Post-Quantum Secure Handover Mechanism for Next-Generation Aviation Communication Networks

Suleman Khan, Gurjot Singh Gaba, *Senior Member, IEEE*, Andrei Gurtov, *Senior Member, IEEE*, Leonardus J. A. Jansen, Nils Mäurer, *Member, IEEE*, and Corinna Schmitt

***Abstract*—The L-band Digital Aeronautical Communications System (LDACS) is a key advancement for next-generation aviation networks, enhancing Communication, Navigation, and Surveillance (CNS) capabilities. It operates with VHF Datalink mode 2 (VDLm2) and features a seamless handover mechanism to maintain uninterrupted communication between aircraft and ground stations (GSs), improving safety and efficiency in air traffic management (ATM). However, LDACS' handover process encounters significant security risks due to inadequate authentication and key agreement between aircraft and ground station controllers (GSCs) during handovers. This vulnerability threatens communications' confidentiality, integrity, and authenticity, posing risks to flight safety and sensitive data. Therefore, developing and implementing a robust security framework to protect aviation communications is essential. In response, we have proposed a security solution specifically designed to protect LDACS handovers. Our solution uses a mutual authentication and key agreement mechanism tailored for LDACS handovers, ensuring robust security for all types of handovers, including Intra GSC - Intra Aeronautical Telecommunication Network (ATN), Inter GSC - Intra ATN, and Inter GSC - Inter ATN. Our approach utilizes post-quantum cryptography to protect aviation communication systems against potential post-quantum threats, such as unauthorized access to flight data, interception of communication, and spoofing of aircraft identity. Furthermore, our proposed solution has undergone a thorough informal security analysis to ensure its effectiveness in addressing handover challenges and offering robust protection against various threats. It seamlessly integrates with the LDACS framework, delivering low Bit Error Rate (BER) and latency levels, making it a highly reliable approach in practice.**

Manuscript received 15 January 2024; revised 5 May 2024 and 31 May 2024; accepted 13 June 2024. Date of publication 19 June 2024; date of current version 27 August 2024. This work was supported in part by Trafikverket, Sweden, and Luftfartsverket, Sweden, under Automation Program II; in part by the Wallenberg AI, Autonomous Systems and Software Program (WASP), Sweden; and in part by the SESAR Joint Undertaking through the SEC-AIRSPACE Project funded by the European Union's Horizon Europe Research and Innovation Programme under Grant 101114635. *(Corresponding author: Gurjot Singh Gaba.)*

Suleman Khan, Gurjot Singh Gaba, and Andrei Gurtov are with the Department of Computer and Information Science (IDA), Linköping University, 58183 Linköping, Sweden (e-mail: suleman.khan@liu.se; gurjot.singh@liu.se; andrei.gurtov@liu.se).

Leonardus J. A. Jansen is with the Institute of Communication and Navigation, German Aerospace Center (DLR), 82234 Wessling, Germany (e-mail: leonardus.jansen@dlr.de).

Nils Mäurer was with the Institute of Communication and Navigation, German Aerospace Center (DLR), 82234 Wessling, Germany. He is now with the Department of Cyber Programmes, Airbus Defence and Space, 82024 Taufkirchen, Germany (e-mail: maeurer.nils@gmail.com).

Corinna Schmitt is with the Research Institute CODE, Universität der Bundeswehr München, 85579 Neubiberg, Germany (e-mail: corinna.schmitt@unibw.de).

Digital Object Identifier 10.1109/TGCN.2024.3417298

***Index Terms*—Aviation network, aviation security, BIKE, FCI, LDACS.**

I. Introduction

THE EUROPEAN air travel industry experienced a remarkable recovery in early 2023, with passenger numbers rising by 56% compared to the previous year, reaching 179 million [1]. This resurgence, indicating a return to pre-pandemic activity levels, has highlighted the limitations of the current ATM infrastructure, particularly the VDLm2 system [2]. VDLm2, designed for lower levels of air traffic, facilitates communication between aircraft and GSs. However, the sudden increase in air travel has exposed significant challenges to the VDLm2 system [3]. The primary issue is frequency congestion, where the limited available spectrum struggles to accommodate the increased volume of messages, leading to potential delays and communication inefficiencies such as increased latency, higher message loss rates, and overall reduced system throughput [4].

These issues can significantly affect the reliability and performance of air traffic control operations. Additionally, VDLm2's limited data capacity needs to be improved to manage the current volume and complexity of air traffic communications. These challenges highlight the urgent need to update and enhance the ATM infrastructure to efficiently manage increasing air traffic and ensure smooth operations in European skies. To address the challenges and shortcomings of the current ATM system, a new system called LDACS has been introduced. LDACS, an advanced communication system designed to meet the rising demands of air traffic and complex airspace management, operates alongside the existing VDLm2 system. Its main objectives include enhancing scalability, increasing data capacity, and improving resistance to interference [5]. The importance of LDACS in managing air traffic across European skies has been highlighted lately [6]. As a digital communication system used both on the ground and in the air, LDACS communicates with civil aircraft during flight, ensuring consistency throughout the journey [7]. During LDACS communication, aircraft on long routes maintain connectivity by switching between different GSs as they travel through various geographical regions. This process, known as handover, is crucial as it allows aircraft to establish connections with new GSs, ensuring uninterrupted communication throughout their flight [8].

There are different types of session handovers in LDACS communication, each serving specific purposes. Intra-GSC to intra-ATN handovers occur when communication shifts within a single GSC and remains within the same ATN region. Inter-GSC to intra-ATN handovers occurs when the aircraft moves between different GSCs within the same ATN region. Inter-GSC to inter-ATN handovers refers to transitions between GSCs in different ATN regions [9], [10]. LDACS uses two primary handover connection strategies to accomplish these transitions: "Make-before-Break" and "Break-before-Make". The Make-before-Break handover establishes a new communication link before terminating the existing one, ensuring seamless communication during the transition. On the other hand, the Break-before-Make handover involves breaking the existing communication link before establishing a new one. These handover strategies are crucial for maintaining seamless communication between aircraft and GSs, particularly during critical phases of flight when uninterrupted communication is vital for safety and operational efficiency.

However, in the context of ATM, the handover phase—when aircraft transition between GS, GSC, and across ATN—presents a critical vulnerability due to the lack of essential security measures like encryption and authentication. This vulnerability leaves these essential communication exchanges open to cyber threats, allowing attackers the potential to intercept, alter, or disrupt critical information. Such risks could compromise the safety and security of air operations. Addressing these threats requires the adoption of cryptographic primitives, including encryption for communication confidentiality, authentication to verify parties' identities, and integrity checks to maintain data authenticity during transmission [11]. Implementing these security measures is crucial for countering cyber threats and preserving the integrity and continuity of ATM systems.

To address this issue, the authors in [8] proposed a protocol for the intra-GSC handover. Their approach involves a "Make-before-Break" approach and uses a digital certificate-based solution. The solution comprises a Certificate Distribution Center (CDS) for secure certificate distribution to aircraft and GSs, an Online Certificate Status Protocol (OCSP) server for certificate revocation, and encrypted communication channels between GS and CDS. Both AS and GSs store valid CA certificates and have their own certificates with private keys for secure communication. However, their solution falls short in managing handovers between Inter GSC - Intra ATN and Inter GSC - Inter ATN, and fails to ensure anonymity. In the aviation industry, maintaining secure and anonymous handovers is crucial to prevent unauthorized access, data breaches, and potential threats to VIP transport or valuable cargo. Without robust security and anonymity measures, the consequences could include compromised passenger safety, loss of sensitive information, and increased vulnerability to cyber-attacks. Therefore, a more robust and cost-effective solution that addresses all handover types while ensuring anonymity is essential.

In response to the cybersecurity vulnerabilities identified during LDACS handovers (Intra GSC - Intra ATN, Inter GSC - Intra ATN, and Inter GSC - Inter ATN), we propose a cutting-edge security framework. To the best of our knowledge, this is the first security solution to comprehensively cover all types of session handovers for LDACS. Our framework employs a "Make-before-Break" strategy combined with Bit-flipping Key Encapsulation (BIKE), ensuring seamless transitions and providing a robust layer of security resilience against quantum computing threats—a crucial consideration in the post-quantum era. The proposed approach optimizes the handover process by reducing computational and communication costs compared with existing literature, while maintaining aircraft anonymity and providing robust quantum-resistant protection. Fully compliant with aviation industry standards, this advanced security solution significantly enhances the trustworthiness and reliability of LDACS. Protecting ATM systems against sophisticated cybersecurity challenges marks a significant leap forward in the security of aviation communications.

A. Our Contributions

- In this study, we have thoroughly investigated three possible cases for session handover in LDACS, namely (1) intra GSC - intra ATN, (2) inter GSC - intra ATN, and (3) inter GSC - inter ATN. Our research contribution lies in developing a robust security framework that is tailored to protect these session handovers. The proposed framework ensures secure and uninterrupted communication flows throughout the ATM network.
- Our proposed security framework for LDACS uses a "Make-before-Break" approach supported by two key security components: Physical Unclonable Functions (PUF) and BIKE protocol. PUF enables mutual authentication between the aircraft and GSC, while the BIKE protocol facilitates the generation of a strong and secure session key. By integrating the BIKE protocol, the system becomes more resilient against post-quantum cyber attacks, thereby ensuring the long-term security of LDACS.
- In order to comply with LDACS standards, we conducted thorough tests to ensure that our security framework meets the required communication and computational performance standards. We also evaluated the BER and latency to ensure the proposed security framework is compatible and efficient in LDACS environments.
- We conducted an informal analysis to verify the robustness of our proposed solution against various attacks.

B. Paper Organization

The paper is divided into several sections. Section II provides a background overview of LDACS. Section III covers handover types and methods, adversarial model, and security objectives. Section IV introduces the necessary preliminaries for our protocol. The proposed solution for session handover is presented in Section V, followed by an informal security analysis in Section VI. Section VII discusses the performance of the proposed solution, and finally, Section VIII concludes the paper and discusses future work.

II. LDACS Background

LDACS, a digital communication system, facilitates bidirectional communication and is ideal for flight guidance, safety, and regulatory flight communications. It caters to the current needs of Air Traffic Services (ATS) and Aeronautical Operational Control (AOC) data and supports digital voice communications. Additionally, LDACS is designed to be future-proof, accommodating upcoming applications like managing 4D trajectories. Recognized within the International Civil Aviation Organization's (ICAO) Future Communications Infrastructure (FCI), LDACS serves as a crucial link-layer network access technology, integrating the ATN with the IP-Protocol Suite (IPS). Each LDACS cell, operated by an LDACS GS, can serve up to 512 Aircraft Stations (AS), enabling communication through two directions: the Forward Link (FL) for ground-to-aircraft transmissions and the Reverse Link (RL) for aircraft-to-ground communications.

LDACS sets itself apart by offering variable channel quality, adjusted based on the employed Coding and Modulation Scheme (CMS). This adaptability enables LDACS to provide a net user-data rate between 230.53 to 1428.27 kbps in the FL and 235.30 to 1390.40 kbps in the RL per cell. These rates signify an enhancement of up to 90 times the net capacity compared to the previous VDLm2, marking a significant advancement in the efficiency and capacity of aeronautical communication systems [12]. In addition to its adaptability in channel quality, LDACS introduces a novel approach to aviation communication by integrating message prioritization. This feature significantly enhances the handling and delivery of critical information. LDACS prioritizes messages crucial for its operation, such as cell-entry, authentication, or handover messages, followed by mission-critical data like ATS data. By prioritizing these messages, LDACS ensures establishing a reliable connection before transmitting mission-critical data in its order of importance, focusing on maintaining seamless communication during transitions like handovers. In terms of data transmission, LDACS uses a structured approach. User data, including voice communications and other types, are transmitted via the Data Channel (DCH).

Conversely, control data is distributed across four logical channels, each serving a specific purpose [12]. Firstly, the FL's Broadcast Channel (BCCH) allows GS to broadcast important information about the cell to AS. Secondly, the FL's Common Control Channel (CCCH) empowers GS to allocate necessary resources to specific AS, enabling them to transmit user data in the RL DCH. Thirdly, the Random Access Channel (RACH) facilitates AS in requesting cell entry, an essential step for initiating communication. Lastly, the Dedicated Control Channel (DCCH) in the RL is designed to allow AS to request additional resources for sending user data, ensuring efficient data transmission processes. These channels appear at fixed locations in time, as defined by the LDACS frame structure. The superframe, consisting of a 6.72 ms BCCH (in FL)/RACH (in RL) block followed by four multi-frames, spans over 240 ms. The multi-frames are 58.32 ms long and consist of blocks of DCH (in FL and RL) and CCCH (in FL)/DCCH (in RL).

LDACS enhances aviation communication through sophisticated data prioritization and channel management and strongly emphasizes security. It employs a Public Key Infrastructure (PKI) to protect communications and uses both pre- and post-quantum certificates across AS and GS to secure the system against emerging cryptographic threats. To maintain trust within the network, certificate revocation checks are conducted during communications to confirm the validity of used certificates [9]. In addition to the PKI-based security, research is ongoing to explore alternative, more efficient methods such as a certificate-less Mutual Authentication and Key Exchange (MAKE) process. This innovative approach leverages PUFs and the BIKE protocol to provide security against various post-quantum threats, potentially reducing the overhead associated with traditional certificate management, making it more efficient and easier to maintain [13].

Another crucial component of LDACS is its secure handover capability, which ensures uninterrupted and safe communication as AS transitions between GS coverage areas. While LDACS currently offers robust security measures for handovers within the same GSC—termed intra-GSC handovers—it faces challenges in handling handovers across different GSCs, including intra-ATN and inter-GSC-Inter ATN handovers [8]. This security gap underscores the necessity for a comprehensive solution to protect communications across all handover scenarios and mitigate potential vulnerabilities. To address this, this paper proposes an end-to-end secure handover protocol for LDACS. This protocol encompasses all handover types (Intra GSC—Intra ATN, Inter GSC—Intra ATN, and Inter GSC—Inter ATN), ensuring complete security without compromising operational efficiency.

III. Handover Types and Methods, Adversary Model, and Security Objectives

This section discusses different types of session handovers in LDACS, explains handover connections, examines the adversary model, and outlines key security objectives to mitigate threats during session handovers.

A. Handover Types

This subsection discusses different types of session handovers in LDACS, including intra GSC - intra ATN, inter GSC - intra ATN, and inter GSC - inter ATN, shown in Fig. 1.

- The *Intra GSC - Intra ATN Session Handover* is a process that enables the transfer of an aircraft's communication from one GS, i.e., (GS_{111}) to another (GS_{112}) under the jurisdiction of GSC_{11} and ATN_1 as shown in Fig. 1. This handover is crucial in ensuring the aircraft maintains continuous and uninterrupted communication as it navigates through different sectors of the same GSC territory. This process is essential for ensuring seamless and reliable communication.
- In the scenario of *Inter GSC - Intra ATN Session Handover*, aircraft moves out of the coverage area of one GS, i.e., (GS_{111}) managed by a specific GSC_{11} and enters the range of a different GS, i.e., (GS_{121})

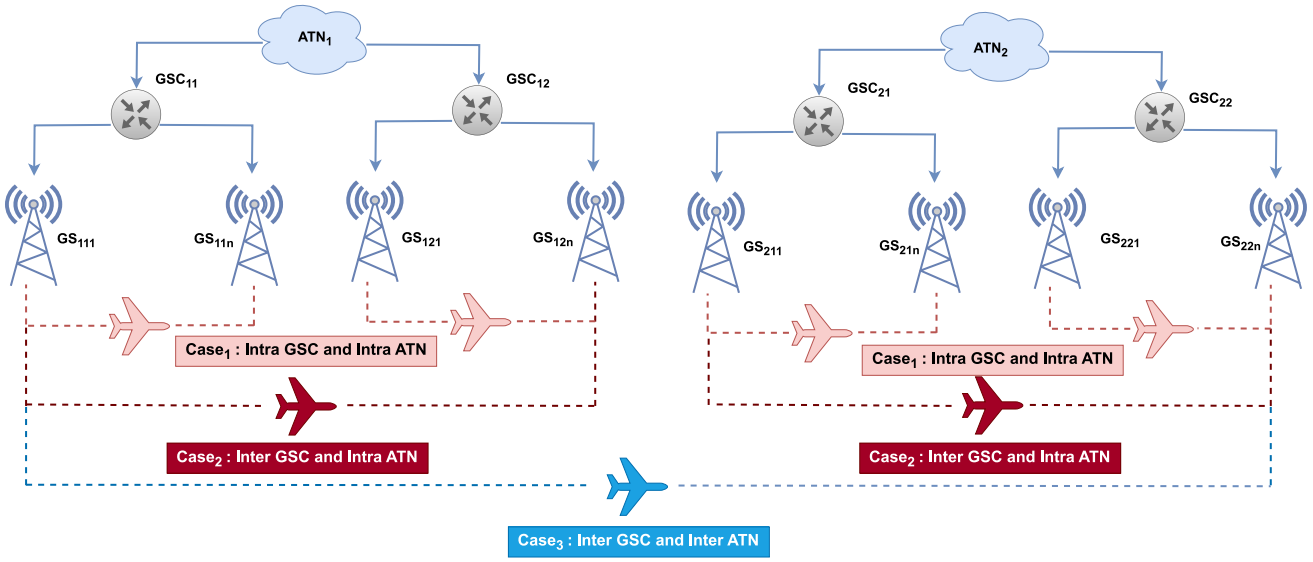


Fig. 1. Handover Scenarios in LDACS.

supervised by GSC_{12}under the same ATN_1 management. This handover is crucial for maintaining continuous communication, which is critical for safe navigation. It ensures uninterrupted connectivity for the aircraft within the ATN region, ensuring smooth navigation through different airspace domains.

- During a *Inter GSC—Inter ATN Session Handover*, aircraft move from one GS, i.e., (GS_{111}) managed by a GSC_{11} to another GS, i.e., (GS_{211}) under a different GSC_{21} within a different ATN_2. This handover allows the aircraft to move smoothly across different control regions and networks, ensuring uninterrupted communication throughout its journey, even when it crosses into territories managed by different ATN authorities.

B. Handover Methods

In LDACS, there are two primary method through which session handover is performed [14]:

1) In the **"Break Before Make"** scenario, the aircraft switches communication from one GS to another by disconnecting from the current GS and connecting to the next GS. It is crucial to manage this transition phase carefully, as the aircraft has no active communication link with any GS during this period. This disconnection can impact communication continuity, so minimizing disruptions is essential.
2) In the **"Make Before Break"** method, the aircraft establishes a connection with the next GS while still connected to the current one. This method ensures no interruption in communication during the transition, leading to a seamless session handover and eliminating potential communication gaps. This continuous connectivity is crucial for accurately relaying flight information and maintaining stable communication links during the transition, which is of utmost importance for safe flights and smooth operations.

We have considered the "Make Before Break" method as a use case to validate our proposed solution.

C. Adversary Model

Our security model for the LDACS is inspired by the Dolev-Yao (DY) adversarial framework [15]. The model focuses on a complex cyber-attack that aims to disrupt the communication between AS and GSC_{11} during the critical session handover phase from one GS, GS_{111}, to another, GS_{112}. Let's consider a scenario; the attacker is a malicious GS represented as GS_e. This scenario demonstrates how GS_e can exploit vulnerabilities in the wireless communication channel to interfere with and manipulate the handover process. In this scenario, an attack is carried out by GS_e, which intercepts the communication between the AS and GS_{112} to capture sensitive information transmitted during the session handover process. This initial interception is the first step in a series of potential malicious activities aimed at causing harm to the communication without disrupting it.

Once GS_e has successfully eavesdropped, it can use methods like signal jamming and noise injection to disrupt the handover request from the AS to GS_{112}, preventing it from reaching its intended destination. Additionally, GS_e can also interfere with messages from GS_{112} intended for the AS, ensuring that the AS receives communications from the attacker instead. GS_e can use its expertise and sophisticated tools to impersonate GS_{112} and deceive the AS into establishing a communication link with the adversary. This impersonation is a type of man-in-the-middle (MITM) attack, where GS_e positions itself between the communicating parties [16]. After successfully positioning themselves in the middle of a communication stream, GS_e can inject false information and malicious commands. This can disrupt normal operations, potentially leading to unauthorized command execution and compromising the safety and security of the operation. Additionally, the attacker can engage in replay attacks, where they re-transmit previously captured messages.

This adversary model highlights GS_e broad capabilities to eavesdrop, disrupt, deceive, modify, inject, and replay communications. Such actions pose significant threats to the security and safety of LDACS, underscoring the need for robust protective measures.

D. Security Objectives

The following security objectives in LDACS session handovers are critical for protecting communication [18], [19], [20]:

- *Mutual Authentication:* Involves verifying the identities of both communicating parties, the AS and the GSC, is crucial to ensure that they are legitimate and authorized entities. Mutual authentication is essential during handovers to prevent unauthorized entities from accessing the communication network and to prevent potential security breaches or malicious activities.
- *Integrity:* It is important to ensure that the information exchanged between AS and GSCs—such as flight paths and speed instructions—remains precise and unmodified, which is critical for the safety and reliability of communications. To protect against unauthorized changes that could pose significant safety and operational risks, implementing secure hash algorithms and Message Authentication Codes (MAC) is crucial. These security measures are key to ensuring that data is transmitted intact and unaltered, supporting the integrity of flight operations and the overall efficiency of ATM.
- *Confidentiality:* It is crucial to maintain confidentiality in LDACS message exchanges to secure information transfer between AS and GSCs during session handovers. During these exchanges, critical operational data such as aircraft positioning, communication frequencies, and navigation updates are transmitted. To prevent unauthorized access and protect confidentiality, it is essential to employ encryption and session keys, allowing access only to verified AS and GSCs. This security measure is necessary to uphold the safety and security of flight operations.
- *Freshness:* In LDACS communications, ensuring message freshness, especially during session handovers, is essential for preventing replay attacks. This can be achieved by making each message unique using timestamps or nonces. The system can detect attempts to compromise air traffic operations security with previously intercepted data by rejecting outdated messages. Ensuring message freshness during session handovers is crucial for communication reliability.
- *Anonymity:* Anonymity is essential in the LDACS framework as it hides the aircraft's identity during transmissions, protecting specific identifiers such as the ICAO address from potential cyber threats. This protection is crucial for ensuring the safety and security of VIP travel, sensitive missions, cargo, and confidential information. By making it difficult for attackers to target specific aircraft, anonymity significantly enhances the safety of passengers and high-value shipments.

By adhering to these few security objectives and understanding the adversary model, LDACS can effectively counter various cyber threats during handovers, ensuring the integrity and reliability of air traffic communications [21].

IV. Preliminaries

In this section, we discuss the main components of our proposed approach, including the use of PUFs for mutual authentication and the BIKE protocol for key generation. Additionally, we provide detailed information about our previous research, which forms the basis for this extended version.

A. Physically Unclonable Function (PUF)

PUFs are an essential component in protecting communication within cryptographic systems. Our approach involves incorporating PUFs because of their exceptional ability to generate unique responses (R) to specific challenges (C), represented as $R = PUF(C)$. The unique responses result from the inherent physical properties of the circuit or system where the PUF is implemented, making each PUF's challenge-response pair unique and challenging for adversaries to predict or replicate. The strength of PUFs lies in their unpredictability and the complexity involved in their creation, providing a robust security layer. In particular, PUFs are helpful in the LDACS system because of their resistance to being cloned, making them an ideal choice for secure authentication and communication. PUFs are designed so that any attempt to tamper with the physical setup of the system would lead to a change in the PUF output, making the tampered PUF unusable for duplication. This sensitivity to physical changes enhances the system's security against malicious attempts, ensuring that communication remains protected and authentic [22].

B. Bit Flipping Key Encapsulation (BIKE)

The BIKE Key Encapsulation Mechanism (KEM), under consideration for standardization by the National Institute of Standards and Technology (NIST), leverages Quasi-Cyclic Moderate Density Parity Check (QC-MDPC) for public key encryption, aiming to secure communications against quantum threats. Integral to BIKE, like other KEMs, are three pivotal functions: key generation, encapsulation, and decapsulation, ensuring a robust framework for secure message exchange in a future-proof cryptographic system, as shown in Fig. 2.

1) Key Generation: In the BIKE protocol, key generation begins with the creation of a private key S_K and a public key P_K, anchored by a parity-check matrix $\mathcal{H}$ derived from the QC-MDPC code. This matrix is structured around two circulant blocks, represented by polynomials h_0 and h_1, each designated with a specific Hamming weight w. The private key S_K comprises these polynomials along with an additional random sequence σ, formalized as $S_K = (h_0, h_1, \sigma)$. To formulate the public key P_K, the inverse of h_0 is computed (h_0^{-1}), and then multiplied by h_1, resulting in $h = h_1 \cdot h_0^{-1}$, which is publicly shared. The foundational security premise

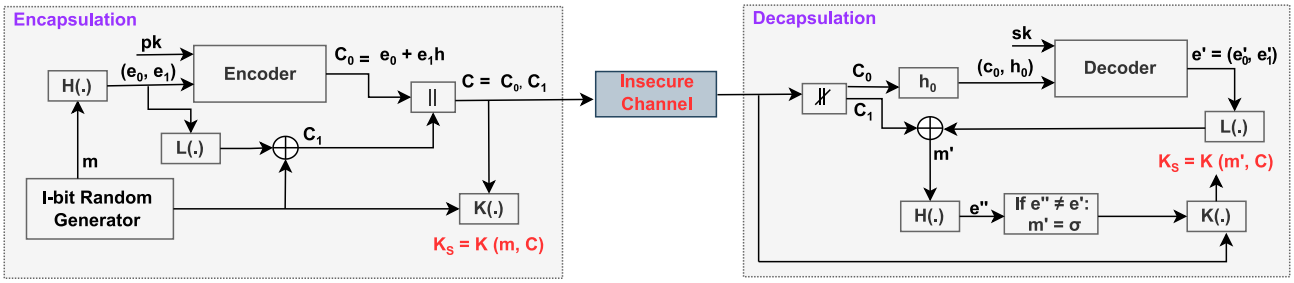


Fig. 2. Working principle of the BIKE protocol [17].

TABLE I
NOTATION AND DESCRIPTION

Notation	Description	Notation	Description
Private and Public element of AS	S_{EAS}, P_{EAS}	Messages	M_1, M_2 , M_3 , M_4
Temporary (pseudo) identity of AS	τ_{AS}	Addition, Subtraction, Bit-wise XOR	+, -, $\oplus$
Operator: Equality, Not Equal	$\stackrel{?}{=}$, $\neq$	Hash, Concatenation, Scalar Product	h, $\|\|$, $\cdot$
Truncated to first 24 bits	T_{24}	Message digest	β, θ, ψ
Random Numbers	R_v, R'_v	Variables	g, p, n, q, y, z
Session Key, Secret Key, Syndrome	K_S, S_K, S	If true then continue else abort	$T - C/E - A$
Message Authentication Code	h_{K_S}, h_ψ, h'_ψ	Nonces	N_1, N_2, N_3, N_4
Authentication Successful, Connection Termination	AS, CT	Ciphertexts, Error Vectors	C_0, C_1, C
Error Vectors	e_0, e_1	Fetch From Memory	$Fetch_M$
Random Oracles (Hash Functions)	H, L, K	Encryption, Decryption	E, D
International Civil Aviation Organization Address	$ICAO_A$	Embedded PUF	P_{AS}
Aeronautical Fixed Telecommunication Network	$AFTN_A$	Challenge, Response	C_{AS}, R_{AS}

of BIKE hinges on the computational difficulty of inferring S_K from P_K, a task presumed to be infeasible, especially for adversaries equipped with quantum computing resources. Additionally, BIKE employs a random oracle *K*, utilizing the message *m* and ciphertext $\mathcal{C}$ to generate a session key K_S, which is then used to secure subsequent communications.

2) Encapsulation: In the BIKE cryptographic scheme, the process begins with the generation of an ℓ-bit vector, denoted as *m*, which serves as the initial message. This message *m* is then inputted into a random oracle, modeled as a hash function $\mathcal{H}$, to compute two error vectors, e_0 and e_1. These error vectors undergo another transformation through a second random oracle *L*, resulting in $L(e_0, e_1)$. This output is then XOR-ed with the original message $\mathbf{m}$ to produce one component of the ciphertext, $\mathcal{C}_1$. Simultaneously, the public key P_K and the error vectors e_0 and e_1 are processed through an encoder function. This operation generates another component of the ciphertext, $\mathcal{C}_0$, which is mathematically represented as $e_0 + e_1 \cdot h$, where *h* is a part of the public key. The complete $\mathcal{C}$ is then constructed as a pair $(\mathcal{C}_0, \mathcal{C}_1)$, where $\mathcal{C}_0$ is $e_0 + e_1 \cdot h$ and $\mathcal{C}_1$ is $m \oplus L(e_0, e_1)$. Finally, the K_S is computed using both the original message *m* and the complete $\mathcal{C}$ through a function $K(m, \mathcal{C})$. This sequence of operations ensures the encryption of the message within the BIKE cryptographic framework, leveraging the properties of random oracles and specific encoding functions to generate $\mathcal{C}$ and session keys.

3) Decapsulation: The BIKE cryptographic scheme has a complex decapsulation process that securely generates the symmetric key K_S from the received ciphertext $\mathcal{C}$ using the receiver's private key S_K. The procedure consists of several well-defined steps, starting with decoding the first component of the ciphertext, $\mathcal{C}_0$. The decoding process involves calculating the syndrome *S* as the dot product of $\mathcal{C}_0$ and h_0, which is part of the private key. After that, the syndrome is passed through a Black-Grey-Flip (BGF) decoder to recover the estimated error vectors, represented as e_0 and e_1. The process proceeds to the critical phase of message reconstruction and verification, where $\mathcal{C}_1$, the second component of the ciphertext, is XOR-ed with the output of a random oracle function $L(e_0, e_1)$, aiming to reconstruct the original message, represented as $m = \mathcal{C}_1 \oplus L(e_0, e_1)$. This step includes a pivotal verification mechanism; if the hash function $H(m)$ does not yield a match with the recovered error vectors (e_0, e_1), indicating a potential discrepancy or tampering, *m* is then set to a predefined constant σ, signaling an unsuccessful decryption attempt. The culmination of this decapsulation procedure is the generation of the symmetric key K_S, executed by applying a *K* to both the potentially corrected message *m* and the original ciphertext $\mathcal{C}$, formulated as $K_S = K(m, \mathcal{C})$. This process ensures that K_S can be securely derived only by entities with the appropriate private key, protecting the integrity and confidentiality of communication within the BIKE cryptographic framework.

C. Background for LDACS MAKE

This subsection will provide an overview of our previous work [13], which serves as the foundation for this study.

In our previous work [13], we introduced a security solution that aims to enhance the security of LDACS against cyber attacks that target aviation communications between AS and

GSC_{11} via GS_{111}. To achieve this, our solution focuses on a MAKE process that employs PUF and BIKE to ensure the authenticity and integrity of the AS and GSC_{11}. The authentication process begins with the AS creating a nonce N_1 to ensure the message is fresh. It then retrieves its temporary identity τ_{AS} and message digest, $\theta = h(R_{AS})$ from memory. To ensure the integrity and authenticity of the data, the AS computes a MAC, $\alpha = h_\theta(N_1 \| \tau_{AS})$. The AS then prepare the message $M_1 = N_1 \| \tau_{AS} \| \alpha$ and sends it to GSC_{11}.

Upon receiving M_1, GSC_{11} first checks the freshness of N_1 to confirm that the message is fresh and not a replay of a prior session. If N_1 is found to be outdated, the connection is terminated to prevent any security breach. If N_1 is confirmed as fresh, GSC_{11} proceeds to retrieve τ_{AS} from its memory and selects the corresponding challenge-response pair, C_{AS} and R_{AS} and computes another message digest, $\beta = h(R_{AS})$. GSC_{11} then computes its own version of the MAC, $\alpha' = h_\beta(N_1 \| \tau_{AS})$, and compares it with the α received from the AS. This comparison is crucial because the shared secret θ/β, known only to the legitimate AS and GSC_{11}, is used to verify the authenticity of the communication. A mismatch in the MAC comparison results in the termination of the connection to protect against unauthorized access or tampering, ensuring that both the AS and GSC_{11} involved are legitimate and the data exchanged is secure and authenticated. To authenticate itself to the AS, GSC_{11} first generates a new nonce N_2 to ensure message freshness. Then, to protect against unauthorized access, the GSC_{11} encodes C_{AS} by performing a bitwise XOR operation with β, resulting in the encoded value ω. Using this encoded ω, the GSC computes a MAC as $\gamma = h_\beta(N_2 \| \omega)$, confirming authenticity and data integrity to the AS. Lastly, GSC_{11} prepares the message M_2, containing N_2, ω, and γ, and sends it to the AS, effectively concealing any sensitive information about C_{AS} and R_{AS} and maintaining the confidentiality of the system's security parameters.

Upon receiving message M_2, the AS verifies the freshness of nonce N_2 to ensure the message is not replayed. If the message is fresh, it computes the MAC $\gamma' = h_\theta(N_2 \| \omega)$. The AS then confirms the integrity and authenticity of the received message M_2 by comparing γ' with γ. If the information is authenticated, the AS will authenticate the GSC_{11}. For this purpose, it first derives the challenge $C_{AS} = \omega \oplus \theta$ and inputs C_{AS} into the PUF function, $P_{AS}(C_{AS})$ to generate the response R_{AS}. The AS then computes the message digest $\beta' = h(R_{AS})$, derives β by calculating $C_{AS} \oplus \omega$, and ensures $\beta' \stackrel{?}{=} \beta$ to authenticate the GSC_{11}. If the check fails, the connection is terminated to maintain security.

After successfully authenticating each other, AS and GSC_{11} proceed to the key generation and exchange process. The BIKE protocol generates a secure session key K_S. The BIKE protocol is highly resilient against post-quantum threats, making it an ideal choice for generating and exchanging K_S. Once the K_S is generated, the AS and the GSC_{11} encrypt their communications, ensuring that the data transmitted between them is secure from eavesdropping and tampering. Since the AS is already in flight and has a secure connection with GSC_{11} using a session key K_S, there may be situations where it needs to switch to another GSC_{12} within the same ATN_1 or to a different ATN_2. However, our previous work focused only on the MAKE process and did not address the complexities of session handover for LDACS, leaving the end-to-end security framework incomplete. To address these security gaps in LDACS, we have proposed a solution utilizing PUF and BIKE's existing infrastructure. Our primary focus is on securing the session handover process to ensure that the aircraft's communications remain secure and uninterrupted during its transition between GSCs, whether under the same or different ATN jurisdiction. By implementing this approach, we provide a comprehensive security framework for aviation communications.

V. Proposed Solution for Session Handover

This section proposes a new security framework to address the complex challenges associated with session handover scenarios within the LDACS network. These scenarios include Intra GSC—intra ATN, Inter GSC—intra ATN, and Inter GSC—inter ATN handovers. The proposed framework provides a comprehensive solution for these scenarios by establishing secure connections between the AS and the GSC. This end-to-end security is of the utmost importance in enhancing the security structure of the network and increasing the overall defense mechanisms of the LDACS network.

A. Assumptions

In LDACS, ATNs and GSCs are assumed to communicate via secure channels with other ATNs and GSCs, respectively. These channels are established using a shared secret key (S_K), a key element known only within the respective group of ATNs or GSCs. The S_K is the backbone of these secure exchanges, ensuring that ATNs can securely exchange important setup information, such as cryptographic parameters and authenticity-related information. Importantly, GSCs also utilize S_K, which is established through a Multi-Party Key Exchange (MPKE) approach within a single ATN's infrastructure. This approach plays a crucial role in the system, ensuring that GSCs can securely agree on an S_K, thereby protecting against eavesdropping and other security threats by ensuring that GSCs within the same ATN share all information securely.

B. Proposed Secure Handover Mechanism

This subsection discusses the proposed security framework for all three possible types of session handover in LDACS.

LDACS represents a significant advancement in modern aviation communication networks, providing a reliable infrastructure to support extensive and scalable connectivity. Each GS within this system can efficiently manage communications with up to 512 aircraft, which is crucial for handling high-volume and dynamic air traffic globally. However, due to a GS's limited coverage area, the establishment of a vast network overseen by a GSC becomes imperative. This network, comprising multiple GSs, plays a pivotal role in ensuring uninterrupted communication over vast distances. During flights, the ability to transition securely between these GSs is crucial for maintaining a continuous connection and protecting against security threats. To address the challenges

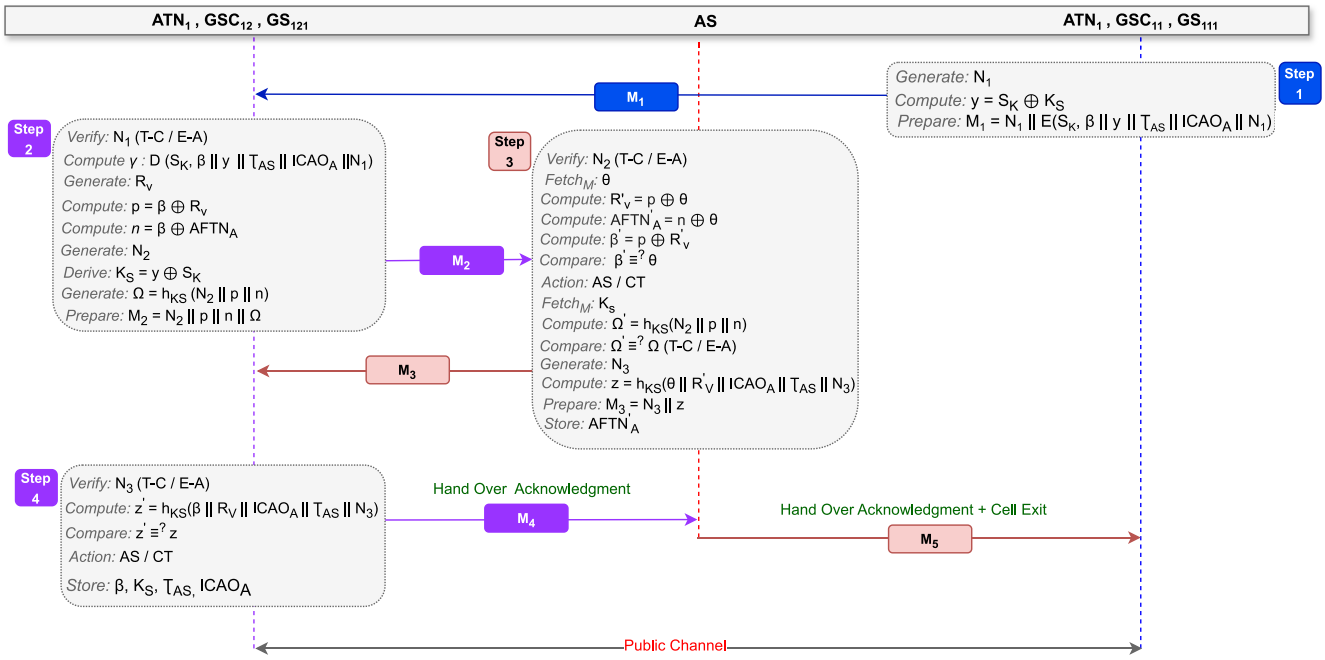


Fig. 3. Case 2: Exchange of messages by AS with approaching and previous GSC to perform Inter GSC - Intra ATN session handover.

associated with aircraft transitions between GSs within the LDACS framework, we propose using our previously introduced MAKE solution [13]. The MAKE solution leverages the structural hierarchy of the LDACS framework, where a single GSC manages multiple GSs, to enhance the security and efficiency of communication transitions. By integrating MAKE, we can ensure that as an aircraft moves from one GS to another, the connection switch is secure, maintaining continuous and protected communication throughout the flight.

Case 1: **Intra GSC - Intra ATN:** In this scenario, our proposed MAKE solution is employed since an AS is already in flight, connected through GS_{111} under the ATN_1 network, and managed by GSC_{11}. Both the AS and GSC_{11} have previously authenticated each other and established a session key (K_S), which is used to secure their communications. As the AS moves from GS_{111} to GS_{112}, it sends a handover notification encrypted with K_S to GSC_{11} through GS_{111}. Upon receipt, GSC_{11} facilitates the transition by informing GS_{112} of the ongoing session, preparing it to assume the communication relay responsibilities. GS_{112}, performing similarly to GS_{111}, acts solely as a relay. The AS continues its secure communication through GS_{112}, using K_S. Throughout this transition, GSC_{11} maintains the session context, ensuring seamless handovers as needed without the necessity for re-authentication. The persistent use of K_S by both the AS and GSC_{11} for encryption and decryption secures the communications throughout the flight. As the session ends, K_S is invalidated to terminate the communication securely. This method enhances security during AS transitions between GSs by streamlining key management, reducing latency, and minimizing communication and computational overheads.

Case 2: **Inter GSC - Intra ATN:** In the scenario depicted in Fig. 1, an AS is transitioning into airspace and needs to establish a connection with a GS_{121}, which is under the control of a different GSC_{12} but within the same ATN_1. Since GS_{121} is managed by a separate GSC_{12}, the AS must authenticate with GSC_{12} to enable secure and seamless communication within the same ATN_1 framework. The authentication steps performed by the AS and GSC_{12} are shown below:

Step 1: To ensure a secure handover between AS and GSC_{12}, both AS and GSC_{12} follow a series of steps as illustrated in Fig. 3. The process is initiated by GSC_{11}, which generates a nonce N_1 to confirm the communication's freshness and prevent replay attacks. Instead of transmitting the session key K_S in plaintext, which would be vulnerable to interception, GSC_{11} uses a secure method to conceal it. K_S is combined with S_K (a secure secret key shared among the relevant GSC entities) through a bitwise XOR $(\oplus)$ operation, resulting in a variable y that effectively hides K_S from potential adversaries. Following the key masking, GSC_{11} constructs a message M_1, which is structured as follows: $M_1 = N_1 \| E(S_K, \beta \| y \| \tau_{AS} \| ICAO_A \| N_1)$. This message encapsulates the message digest β, the encoded session key y, the AS temporary identity τ_{AS}, its International Civil Aviation Organization (ICAO) address $ICAO_A$, and the nonce N_1. All these components are encrypted with S_K and sent to GSC_{12} for further processing.

Step 2: Upon receiving M_1 via GS_{121}, GSC_{12} first validates the nonce N_1 to ensure message freshness. If the nonce is not fresh, the connection will be terminated. If the message is fresh, it decrypts M_1 using S_K, accessing the confidential data. Subsequently, for authentication with the AS, GSC_{12} generates a random number R_v and employs the XOR operation to combine R_v with β, yielding $p = \beta \oplus R_v$. Additionally, GSC_{12} uses XOR to combine β with the AFTN address, creating $n = \beta \oplus AFTN_A$. The XOR operation is selected for its ability to encode data securely; it effectively masks sensitive information, preventing its disclosure and ensuring that only

GSC_{12} and the AS with the correct data can decode and access the data. To ensure the message's freshness, GSC_{12} generates a new nonce, N_2, and then derives a K_S using the expression $K_S = y \oplus S_K$. This re-derived K_S is subsequently used to generate a new MAC, denoted as Ω, which incorporates N_2, p, and n. This Ω allows the AS to verify the message's integrity and confirm GSC_{12}'s legitimacy. Finally, GSC_{12} sends a message M_2 containing N_2, the encoded values p and n, and Ω through GS_{121}, effectively securing the communication and protecting against tampering and unauthorized access.

Step 3: Upon receiving message M_2, the AS begins the verification process by first checking the nonce N_2 for message freshness. This step is essential to prevent replay attacks. The AS then retrieves a message digest θ ($\theta = h(R_{AS})$) from its memory, generated during the registration phase in our previous MAKE [13], and is crucial for the authentication process. The next step involves the AS performing an exclusive OR (XOR) operation to derive $R'_v = p \oplus \theta$ and $AFTN'_A = n \oplus \theta$, both critical steps in the authentication process. Following this, $\beta' = p \oplus R'_v$ is computed. The critical moment in the verification process arises when β' is compared with the stored θ. If β' matches θ, it conclusively verifies the authenticity of GSC_{12}, ensuring the communication is secure and trustworthy. Conversely, a mismatch leads to the termination of the connection to safeguard against potential security threats. This mechanism ensures that only authenticated communications are accepted, effectively protecting against unauthorized access. To verify the integrity of the message M_2, the AS retrieves the K_S from its memory.

Using K_S, the AS generates a new MAC Ω', based on the combination of N_2, p, and n. This newly computed Ω' is then compared to the Ω received with message M_2. A successful match confirms the message's authenticity, indicating that it genuinely originated from GSC_{12} and has not been tampered with, thus ensuring the integrity and authenticity of the communication. Similarly, a mismatch will terminate the connection. Subsequently, to authenticate itself to GSC_{12}, the AS generates a new nonce, N_3, and constructs another MAC, $z = h_{K_S}(\theta \| R'v \| ICAO_A \| \tau_{AS} \| N_3)$, utilizing K_S. The AS then prepares a new message, M_3, which includes N_3 and z, and sends this to GSC_{12} via GS_{121} for further authentication checks. This step verifies the AS identity to GSC_{12} and sets the stage for secure future communications by confirming the AS $AFTN_A$ address.

Step 4: Upon receiving message M_3, GSC_{12} first checks the freshness of the nonce N_3 to ensure the message's timeliness and protect against replay attacks. Following this initial check, GSC_{12} proceeds to generate a new MAC z', using the derived session key K_S. The new MAC is created by combining elements such as β, R_v, $ICAO_A$, τ_{AS}, and N_3, resulting in $z' = h_{K_S}(\beta \| R_v \| ICAO_A \| \tau_{AS} \| N_3)$. Next, GSC_{12} compares this newly computed MAC z' with the MAC z received within message M_3. If z' matches z, it confirms that message M_3 is authentic and originates from a legitimate AS; thus, the authentication process is deemed successful. Conversely, a mismatch indicates a potential security threat, leading to the termination of the connection. After successfully authenticating the message, GSC_{12} stores critical information including β, K_S, τ_{AS}, and $ICAO_A$. In the next step, ATN_1 transmits M_4, a short ***HO_COM*** message as specified in [12] to the AS to acknowledge the handover. Finally, the AS sends message M_5, which informs GSC_{11} about the successful handover acknowledgment between the AS and GSC_{12} and exits the current cell. This is how the handover is performed between approaching GSC_{12} and AS within the same ATN without requiring a fresh MAKE.

The important point related to Case 2 is that the GSCs securely exchange the session keys until they expire. However, in case of no cooperation between GSCs within the same ATN or key expiry, the GSC and AS follow the solution we provided in Case 3, i.e., they establish a fresh session key. The proposed solution even allows the re-negotiation of keys for each GSC within the same ATN; however, it would add a lot of computation and communication costs.

Case 3: **Inter GSC - Inter ATN:** Consider a scenario where an AS, initially connected to GS_{111} under the control of GSC_{11} managed by ATN_1, is transitioning into the airspace serviced by GS_{211}, control by a different GSC_{21}, and managed by ATN_2. As the AS enters this new airspace, mutual authentication between the AS and the new GSC_{21} is required to establish a secure connection.

Step 1: The GSC_{11} initiates the Inter GSC - Inter ATN session handover process by generating a nonce N_1 to ensure message freshness and prevent replay attacks. Following this, GSC_{11} computes a hash of the session key, denoted as $\psi = h(K_S)$. This ψ is critical for two reasons: it enables the AS to verify the authenticity of the approaching GSC_{21} and likewise allows GSC_{21} to verify the authenticity of the AS. Subsequently, GSC_{11} prepare a message, M_1, structured as follows: $M_1 = N_1 \| E(S_K, \beta \| \psi \| \tau_{AS} \| \mathrm{ICAO}_A \| N_1)$. This message M_1 incorporates the N_1, alongside several key elements: β, ψ, τ_{AS}, and the ICAO_A. All these components are encrypted with S_K, ensuring the confidentiality of the information exchanged between GSC_{11} and GSC_{21}. Finally, the message is sent to GSC_{21} for further processing.

Step 2: Upon receiving the message M_1, GSC_{21} first checks the freshness of the message to confirm that it has not been replayed. If the message is outdated, the connection is terminated to protect against potential replay attacks. Once the freshness of the message is confirmed, GSC_{21} proceeds to decrypt M_1 using the shared secret key S_K. This decryption is crucial as it allows access to important information provided by GSC_{11}, revealing key components necessary to advance the security protocol. For authentication purposes with the AS, GSC_{21} generates a random number R_v and employs the XOR operation to securely combine R_v with β and β with the AFTN address. This operation yields two results: $p = \beta \oplus R_v$ and $n = \beta \oplus AFTN_A$. Additionally, GSC_{21} uses the XOR operation between R_v and the PUF challenge C_{AS} to encode the C_{AS}. The XOR operation is chosen for its ability to encode data securely, effectively masking sensitive information and ensuring that only GSC_{21} and the AS with the correct data can access and decode the data. Following this, GSC_{21} generates a new nonce N_2 to ensure the message freshness on the AS side. Subsequently, GSC_{21} prepares a

MAC, incorporating N_2, p, n, and g. This MAC, denoted as $\Omega = h_\psi(N_2\|p\|n\|g)$, serves as a compact proof of the message's critical components, securing their integrity and authenticity. Finally, GSC_{21} prepares the message M_2, which includes the new nonce N_2 along with computed values p, n, g, and the Ω. This message, M_2, is then sent to the AS for further authentication. *Step 3:* When the AS receives the message M_2, initially, the AS checks the message's freshness to protect against replay attacks, confirming that the communication is current and valid. Next, the AS retrieves a stored message digest θ from its memory, which plays a critical role in the authentication process. Using θ, the AS computes a random number R'_v through an XOR operation with the received value p. Subsequently, it computes $AFTN'_A$ by performing an XOR operation on n and θ, resulting in $AFTN'_A = n \oplus \theta$. This value is stored following a successful authentication process. To authenticate GSC_{21}, the AS then computes β' by XORing the received value p with R'_v, formulated as $\beta' = p \oplus R'_v$. A match between β' and θ confirms the legitimacy of GSC_{21}. Conversely, a mismatch would lead to the termination of the connection, preserving the network's security.

After successfully authenticating GSC_{21}, the AS proceeds to confirm the message's integrity and authenticity by fetching its session key K_S and preparing a new message digest ψ' as $\psi' = h(K_S)$. It then computes a new MAC, Ω', using ψ' and the received message components, expressed as $\Omega' = h_\psi(N_2\|p\|n\|g)$. Comparing this computed Ω' with the received Ω ensures that the message M_2 has not been tampered with and originates from a legitimate source, GSC_{21}. Following successful validation, the AS derives an important value C_{AS} by performing an XOR operation between R'_v and g. This operation facilitates the computation of R_{AS} through the function $P_{AS}(C_{AS})$ and allows for the determination of θ_n using the hash function $h(R_{AS})$. In order to securely share R_{AS} and P_{EAS} XOR operation is performed involving R_{AS} with R'_v and P_{EAS} with β', respectively. R_{AS} is sent to enable future authenticity verification, whereas P_{EAS} is sent to establish a session key with the GSC_{21} managed by ATN_2. Subsequently, to authenticate itself to the GSC_{21}, AS generates a new nonce N_3 and computes a MAC, z. This z incorporates several elements, including θ, R'_v, $ICAO_A$, τ_{AS}, y, v, and N_3. Next AS calculates τ_{ASn} as $h(ICAO_A\|R_{AS})_{T24}$ and stores $AFTN'_A$, τ_{ASn} and θ_n in its memory. Finally AS prepares the message M_3 which includes N_3, y, v, and z. This message, M_3, is then sent to GSC_{21} to affirm the AS authenticity, ensuring the integrity and continuity of secure communications.

Step 4: Upon receiving the message M_3 via GS_{211}, GSC_{21} first ensures the message's timeliness by verifying the freshness of the nonce N_3, protecting against replay attacks from previous sessions. Next, GSC_{21} calculates a new MAC $z' = h_\psi(\beta\|R_V\|ICAO_A\|\tau_{AS}\|y\|v\|N_3)$, to authenticate the origin and integrity of the message. This newly computed z' is compared with the received z. A match confirms the authenticity of the message M_3 from the AS, indicating that it is from a legitimate AS, while a discrepancy leads to the termination of the connection due to security concerns. Following successful authentication, GSC_{21} proceeds to further secure the communication by computing R'_{AS} through an XOR operation between R_v and y. It uses this R'_{AS} to calculate a new message digest of response β_n and derives the public key element P_{EAS} by XORing the computed β with the received v. Additionally, it generates a temporary AS identity by hashing $ICAO_A$ and R'_{AS}, truncated to 24 bits. These values are stored for future use.

Subsequently, GSC_{21} initiates the establishment of the session key (K_S) using the BIKE protocol, known for its secure key exchange mechanism. This process starts with generating a random vector m_{PRNG}, which is processed through three cryptographic random oracles named *H*, *L*, and *K*. The first oracle, *H*, outputs two error vectors $(e_0, e_1) = H(m_{PRNG})$. Using these error vectors along with the public element P_{EAS}, GSC_{21} creates the ciphertext components $C_0 = e_0 + e_1 \cdot h$ and $C_1 = m_{PRNG} \oplus L(e_0, e_1)$, resulting in the complete ciphertext $C = (C_0, C_1)$. Lastly, the third oracle, *K*, using inputs m_{PRNG} and *C*, generates the session key $K_S = K(m, C)$. GSC_{21} then computes the new MAC, $h_{K_S}(N_4\|C\|\beta_n)$, and prepares the message $M_4 = N_4\|C\|h_{K_S}(N_4\|C\|\beta_n)$, which is sent to the AS for further processing. This process enables AS to generate K_S and ensures the generation of the same key at GSC_{21} side.

Step 5: Upon receiving message M_4, the AS first verify the freshness of nonce N_4. If the message is not fresh, the connection will be terminated. If the message is fresh, It computes the syndrome through the expression $S = C_0 \cdot h_0$. It is important to note that an attacker cannot generate the syndrome, as it requires knowledge of h_0, which is only known to the AS. Next, the AS uses its secret element $S_{EAS} = (h_0, h_1, \sigma)$ and syndrome to deduce the error vectors using a BGF decoder, $e' = e'_0, e'_1 = DEC(S_{EAS}, (C_0, h_0))$. The AS then evaluates the integrity of the received ciphertext by computing $m'_{PRNG} = C_1 \oplus L(e')$, $e'' = H(m'_{PRNG})$ and comparing them: $e'' \equiv^? e'$. If $e'' \equiv^? e'$ are equal, it indicates that the received error vectors e_0, e_1 and the random vector m'_{PRNG} have been correctly recovered. In case of inequality, implying a mismatch or alteration, the connection is terminated to prevent further communication with an untrusted entity. Finally, the AS computes the session key by calculating $K_S = K(m'_{PRNG}, C)$. The AS then computes the MAC using the freshly generated key $h_{K_S}(N_4||C||\theta_n)$ and compares it with the received information from the GSC, $h_{K_S}(N_4||C||\theta_n) \equiv^? h_{K_S}(N_4||C||\beta_n)$. Equality indicates that the AS has generated a similar key and stores the key K_S, whereas inequality necessitates re-establishing the key. After successfully generating the K_S, AS will send a handover acknowledgment message M_5 to GSC_{21}. Finally, AS will complete the session handover by sending message M_6, which informs GSC_{11} about the successful handover acknowledgment between the AS and GSC_{21} and exits the current cell.

VI. Informal Security Analysis

In this section, we performed an informal security analysis of our proposed scheme for Cases 2 and 3. Since we already provided an informal analysis for Case 1 in our previous work [13], therefore it is not discussed here.

Theorem 1: Ensuring Mutual Authentication.

Proof (Case 2 and Case 3): In the proposed solution, only authorized entities like GSCs and AS are involved in communications. When AS receives the message $M_2 = N_2\|p\|n\|\Omega$, it retrieves a pre-stored value θ and computes $R'_v = \theta \oplus p$ and $\beta' = p \oplus R'_v$. If β' matches θ, AS authenticates GSC_{12}; otherwise, the mismatch will lead to the connection being terminated. To authenticate itself to GSC_{12}, AS then computes a MAC $z = h_{K_S}(\theta\|R'_v\|\,\mathrm{ICAO}_A\,\|\tau_{AS}\|N_3)$ and includes it in the response message M_3. Upon receiving M_3, GSC_{12} computes its own MAC $z' = h_{K_S}(\theta\|R'_v\|\,\mathrm{ICAO}_A\,\|\tau_{AS}\|N_3)$ and compares it with the received MAC z. A match confirms AS's authenticity, ensuring secure communication between AS and GSC_{12}, while a discrepancy terminates the connection. In the Case 3, when AS receives the message $M_2 = N_2\|p\|n\|g\|\Omega$, it retrieves θ and computes $R'_v = \theta \oplus p$ and $\beta' = p \oplus R'_v$. If β' matches θ, AS authenticates GSC_{21}; otherwise, the connection is terminated. To verify itself to GSC_{21}, AS computes a message digest $\psi' = h(K_S)$ and a MAC $z = h_{\psi'}(\theta\|R'_v\|\,\mathrm{ICAO}_A\,\|\tau_{AS}\|y\|v\|N_3)$, including it in the response message M_3. Upon receiving M_3, GSC_{21} computes its own MAC $z' = h_{\psi}(\beta\|R_V\|\,\mathrm{ICAO}_A\,\|\tau_{AS}\|y\|v\|N_3)$ and compares it with the received MAC z. A match confirms AS's authenticity, while a mismatch aborts the communication. ■

Theorem 2: Preserving Message Integrity.

Proof (Case 2 and Case 3): The proposed solution aims to ensure the integrity of messages in both Case 2 and Case 3 by employing MAC (Ω, Ω', z, z'). Consider an attacker trying to modify Case 2, message $M_2 = N_2||p||n||\Omega$, sent from GSC_{12} to AS. Suppose the attacker modifies the message component p. Upon receiving the modified M_2, AS will compute a new MAC, $\Omega' = h_{K_S}(N_2\|p\|n)$. Since the attacker has changed p, the newly computed Ω' will differ from the received Ω. This mismatch will lead to the termination of the connection. If there is a match, it confirms the message's authenticity and integrity, indicating it has not been tampered with and originates from GSC_{12}. Similarly, other messages such as M_1 and M_3 in Case 2 preserve their integrity and resist modification attacks. In Case 3, consider the message $M_2 = N_2\|p\|n\|g\|\Omega$ sent from GSC_{21} to AS. If an attacker modifies g, AS will use its session key K_S to generate a message digest $\psi' = h(K_S)$ and then compute a new MAC $\Omega' = h_{\psi'}(N_2\|p\|n\|g)$. Since g was altered, Ω' will not match the received Ω, resulting in connection termination and ensuring message integrity. Similarly, messages M_1, M_3, and M_4 in Case 3 maintain their integrity. ■

Theorem 3: Assuring Confidentiality.

Proof (Case 2 and Case 3): The proposed protocol aims to protect sensitive data, such as the PUF challenges (C_{AS}), responses (R_{AS}), and session keys (K_S), in both Case 2 and Case 3. In Case 3, consider if an attacker intercepts the message $M_3 = N_3\|y\|v\|z$ sent from AS to GSC_{12}. To ensure the confidentiality of information, our scheme encodes the sensitive information (P_{EAS}, R_{AS}) before sending it over the unsecure channel. The use of collision-resistant hash and MAC functions, along with XOR operations, ensures that no one can extract sensitive information from that message. Attackers cannot decode the information because they do not have any knowledge of R_v and β. Therefore, the confidentiality of these messages is preserved. Likewise, other messages in both cases preserve message confidentiality. ■

Theorem 4: Establishing Session Key Agreement.

Proof (Case 2 and Case 3): In Case 2, GSC_{11} securely shares the encoded K_S with GSC_{12} inside the encrypted message $M_1 = N_1\|E(S_K, \beta\|\psi\|\tau_{AS}\|\,\mathrm{ICAO}_A\,\|N_1)$. Upon receiving M_1, GSC_{12} decrypts it and extracts K_S using $K_S = y \oplus S_K$. This K_S is then used to create a MAC (Ω) for the message $M_2 = N_2\|p\|n\|\Omega$, which is sent to the AS. The AS verifies GSC_{12}'s possession of K_S by generating a corresponding MAC (Ω') and comparing it with the received MAC (Ω). Following this validation, AS generates a MAC (z) using K_S and sends it to GSC_{12}, who then computes its own MAC (z'). If z' matches z, it confirms that both parties hold the correct session key, ensuring secure communication.

In Case 3, GSC_{21} and AS establish the K_S using the BIKE protocol. GSC_{21} initiates by generating a random vector m_{PRNG} and calculating error vectors $(e_0, e_1) = H(m_{\mathrm{PRNG}})$. It then forms the ciphertext components $C_0 = e_0 + e_1 \cdot h$ and $C_1 = m_{\mathrm{PRNG}} \oplus L(e_0, e_1)$, which are sent to the AS. Upon receipt, AS computes the syndrome $S = C_0 \cdot h_0$ and uses a secret S_{EAS} to recover e'_0 and e'_1. Integrity checks are performed by verifying $m'_{\mathrm{PRNG}} = C_1 \oplus L(e'_0, e'_1)$ and ensuring $e'' = H(m'_{\mathrm{PRNG}})$ matches e'. Both GSC_{21} and AS then independently compute the K_S using the random oracle *K* with inputs m_{PRNG} and *C*. This method ensures that the session key remains secure, even in the presence of potential eavesdropping, due to the complexity and security embedded in the BIKE protocol. ■

Theorem 5: Ensuring Message Freshness.

Proof (Case 2 and Case 3): The proposed solution effectively prevents replay attacks through a nonce verification mechanism. Consider a Case 3 scenario, when an attacker intercepts and attempts to replay the message $M_4 = N_4\|C\|h_{K_S}(N_4\|C\|\beta_n)$, the nonce serves as a defense against such attacks. Both the AS and GSC_{21} are programmed to verify the freshness of each received message by checking its nonce. Upon receiving the replayed message, the AS extracts and examines the nonce N_4. If N_4 has been previously used, the AS identifies the replay attack and terminates the connection. This protective measure is consistently applied to all messages, including M_1, M_2, and M_3 for both Case 2 and Case 3 scenarios, ensuring comprehensive security. ■

Theorem 6: Ensuring Forward and Backward Secrecy.

Proof (Case 2 and Case 3): We propose adopting the BIKE protocol with ephemeral keys, ensuring each key exchange session utilizes a new and independent session key pair. This approach guarantees that the keys preceding and succeeding any given key are unrelated. The session key generation mechanism, denoted as $K_S = K(m'_{PRNG}, C)$, incorporates a random value in every instance, ensuring the independence of session keys. Consequently, compromising a single session key does not compromise the integrity of other session keys or sessions, thereby establishing both forward and backward secrecy within our protocol. The use of ephemeral keys ensures that a compromise in the future only affects the data from the compromised session, significantly minimizing potential data

exposure. As an additional layer of security, though not necessary, we recommend employing the private (S_{EAS}) and public (P_{EAS}) key pair for a single encapsulation and decapsulation process $(ENC(P_{EAS}, (e_0, e_1))$ and $DEC(S_{EAS}, (C_0, h_0)))$. Despite being complex and computationally intensive, this approach further ensures that compromising long-term key (S_{EAS}) does not affect the security of past and future session keys and sessions, achieving comprehensive forward and backward secrecy. ■

Theorem 7: Providing Anonymity.

Proof (Case 2 and Case 3): The proposed protocol adopts a unique approach to maintain identity anonymity. During the registration phase, the AS generates a temporary identity, named τ_{AS}, through the hash function $h(ICAO_A||R_{AS})_{T_{24}}$, where $ICAO_A$ is the AS identifier, and R_{AS} is a PUF response. This temporary identity τ_{AS} is then used to replace the AS's real identity in all subsequent communications. For example in Case 2, when GSC_{11} communicates with the approaching GSC_{12}, it sends a message $M_1 = N_1 \| E(S_K, \beta \| \psi \| \tau_{AS} \| \text{ICAO}_A \| N_1)$. In this message, the actual identity of the AS is encrypted, and τ_{AS} is included for future interactions. This ensures that the real identity of the AS is concealed, while τ_{AS} facilitates ongoing anonymous communication between GSC_{12} and AS. This method of using τ_{AS} in place of the real identity is consistently applied across the protocol, effectively preserving anonymity throughout the communication process. ■

VII. Performance and Comparative Analysis

In this section, we will discuss the security features of our proposed session handover solution and the associated computational costs. Additionally, we will examine message size, data overheads, and latency.

Table II demonstrates that the proposed session handover solution has successfully achieved a comprehensive set of security properties, including data confidentiality, message integrity, message freshness, anonymity, untraceability, and mutual authentication. Incorporating these security properties strengthens the proposed security solution, enabling it to effectively counter attacks such as replay, impersonation, modification, MITM, cloning, etc. By comparing the results in Table II, it becomes evident that the existing scheme [8] does not provide an anonymity feature for aircraft, which is crucial for protecting VIP travel schedules, sensitive missions, cargo, and confidential information.

Computation Costs: The detailed cost breakdown for our proposed solution across various LDACS session handover scenarios is shown in Table III. This includes the computational expenses of MACs, nonces, XOR operations, encryption, decryption, random number generation, hashing, and PUF during the mutual authentication and key agreement. Our analysis demonstrates that the proposed solution offers advantages in efficiency over traditional methods. According to computational costs mentioned in [8], our protocol is cost-effective, provides anonymity features, and offers robust defenses suitable for the post-quantum era.

TABLE II
Comparison of Security Features: Proposed Solution vs. Conventional Protocol

Goals		[8]	Proposed
Protection Against	Replay	✓	✓
	DoS	⋆	⋆
	Impersonation	✓	✓
	Modification	✓	✓
	MITM	✓	✓
	Cloning	✓	✓
	Known Key	✓	✓
Post Quantum Security		✓	✓
Message Integrity		✓	✓
Confidentiality		✓	✓
Message Freshness		✓	✓
Anonymity		⋆	✓
Mutual Authentication & Key exchange		✓	✓

✓: goal accomplished, ×: goal unaccomplished, ⋆: To some extent

Latency Model: Calculating latency is crucial for LDACS to ensure fast and reliable communications, which is essential for managing air traffic safely and efficiently. This improves system performance and maintains operations within regulatory standards. In order to compute the latency for our proposed solution, we used a latency emulation model that was specifically designed for the LDACS network. This model was first introduced in [23] and has since been updated in [24]. It is essential to accurately evaluate latency, particularly when dealing with data retransmission scenarios. The model's Forward Link latency is calculated using the following equation:

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

Here, $m_{FL}(t)$ represents the time until the start of the next CC frame, δ_{RX} is a binary indicator of the need for a retransmission, d_{MF} denotes the duration of a management frame, and *n* is derived from the length of the reverse link's medium access cycle from the perspective of the forward link.

Similarly, the Reverse Link latency is calculated using:

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

Here, $m_{RL}(t)$ marks the time until the start of the next DC slot, with δ_{RX} again indicating retransmissions. The term d_{MF} is consistent with its use in the Forward Link equation, and *N* corresponds to the reverse link medium access cycle length. We model δ_{RX}, which indicates the need for retransmission as a binary stochastic process influenced by the packet error rate. This rate is calculated using the BER and the packet length *l*. The probability that a packet is transmitted without any errors is given by: $P(\text{no error in packet}) = (1 - \text{BER})^l$. The probability that a packet contains errors necessitating potential retransmission is $P(\text{error in packet}) = 1 - (1 - \text{BER})^l$. These probabilities determine the value of δ_{RX}. If $\delta_{RX} = 1$, it indicates an error has occurred in the packet, and retransmission is necessary. Retransmitting the packet ensures the reliability and accuracy of data communication, which is

TABLE III
PROPOSED SOLUTION COMPUTATIONAL COST FOR ALL THE THREE SESSION HANDOVER CASES

Cases	AS	GSC	Total Cost
$Case_1$	D_{1T} + MAC_{4T} + XOR_{3T} + PUF_{1T} + R_{2T} + SM_{1T} + H_{4T}	E_{1T} + MAC_{4T} + XOR_{2T} + R_{3T} + H_{4T}	D_{1T} + E_{1T}+ MAC_{8T} + XOR_{5T} + PUF_{1T} + R_{5T} + SM_{1T} + H_{8T}
$Case_2$	MAC_{2T} + XOR_{3T} + R_{1T}	E_{1T} + D_{1T} + MAC_{2T} + XOR_{2T} + R_{2T}	D_{1T} + E_{1T}+ MAC_{4T} + XOR_{5T} + R_{3T}
$Case_3$	D_{1T} + MAC_{3T} + XOR_{7T} + PUF_{1T} + R_{1T} + SM_{1T} + H_{6T}	D_{1T} + E_{2T} + MAC_{3T} + XOR_{5T} + R_{4T} + H_{6T}	D_{2T}+ E_{2T} + MAC_{6T} + XOR_{12T} + PUF_{1T} + R_{5T} + SM_{1T} + H_{12T}

Acronyms: PUF: Physically Unclonable Function, *MAC:* Message Authentication Code, *H*: Hash, *XOR*: Bit-wise XOR, *R*: Random number *E*: Encryption, *D*: Decryption, *SM*: Scalar Multiplication.

TABLE IV
PARAMETER VALUES FOR LATENCY TIMING FOR THE LDACS [9]

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

crucial in scenarios where errors could lead to significant data loss or operational issues. For a comprehensive understanding of this model, we refer the reader to [23], [24] and Table IV, where relevant parameters are listed.

Message size: Each message in the LDACS begins with a 48-bit header. In compliance with the recommendations from [25], our proposed solution employs 128-bit MAC tags for secure communication. All communication entities use 24-bit identifiers for identification purposes, and nonces are 64 bits. Additionally, the size of the public key and cipher is 22,054 bits, respectively, ensuring robust encryption and security measures throughout the communication system.

Data overhead: We assign the aforementioned message sizes to every message in the proposed protocol.

Case 1: In Case 1, a total of four messages are exchanged between the AS and the GSC_{11} [13]. Message M_1 is transmitted from AS to GSC_{11} and comprises 264 bits. This message includes a 48-bit header, a 24-bit identity, a 64-bit nonce, and a 128-bit MAC tag. Message M_3 is also sent from AS to GSC_{11}, comprising 240 bits, a 48-bit header, a 64-bit nonce, and a 128-bit MAC tag. The total size of the messages sent by AS is 504 bits. Message M_2 and Message M_4 are sent from GSC_{11} to AS. M_2 includes a 48-bit header, a 64-bit nonce, a 128-bit XOR operation, and a 128-bit MAC tag. M_4 consists of a 48-bit header, a 64-bit nonce, a 22,054-bit cipher denoted as *C*, and a 128-bit MAC tag. The total bit count for the messages sent by GSC_{11} is 22,662 bits.

Case 2: The handover process in Case 2 starts with GSC_{11} sending message M_1 to GSC_{12}, initiating the communication transition. The message M_1 includes a 48-bit LDACS header, a 64-bit nonce N_1, and an encrypted section comprising a 128-bit digest β, a 128-bit encoded session key K_S, the AS temporary ID τ_{AS} and its ICAO number (each 24 bits), and a 64-bit nonce N_1, totaling 480 bits. Subsequently, GSC_{12} sends message M_2 to the AS, which contains a 48-bit header, a new 64-bit nonce N_2, a 128-bit MAC tag Ω, and two 128-bit XOR components *p* and *n*, for a total of 496 bits. The AS then responds with M_3, which includes a 48-bit header, a 64-bit nonce N_3, and a 128-bit MAC tag *z*, totaling 240 bits. Following this, GSC_{12} sends a concise 48-bit message, M_4, to the AS to confirm the successful handover. The session handover is completed as the AS sends M_5, a 64-bit acknowledgment, and a cell exit message to GSC_{11}.

Case 3: The Case 3 session handover process facilitates the transition of control between the AS and GSC_{21} through a series of message exchanges. Initially, in Step 1, a message M_1 containing 480 bits is sent from GSC_{11} to GSC_{21}. This message comprises a 48-bit header, a 64-bit nonce N_1, encrypted parameters including 128-bit digests β and ψ, as well as τ_{AS}, the ICAO number (each 24 bits), and another 64-bit nonce N_1. In Step 2, GSC_{21} sends a 624-bit message M_2 to the AS, featuring a 48-bit header, a 64-bit nonce, a 128-bit MAC tag Ω, and three 128-bit XOR components identified as *p*, *n*, and *g*. Step 3 sees the AS responding with a large 22,422-bit message M_3, which includes a 48-bit header, a 64-bit nonce, a 128-bit MAC tag *z*, and two XORs: *y* at 22,054 bits and *v* at 128 bits, sent to GSC_{21}. In Step 4, GSC_{21} sends a 22,294-bit message M_4 back to the AS, comprising a 48-bit

TABLE V
PROPOSED SOLUTION COMMUNICATION COST COMPARISON

Cases	Messages	Cost Addition	Individual Cost	Total Cost
$Case_1$	M_1	48+64+24+128	264	23166
	M_2	48+64+128+128	368	
	M_3	48+64+128	240	
	M_4	48+64+22054+128	22294	
$Case_2$	M_1	48+64+128+128+24+24+64	480	1328
	M_2	48+64+128+128+128	496	
	M_3	48+64+128	240	
	M_4	48	48	
	M_5	48+16	64	
$Case_3$	M_1	48+64+128+128+24+24+64	480	45932
	M_2	48+64+128+128+128+128	624	
	M_3	48+64+128+22054+128	22422	
	M_4	48+64+22054+128	22294	
	M_5	48	48	
	M_6	48+16	64	

Message components sizes: LDACS Header: 48 bits, *ICAO*: 24 bits, *Nonce*: 64 bits, *Handover Acknowledgement:* 48 bits, *Cell Exit:* 16 bits *MAC:* 128 bits, *XOR Operation*: 128 & 22054 bits, *Bike public key* (P_{EAS}): 22054, *Bike cipher size* (C): 22054, *Session key size* (K_S): 128 bits.

header, a 64-bit nonce, a 128-bit MAC tag h_{K_S}, and a cipher component C of 22,054 bits. Step 5 involves the AS sending a 48-bit handover acknowledgment message M_5 to GSC_{21}, and finally, in Step 6, the AS sends a 64-bit message M_6 to GSC_{21}, which includes both the handover acknowledgment and a cell exit message, effectively concluding the session handover. The total communication cost for this session handover amounts to 45,932 bits, ensuring a secure and efficient transfer of control.

Compared to traditional certificate-based solutions [8], our proposed solution demonstrates significant improvements in communication cost, as illustrated in Fig. 5. It effectively reduces the communication load from 13,288 bits to 504 bits on the AS side and from 51,584 bits to 22,662 on the GSC side. This results in a total communication cost reduction of 41,706 bits. Our proposed solution maintains equivalent security levels while providing anonymity, significantly lowering communication requirements for both AS and GSC and enabling secure and efficient handovers. Figure 6 illustrates our proposed session handover solution for LDACS systems, highlighting improved message exchange efficiency during the handover phases. Our proposed solution is more comprehensive than the protocol by Mäurer et al. [8], which only deals with Case 1 handover and requires six messages for completion. For Case 1, our solution accomplishes a session handover with just four messages—two from the AS and two from the GSC. For Case 2, our proposed solution requires five messages, with two originating from the AS and three from the GSC. In Case 3, the handover process is effectively completed with six messages, evenly distributed with three each from the AS and GSC. These improvements significantly reduce communication overhead and enhance operational efficiency.

Latency calculation: Table IV presents the initial BER conditions for the LDACS. It specifies an optimal BER of 10^{-6} and a more challenging scenario at 10^{-5}. As stated in Section II LDACS features a message prioritization system that organizes messages into eight service categories, assigning the highest priority to authentication and handover messages. This ensures they are the first to be processed by the LDACS scheduler. Nevertheless, if resource allocations have already been made in the DC and scheduled via the CC, the scheduler is bound to respect these existing commitments. The relevant conditions and equations outlined in Table IV must still be adhered to, as specified in the LDACS documentation [12].

Applying the Forward- and Reverse Link models from Table IV onto the proposed solution communications cost from Table V allows us to model the latency for cases 1 to 3. Please note the following assumptions are necessary:

- Inter GSC communications (i.e., message M_1 in Figure 3 and Figure 4) are assumed to take 5 ms total.
- Computation times at each node (i.e., AS, GS, GSC) are also estimated to take 5 ms, following [24].
- Since re-transmissions are modeled via a probabilistic model, all handover scenarios, i.e., cases 1 to 3, are re-run 1000 times each.

Please further note, as equations in Table IV depend on the amount of aircraft in the cell, the amount of aircraft gradually increases from 1 to 512. Lastly, all cases are tested for both stated BERs. The results are listed in Figure 7.

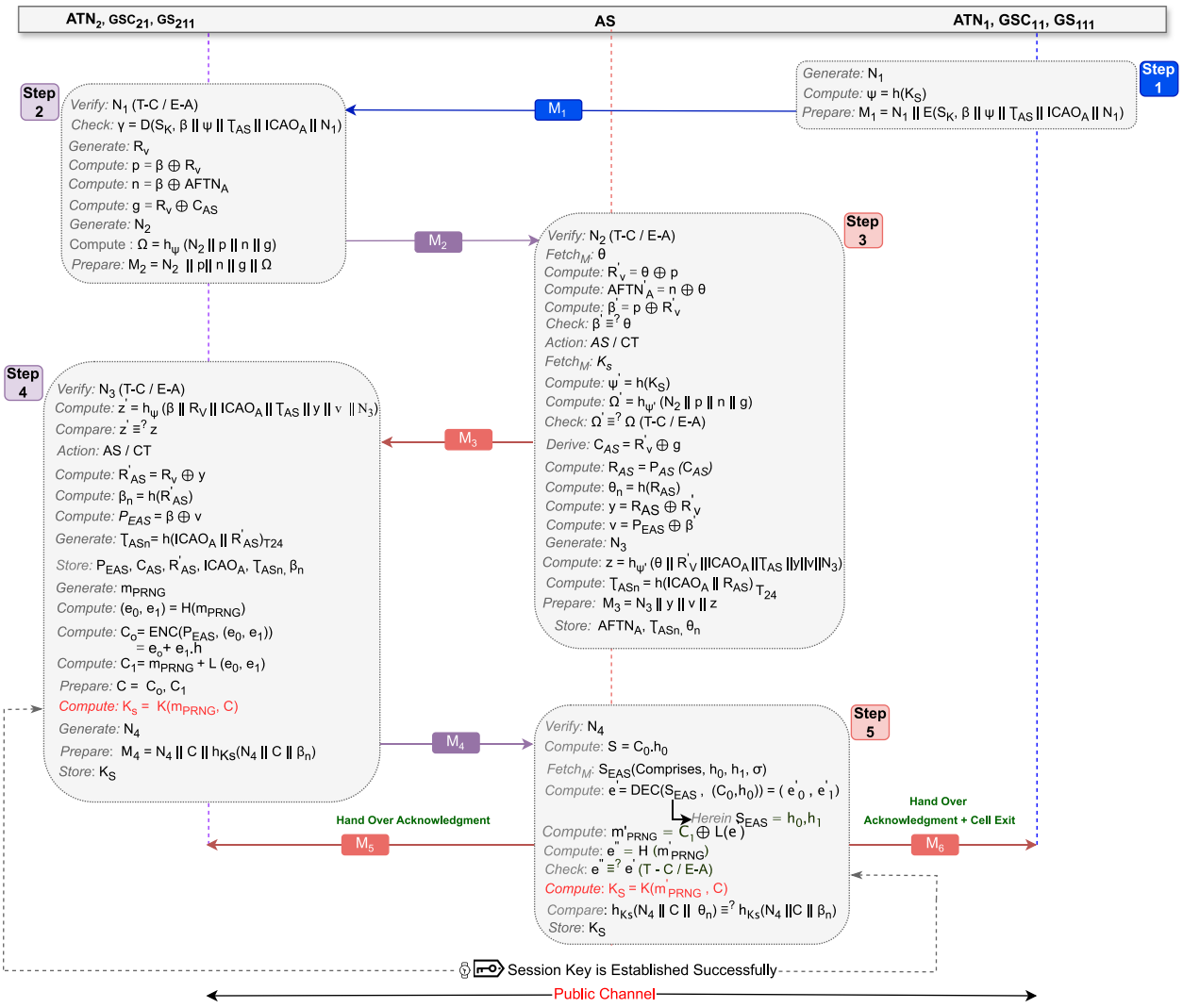


Fig. 4. Case 3: Exchange of messages by AS with approaching and previous GSC to perform Inter GSC - Inter ATN session handover.

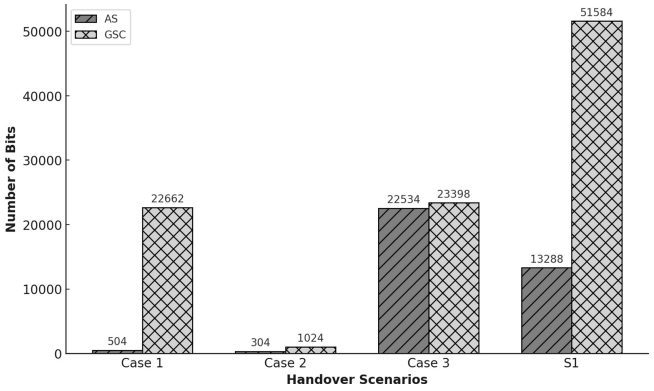


Fig. 5. Comparison of communication cost: Proposed Solution vs. Traditional Protocol. Protocol in comparison, S_1 (Intra GSC): [8].

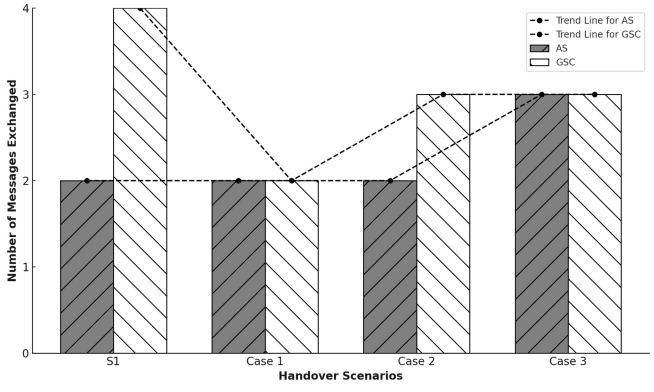
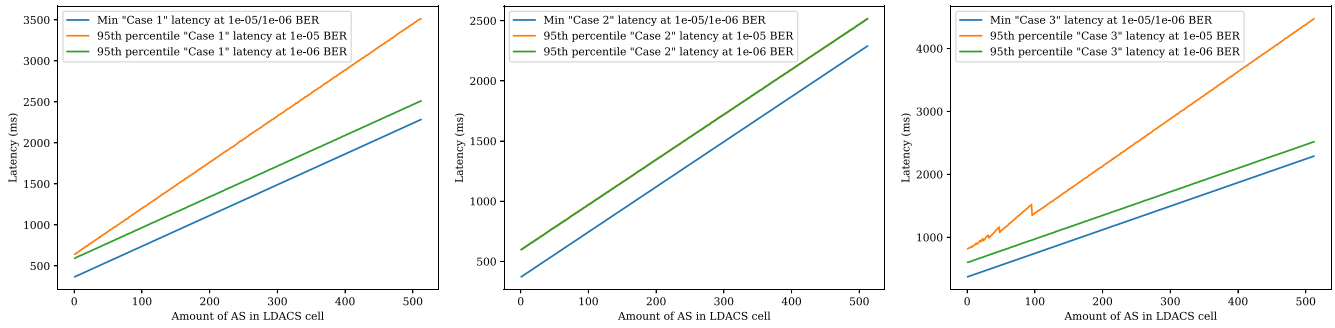


Fig. 6. Comparative Analysis of Message Exchanges: Proposed vs. Traditional Scheme S_1 [8].

As seen in Fig. 7(a), the Case 1 latency ranges between 450 and 670 ms with few aircraft and even reaches 2000 to 3500 ms with many aircraft in the cell, all handling high-priority traffic simultaneously. The BER of 10^{-5} does have an approximately 1000 ms latency worsening effect due to the comparatively large BIKE ciphertext message. In Fig. 7(b), the Case 2 latency ranges between 450 and 550 ms with few aircraft in the cell and reaches 2200 to 2500 ms with many aircraft in the cell, all handling high-priority traffic simultaneously. Most notably, the BER has no measurable



(a) Case 1 Latency (b) Case 2 Latency (c) Case 3 Latency

Fig. 7. Handover latencies in milliseconds depending on the amount of aircraft in the cell per Cases 1, 2, and 3.

influence here, much to the very small messages used in Case 2. In Fig. 7(c), the Case 3 latency ranges between 660 and 900 ms with few aircraft in the cell and even reaches 2000 to 4300 ms with many aircraft in the cell, all handling high-priority traffic simultaneously. Most notably, the BER here has more impact on all three cases, which is explainable by the two large BIKE public key and ciphertext messages.

The proposed solutions, Cases 1-3, impact LDACS latency due to the need to complete the initial authentication and key agreement sequence before securely exchanging LDACS user/control data. Once this sequence is completed, both the AS and GSC have a session key that can be used to protect LDACS user/control data. Therefore, the latency performance impact of the presented solutions is an additional approximately 450, 450, or 660 ms (for Cases 1-3) for initial authentication and key agreement before securely exchanging any LDACS user/control data. This is the most likely scenario at an LDACS working point BER of 10^{-6} with only a few aircraft in the cell simultaneously exchanging high-prioritized traffic. However, it should be noted that in a scenario with many aircraft in the cell exchanging high-prioritized traffic and at the worst LDACS BER of 10^{-5}, the latency can increase to approximately 4300 ms due to bit-flips and necessary message re-transmissions because of the relatively large post-quantum messages. Despite this, the latency remains below the required 10-second threshold to comply with RTCA DO-350A [26].

VIII. Conclusion and Future Work

LDACS represents a substantial advancement in next-generation aviation communication networks by offering enhanced bandwidth, capacity, and coverage compared to traditional systems. It is designed to overcome current limitations in aviation communications by incorporating advanced security measures, ensuring the confidentiality, integrity, and availability of aeronautical communications. One critical aspect of LDACS is its seamless handover mechanism, which is essential for maintaining continuous and secure communication as aircraft transition between different GSs. However, this process is challenged by significant security vulnerabilities due to the absence of robust security protocols during intra- and inter-ATN transitions. This lack of security exposes the system to various cyber threats, including eavesdropping, data injection, modification, MITM, etc. These vulnerabilities can lead to unauthorized access and manipulation of sensitive data, undermining communications' confidentiality, integrity, and authenticity. This not only compromises flight safety but also poses a risk to ATM's overall security. To address these issues, we have proposed a secure handover solution that utilizes lightweight cryptographic elements such as MAC, nonce, PUF, and bitwise XOR operations. Additionally, we have integrated the BIKE protocol to enhance defenses against post-quantum cyber threats. Our research findings indicate that during session handovers, the AS requires only 504 bits in Case 1, 304 bits in Case 2, and 22534 bits in Case 3. Conversely, the GSC requires 22662 bits in Case 1, 1024 bits in Case 2, and 23398 in Case 3. This strategic management of message sizes and counts has significantly optimized the handover process, reducing the number of required messages to four in Case 1, five in Case 2, and six in Case 3, thereby minimizing the communication load. The proposed solution adds a latency of approximately 450 to 2500 milliseconds before any LDACS user or control data exchange occurs at the working BER point of LDACS within the RTCA DO-350A 10-second threshold. Our future plans for LDACS include the development of air-to-air authentication, a groundbreaking feature that will create a network in the sky and improve the security and versatility of aircraft communications. This authentication mechanism will allow aircraft to communicate directly with each other, enhancing the efficiency and safety of the aviation system. This step is essential to meet modern aviation's evolving demands and ensure that LDACS continues to play a significant role in the future of aviation communication networks.

References

[1] (Eurostat, Luxembourg City, Luxembourg). *Air Passenger Transport-Monthly Statistics*, Dec. 2023, Accessed: Dec. 19,2023. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Air_passenger_transport_-_monthly_statistics

[2] (SESAR JU, Brussels, Belgium). *VDL Mode 2 Capacity and Performance Analysis*, Nov. 2015, Accessed: Dec. 19, 2023. [Online]. Available: https://www.sesarju.eu/sites/default/files/documents/news/SJU_VDL_Mode_2_Capacity_and_Performance_Analysis.pdf

[3] S. Eskilsson, H. Gustafsson, S. Khan, and A. Gurtov, "Demonstrating ADS-B AND CPDLC attacks with software-defined radio," in *Proc. Integr. Commun. Navigat. Surveillance Conf. (ICNS)*, 2020, pp. 1B2–1.

[4] N. Mäurer, T. Guggemos, T. Ewert, T. Gräupl, C. Schmitt, and S. Grundner-Culemann, "Security in digital aeronautical communications a comprehensive gap analysis," *Int. J. Crit. Infrastruct. Protect.*, vol. 38, Sep. 2022, Art. no. 100549.

[5] M. Schnell, "Update on LDACS-the FCI terrestrial data link," in *Proc. 19th Integr. Commun., Navigat. Surveillance Conf. (ICNS)*, 2019, pp. 1–10.
[6] R. J. Kerczewski, J. M. Budinger, and T. J. Gilbert, "Technology assessment results of the Eurocontrol/FAA future communications study," in *Proc. IEEE Aerosp. Conf.*. IEEE, 2008, pp. 1–13.
[7] M. A. Bellido-Manganell et al., "LDACS flight trials: Demonstration and performance analysis of the future aeronautical communications system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 1, pp. 615–634, Feb. 2021.
[8] N. Mäurer, T. Gräupl, C. Schmitt, C. Rihacek, and B. Haindl, "A secure ground handover protocol for LDACS," in *Proc. Int. Workshop ATM/CNS*, 2022, pp. 1–8.
[9] N. Mäurer, T. Gräupl, C. Schmitt, and G. D. Rodosek, "PMAKE: Physical unclonable function-based mutual authentication key exchange scheme for digital aeronautical communications," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, 2021, pp. 206–214.
[10] N. Mäurer, T. Gräupl, C. Schmitt, G. D. Rodosek, and H. Reiser, "Advancing the security of LDACS," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 4, pp. 5237–5251, Dec. 2022.
[11] A. D. Dwivedi, R. Singh, U. Ghosh, R. R. Mukkamala, A. Tolba, and O. Said, "Privacy preserving authentication system based on non-interactive zero knowledge proof suitable for Internet of Things," *J. Ambient Intell. Human. Comput.*, vol. 13, pp. 4639–4649, Oct. 2022.
[12] T. Gräupl, N. Mäurer, and M. Schnell, "Final LDACS A/G specification," document PJ.14-W2-60 TRL6, SESAR JU, Brussels, Belgium, 2023.
[13] S. Khan, G. S. Gaba, A. Gurtov, N. Mäurer, T. Gräupl, and C. Schmitt, "Enhancing cybersecurity for LDACS: A secure and lightweight mutual authentication and key agreement protocol," in *Proc. IEEE/AIAA 42nd Digit. Avionics Syst. Conf. (DASC)*, 2023, pp. 1–10.
[14] N. Mäurer, T. Gräupl, and C. Schmitt, "L-band digital aeronautical communications system (LDACS)," Internet Eng. Task Force, RFC 9372, 2023.
[15] S. Khan, A. Gurtov, A. Breaken, and P. Kumar, "A security model for controller-pilot data communication link," in *Proc. Integr. Commun. Navigat. Surveillance Conf. (ICNS)*, 2021, pp. 1–10.
[16] S. Khan, G. S. Gaba, A. Braeken, P. Kumar, and A. Gurtov, "AKAASH: A realizable authentication, key agreement, and secure handover approach for controller-pilot data link communications," *Int. J. Crit. Infrastruct. Protect.*, vol. 42, Sep. 2023, Art. no. 100619.
[17] M. R. Nosouhi et al., "Weak-key analysis for BIKE post-quantum key encapsulation mechanism," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2160–2174, 2023.
[18] A. Bilzhause, B. Belgacem, M. Mostafa, and T. Gräupl, "Datalink security in the L-band digital aeronautical communications system (LDACS) for air traffic management," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 32, no. 11, pp. 22–33, Nov. 2017.
[19] R. Koodli, "Mobile IPv6 fast handovers," Internet Eng. Task Force, RFC 4068, 2009.
[20] C. Boyd, A. Mathuria, and D. Stebila, *Protocols for Authentication and Key Establishment*, vol. 1. Berlin, Germany: Springer, 2023.
[21] *Industrial Communication Networks—Network and System Security—Part-1-1: Terminology, Concepts and Models*, IEC Standard 62443-1-1, 2009.
[22] Q. T. Ngo, K. T. Phan, A. Mahmood, and W. Xiang, "Physical layer security in IRS-assisted cache-enabled satellite communication networks," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 4, pp. 1920–1931, Dec. 2023.
[23] T. Gräupl and M. Mayr, "Method to emulate the L-band digital aeronautical communication system for SESAR evaluation and verification," in *Proc. IEEE/AIAA 34th Digit. Avionics Syst. Conf. (DASC)*, 2015, pp. 2D1–1.
[24] N. Mäurer, T. Gräupl, C. Gentsch, and C. Schmitt, "Comparing different Diffie-Hellman key exchange flavors for LDACS," in *Proc. AIAA/IEEE 39th Digit. Avionics Syst. Conf. (DASC)*, 2020, pp. 1–10.
[25] C. M. Bsi, "Cryptographic mechanisms: Recommendations and key lengths," Bundesamt für Sicherheit in der Informationstechnik, Bonn, Germany, Rep. TR-02102-1, 2020.
[26] *Safety and Performance Standard for Baseline 2 ATS Data Communications, Initial Release (Baseline 2 SPR Standard)*, Standard RTCA DO-350A Volume I and II, 2016, Accessed: Jan. 20, 2023. [Online]. Available: https://www.rtca.org/products/do-350a-volume-1-2-electronic/