keys: (1) the session key $K_{AS,GS}$, (2) the MAC key K_M and (3) the encryption key K_E, of which (2,3) are only used in the MAKE phase. Using K_M an HMAC m_{AS} over all identifiers is calculated and signed together with *algo*, t_{AS}, N_{AS}, N_{GS} in σ_{AS} Finally σ_{AS} is encrypted with K_E and sent to the GS. Please note: the AS uses the algorithms from *algo*. In case no confidentiality option was negotiated and the MAC part in EPLDACS does not contain an encryption option (i.e., some HMAC variation), then AES-128-GCM is used as minimum default.

Steps 16-18: First the GS now calculates the shared ephemeral secret z, from which it derives three keys: (1) the session key $K_{AS,GS}$, (2) the MAC protocol key K_M and (3) the encryption protocol key K_E. Now the message $\mathrm{ENC}_{K_E}(\sigma_{AS})$ can be decrypted, m'_{AS} be generated, and the signature σ_{AS} can be verified. Upon success, the AS is authentic and the GS builds a MAC of its own with using the identifiers in GS then AS order, m_{GS} and builds σ_{GS} with EPLDACS, CCLDACS, P_{GS}, P_{AS}, N_{GS}, N_{AS} and m_{GS}. Finally σ_{GS} is encrypted with K_E and sent to the AS.

Step 19: The AS decrypts $\mathrm{ENC}_{K_E}(\sigma_{GS})$, builds m'_{GS} and verifies σ_{GS}. Upon success, the GS is authentic, both share the session key $K_{AS,GS}$ and have already reached key confirmation with K_M and K_E.

The DCH after cell entry procedure is chosen for the MAKE protocol, as the LDACS DCH is flexible in size and can react to changes in the underlying cryptography.

6. Formal Security Verification of the LDACS Cell-Attachment Procedure

Here the LDACS cell-attachment procedure is formally verified with the symbolic model checker "Tamarin" [47]. In symbolic verification, the modeled cryptographic schemes are treated as black boxes meaning that signature and encryption mechanisms are assumed secure as long as the appropriate keys are unknown to the adversary.

6.1. Tamarin Model of the LDACS Cell-Attachment Procedure

The protocol is modelled as depicted in Figure 2 in Tamarin's specific rule-based declarative language. A rule can consume and produce so-called *facts*, whereby emitting so-called *events*. The latter can be reasoned with via the specified *lemmas*. Every security property like *secrecy* or *authentication*, which the protocol is supposed to fulfill, has to be reasoned and specified by the user.

Following the standards and recommendations from the Tamarin documentation[3] led to modelling each role (CA, AS, GS) using state-facts that can be instantiated infinitely, separated by public IDs and private session-IDs. Each role starts with its own (ICAO)-ID and public-key pair, which is generated by the standard PKI rule, taken from the documentation. The CA is modeled only in terms of a public-key pair which is generated once, and then can be used by AS and GS to verify corresponding certificates. To keep the model concise, OCSP-messages were not modelled.

3. https://tamarin-prover.github.io/manual/, accessed 07/16/2021

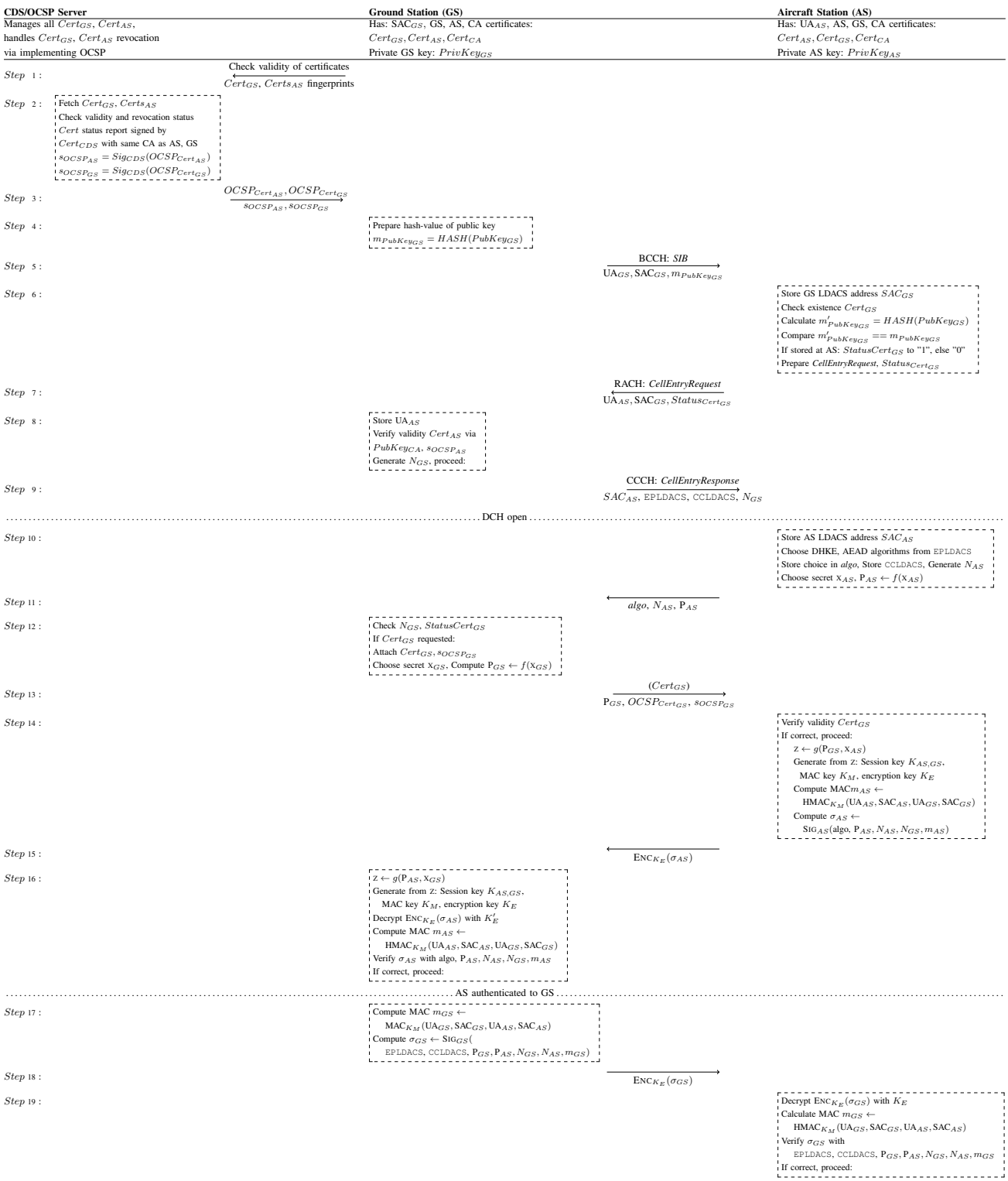

Figure 2: LDACS Cell-Attachment Procedure.

```
rule Register_pk:
  [ Fr(~ltkX) ]
  --[ OnlyOnceV($X) ]->
  [
    !Ltk($X, ~ltkX)
    , !Pk($X, pk(~ltkX))
    , Out(pk(~ltkX))
  ]

rule CA_init:
  [ Fr(~ltk) ]
  --[ OnlyOnce() ]->
  [
    !LtkCA(~ltk)
    , !PkCA(pk(~ltk))
    , Out(pk(~ltk))
  ]
```

Listing 1: Tamarin rules to model a PKI

```
// Compromising an agent's long-term key
rule Reveal_ltk:
  [ !Ltk($X, ltkX) ] --[ Corrupted($X) ]-> [ Out(ltkX) ]
```

Listing 2: Tamarin rules for key compromise of long-term key

```
lemma executable_a:
  exists-trace
  "Ex A B ia ib x y #i #j #k #l #m #n #o #p #q.
    CreateA(A, ia)@i & CreateG(B, ib)@j & AttachingWCert(A, B, ia)@k
    & Running(A, B, ia, x)@l & CheckG1(B, ib)@m & CheckA1(A, ia)@n
    & Running(B, A, ib, y)@o
    & Commit(A, B, ia, <x, y>)@p & Commit(B, A, ib, <y, x>)@q"

lemma executable_b:
  exists-trace
  "Ex A B ia ib x y #i #j #k #l #m #n #o #p #q.
    CreateA(A, ia)@i & CreateG(B, ib)@j & AttachingNoCert(A, B, ia)@k
    & Running(A, B, ia, x)@l & CheckG1(B, ib)@m & CheckA1(A, ia)@n
    & Running(B, A, ib, y)@o
    & Commit(A, B, ia, <x, y>)@p & Commit(B, A, ib, <y, x>)@q"
```

Listing 3: Lemma 1

```
lemma mutual_authentication:
  "AllA B x y ia #i. Commit(A, B, ia, <x, y>)@i ==>
    ( Ex ib #j. Running(B, A, ib, y)@j
      & j<i
      & not (Ex 2 B2 ia2 #i2. Commit(A2, B2, ia2, <x, y>)@i2 & not(#i2=#i))
    )
    | (Ex C #r. Corrupted(C)@r & Honest(C)@i & #r<#i)"
```

Listing 4: Lemma 2

```
lemma secure_key_establishment:
  "All A B ia x #i. Commit(A, B, ia, x)@i ==>
    (Ex P ib #j #m. Knows(A, ia, P, B)@m & Knows(B, ib, P, A)@j
      & not (Ex D E id #k. Knows(D, id, P, E)@k & not(#m=#k) & not(#j=#k))
    )
    | (Ex C #r. Corrupted(C)@r & Honest(C)@i & #r<#i)"
```

Listing 5: Lemma 3

```
lemma secrecy:
  "All x #i.
    Secret(x)@i ==>
      not (Ex #j. K(x)@j)
      | (Ex B #r. Corrupted(B)@r & Honest(B)@i & #r<#i)"
```

Listing 6: Lemma 4

TABLE 7: Tamarin verification results

Lemma	Scope	Result	Steps
Executable (AS has Cert)	Exists-trace	✔Verified	38
Executable (AS needs Cert)	Exists-trace	✔Verified	37
Mutual Authentication	All-traces	✔Verified	92
Secure Key Exchange	All-traces	✔Verified	1042
Perfect Forward Secrecy	All-traces	✔Verified	104

Unfortunately, some protocol feature that are introduced for efficiency make the model more complex. The possible existence of a local copy of the GS' certificate at the AS in step 7, requires distinguishing in step 13, whether or not a new certificate needs to be verified. This is modeled by alternative rules, which can both be applied at the appropriate stage of the protocol leading to different paths and is a unique feature of the Tamarin model checker. One benefit of this approach is, it can be proven that the security properties of both paths hold, with just one lemma per property - which shows that both paths are equally secure. One exclusion is the executability-lemma, which exists twice, as both paths need to be formally verified.

Tamarin follows the attacker model proposed by Dolev-Yao [48], where an ideal, powerful attacker is assumed, who can read, block and send any message to and from each agent. Additionally, the possibility is assumed that the adversary can corrupt any station and obtain their long-term secret (certificate), to prove if the protocol fulfills the perfect-forward-secrecy requirement.

6.2. Tamarin Lemmata

In this section the proven lemmata are given in a literal and formal representation.

Lemma 1: "Executable" (cf. Listing 3): There exists a trace where instance A in role AS participates in session i_a and instance B in role GS participates in session i_b, A is requesting B for cell entry, both are starting the protocol by exchanging P_{AS} and P_{GS}, and finally both commit by having the same shared data. Please note: the lemma is split in *executable_a* and *executable_b* to reflect the option of sending the GS certificate in step 11.

Lemma 2: "Mutual Authentication via Injective Agreement" (cf. Listing 4): If A finishes a run with B by exchanging y, it can be sure, B also ran the protocol with A and y has not been exchanged before in any other run. Only exception: the private key of an honest agent has been compromised before.

Lemma 3: "Secure Key Establishment" (cf. Listing 5): If A finishes a run with B, it can be sure, that it has a fresh key P and that B also has this key for use with A, and this key has not been established before, except with negligible probability, implicating that also no other agent knows it. Only exception: the private key of an honest agent has been corrupted before.

Lemma 4: "Perfect Forward Secrecy" (cf. Listing 6): The exchanged session key ($K_{AS,GS}$) cannot be known by the attacker, even when he acquires the private key of one or both parties later on. The case, the session key was leaked to the attacker, is excluded.

6.3. Result

For evaluation purposes, the Tamarin prover version 1.6.0 in automatic mode was used to prove the five lemmata presented in Section 6.2. The verification took 2 m 10 s on a Ubuntu 18.04 Laptop with an Intel(R) Core(TM) i7-8650U CPU and 16GB of RAM. All five lemmata could be verified without interaction.

The source code of the Tamarin model is available for download at GitHub[4]. The "scope" column states which type of proof has been done: 'exists-trace'-proofs verify, that the given property or lemma holds at least for one trace of the protocol; 'all-traces'-proofs respectively verify that the property holds for all traces. The last column displays the number of verification steps that were executed by Tamarin to verify the lemma. As all lemmata

4. https://github.com/kr4ck-com/LDACSCellAttProof, accessed 07/16/2021

have been proven to hold, all required security controls of the LDACS cell-attachment procedure also hold in the symbolic model.

7. Conclusion

Throughout this work, the full LDACS cell-attachment procedure, together with AS, GS certificates, possible cipher suites, and options for certificate revocations and validity checks were presented. Looking at two other aeronautical communications systems for terrestrial communications, namely AMS and AeroMACS, pros- and cons of their security design were discussed. This resulted in identifying ephemeral key establishment, multiple cipher suites and regular validity checks for certificates as design goals for the LDACS cell-attachment procedure. As LDACS will have its own dedicated PKI, the content of end-entity AS and GS certificates, together with pre- (*ECDSA*) and post-quantum (*Falcon*) signature options was discussed. Then cipher suite options for user-data AEAD and MAKE algorithms were summarized in `EPLDACS` and control-data protection algorithms in `CCLDACS`, with the main focus of this work on the first. Finally the entire LDACS cell-attachment procedure was introduced, spanning the original LDACS cell entry procedure, extending it with a 4-pass SIGMA/IKEv2 inspired MAKE protocol. Finally, the design of that procedure was formally verified with Tamarin, where it passed all tests for *Executability*, *Mutual Authentication*, *Secure Key Establishment* and *Perfect Forward Secrecy*.

For future work, the investigation of exact key derivation methods for $K_{AS,GS}$, K_E and K_M, as well as the detailed LDACS control channel protection is paramount, to arrive at a full LDACS cybersecurity architecture.

Appendix

ACARS	Aircraft Communications Addressing and Reporting System
AEAD	Authenticated Encryption with Associated Data
AeroMACS	Aeronautical Mobile Airport Communication System
AMS	ACARS Message Security
AOC	Aeronautical Operational Control
AS	Aircraft Station
ATN/IPS	Air Traffic Network/IP-Protocol Suite
ATN	Air Traffic Network
ATM	Air Traffic Management
ATS	Air Traffic Services
BC	Broadcast
CA	Certificate Authority
CC	Common Control
CDS	Certificate Distribution Center
CMAC	cipher-based MAC
CMS	Coding and Modulation Scheme
CRL	Certificate Revocation List
DC	Dedicated Control
DCH	Data Channel
DH	Diffie-Hellman
DHKE	Diffie-Hellman Key Exchange
EAP	Extensible Authentication Protocol
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ENR	En-Route
FL	Forward Link
GKM	Group Key Management
GS	Ground Station
HKDF	HMAC Key Derivation Function
HMAC	Hash-based Message Authentication Code
ICAO	International Civil Aviation Organization
IPS	IP-Protocol Suite
KDF	Key Derivation Function
KEK	Key Encryption Key
LDACS	L-band Digital Aeronautical Communication System
MAC	Message Authentication Code
MAKE	Mutual Authentication and Key Establishment
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
PHY-SDU	Physical Layer Service Data Unit
PKI	Public Key Infrastructure
RA	Random Access
RL	Reverse Link
SA	Security Association
SIKE	Supersingular Isogeny Key Encapsulation
STS	Station to Station
TEK	Traffic Encryption Key
VDLm2	VHF Digital Link Mode 2
VHF	Very High Frequency

References

[1] EUROCONTROL, "EUROCONTROL Forecast Update 2021-2024," https://www.eurocontrol.int/sites/default/files/2021-05/eurocontrol-four-year-forecast-2021-2024-full-report.pdf, accessed 07/16/2021, EUROCONTROL, Tech. Rep., 2021.

[2] ICAO, "Handbook On Radio Frequency Spectrum Requirements For Civil Aviation, Volume I, ICAO Spectrum strategy, Policy Statements And Related Information," https://standards.globalspec.com/std/10402555/ICAO9718VOLUMEI, accessed 07/16/2021, International Civil Aviation Organization (ICAO), Doc 9718, 2018.

[3] M. S. B. Mahmoud, A. Pirovano, and N. Larrieu, "Aeronautical Communication Transition From Analog to Digital Data: A Network Security Survey," *Computer Science Review*, vol. 11, pp. 1–29, 2014.

[4] SESAR JU, "European ATM Master Plan," https://www.atmmasterplan.eu/, accessed 07/16/2021, European Union (EU), Tech. Rep., 2020.

[5] M. Schnell, U. Epple, D. Shutin, and N. Schneckenburger, "LDACS: Future Aeronautical Communications For Air-Traffic Management," *Communication Magazine*, vol. 52, no. 5, pp. 104–110, 2014.

[6] ICAO, "ADS-B Implementation and Operations Guidance Document," https://www.icao.int/APAC/Documents/edocs/AIGDEdition11.pdf, accessed 07/16/2021, International Civil Aviation Organization (ICAO), Tech. Rep., 2018.

[7] ICAO, "Manual On The Aeronautical Telecommunication Network (ATN) Using Internet Protocol Suite (IPS) Standards and Protocols," https://standards.globalspec.com/std/10026940/icao-9896, accessed 07/16/2021, International Civil Aviation Organization (ICAO), Doc 9896, 2015.

[8] RTCA, "Internet Protocol Suite Profiles," https://www.rtca.org/products/do-379/, accessed 07/16/2021, Radio Technical Commission for Aeronautics (RTCA), DO-379, 2019.

[9] N. Mäurer and A. Bilzhause, "Paving The Way For An IT Security Architecture For LDACS: A Datalink Security Threat And Risk Analysis," in *18th Integrated Communications, Navigation and Surveillance Conference*. IEEE, 2018, pp. 1A2/1–1A2–11.

[10] N. Mäurer and C. Schmitt, "Towards Successful Realization Of The LDACS Cybersecurity Architecture: An Updated Datalink Security Threat- And Risk Analysis," in *19th Integrated Communications, Navigation and Surveillance Conference*. IEEE, 2019, pp. 1–13.

[11] Mäurer, N. and Bilzhause, A., "A Cybersecurity Architecture For The L-band Digital Aeronautical Communications System (LDACS)," in *37th Digital Avionics Systems Conference*. IEEE, 2018, pp. 1–10.

[12] Mäurer, Nils and Gräupl, Thomas and Schmitt, Corinna, "Cybersecurity For The L-band Digital Aeronautical Communications System (LDACS)," in *Aviation Cybersecurity: Foundations, Principles, and Applications*, H. Song, K. Hopkinson, T. de Cola, T. Alexandrovich, and L. D., Eds. Institute of Engineering and Technology (IET), 2021, pp. 1–38.

[13] Mäurer, N., Gräupl, T. and Schmitt, C., "Comparing Different Diffie-Hellman Key Exchange Flavors For LDACS," in *39th Digital Avionics Systems Conference*. IEEE, 2020, pp. 1–10.

[14] N. Mäurer, T. Gräupl, C. Schmitt, and G. Dreo Rodosek, "PMAKE: Physical Unclonable Function-Based Mutual Authentication Key Exchange Scheme For Digital Aeronautical Communications," in *IFIP/IEEE International Symposium on Integrated Network Management*. IEEE, 2021, pp. 1–9.

[15] Mäurer, Nils and Gentsch, Christoph and Gräupl, Thomas and Schmitt, Corinna, "Formal Security Verification Of The Station-to-Station Based Cell-Attachment Procedure Of LDACS," in *18th International Conference on Security and Cryptography*. SCITEPRESS Digital Library, 2021, pp. 1–8.

[16] T. Gräupl, C. Rihacek, and B. Haindl, "LDACS A/G Specification," https://www.ldacs.com/wp-content/uploads/2013/12/SESAR2020_PJ14-W2-60_D3_1_210_Initial_LDACS_AG_Specification_00_01_00-1_0_updated.pdf, accessed 07/16/2021, German Aerospace Center (DLR), SESAR2020 PJ14-02-01 D3.3.030, 2020.

[17] ARINC, "Datalink Security Part 1 - ACARS Message Security," https://standards.globalspec.com/std/1039315/ARINC823P1, accessed 07/16/2021, Aeronautical Radio, Incorporated (ARINC), ARINC SPECIFICATION 823P1, 2007.

[18] ARINC, "Datalink Securoity Part 2 - Key Management ," https://standards.globalspec.com/std/1039315/ARINC823P2, accessed 07/16/2021, Aeronautical Radio, Incorporated (ARINC), ARINC SPECIFICATION 823P2, 2008.

[19] 3GPP, "3GPP System Architecture Evolution (SAE); Security Architecture (Release 16)," https://www.3gpp.org/ftp/Specs/archive/33_series/33.401/33401-g30.zip, accessed 07/16/2021, 3rd Generation Partnership Project (3GPP), Tech. Rep., 2020.

[20] ICAO, "Manual Of Technical Provisions For The Aeronautical Telecommunication Network (ATN)," https://standards.globalspec.com/std/740124/icao-9705, accessed 07/16/2021, International Civil Aviation Organization (ICAO), Doc 9705, 2002.

[21] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," Internet Requests for Comments, RFC Editor, RFC 2401, November 1998.

[22] B. Blanchet, "Symbolic And Computational Mechanized Verification Of The ARINC823 Avionic Protocols," in *30th Computer Security Foundations Symposium*. IEEE, 2017, pp. 68–82.

[23] RTCA, "Minimum Operational Performance Standards (MOPS) for the Aeronautical Mobile Airport Communication System (AeroMACS," https://www.rtca.org/products/do-346-electronic/, accessed 07/16/2021, Radio Technical Commission for Aeronautics (RTCA), DO-346, 2014.

[24] IEEE, "IEEE Standard For Local And Metropolitan Area Networks Part 16: Air Interface For Broadband Wireless Access Systems," https://standards.ieee.org/standard/802_16-2009.html, accessed 07/16/2021, Institute of Electrical and Electronics Engineers (IEEE), IEEE Std 802.16-2009, 2009.

[25] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "Extensible Authentication Protocol (EAP)," Internet Requests for Comments, RFC Editor, RFC 3748, June 2004.

[26] 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," Internet Requests for Comments, RFC Editor, RFC 5280, May 2008.

[27] J. Jonsson and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1," Internet Requests for Comments, RFC Editor, RFC 3447, February 2003, http://www.rfc-editor.org/rfc/rfc3447.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3447.txt

[28] F. Yang, "Comparative Analysis On TEK Exchange Between PKMv1 And PKMV2 For WiMAX," in *7th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 2011, pp. 1–4.

[29] NIST, "Secure Hash Standard (SHS)," https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf, accessed 07/16/2021, National Institute of Standards and Technology (NIST), FIPS 180-4, August 2015.

[30] A. Bilzhause, B. Belgacem, M. Mostafa, and T. Gräupl, "Datalink Security In The L-band Digital Aeronautical Communications System (LDACS) For Air Traffic Management," *Aerospace and Electronic Systems Magazine*, vol. 32, no. 11, pp. 22–33, 2017.

[31] T. Ewert, N. Mäurer, and T. Gräupl, "Group Key Distribution Procedures For The L-Band Digital Aeronautical Communications System (LDACS)," in *40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021, pp. 1–10.

[32] H. Krawczyk, "SIGMA: The 'SIGn-and-MAc'Approach To Authenticated Diffie-Hellman And Its Use In The IKE Protocols," in *Annual International Cryptology Conference*. Springer, 2003, pp. 400–425.

[33] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)," Internet Requests for Comments, RFC Editor, STD 79, October 2014.

[34] RTCA, "Minimum Operational Performance Standards (MOPS) for Aircraft VDL Mode 2 Physical Link and Network Layer," https://www.rtca.org/products/do-281c-electronic/, accessed 07/16/2021, Radio Technical Commission for Aeronautics (RTCA), DO-281C, 2018.

[35] E. Rescorla, "The transport layer security (tls) protocol version 1.3," Internet Requests for Comments, RFC Editor, RFC 8446, August 2018.

[36] D. Jao, "Supersingular Isogeny Key Encapsulation," https://sike.org/files/SIDH-spec.pdf, accessed 07/16/2021, National Institute of Standards and Technology (NIST), Tech. Rep., 2020.

[37] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP," Internet Requests for Comments, RFC Editor, RFC 6960, June 2013.

[38] E. Barker, "Digital Signature Standard (DSS)," https://doi.org/10.6028/NIST.FIPS.186-4, accessed 07/16/2021, National Institute of Standards and Technology (NIST), FIPS.186-4, 2013.

[39] D. Soni, K. Basu, M. Nabeel, N. Aaraj, M. Manzano, and R. Karri, "FALCON," in *Hardware Architectures for Post-Quantum Digital Signature Schemes*. Springer, 2021, pp. 31–41.

[40] M. Lochter and J. Merkle, "Elliptic curve cryptography (ecc) brainpool standard curves and curve generation," Internet Requests for Comments, RFC Editor, RFC 5639, March 2010.

[41] D. J. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, no. 7671, pp. 188–194, 2017.

[42] M. Raavi, S. Wuthier, P. Chandramouli, Y. Balytskyi, X. Zhou, and S.-Y. Chang, "Security Comparisons and Performance Analyses of Post-quantum Signature Algorithms," in *International Conference on Applied Cryptography and Network Security*. Springer, 2021, pp. 424–447.

[43] H. Krawczyk and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)," Internet Requests for Comments, RFC Editor, RFC 5869, May 2010.

[44] J. Song, R. Poovendran, J. Lee, and T. Iwata, "The AES-CMAC Algorithm," Internet Requests for Comments, RFC Editor, RFC 4493, June 2006.

[45] D. McGrew, D. Bailey, M. Campagna, and R. Dugal, "Aes-ccm elliptic curve cryptography (ecc) cipher suites for tls," Internet Requests for Comments, RFC Editor, RFC 7251, June 2014.

[46] Y. Sun, M. Chen, A. Bacchus, and X. Lin, "Towards Collusion-Attack-Resilient Group Key Management Using One-Way Function Tree," *Computer Networks*, vol. 104, pp. 16–26, 2016.

[47] S. Meier, B. Schmidt, C. Cremers, and D. Basin, "The TAMARIN Prover For The Symbolic Analysis Of Security Protocols," in *25th International Conference on Computer Aided Verification (CAV)*. Springer, 2013, p. 696–701.

[48] D. Dolev and A. Yao, "On The Security Of Public Key Protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.