End-to-End Quantum-Safe Security for Satellite Data Links (E2EQSS)

1st Christoph Wildfeuer
School of Engineering and Environment
Fachhochschule Nordwestschweiz
Windisch, Switzerland
christoph.wildfeuer@fhnw.ch

2nd Timeo Jauslin
School of Engineering and Environment
Fachhochschule Nordwestschweiz
Windisch, Switzerland
timeo.jauslin@fhnw.ch

3rd Alain Lavoyer
School of Engineering and Environment
Fachhochschule Nordwestschweiz
Windisch, Switzerland
alain.lavoyer@fhnw.ch

4th Milenko Starcik
Cybersecurity Department
VisionSpace Technologies GmbH
Darmstadt, Germany
milenko.starcik@visionspace.com

5th Afonso Serra
Cybersecurity Department
VisionSpace Technologies GmbH
Darmstadt, Germany
afonso.serra@visionspace.com

6th Laszlo Etesi
Ateleris GmbH
Brugg, Switzerland
laszlo.etesi@ateleris.ch

7th Valentina Tamburello
Ateleris GmbH
Brugg, Switzerland
valentina.tamburello@ateleris.ch

8th Bruno Huttner
ID Quantique
Geneva, Switzerland
bruno.huttner@idquantique.com

***Abstract*—As quantum computing advances, the need for quantum-safe cryptographic solutions in space systems becomes increasingly urgent. Current CCSDS/SDLS protocols rely exclusively on symmetric key cryptography, which poses challenges for key distribution, and scalability. This contribution presents an extension of the CCSDS/SDLS protocol stack with post-quantum public-key cryptography (PQC), enabling secure and scalable key exchange and authentication for satellite communications. Building on NIST Round 3 PQC candidates, we integrate algorithms such as ML-KEM (Kyber) and ML-DSA (Dilithium) into a hybrid security architecture that supports both legacy and future space missions. Our implementation is designed for compatibility with existing satellite systems and optimized for resource-constrained environments. It comprises a quantum random number generator for improved key generation. This work is part of ESA's E2EQSS [1] initiative and demonstrates a practical pathway to end-to-end quantum-safe satellite data links.**

***Index Terms*—Post-quantum cryptography (PQC), satellite communications, CCSDS, SDLS, ML-KEM/Kyber, ML-DSA/Dilithium, public key infrastructure (PKI), QRNG**

I. Introduction

In 2016, the U.S. National Institute of Standards and Technology (NIST) launched its Post-Quantum Cryptography (PQC) initiative to standardize public-key cryptographic algorithms that remain secure in the presence of quantum computing capabilities [2], [3]. NIST defined five security levels to compare the hardness of quantum-resistant asymmetric algorithms with well-known symmetric-key primitives. Table I summarizes the NIST PQC security levels.

TABLE I
NIST PQC SECURITY LEVELS AND EXAMPLE SYMMETRIC PRIMITIVES. ML-KEM-512 (SECURITY CATEGORY 1), ML-KEM-768 (SECURITY CATEGORY 3), ML-KEM-1024 (SECURITY CATEGORY 5)

Lvl	**Equivalent Symmetric Security**	**Example Algorithms**
1	At least as hard as AES-128	AES-128, SHA-256
2	Stronger than Level 1	SHA-384
3	At least as hard as AES-192	AES-192
4	Stronger than Level 3	SHA-512
5	At least as hard as AES-256	AES-256

Until now, satellite communications have relied almost exclusively on symmetric cryptographic algorithms, such as the Advanced Encryption Standard (AES) with 128-bit or 256-bit keys, to secure data links. While these remain secure even against quantum adversaries (if key lengths are chosen properly), symmetric schemes alone pose scalability and operational challenges, particularly in managing keys across large and dynamic satellite constellations.

In traditional pre-shared key systems, each pair of users must share a unique symmetric key to ensure secure communication. This approach does not scale well: for a network of n users, $\binom{n}{2} = \frac{n(n-1)}{2}$ distinct keys are required. For example, a constellation of 100 satellites would require 4,950 unique symmetric key pairs, with each satellite needing to store and manage 99 separate keys securely. By contrast, a Post-Quantum Public Key Infrastructure (PQC/PKI) requires each satellite to store only its own long-term private key together with a small set of trust anchors (e.g., CA root and intermediate certificates). Peers present X.509 certificates during the handshake; authenticity is derived from the certification chain rather than from pre-storing every peer's public key. This reduces the number of long-term keys per node to one and the total number of private keys in the network to $\mathcal{O}(n)$. Such an approach provides both scalability and flexibility, simplifying key management in large or evolving satellite networks while enabling seamless integration of post-quantum algorithms.

The End-to-End Quantum-Safe Security for Satellite Data Links project (E2EQSS) addresses this gap by proposing the integration of asymmetric, quantum-safe cryptography into the satellite security architecture. In addition, it includes a Quantum Random Number Generator (QRNG) in both segments for more secure key generation processes. This pioneering project aims to develop a quantum-safe framework that enables secure and scalable key management and data protection, from ground to space.

As an initial step in this direction, we performed a practical demonstration of a post-quantum key exchange between a ground station and the SpooQy-1 nanosatellite in low Earth orbit using Kyber-512 and the CubeSat Space Protocol (CSP) [4]. This highlighted the feasibility of integrating quantum-resistant cryptography into operational space systems for Size, Weight, and Power (SWaP)-constrained nanosatellites.

II. Software Design Overview

The software architecture of the E2EQSS system is structured into two primary domains: the *Ground PQC* segment and the *Space PQC* segment in Fig. 1. These domains are designed to operate independently while maintaining tight integration through a standardized post-quantum cryptographic protocol stack compliant with the Consultative Committee for Space Data Systems (CCSDS) standards.

The Mission Control System (MCS) and the PKI form the core components on the ground. The MCS handles mission operations and orchestrates secure communication. At the same time, the PKI subsystem issues, renews, and revokes X.509 certificates signed with the Module-Lattice-Based Digital Signature Algorithm, ML-DSA-65, a post-quantum signature scheme. Communication between the MCS and satellite systems relies on mutual certificate validation, Online Certificate Status Protocol (OCSP) [5] responses, and a post-quantum handshake based on the Module-Lattice-Based Key Encapsulation Mechanism, ML-KEM-768, key encapsulation mechanism.

In space, the Ateleris PQC Subsystem (APQS) executes the cryptographic handshake, stores session keys, and performs AES-256 symmetric encryption on telemetry and telecommand data streams. Due to limited onboard resources, computationally intensive symmetric encryption tasks are offloaded to FPGA cores where possible. The software is integrated monolithically into the flight software or as a modular component of the flight software. We have been developing an extension of the CCSDS/SDLS protocol by incorporating it into NASA's Core Flight System (cFS) using C. The cFS provides a modular and reusable flight software architecture well-suited for space applications. Leveraging its message-based middleware, we implemented the protocol extension as a set of loosely coupled flight applications. This architecture enables scalable integration, easy maintenance, and clear separation between communication logic and system-level functions, such as telemetry and telecommand handling.

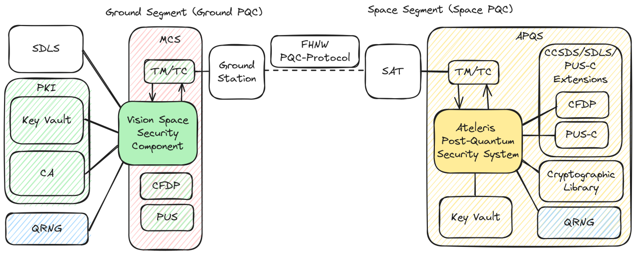
Ground Segment (Ground PQC)
Space Segment (Space PQC)
SDLS
PKI
Key Vault
CA
QRNG
MCS
TM/TC
Vision Space Security Component
CFDP
PUS
Ground Station
FHNW PQC-Protocol
SAT
APQS
TM/TC
Ateleris Post-Quantum Security System
CCSDS/SDLS/ PUS-C Extensions
CFDP
PUS-C
Cryptographic Library
Key Vault
QRNG

Fig. 1. E2EQSS System Overview. Interaction between Ground PQC (MCS, PKI) and Space PQC (APQS) components (flight segment) over a CCSDS-compatible PQC protocol stack.

Our modular architecture ensures crypto-agility, fault resilience, and compatibility with existing and future mission infrastructure.

III. Flight Segment

A. Introduction

The flight segment of the E2EQSS system handles post-quantum cryptographic functions onboard the satellite, including key exchange, symmetric encryption, and secure key management. Implemented on the On-Board Computer (OBC), these functions ensure secure links with ground control and peers. Due to limited resources and radiation exposure, the protocols are optimized for performance and reliability.

B. Space Data Link Security Protocol (SDLS)

Our approach builds on the existing CCSDS Space Data Link Security (SDLS) protocol in Fig. 2. During the handshake phase, our system operates as an application that utilizes the CCSDS File Delivery Protocol (CFDP) for communication. During the symmetric encryption, the system is located between the Space Packet Protocol (SPP) and the Telemetry and Telecommand (TM/TC) layer. Its function is to encrypt and decrypt, according to the SDLS standard, the incoming and outgoing SPP-frames with the current symmetric session key.

The SDLS standard defines mechanisms for authentication and encryption at the data link layer in space communications and is widely used across ESA and other satellite missions. We extend this well-established protocol stack by integrating post-quantum cryptographic primitives. However, SDLS is limited to symmetric key cryptography, assuming pre-shared keys between ground and flight segments. Although robust, this approach imposes severe constraints on scalability, flexibility, and post-deployment key management, especially for large satellite constellations and multiparty ground segments.

C. Handshake

At the heart of the flight segment's secure communication architecture lies the handshake protocol, designed to establish symmetric session keys using post-quantum key encapsulation. The flight segment supports ML-KEM-768, standardized by NIST (FIPS 203) [2], as the primary asymmetric primitive

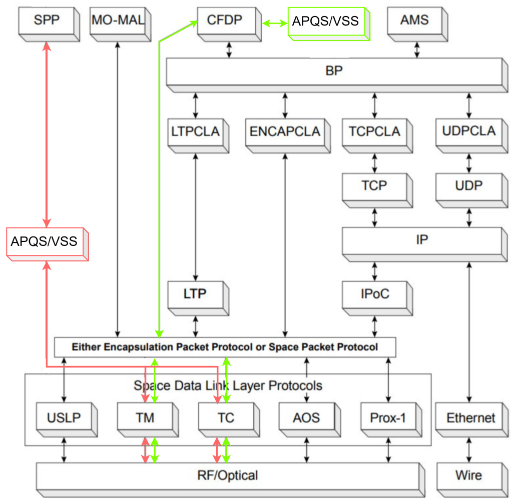

Fig. 2. Modified CCSDS Protocol Stack [6]. Interaction of the Ground PQC (VSS) and Space PQC (APQS) components within the CCSDS protocol stack. Green denotes the path during handshake, and red the symmetric bulk encryption flow.

for the secure key exchange. The satellite maintains a long-term public-private key pair, with its public key distributed in an X.509 certificate signed using ML-DSA-65 (FIPS 204) [3]. The handshake process ensures a mutually authenticated and quantum-safe establishment of encryption keys. All handshake data exchanges are performed over the CCSDS CFDP protocol to accommodate bandwidth constraints and enable file-based transmission. Resilience features include detecting mismatched key material and autonomously reinitiating the needed process. This architecture ensures that each communication session begins with a fully authenticated, quantum-safe, and verified key exchange, forming the basis for all subsequent secure telemetry, telecommand, and payload data transmission.

D. Hardware Environment

The hardware environment for the flight segment of the E2EQSS system is designed to support secure cryptographic operations under the constraints of space-based embedded systems. The primary processing unit is a CubeSat-class OBC, typically based on a Xilinx Zynq-7000 System on Chip (SoC), which integrates a dual-core ARM Cortex-A9 processor operating at 667MHz alongside an FPGA for hardware acceleration. This architecture enables computationally demanding post-quantum cryptographic operations, such as ML-KEM-768 key generation, encapsulation, and decapsulation, within strict timing constraints, targeting an end-to-end session key setup of under 0.5 seconds. Symmetric bulk encryption using AES-256 is offloaded to FPGA IP cores to reduce CPU load and increase throughput, making the system suitable for continuous secure telemetry and telecommand streams. The platform supports communication with hardware entropy sources, such as QRNGs, and secure storage elements like key vaults. Given the exposure to radiation and Single Event Effects (SEEs), the hardware must include mitigation strategies such as Error-Correcting Code (ECC) memory, watchdogs, and redundancy to ensure data integrity and fault-tolerant cryptographic execution in space environments.

E. Entropy from a QRNG

ML-KEM and ML-DSA both depend on high-quality randomness to ensure security. ML-KEM requires fresh entropy for key generation and encapsulation, using 32-byte seeds to derive secret and ephemeral values, while decapsulation is deterministic. ML-DSA needs randomness for key generation and for each signature. Weak or repeated nonces can lead to key compromise. To ensure the unpredictability needed for secure operation, a QRNG is used to supply high-entropy input. The QRNG is a NIST certified entropy source according to SP 800-90B [7]. Having a local source of entropy prevents possible corruption or disclosure, which could happen with a remote source. In contrast to other physical random number generators, based on classical systems, the unpredictability of the output is a natural consequence of quantum mechanics and provides an abundant and immediate source of randomness.

IV. Constraints

The E2EQSS software is designed under strict constraints to ensure compatibility, interoperability, and security in space environments:

- **CCSDS Compliance:** Adheres to CCSDS standards for secure data exchange and protocol interoperability.
- **System Interoperability:** Integrates with ESA's MCS, ground stations, and onboard systems, supporting legacy fallback modes.
- **PQC Standards:** Uses NIST-approved post-quantum algorithms: ML-KEM (mandatory) and HQC (optional fallback).
- **Resource Constraints:** Optimized for limited compute, memory, and power; uses FPGA for AES and efficient PQC on embedded ARM.
- **Scalability:** Supports constellations (≥ 5 satellites, star topology) with scalable key and certificate management.
- **Reliability:** Includes fault tolerance and graceful degradation to handle radiation, hardware faults, and link loss.
- **Data Integrity:** Ensures integrity of encrypted telemetry and command data, with detection and correction mechanisms.

V. Ground Segment

The ground segment in our post-quantum secure communication architecture ensures end-to-end cryptographic protection between space and ground nodes. It provides a robust and extensible quantum-safe software infrastructure focused on secure key management, telemetry, telecommand data encryption, and integration with existing mission control systems. The ground segment includes a software-based PKI offered as a service to ground and space components. The PKI comprises a CA to issue, validate, and staple the OCSP. Key generation is handled by a QRNG. Secure key storage and retrieval are managed through a software-based key vault. This solution enables the system to maintain the confidentiality and integrity of cryptographic material during key lifecycle operations. The VisionSpace Security (VSS) component handles encryption and decryption of mission data. This custom bundle integrates with the European Ground Segment Common Core (EGS-CC) [8]. VSS acts as middleware between the PKI infrastructure and the EGS-CC telemetry and telecommand processing chain, offering three primary services: encryption and authentication, as well as decryption services. During active communication sessions, session keys are temporarily retained in memory to optimize performance.

The system supports dynamic behaviors such as quantum-safe handshake initiation and certificate renewal workflows. It interfaces with key EGS-CC components, such as CFDP and the Packet Utilization Standard (PUS) Service 7 (for event reception), ensuring secure protocol-level integration for command and telemetry exchanges. To maintain resilience and service continuity, the architecture includes REST interfaces for PKI interaction and error handling mechanisms to deal with service outages or the inconsistent state of the system. The entire design of the ground segment supports the flexibility of deployment, including on-premise and cloud-based configurations. It adheres to modular and service-oriented architecture principles, ensuring scalability and maintainability.

VI. Cryptographic Protocol

A. *Introduction*

The cryptographic protocol is designed to comply with existing CCSDS standards, enabling seamless integration into current and future infrastructure. It is built with crypto-agility in mind, allowing for the update or complete replacement of all cryptographic algorithms and primitives without reengineering the protocol flow. Our approach incorporates mutual authentication using X.509 certificates issued, revoked, and managed by a PKI. Using OCSP provides the advantage of real-time certificate status verification with low bandwidth and storage requirements [9]. The protocol is divided into two phases: the first establishes a shared key between two parties, then it transitions to the more efficient symmetric encryption using quantum-safe AES-256.

B. *Shared Key Establishment*

The shared key establishment process is divided into three stages as seen in Fig. 3:

1) Certificate Request: The MCS and the satellite both request a certificate from the Certification Authority (CA). Initial authentication is achieved with a Message Authentication Code (MAC) and a single pre-shared secret key between the users and the CA.

2) Certificate Exchange: The MCS initiates the certificate exchange by transmitting a request packet that includes its identity, certificate, and corresponding OCSP response. Upon reception, the satellite verifies the certificate's signature and the validity of the OCSP response. Once verification is complete, the satellite replies with its own identity and certificate. Subsequently, the MCS validates the received certificate and queries the certification authority (CA) for the associated OCSP response.

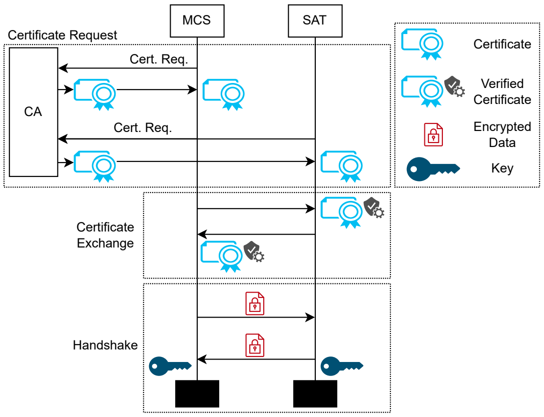

Fig. 3. Overview of the E2EQSS protocol divided into three phases

3) Handshake: The handshake is the primary process of the cryptographic protocol, as it generates a symmetric shared key that is used for subsequent symmetric bulk encryption. The handshake requires both the satellite and the MCS to have an authenticated long-term ML-KEM-768 key pair. The public keys are embedded with their identity in a certificate and signed with ML-DSA-65. A trusted CA authenticates them, and they are valid for a period of several years. After exchanging these certificates and validating correctness, the MCS initiates the handshake. Another ephemeral ML-KEM-768 key pair is created to prevent replay attacks and sent to the satellite alongside ciphertexts generated by the long-term keys.

Besides those ephemeral ML-KEM keys, there is an additional Elliptic Curve Diffie-Hellman (ECDH) key exchange included in the handshake. This enhances overall robustness and security, as ML-KEM has not yet been widely adopted in large-scale deployments or fully researched, and potential attack vectors may still exist. The two mechanisms are combined in a hybrid fashion, ensuring that the overall cryptographic security is preserved even if one of the underlying algorithms is compromised [10]. Consequently, the handshake response consists of three components: the ML-KEM ephemeral ciphertext, the ECDH ephemeral ciphertext, and a key confirmation tag. The key confirmation scheme follows NIST recommendations [11] and is implemented bilaterally to improve reliability. In this approach, both parties generate a MAC tag over defined data sets, resulting in distinct tags. Each party then verifies the received MAC using the key confirmation key derived alongside the shared secret using the key derivation function (KDF). The data used in the computation of the MAC tag is carefully selected to maximize the likelihood that both parties have derived identical cryptographic keys. To achieve this, the MAC is computed over multiple components extracted from the Input Keying Material (IKM), ensuring that any discrepancies in the key derivation process are detected early.

This approach strengthens the integrity of the key confirmation step by tightly coupling the confirmation data to the internal state of the key agreement process.

C. Symmetric Encryption

Upon completing the handshake procedure, the participants possess a shared secret of 256 bits. This shared secret serves as the master key as defined by the CCSDS Space Data Link Security Extended Procedures (SDLS-EP) [12]. Within this framework, the master key is utilized to derive session-specific encryption keys through a KDF. These session keys are subsequently used for bulk data encryption using the AES-256 cipher, as specified in the SDLS. The lifetime and rotation frequency of both the master key and the derived session keys are determined by the desired security assurance level of the system and the operational threat model. The handshake yields a per-session master secret (MS). We immediately derive SDLS traffic keys from MS via HKDF-SHA-384 and securely erase MS. We never use the same keys for multiple sessions from a single long-lived master; thus compromise of any one session key does not endanger past sessions, and compromise of long-term identity keys does not retroactively reveal session keys (PFS).

D. Formal Verification

To evaluate the resilience of the protocol against common attacks at the protocol level, such as replay and man-in-the-middle attacks, is modeled and analyzed using the Tamarin Prover [13]. The model was specified through multiset rewriting rules [14]. Formal verification primarily focuses on the handshake phase. Special attention is paid to the hybrid encryption mechanism. This formal verification should not be interpreted as proof that the protocol is unbreakable. It does not assess the strength of the underlying algorithms or whether they can be broken. Instead, it provides insight into whether the protocol leaks information or is vulnerable to known attack patterns. This work is still ongoing.

VII. Conclusions

This work introduces a quantum-safe extension to the CCSDS/SDLS protocol stack for satellite communications, integrating post-quantum cryptographic primitives ML-KEM and ML-DSA into a hybrid architecture. Ground systems use ML-DSA-signed X.509 certificates managed by a software PKI. At the same time, the flight segment performs ML-KEM-based key exchanges and AES-256 encryption, accelerated via FPGA for CubeSat constraints.

The protocol establishes symmetric session keys through hybrid ML-KEM/ECDH encapsulation and mutual authentication with a quantum-safe Hash-based Message Authentication Code (HMAC) using the SHA algorithm, specifically HMAC-SHA-384. Built on NASA's cFS, the system ensures crypto-agility and supports updates without core redesign.

The handshake protocol was formally verified with the Tamarin Prover to resist replay and man-in-the-middle attacks. This architecture, developed under ESA's E2EQSS initiative, enables practical, standards-compliant quantum-safe satellite communication.

Acknowledgment

This work was supported by ESA under the ARTES 4S (Secure Satcom for Safety and Security) program.

References

[1] "E2EQSS." [Online]. Available: https://connectivity.esa.int/projects/e2eqss

[2] National Institute of Standards and Technology (US), "Module-lattice-based key-encapsulation mechanism standard," National Institute of Standards and Technology (U.S.), Washington, D.C., Tech. Rep. NIST FIPS 203, Aug. 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf

[3] ——, "Module-lattice-based digital signature standard," National Institute of Standards and Technology (U.S.), Washington, D.C., Tech. Rep. NIST FIPS 204, Aug. 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf

[4] S. M. Burkhardt, A. Reezwana, T. Islam, W. Meier, A. Ling, and C. F. Wildfeuer, "First demonstration of a post-quantum key-exchange with a nanosatellite," Jun. 2022, arXiv:2206.00978. [Online]. Available: http://arxiv.org/abs/2206.00978

[5] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP," Internet Engineering Task Force, Request for Comments RFC 6960, Jun. 2013, num Pages: 41. [Online]. Available: https://datatracker.ietf.org/doc/rfc6960

[6] The Consultative Committee for Space Data Systems (CCSDS), "Overview of Space Communications Protocols," The Consultative Committee for Space Data Systems (CCSDS), Washington, DC, USA, Informational Report Issue 4, Apr. 2023. [Online]. Available: https://ccsds.org/Pubs/130x0g4e1.pdf

[7] I. T. L. Computer Security Division, "Cryptographic Module Validation Program | CSRC | CSRC," Oct. 2016. [Online]. Available: http://csrc.nist.gov/projects/cryptographic-module-validation-program/entropy-validations/certificate/63

[8] M. Pecchioli, A. Walsh, J. M. Carranza, R. Blommestijn, M.-C. Charmeau, M. Geyer, C. Stangl, P. Parmentier, H. Eisenman, J. Rueting, and others, "Objectives and Concepts of the European Ground Systems Common Core (EGS-CC)," *Simulation and EGSE facilities for Space Programmes*, 2012. [Online]. Available: http://www.egscc.esa.int/downloads/20120925_SESP_2012_egscc_objectives_and_concepts.pdf

[9] D. Koisser, D. Fischer, M. Wallum, and A.-R. Sadeghi, "TruSat: Building Cyber Trust in Collaborative Spacecraft Networks," in *2022 IEEE Aerospace Conference (AERO)*. Big Sky, MT, USA: IEEE, Mar. 2022, pp. 1–12. [Online]. Available: https://ieeexplore.ieee.org/document/9843330/

[10] National Institute of Standards and Technology (US), "Recommendations for Key-Encapsulation Mechanisms," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. NIST SP 800-227 ipd, 2025. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-227.ipd.pdf

[11] ——, "Recommendation for pair-wise key-establishment schemes using discrete logarithm cryptography," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. NIST SP 800-56Ar3, Apr. 2018. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf

[12] The Consultative Committee for Space Data Systems (CCSDS), "Space Data Link Security Protocol—Extended Procedures," The Consultative Committee for Space Data Systems (CCSDS), Washington, DC, USA, Recommended Standard Issue 1, Feb. 2020. [Online]. Available: https://ccsds.org/Pubs/355x1b1.pdf

[13] B. Schmidt, "Formal analysis of key exchange protocols and physical protocols," Doctoral Thesis, ETH Zurich, 2012, accepted: 2018-02-15T09:01:39Z. [Online]. Available: https://www.research-collection.ethz.ch/handle/20.500.11850/72713

[14] C. Cremers and S. Mauw, *Operational Semantics and Verification of Security Protocols*, ser. Information Security and Cryptography. Berlin, Heidelberg: Springer, 2012. [Online]. Available: https://link.springer.com/10.1007/978-3-540-78636-8