Comparing Different Diffie-Hellman Key Exchange Flavors for LDACS

Nils Mäurer and 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

Corinna Schmitt
Research Institute CODE
Universität der Bundeswehr München
Munich, Germany
corinna.schmitt@unibw.de

***Abstract*—Growth of civil air traffic worldwide poses a great challenge for the supporting Communication, Navigation and Surveillance (CNS) infrastructure. Analogue systems have to be replaced by digital means to optimize spectrum efficiency and automation is becoming much more important to be able to handle the amount of participants in the air traffic system. As safety and security are strongly intertwined in aviation, cybersecurity is one key enabler for digitalization in civil aviation. As such we investigate mutual authentication and key agreement methods for the digital aeronautical ground-based communications system L-band Digital Aeronautical Communication System (LDACS). Thereby, we compare the suitability of three different Diffie-Hellmann (DH) key exchange flavors used in a modified version of the Station-To-Station (STS) protocol, for digital aeronautical communication in terms of latency and security data overhead. We conclude, the STS protocol based on a central Public Key Infrastructure (PKI) trust solution with Supersingular Isogeny Diffie–Hellman (SIDH) for post-quantum security to be best suited for long term security. However, due to the smaller key sizes, Elliptic Curve Diffie-Hellman (ECDH) is the more resource efficient candidate and may play a role in low resource authentication scenarios for LDACS.**

***Index Terms*—Cybersecurity, LDACS, Authenticated Key Exchange, STS protocol**

I. Introduction

The COVID-19 crisis has vastly impacted worldwide civil air traffic, reducing world passenger numbers by 35% to 65% compared to the pre-COVID-19 level [1]. Despite the decline of passenger numbers, the entire industry is currently undergoing a digital transformation, especially now to become more efficient and cost-saving in the long term. One area mostly affected by this trend is the Communication, Navigation and Surveillance (CNS) infrastructure. With the Single European Sky ATM Research (SESAR) program in the EU and NextGEN in the US, several new digital aeronautical communication technologies shall be developed in the framework of the Future Communications Infrastructure. Candidates in this framework are Aeronautical Mobile Airport Communication System (AeroMACS) for airport communications, SatCOM, for oceanic, polar and remote domains, and L-band Digital Aeronautical Communication System (LDACS) for long-range terrestrial aeronautical communications [2].

As safety and security are strongly interrelated in aviation, strong cybersecurity is the foundation and enabler for digitalization in aviation that is agreed on by the World Economic Forum, the International Civil Aviation Organization (ICAO), and the cybersecurity research community [3]. Unfortunately cybersecurity for CNS is still not realized in most deployed systems as depicted in [4] and [5]. Thus future CNS systems like LDACS require profound cybersecurity measures allowing automated data processing and protection of the system against threats from the IT sector.

In previous works we proposed cybersecurity functions and measures for LDACS, such as Mutual Authentication and Key Exchange (MAKE) protocols, authenticated encryption for messages in transit and different trust solutions [6], [7]. However, there has been no discussion yet about the optimal key agreement variation of the Diffie-Hellman Key Exchange (DHKE) for digital aeronautical communications.

The objective of this paper is to compare three variations of the Diffie-Hellman Key Exchange (DHKE) for the Station-To-Station (STS) protocol in terms of induced latency and security data overhead and decide their suitability for LDACS. The investigated variations are (i) ephemeral DHKE, (ii) Elliptic Curve DHKE (ECDH), and (iii) Supersingular Isogeny DHKE (SIDH).

The rest of the paper is structures as follow: Section II presents LDACS and its cybersecurity measures, as well as relevant math and key sizes of the different DHKE flavors. In Section III, we discuss a methodology to calculate latency times for data exchange of LDACS and introduce the STS protocol in depth, including message formats and sizes. With the formulas from Section III and based on the message formats, we calculate authentication latency and data overhead and list our results in Section IV. Section V summarizes the received results and concludes the paper. All used acronyms are listed in the Appendix.

II. Background

LDACS is a ground-based digital communications system for flight guidance and communications related to the safety and regularity of flight; developed in Europe and is currently under standardization by ICAO [8]. It covers current Air Traffic Services (ATS) and Aeronautical Operational Control (AOC) data including future applications. Further, LDACS

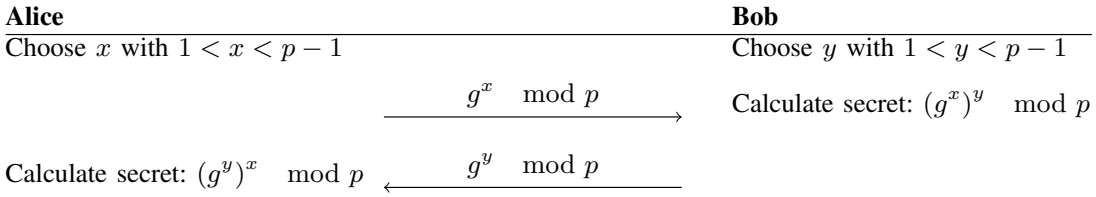

Fig. 1. Ephemeral DHKE protocol

enables new concepts, like sectorless Air Traffic Management (ATM), and has at least 50 times more network capacity than the currently used terrestrial links like the VHF Data Link (VDL) Mode 2 system [8]. Instead of some kBit/s, LDACS offers up to 2 MBit/s. By enabling not only communication but also navigation and surveillance at the same time, it is the world's first integrated CNS system [2]. A basic LDACS network is formed by up to 512 Aircraft Station (AS), which are served by a Ground Station (GS) and multiple GS are connected to a Ground Station Controller (GSC). In Section III we give a more detailed description about how data is handled by LDACS. Cybersecurity considerations for LDACS were first published in [9] and list five main objectives:

1) A guarantee of safe and effective LDACS system operations together with system security functions,
2) supporting reliability and robustness,
3) supporting message authentication and integrity,
4) supporting confidentiality, and
5) supporting mutual entity authentication.

From these high level objectives, we derived security functions and assigned algorithms, protocols and procedures to them. First, we have to rely on an overall structure that enables us to establish trust among communication entities. Therefore, a certificate-based Public Key Infrastructure (PKI) solution, necessary for the STS protocol, is conceivable. Ensuring that all involved entities in the envisioned communication have a valid identity and established a common secret, mutual entity authentication and key negotiation are required. This can be achieved by the STS [6] protocol together with different possible variations of the DHKE. In order to provide confidentiality, integrity, and authenticity protection of messages in transit AES-256 Galois Counter Mode (GCM) is applied [7].

Section II-A presents now background information to the DHKE variants investigated, which are promising candidates for our envisioned solution presented in Section III. This is followed by a brief introduction to the assumed authenticated key exchange protocol in Section II-B.

A. Overview of Diffie-Hellman Key Exchange Variants

As central element, STS requires a DHKE, we want to present three possible variants and compare them in terms of amount of bits exchanged between communication partners, denoted here as $Alice$ and Bob.

(i) Ephemeral DHKE:
The original Diffie-Hellman or Diffie-Hellman-Merkle Key Exchange was first published in 1976 [10] and is based on the discrete logarithm or Diffie-Hellman problem: given a cyclic group G of prime order n, a generator g of G and elements $g^x, g^y \in G$, find g^{xy}. Man-in-the-Middle attacks are still possible when no authentication or additional security features are used [11]. However, using the authenticated DHKE or STS protocol with certificates and supported by a PKI [12] is still a secure key exchange protocol today [13]. Public parameters for the DHKE are (p, q, g): a large prime p, a prime divisor of $p-1$: q, and an element g of order q in ${\mathbb{Z}_p}^\times$ and generator for cyclic multiplicative group G. The secret keys for Alice and Bob are $x, y \in {\mathbb{Z}_p}^\times$ and public keys are $g^x, g^y \in {\mathbb{Z}_p}^\times$. The protocol run is depicted in Figure 1. If the STS protocol is used, the sizes of the DHKE parameters still need to be large enough. The Federal Office for Information Security Germany suggests a key length of at least 3000 Bit for the use beyond 2022 [14]. As LDACS is foreseen to be realized after 2022, we use sizes of 3072 Bit in this paper.

(ii) Elliptic Curve DHKE (ECDH):
As ephemeral DHKE requires large key spaces, other abelian groups were researched, where the same idea could be applied. One very successful cryptographic platform was the use of elliptic curves over finite fields [15], [16], resulting in the ECDH key agreement protocol. The public key sizes and thus the group sizes could be reduced from 1024 Bit to 160 Bit, from 3072 Bit to 256 Bit and from 15,360 Bit to 512 Bit [17] respectively by using the discrete logarithm problem on elliptic curves over finite fields. Please note that the protocol run remains the same as depicted in Figure 1, only the symbols have a different meaning: p is again a prime defining the field $\mathbb{F}_{\text{ı}}$, $a, b \in \mathbb{F}_{\text{ı}}$ define an elliptic curve, the cyclic subgroup is defined by a generator g is now defined by the base point P on $E(\mathbb{F}_p)$ and $q := ord(P)$ is the order of the base point P in $E(\mathbb{F}_{\text{ı}})$. The secret parameters for $Alice$ and Bob are again $x \in 1, ..., q-1$ and $y \in 1, ..., q-1$ respectively and the public keys g^x, g^y are now $Q_A := x \cdot P$ and $Q_B := y \cdot P$ [16]. In terms of parameter sizes [14], the Federal Office for Information Security Germany recommends the use of RFC 5639 [18] and a minimum size of 256 Bit for x, y, Q_A and Q_B. As we chose 3072 Bit for the ephemeral DHKE, we choose now its equivalent in security on an elliptic curve, which is 256 Bit sizes in this paper [17].

(iii) Supersingular Isogeny DHKE (SIDH):
To harden cryptographic protocols for quantum resistance the idea of quantum-resistant public-key cryptosystems based on the conjectured difficulty of finding isogenies between supersingular elliptic curves was formulated in [19] and extended for key exchange applications [20]. The basic idea is to compose two random walks on an isogeny graph of elliptic curves so that the end node of both ways is the same. As the graph used for SIDH is chaotic, auxiliary points are required to help Alice and Bob walking the respective other's public key [21]. For mathematical details in depth it is referred to [22] Important to know is that the basic protocol run with Alice and Bob choosing a secret key, calculating and exchanging a public key remains the same as depicted in Figure 1. It can be used to derive a shared secret. To demonstrate the effectiveness and popularity of SIDH, we want to mention the Supersingular Isogeny Key Encapsulation (SIKE) - a post-quantum cryptography candidate. It mitigates weaknesses of SIDH such as Man-in-the-Middle or active reaction attacks [22]. It uses a 335 Byte compressed key. However, as we mitigate the weaknesses of SIDH making use of the combination of SIDH and the STS protocol, we can use even smaller keys! We can calculate public key sizes for SIDH as follows: During a key exchange entities Alice and Bob will each transmit two coefficients defining an elliptic curve and two elliptic curve points. Each elliptic curve coefficient requires $\log_2 p^2$ and each elliptic curve point can be transmitted in $\log_2 p^2 + 1$ Bit. To obtain a 128 Bit security level a 768 Bit modulus is required, hence the transmission is $4 \times \log_2(2^{768})^2 + 4 = 6148$ Bit [23]. However, there are key compression techniques for SIDH public keys resulting in 385 Byte [23] and 328 Byte (2624 Bit) for Alice' public key and 330 Byte (2640 Bit) for Bob's public key [24]. Thus, for this paper we will assume SIDH public key sizes to be 2624 and 2640 Bit respectively.

B. Authenticated Key Exchange Protocol

The origin of LDACS mutual authentication and key exchange protocol, first mentioned in [6], is a variation of the STS protocol [13]. Since the publication of [6], we investigated different STS variants and protocol 5.25 "Modified STS protocol" in [13] proves to be more secure and concise than that mentioned in [6]. It prevents the possibility of a Man-in-the-Middle attack during the exchange of the key material by signing the respective material with the help of exchanged or prestored public key certificates of the respective communication partner. Details about the use of the authenticated key exchange, STS protocol for LDACS, are listed in section III-C. However, in order to ensure trust in public keys from the respective communication partner, a PKI is required [6]. For a root of trust a PKI is widely considered as a good solution for the origin of trust as justified in [25], but comes with some drawbacks for digital aeronautical communications:

1) Massive rollout, management, and revocation of certificates are required.
2) A root of trust has to be declared and accepted by state actors worldwide potentially requiring secure cross certification among all countries worldwide respecting political situations and regulations in aviation.

With the two drawbacks mentioned PKI may be a challenging solution for an aeronautical trust framework. But keeping in mind that digital data links for civil aeronautical traffic (i.e. AeroMACS) use a PKI as their trust solution [26], it looks promising to use PKI also for LDACS. Thus, the modified STS protocol becomes a good candidate for mutual authentication and key agreement for LDACS.

III. Method

Building on the knowledge gained in Section II, this section concentrates on the applied method for our approach presented in this paper. Here we introduce the math enabling us to estimate authentication and key agreement times for LDACS communicating parties, as well as introducing the STS protocol for LDACS together with its respective messages and message sizes.

A. Details of LDACS Framing

The LDACS protocol is structured in time with a frame structure of slots. In the Forward Link (FL) direction, each Super Frame (SF) starts with a Broadcast (BC) slot, where the GS announces their existence to the AS and send physical parameters for link establishment. The rest of the FL SF is split into four Multi Frame (MF), each containing nine Orthogonal Frequency-Division Multiplexing (OFDM) frames with a frame duration of $T_{DF/CC} = 6.48ms$ (54 OFDM symbols). Each frame has a capacity for 2442 symbols and comprises three FL Physical layer Service Data Unit (PHY-SDU). Every FL PHY-SDU can be used to transmit FL user data or Common Control (CC) data, where GS can allocate resources to an AS. On the Reverse Link (RL), a SF starts with a Random Access (RA) slot, where AS can request access to an LDACS cell and continues with four MF. Each RL MF is constructed from 162 RL PHY-SDU equivalent to OFDM Access (OFDMA) tiles. They are used to transmit Dedicated Control (DC) data, where AS can request the allocation for resources allowing them to send on the RL, and RL user data. Those details are depicted in Figure 2 [8].

Data is transported in the Data Channel (DCH) via different FL PHY-SDUs and RL PHY-SDUs of different sizes. Depending on coding and modulation, thus the channel quality, the FL PHY-SDUs sizes range from 728 to 3296 Bit, and the RL PHY-SDUs range from 112 to 528 Bit. With 27 frames per MF in total and one to eight FL PHY-SDUs being reserved on the FL for the Common Control messages, this leaves 19 to 26 FL PHY-SDUs per MF for data transport. The minimum amount of Bit per MF can thus be calculated with $19 * 728 = 13,832$ Bit and the maximum Bit per MF with $26 * 3296 = 85,696$ Bit. On the reverse link, the RL PHY-SDUs are separated into 162 tiles. The first two tiles are sync tiles, followed by a minimum of two DC and a maximum of 32 DC tiles, which limits the minimum usable

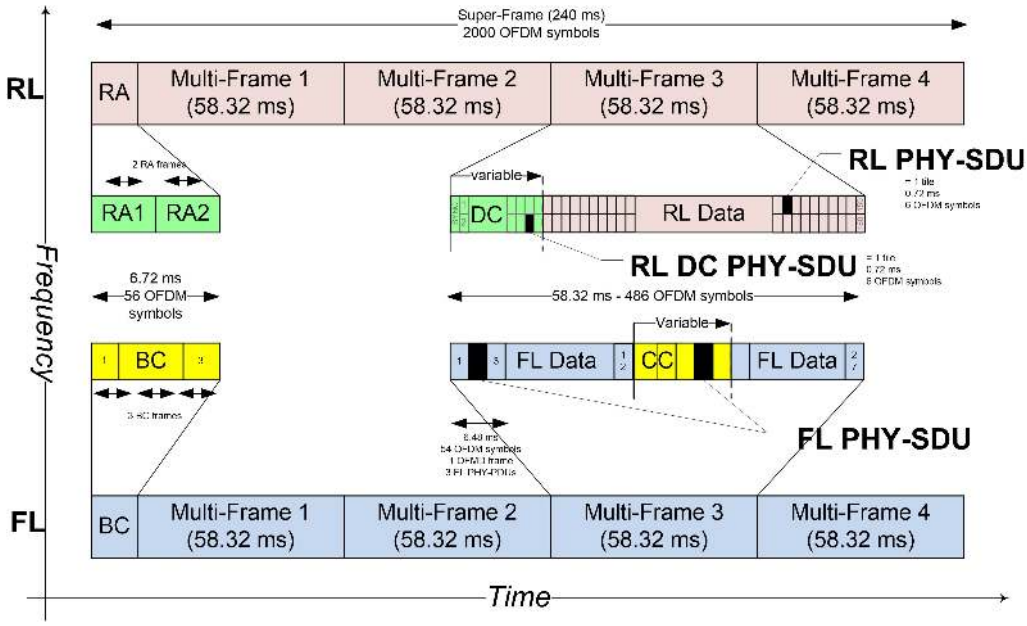

Fig. 2. LDACS frame structure within the Forward and Reverse Link.

user data per MF to $(162 - 2 - 32) * 112 = 14,336$ Bit and allows a maximum of $(162 - 2 - 2) * 528 = 83,424$ Bit per MF [8].

This is equivalent to a minimum data rate of 230.5 kbit/s on the FL and 238.9 kbit/s on the RL, with maximum control channel use. Respectively, the maximum data rate is 1428.3 kbit/s on the FL and 1390 kbit/s on the RL, with minimum control channel use.

B. Model to Emulate Latencies for LDACS

In 2015, Gräupl et al. [27] presented a full methodology on how to emulate latencies for user data in the forward and reverse link of LDACS depending on the Bit Error Rate (BER) and message size. The required equations to calculate FL/RL latencies for LDACS are briefly described in the following and for more details of LDACS framing, we refer to [8]. Taking retransmissions into account the FL latency can be calculated with

$$L_{FL}(t) = m_{FL}(t) + (1 + \delta_{RX}(1 + n)) \times d_{MF} \quad (1)$$

and the RL latency with

$$L_{RL}(t) = m_{RL}(t) + (2 + \delta_{RX}(N + 3)) \times d_{MF}. \quad (2)$$

In Equation 1, we use $m_{FL}(t)$ to classify the time until the start of the next CC frame, $\delta_{RX} \in \{0, 1\}$ to indicate a retransmission, d_{MF} denotes the length of a MF and n is derived from the length of the reverse link medium access cycle from forward link perspective. In Equation 2, we use $m_{RL}(t)$ to denote the time until the start of next DC slot, $\delta_{RX} \in \{0, 1\}$ to indicate a retransmission, d_{MF} denotes the length of a MF and N is derived from the length of the reverse link medium access cycle from reverse link perspective.
We model $\delta_{RX} \in \{0, 1\}$ as stochastic process, based on the packet error rate. Given a Bit Error Rate (BER), we can calculate the packet error rate based on the length of a packet l: $P(\{\text{no error in packet}\}) = (1 - BER)^l$. Thus the opposite event, that a packet indeed contains an error is: $P(\{\text{error in packet}\}) = 1 - ((1 - BER)^l)$. These two probability decide the value of δ_{RX}, whether a retransmission is necessary and, thus, an error appeared in the packet, or not. We are aware of the occurrence of multiple retransmissions, but neglect them at this stage as we assume that they become exponentially more unlikely.

Due to the rapid development of LDACS in the last years, the calculations for n and N changed, due to a reduction of a previous maximum of 52 slot per MF to now 32 AS per DC slot per MF [8]: If we assume that LDACS is configured to use a maximum size DC slot, $n = \lfloor (i + \#\text{AS})/32 \rfloor$ and $N = (\lfloor (i + \#\text{AS})/32 \rfloor - 3) \mod \lfloor \#\text{AS}/32 \rfloor$, with $\#AS$ being the amount of AS per LDACS cell and i indicating the AS' position within the DC slot. i may be any value between 1 and the size of the DC slot. If #AS is not equal to the slot size, the position of an AS in the DC slot will be shifted by this difference each medium access cycle. On average we get therefore $n = \#\text{AS}/32$ and $N = (\#\text{AS}/32 - 3) \mod \#\text{AS}/32$ over time. $m_{FL}(t)$ denotes the time when a CC slot is free for an aircraft to obtain resources to send data. It is dependant on time and we can model it as a stochastic process returning values between 1 to 60ms with uniform distributions:

1) X = amount of milliseconds an AS has to wait until it can send on the CC slot with uniform distribution $U(1, 60)$, with 1 being the lowest waiting values 1 ms and 60 being the largest waiting value 60 ms.
2) $m_{RL}(t)$ denotes the time when a DC slot is free for an aircraft to request resources to send data. It is dependant

TABLE I
PARAMETER VALUES FOR LATENCY TIMING FOR THE LDACS MEDIUM ACCESS CONTROL (MAC) PROTOCOL.

Forward Link Model $L_{FL}(t) = m_{FL}(t) + (1 + \delta_{RX}(1 + n)) \times d_{MF}$		**Reverse Link Model** $L_{RL}(t) = m_{RL}(t) + (2 + \delta_{RX}(N + 3)) \times d_{MF}$	
Parameters	**Values**	**Parameters**	**Values**
d_{MF}	60ms	d_{MF}	60ms
$m_{FL}(t)$	Time until start of next FL MF: Every 1 to 60ms modelled by $U(1, 60)$	$m_{RL}(t)$	Average time until start of next MAC cycle: $\#\text{AS}/32 \times d_{MF} + wait$ $wait$ modelled by $U(1, 60)$
n	Average amount of MF after transmission until next DC slot is scheduled for AS in MAC-cycle: $n = \#\text{AS}/32$	N	Average amount of MF after transmission until next DC slot scheduled for AS in MAC-cycle: $N = (\#\text{AS}/32 - 3) \mod \#\text{AS}/32$
BER	$0, 10^{-6}, 10^{-5}$		
P	$P(\{\text{no error in packet}\}) = (1 - BER)^l$ $P(\{\text{error in packet}\}) = 1 - ((1 - BER)^l)$		

on time and the amount of aircraft in an LDACS cell. Thus we can model it with $\lfloor i + \#\text{AS}/32 \rfloor \times d_{MF} + wait$, with $wait$ being a stochastic process returning values between 1 to 60ms with uniform distribution and i uniformly distributed between 1 and the slot size over time.

We list in Table I all necessary parameters, we need to obtain latency values for the different authentication and key agreement protocols for LDACS.

C. LDACS Certificate Based Authentication Protocol

In Section II-B we already introduced the concept to make use of the STS protocol for LDACS, as already suggested in [6] and [7]. First we want to introduce the exact protocol run foreseen to be used for LDACS and then define message sizes for our emulation of latency times. Please note that steps 2, 3 and 4 in the depicted protocol variant in Figure 3 follow closely protocol 5.25 in [13], which has been proven secure in the Bellare et al. model [28].

Protocol Run: Figure 3 assumes GS and GSC to have already established a secure connection and the GS can now start sending broadcast beacons announcing its existence. Furthermore several parameter need to have been exchanged prior to the protocol run: (1) depending on the choice of the DHKE, necessary public parameters have to be pre-deployed (e.g. p, g), (2) certificates and the public keys of the respective other communication partner have to be at AS and GSC, (3) a selection and agreement of signature and encryption has to have happened at AS and GSC. The cell entry follows details specified in the official LDACS specification [8]: The System Identification Broadcast (SIB) serves as identifier, containing physical parameters and the ID_{GSC} for AS to begin establishing a connection to that GS and ultimately GSC. After this step, authentication messages are only sent in the DCH. As for cryptographic material $Sig_{AS}(\text{DATA})$ and $Sig_{GSC}(\text{DATA})$ denote the signature of DATA of the respective entities. For the DHKE variations, we use x, y as secrets of GSC and AS, t_{AS}, t_{GSC} denote the public key of AS and GSC, MS_{AS-GSC} is the final Master Secret (MS) between AS and GSC derived with HKDF(PMS_{AS-GSC}). HKDF denotes the Hash-based Key Derivation Function [29] and PMS_{AS-GSC} is the Pre-Master Secret (PMS) of AS and GSC. At every verification step, the protocol in Figure 3 assumes the verification to be successful and thus continues. If verification fails, the connection is terminated and the authentication process retried with another suitable GS in range.

Message Data Formats: The details about all message formats and lengths is summarized in Table II. We define data sizes for the authentication and key agreement messages here for the STS for LDACS protocol. For signatures lengths, we assume a total length of 64 Byte for a message signature, produced by current signature procedures such as EdDSA-Ed25519 [30] or even post-quantum procedures such as rainbow [31].

TABLE II
MESSAGE FORMATS FOR THE THREE LDACS AUTHENTICATION MESSAGES, WITH LENGTHS OF RESPECTIVE FIELDS IN Bit.

Message	**Header**	**Field 1**	**Field 2**
$ServerHello$ $KeyExchange$	$header$: 48	t_{GSC}	-
$ClientHello$ $KeyExchange$	$header$: 48	t_{AS}	Sig_{AS}: 512
$ServerKey$ $ExchangeFinished$	$header$: 48	Sig_{GSC}: 512	-

All messages have a header consisting of $TYPE$, ID, UA and $PRIO$ fields. $TYPE$ is a 4 Bit long field and clarifies the message type, ID is 12 Bit long and denotes the ID of that message, UA is the 28 Bit long Unique Address field, containing the LDACS specific addresses of AS and GS and finally the 4 Bit long $PRIO$ field signifies the priority this particular message has. We collect all these

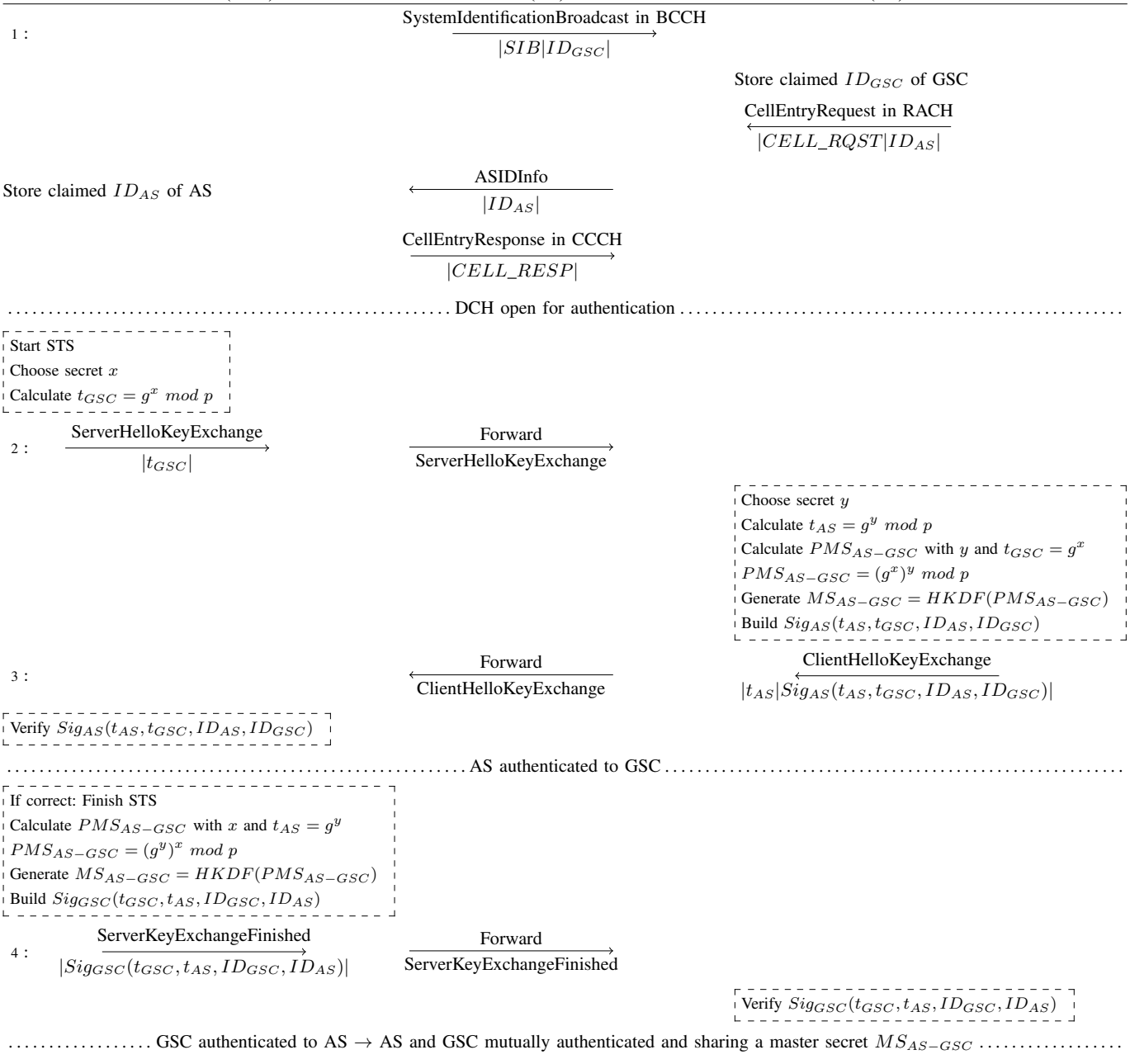

Fig. 3. LDACS Station-To-Station Authentication Protocol

fields into the $header$ resulting in a 48 Bit length. t_{GSC} and t_{AS} are the public keys of the respective entities and have different sizes, depending on the choice of the Diffie-Hellman procedure. The sizes for the public key of the GSC t_{GSC} are: $\{DHKE = 3072|ECDH = 256|SIDH = 2624\}$. The sizes for the public key of the AS t_{AS} are: $\{DHKE = 3072|ECDH = 256|SIDH = 2640\}$.

The $ServerHelloKeyExchange$ message, responsible to initiate the STS protocol between AS and GSC consists of the $header$ and the public key of the GSC t_{GSC}. Depending on the size of the public keys, the sizes for$ServerHelloKeyExchange$ are $\{3120, 304, 2672\}$ Bit. The key exchange message AS to GSC is denoted as $ClientKeyExchangeFinished$ and consists of the $header$, the public key of the AS t_{AS} and an AS signature Sig_{AS}. Depending on the size of the public keys, the sizes for $ClientKeyExchangeFinished$ are $\{3632, 816, 3200\}$ Bit. Finally the $ServerKeyExchangeFinished$ finishes the STS protocol and consists of the $header$ and a GSC signature Sig_{GSC}, totalling in 560 Bit.

IV. Evaluation

We start by listing all assumptions for our evaluation and then proceed by evaluating authentication data overhead and latency. Then we give the overall formula including ground communication and computation delays to calculate the overall authentication latency. Finally we look at the maximum amount of authentication attempts per MF, in dependence to FL/RL PHY-SDU sizes, thus coding and modulation rates of LDACS.

A. Assumptions

For the results of the simulations presented in this section we have the following assumptions: Concerning LDACS (1) all public DHKE parameters have been distributed to AS and GSC previously and (2) GS and GSC have established a secure connection. For the STS results we assume (3) certificates and the public key of the respective other communication partner were handed out in previous steps. Further, we assume (4) processing times at the respective entities to be negligible for the LDACS latency, but denote them with $\triangle_{Comp}(ENTITY)$, with $ENTITY \in \{AS, GS, GSC\}$. Communication times between GS and GSC for STS are denoted with $\triangle_{Comm}(GSC)$. Also we (5) only measure the authentication time after cell entry procedure has been completed (starting from step 2 in Figure 3). Finally (6) due to fibre glass optical cables and fast network ground routing we assume these latency times to be negligible compared to LDACS latency times.

B. Authentication Data Overhead

In Section III-C we already explained the sizes of each message. Without retransmissions we calculate the total amount of data exchanged for each protocol and key exchange flavor, resulting in the values listed in Table III.

TABLE III
Total authentication message sizes in Bit

STS-DHKE	STS-ECDH	STS-SIDH
7312	1680	6432

Following the paradigm "the lower the BER, the lower the amount of retransmissions and vice versa", overall authentication data can increase with increasing retransmissions. Based on the total authentication data calculated, we observed different authentication latency times for LDACS and impact of BER on the suitability of the different protocols and key exchange flavors.

C. Authentication Latency

We investigated an authentication latency baseline by setting the BER=0 and then looked at the more realistic points of operation of LDACS with BER=10^{-5} and BER=10^{-6}.

Authentication Latency Baseline: With $BER = 0$, the different sizes of the DHKE variations have no impact on the latency times, only the amount of exchanged authentication messages. Assuming only one authenticating AS per multiframe (c.f. Section IV-D), each authentication message fits into the FL PHY-SDUs and RL PHY-SDUs of one multiframe.

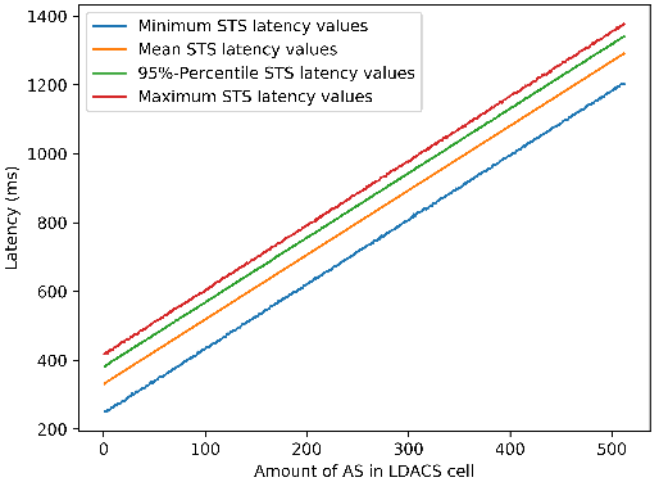

Fig. 4. Baseline authentication latency of STS depending of the amount of AS in an LDACS cell at BER= 0.

Thus we can calculate the latency times for STS at $BER = 0$ and depending of the existing amount of AS per LDACS cell. Results are depicted in Figure 4. We see, that total authentication latencies start at a minimum of 245 ms, average at 332 ms and have a maximum of 422 ms for one aircraft in an LDACS cell. Then latency increase linearly on average, ending at a minimum of 1203 ms, average at 1290 ms and reach a maximum at 1380 ms for 512 aircraft in an LDACS cell. Differences between mean and 95%-percentile latencies of around 60 ms stem from the differences of average and maximum waiting time for an AS, until either a CC or DC slot is free to request or receive resource allocations.

LDACS Point of Operation Authentication Latency: To receive representative results, we assume only one authenticating AS per MF and we emulate 10,000 authentication attempts at BER=10^{-5} and BER=10^{-6} in dependence on the amount of AS in an LDACS cell. We chose this amount of authentication attempts as the probability of a retransmission is calculated as $1 - (1 - BER)^l$ with l being the packet length, following Section III-B. The smallest message for the STS protocol in the FL is 560 Bit and 624 Bit in the RL. This results in a retransmission probability at BER=10^{-6} of 0,0560% in the FL and 0,0624% in the RL. Thus emulating 10,000 authentication attempts suffices to trigger at least one retransmission of the smallest authentication message and is sufficiently accurate for the authentication latency emulation. We perform this emulation with each key exchange flavor respectively.

Overall comparing Figures 5 and 4 with each other, we clearly see, that retransmissions and thus the choice of DHKE flavor does not play a large role at this BER for the authentication latency. All authentication latency graphs start at around 320 ms on average and 400 ms for 95%-precentile for one

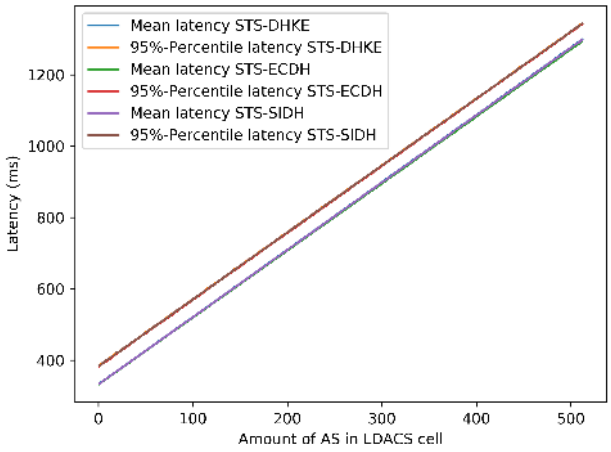

Fig. 5. Authentication latency of the STS protocol depending of the amount of AS in an LDACS cell and DHKE at BER=10^{-6}.

AS in a cell and end at 1300 ms on average and 1340 ms for 95%-precentile for more than 500 AS in a cell. At BER of 10^{-5}, depicted in Figure 6, we notice that the different public key sizes of the different DHKE variations have a major effect on the overall authentication time.

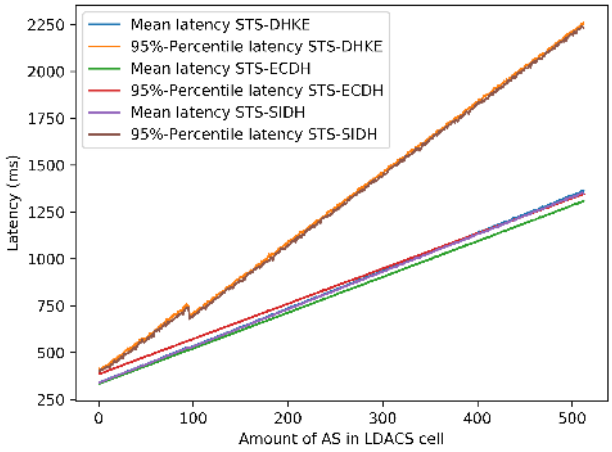

Fig. 6. Authentication latency of the STS protocol depending of the amount of AS in an LDACS cell and DHKE at BER=10^{-5}. Note that the small peaks in the result for less than 3×32 AS are caused by the DC slot falling into an unfavorable position for retransmissions as calculated by N in Table I.

While all DHKE variations start at 300-400 ms on average for one AS and end at 1300 ms on average for more than 500 AS in a cell, the 95% percentiles differ greatly. While 95% percentiles of ECDH follow a similar trajectory to its averages due to its small key sizes, the 95% percentile of DHKE and SIDH look vastly different. DHKE and SIDH start at around 400 ms for one AS in the cell, have a small peak at 750 ms, at 96 AS in a cell due to the DC slot falling into an unfavourable position (c.f. Figure 6) and end at 2220 ms for more than 500 AS per cell. At 500 AS in an LDACS cell, we see a time difference of almost 1200 ms for the authentication to complete between the SIDH/DHKE and ECDH approach, with ECDH being the much faster candidate.

Summary of LDACS Authentication Latency Findings: The requirements document DO-350A imposes a $RCTP_{CSP} = 10s$ threshold for RCP 130/A1 message types [32], meaning that all authentication and connection establishment must be completed below the 10s threshold [8]. Authentication times in all scenarios considered in this paper remain always under the required threshold. This is in itself an important finding of this work.

Overall metrics for LDACS Authentication Latency: In the previous paragraphs, we only looked at the total authentication times induced by the radio gap and the LDACS protocol between AS and GS. To complete this picture, we want to include all additional, albeit simplified, computation ($Comp$) and communication ($Comm$) latencies mentioned at the beginning of Section IV. Hence, for the overall LDACS STS authentication and key agreement time between AS and GSC, denoted L_{STS}, we can use formula (3) based on syntax introduced in Section III-B).

$$\begin{aligned} L_{STS} = {} & \triangle_{Comp}(GSC) + \triangle_{Comm}(GSC) + L_{FL}(t) + \\ & \triangle_{Comp}(AS) + L_{RL}(t) + \triangle_{Comm}(GSC) + \\ & \triangle_{Comp}(GSC) + \triangle_{Comm}(GSC) + L_{FL}(t) + \\ & \triangle_{Comp}(AS) \\ & = 2 \times L_{FL}(t) + 1 \times L_{RL}(t) + \\ & 2 \times \triangle_{Comp}(AS) + \\ & 2 \times \triangle_{Comp}(GSC) + 3 \times \triangle_{Comm}(GSC) \end{aligned} \tag{3}$$

Assuming the communication latency induced by LDACS on the FL and RL is much larger than any of the computational delays (true for DHKE [33], ECDH [33], SIDH [22]) and ground based communication delays, the calculation of L_{STS} can be simplified to:

$$L_{STS} = 2 \times L_{FL}(t) + 1 \times L_{RL}(t) + constant \tag{4}$$

So far, we only regarded one authenticating aircraft. In the following, we want to look at the possible amount of simultaneously authenticating aircraft per multiframe, depending on the authentication method.

D. Coding and Modulation Impact on Authentication Protocol

In Section III-A we calculated the minimum data size in the FL per MF to be $13,832$ Bit and in the RL to be $14,336$ Bit per MF assuming the most conservative coding and modulation, and maximum size control channels. Depending on the DHKE procedure, the sizes of authentication messages for STS range from 304, 560, 2672, to 3120 Bit in the FL and from 816, 3200 to 3632 Bit in the RL according to Section III-C. We can calculate the maximum number of authentication attempts per multiframe with the message sizes and minimum data sizes per multiframe (detailed in Section III-A) given, depending on the choice of DHKE procedure. Please note, that here we also assume a maximum usage of FL/RL PHY-SDU with control channel data, same as listed in Section III-A. We list the results in Table IV.

TABLE IV
MAXIMUM AMOUNT OF AUTHENTICATION ATTEMPTS PER MULTIFRAME FOR LOWEST CODING AND MODULATION RATE OF LDACS AND MAXIMAL CONTROL SLOT OCCUPANCY, DEPENDING ON THE CHOICE OF KEY EXCHANGE PROTOCOL.

DHKE Procedure	**STS**			
	FL		**RL**	
	#Auth. Attempts	Size of message	#Auth. Attempts	Size of message
DHKE	4	3120 Bit	3	3696 Bit
ECDH	45	304 Bit	17	816 Bit
SIDH	5	2672 Bit	4	3200 Bit

With increasingly better coding and modulation the sizes for FL PHY-SDU frames and RL PHY-SDU tiles increase and thus the possible amount of authentication attempts per MF increases. In Figure 7 we show results of possible amounts of authentication attempts for the STS protocol in dependence of our three key agreement flavors, FL/RL PHY-SDU sizes and assume minimal control slot occupancy (FL: one CC slot, RL: two DC slots). Each group depicted in Figure 7 represents one set of FL/RL PHY-SDU sizes, so for the first three bars in Figure 7 we have 112 Bit RL PHY-SDU tiles and 728 Bit FL PHY-SDU frames. Overall we have {728, 960, 1080, 1456, 1936, 2176, 2928, 3296} Bit per FL PHY-SDU frame and {112, 152, 176, 224, 312, 360, 480, 528} Bit per RL PHY-SDU tile. Figure 7 shows especially the ECDH variation to be efficient, due to the small public key sizes. We also see that DHKE or SIDH only marginally differ in amount of possible AS per MF. Overall the biggest difference of the ECDH and DHKE/SIDH, is the maximum possible amount of authentication tries of AS per MF. The STS protocol allows for a maximum of 27 DHKE, 32 SIDH or 281 ECDH based messages in the FL and 22 DHKE, 26 SIDH or 153 ECDH based messages in the RL. This concludes in an important finding of this work, to introduce a maximum amount of AS authentication attempts. Otherwise it is possible for an AS to continuously send authentication requests and thus block communication for other aircraft.

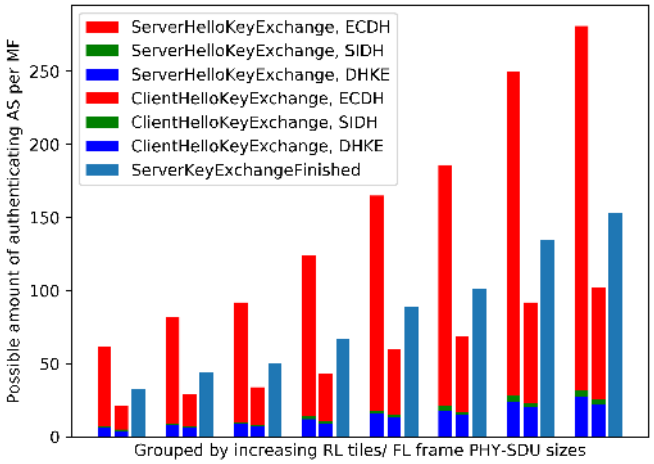

Fig. 7. Maximum possible amount of authentication attempts per MF in dependence on RL PHY-SDU tiles/FL PHY-SDU frames.

V. SUMMARY

The objective of this paper was to compare the suitability of three different variations of the Diffie-Hellman Key Exchange within the STS protocol for LDACS. We introduced the basic math behind the Diffie-Hellman key exchange and explained the variations ECDH and SIDH. Then we explained the choice of parameters and public, private key sizes used in this paper. With introduced formulas to calculate communication latency times for LDACS and an in-depth look at the STS variation for LDACS, we could calculate the overall authentication data sizes for each protocol and key exchange flavor. Comparing the sizes of all three key agreement variations, we see that $DHKE > SIDH >> ECDH$. This is highlighted further when looking at the maximum authentication attempts of an aircraft per LDACS multiframe, depending on the amount of AS already in an LDACS cell and LDACS signal quality. Overall STS-ECDH proved to be the most efficient variation for the most AS authentication attempts per multiframe. One important finding of this paper is to introduce a maximum authentication attempt threshold for authenticating AS, as otherwise an aircraft can block LDACS resources by continuously trying and failing to authenticate.

For a successful implementation of MAKE protocols for LDACS we recommend the key exchange flavors ECDH and SIDH, due to the short key sizes in ECDH and the post-quantum robustness of SIDH. If a central trust approach, such as a PKI, is favoured for the final trust solution for LDACS, we can fully recommend the mentioned STS variant in Figure 3. For future research, an implementation of the STS protocol with ECDH and SIDH key flavors within a software simulation is foreseen, which can further our understanding of the strengths and weaknesses of the protocol and key exchange variations. This also allows us to gather accurate communication and computation times, some of which were simplified in this work.

Overall with a suitable protocol candidate and key agreement variation under investigation, the future cybersecurity architecture of LDACS has a cornerstone for laying the foundation of trust.

APPENDIX

AeroMACS Aeronautical Mobile Airport Communication System
AOC Aeronautical Operational Control
AS Aircraft Station
ATM Air Traffic Management
ATS Air Traffic Services
BC Broadcast
BER Bit Error Rate
CC Common Control
CNS Communication, Navigation and Surveillance
DC Dedicated Control
DCH Data Channel
DHKE Diffie-Hellman Key Exchange
ECDH Elliptic Curve Diffie-Hellman
FL Forward Link

GCM	Galois Counter Mode
GS	Ground Station
GSC	Ground Station Controller
ICAO	International Civil Aviation Organization
LDACS	L-band Digital Aeronautical Communication System
MAC	Medium Access Control
MAKE	Mutual Authentication and Key Exchange
MF	Multi Frame
MS	Master Secret
OFDM	Orthogonal Frequency-Division Multiplexing
PHY-SDU	Physical layer Service Data Unit
PKI	Public Key Infrastructure
PMS	Pre-Master Secret
RA	Random Access
RL	Reverse Link
SF	Super Frame
SIB	System Identification Broadcast
SIDH	Supersingular Isogeny Diffie–Hellman
SIKE	Supersingular Isogeny Key Encapsulation
STS	Station-To-Station
VDL	VHF Data Link

REFERENCES

[1] ICAO. Effects of Novel Coronavirus (COVID-19) on Civil Aviation:Economic Impact Analysis. https://www.icao.int/sustainability/Documents/COVID-19/ICAO_Coronavirus_Econ_Impact.pdf, May 2020 (accessed May 13, 2020).

[2] M. Schnell. Update on LDACS - The FCI Terrestrial Data Link. In *19th Integrated Communications, Navigation and Surveillance Conference (ICNS)*, pages 1–10, New York, NY, USA, April 2019. IEEE.

[3] A. Hall, J. Wingfield, G. De Moura, and K.K. Tiscareno. Advancing Cyber Resilience in Aviation: An Industry Analysis. *World Economic Forum*, pages 1–28, January 2020.

[4] A. Costin and A. Francillon. Ghost in the Air (Traffic): On Insecurity of ADS-B Protocol and Practical Attacks onADS-B Devices. *Black Hat USA*, pages 1–10, August 2012.

[5] M. Strohmeier, M. Schäfer, R. Pinheiro, V. Lenders, and I. Martinovic. On Perception and Reality in Wireless Air Traffic Communication Security. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1338–1357, October 2016.

[6] Mäurer, N. and Bilzhause, A. A Cybersecurity Architecture for the L-band Digital Aeronautical Communications System (LDACS). In *37th Digital Avionics Systems Conference (DASC)*, pages 1–10, New York, NY, USA, September 2018. IEEE.

[7] 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 (ICNS)*, pages 1A2/1–1A2–13, New York, NY, USA, April 2019. IEEE.

[8] T. Gräupl, C. Rihacek, and B. Haindl. LDACS A/G Specification. Sesar2020 pj14-02-01 d3.3.030, German Aerospace Center (DLR), Oberpfaffenhofen, Germany, August 2019.

[9] 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*, 32(11):22–33, November 2017.

[10] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.

[11] S. Blake-Wilson and A. Menezes. Authenticated Diffe-Hellman Key Agreement Protocols. In *International Workshop on Selected Areas in Cryptography*, pages 339–361, Heidelberg, Germany, August 1998. Springer.

[12] W. Diffie, P. C. Van Oorschot, and M. J. Wiener. Authentication and Authenticated Key Exchanges. *Designs, Codes and Cryptography*, 2(2):107–125, March 1992.

[13] C. Boyd, A. Mathuria, and D. Stebila. *Protocols for Authentication and Key Establishment*. Springer, Heidelberg, Germany, November 2019.

[14] BSI. Cryptographic Mechanisms: Recommendations and Key Lengths. Technical Report BSI TR-02102-1, Federal Office for Information Security Germany, March 2020.

[15] N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of computation*, 48(177):203–209, January 1987.

[16] G. Baumslag, B. Fine, M. Kreuzer, and G. Rosenberger. *A Course in Mathematical Cryptography*. Walter de Gruyter GmbH & Co KG, Berlin, Germany, May 2015.

[17] M. Abdalla, T.E. Bjorstad, C. Cid, B. Gierlichs, A. Hülsing, A. Luykx, K.G. Paterson, B. Preneel, A.-R. Sadeghi, T. Spies, M. Stam, M. Ward, B. Warinschi, and G. Watson. D5.4 Algorithms, Key Size and Protocols Report (2018). https://www.ecrypt.eu.org/csa/documents/D5.4-FinalAlgKeySizeProt.pdf, February 2018 (accessed May 20, 2020).

[18] M. Lochter and J. Merkle. Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation. RFC 5639 (Informational), March 2010.

[19] Alexander Rostovtsev and Anton Stolbunov. Public-Key Cryptosystem Based on Isogenies. *IACR Cryptology ePrint Archive*, pages 1–19, May 2006.

[20] David Jao and Luca De Feo. Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies. In *4th International Workshop on Post-Quantum Cryptography*, pages 19–34, Heidelberg, Germany, December 2011. Springer.

[21] Chloe Martindale and Lorenz Panny. How to Not Break SIDH. In *Conference for Failed Approaches and Insightful Losses in cryptology*, pages 1–19, Lyon, France, May 2019. International Association for Cryptologic Research.

[22] David Jao. Supersingular Isogeny Key Encapsulation. https://sike.org/files/SIDH-spec.pdf, April 2020 (accessed May 09, 2020).

[23] Reza Azarderakhsh, David Jao, Kassem Kalach, Brian Koziel, and Christopher Leonardi. Key Compression for Isogeny-based Cryptosystems. In *3rd International Workshop on ASIA Public-Key Cryptography*, pages 1–10, New York, NY, USA, May 2016. ACM.

[24] Craig Costello, David Jao, Patrick Longa, Michael Naehrig, Joost Renes, and David Urbanik. Efficient Compression of SIDH Public Keys. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 679–706, Heidelberg, Germany, April 2017. Springer.

[25] Carlisle Adams and Steve Lloyd. *Understanding PKI: Concepts, Standards, and Deployment Considerations*. Addison-Wesley Professional, Boston, MA, USA, November 2002.

[26] B. Crowe. Proposed AeroMACS PKI Specification is a Model for Global and National Aeronautical PKI Deployments. In *WiMAX Forum at 16th Integrated Communications, Navigation and Surveillance Conference (ICNS)*, pages 1–19, New York, NY, USA, April 2016. IEEE.

[27] T. Gräupl and M. Mayr. Method to Emulate the L-band Digital Aeronautical Communication System for SESAR Evaluation and Verification. In *34th Digital Avionics Systems Conference (DASC)*, pages 1–18, New York, NY, USA, October 2015. IEEE.

[28] R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 453–474. Springer, 2001.

[29] H. Krawczyk and P. Eronen. HMAC-based Extract-and-Expand Key Derivation Function (HKDF). RFC 5869 (Informational), May 2010.

[30] S. Josefsson and I. Liusvaara. Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032 (Informational), January 2017.

[31] Jintai Ding and Dieter Schmidt. Rainbow, a New Multivariable Polynomial Signature Scheme. In *3rd International Conference on Applied Cryptography and Network Security*, pages 164–175, Heidelberg, Germany, June 2005. Springer.

[32] RTCA/EUROCAE. Safety and Performance Requirements Standard for Baseline 2 ATS Data Communications (Baseline 2 SPR Standard). Rtca do-350 volume 1 & 2, RTCA/EUROCAE, Washington, DC / Malakoff, France, March 2016.

[33] Rafael Alvarez, Candido Caballero-Gil, Juan Santonja, and Antonio Zamora. Algorithms for Lightweight Key Exchange. *Sensors*, 17(7):1–14, June 2017.