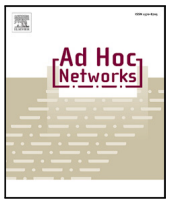
Ad Hoc Networks 174 (2025) 103817

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc



Zero trust-based authentication for Inter-Satellite Links in NextGen Low Earth Orbit networks

Kerry Anne Farrea [a,*], Zubair Baig [a], Robin Doss [a], Dongxi Liu [b]

[a] Deakin Cyber Research and Innovation Centre, Deakin University, Geelong, 3216, Victoria, Australia
[b] Data 61, CSIRO, Sydney, 2000, New South Wales, Australia

ARTICLE INFO

Keywords:
Cyber security
Zero trust
Low earth orbit satellite
Inter-satellite link
Formal method
Attack methods
Performance simulation

ABSTRACT

Next Generation (NextGen) Low Earth Orbit satellite networks are rapidly expanding to support global communication and 6G technology transition. This growth exposes networks to new security challenges due to wide coverage in hostile areas and increased access points in space and on Earth. Traditional static authentication methods prove inadequate in this dynamic environment. We address these challenges by developing a Zero Trust Authentication Protocol for Inter-Satellite Link (ISL) communication. Our protocol implements a novel verification process that leverages orbital signals to authenticate ISLs. This approach ensures secure data access and transmission exclusively among verified satellites, mitigating threats from eavesdropping, signal spoofing, impersonation, and replay attacks. To optimize security and resource efficiency, we integrate Hyperelliptic Curve Cryptography (HECC) into our protocol. We validate our approach through MATLAB and Systems Tool Kit (STK) simulations, complemented by BAN Logic and Scyther analyses. Our findings demonstrate that our protocol enhances the security framework of NextGen LEO networks without compromising their performance or operational capabilities.

1. Introduction

Since their development, satellites have provided coverage to underserved regions such as rural, remote, maritime, and aeronautical routes [1]. Initially, Geostationary Earth Orbit (GEO) satellites at 36,000 kilometers (km.) were preferred for satellite communications due to their stable position relative to Earth, offering wide coverage and consistent communication channels [2]. Recent advancements in multi-beam technology and launch cost reduction have enabled more efficient frequency reuse and increased throughput, mirroring terrestrial cellular network progress [3]. These technological improvements have led to the development of advanced Low Earth Orbit (LEO) networks, with satellites orbiting between 500 and 2000 km. Next Generation *(NextGen)* LEO networks like SpaceX's *Starlink*, Amazon's *Project Kuiper*, OneWeb's *Satellite Constellation*, and Telesat's *Lightspeed* now offer near-real-time, widespread communication [4]. Equipped with onboard processing and Inter-Satellite Links (ISL), these systems create mesh networks supporting diverse applications from large scale Internet of Things (IoT) to autonomous transportation [5]. These networks enables seamless connectivity and efficient resource allocation, with potential integration into Fifth Generation New Radio (5G-NR) and future Sixth Generation (6G) systems, significantly enhancing global telecommunications capabilities [6].

However, the dynamic nature of LEO satellites, characterized by rapid movement and frequent handovers, introduces complex challenges in data security, privacy preservation, and transmission efficiency [7]. NextGen LEO networks require resilient security methods capable of ensuring mutual authentication, confidentiality, integrity, and anonymity for data exchanged over open channels. Traditional security approaches, often relying on static keys or ground-based Network Control Centers (NCCs), have shown limitations in security properties, cyber attack resilience, and performance in the dynamic and open space environment [8]. Moreover, dependence on ground-based infrastructure for authentication introduces delays in responding to emerging threats, as satellites may miss the opportunity to independently verify the authenticity of communications in real-time. While innovations like Quantum Key Distribution (QKD) [9] and Blockchain technologies [10] promise enhanced security, they face substantial implementation and scalability challenges in space environments [7,11].

This paper introduces Zero Trust (ZT) authentication for Inter-Satellite Links (ISL) to address secure, low-latency communication challenges in Low Earth Orbit (LEO) satellite networks, which face dynamic topologies and expanding attack surfaces. Unlike traditional security frameworks that operate on implicit trust once access is granted, ZT

* Corresponding author.
E-mail address: k.farrea@deakin.edu.au (K.A. Farrea).

https://doi.org/10.1016/j.adhoc.2025.103817
Received 7 November 2024; Received in revised form 5 February 2025; Accepted 2 March 2025
Available online 26 March 2025

adheres to a 'never trust, always verify' principle [12]. This model is well-suited to the dynamic and boundary-less environment of outer space, where traditional network perimeters vanish and threats can emerge from any vector [13,14]. By implementing ZT, satellites continuously authenticate and validate every communication link, significantly enhancing the network's resilience against both ground-based and space-based threats. Furthermore, our scheme reduces dependency on ground stations, thereby minimizing signal latency. This approach not only meets the unique operational demands of LEO networks but also aligns with the requirements of emerging 6G technologies [6], enabling more secure, efficient, and reliable satellite communications.

1.1. Contributions

The contributions of this paper are summarized as follows:

1. **Authentication Proposal**: We present a Zero Trust authentication protocol designed for NextGen LEO satellite networks that leverages Inter-Satellite Links (ISL) to secure communications from unauthorized access and service disruptions.
2. **Lightweight Design and Security Validation**: We propose a lightweight and efficient design that minimizes impact on satellite resources, necessary for the resource-constrained environment of LEO satellites. The security of our protocol is validated through a two-pronged approach: heuristic security assessment and formal verification using Scyther and BAN Logic for theoretical security guarantees. This verification ensures the correctness and security of our protocol against various potential attack vectors.
3. **Performance Modeling and Evaluation**: We demonstrate the operational effectiveness and efficiency of our protocol through performance modeling using Systems Tool Kit (STK) and MATLAB. The evaluation includes authentication latency, link budget analysis and computational and communication overhead. Results are compared with existing satellite authentication schemes to demonstrate the superiority of our approach in terms of security strength, latency reduction, and resource efficiency.

The paper is structured as follows: Section 2 reviews related work; Section 3 presents the network model; Sections 4 and 5 describe the threat model and security requirements; Sections 6–7 detail the protocol development, design, and proposed scheme; Sections Section 8–11 cover security analysis using informal methods, Scyther, BAN Logic, and security comparison; Section 12 discusses performance simulations and compares existing methods; the final section concludes the paper and suggests future work.

2. Related works

This section reviews the current state of satellite authentication methods, outlining traditional approaches and the shift towards Zero Trust.

2.1. Traditional authentication methods

NextGen LEO satellites face security challenges due to their orbital position, mobility, resource constraints, open wireless channels, long signal propagation and multiple access points in space and on Earth. These factors complicate the design of efficient and secure authentication schemes for satellite networks [7]. Long-distance telemetry exposes satellite systems to interference, interception, and spoofing [15]. Traditional authentication methods [16,17] using pre-shared keys and certificates face scalability and security challenges [18]. More advanced techniques like Identity-Based Encryption (IBE) [19,20] and Physical Layer Authentication (PLA) [21,22] address some limitations but introduce new challenges. IBE can be computationally intensive for LEO networks [15], while PLA methods struggle in noisy environments.

Recent approaches to satellite network security include several innovative strategies. Lightweight protocols combine Perfect Forward Secrecy (PFS) with efficient key exchange [23,24]. Some researchers have applied deep reinforcement learning to optimize encryption [25]. Security-sensitive task offloading has also been explored [26]. While these methods offer enhanced security, they often face performance issues due to limited system options, emphasizing the need for a better balance between security and performance in satellite networks [25]. With advancements in quantum computing, researchers are developing quantum cryptographic authentication techniques such as Ring Learning with Errors [27] and lattice-based signatures [28] to protect satellite communications from quantum attacks. However, implementing and integrating quantum security in space remains challenging [11].

2.2. Zero trust authentication methods

The dynamic space environment requires security protocols that transcend traditional static frameworks. Zero Trust (ZT) addresses this need by enforcing continuous verification of every access request, regardless of inherited network trust [29]. This approach aligns with satellite networks' distributed nature and evolving security challenges. For satellite networks, ZT implements four security tenets: (1) Continuous Authentication of satellite links, (2) Multi-Factor Authentication (MFA) for layered security, (3) End-to-end encryption across all communications, (4) Network segmentation of communication channels [14]. These principles apply across all communication paths shown in Fig. 1, including ground station links, user device connections, and Inter-Satellite Links (ISL), creating a comprehensive security framework.

The Zero Trust model is gaining prominence in satellite communications [30] as traditional single security techniques like cryptographic user authentication prove insufficient for modern distributed networks [31]. MFA schemes, including biometric-based approaches [32–34], enhance security but face challenges like desynchronization and increased communication costs [35]. Blockchain-based protocols [12,36] offer secure, lightweight authentication, but struggle with scalability. The European Space Agency (ESA) has explored blockchain for decentralized authentication, while Galileo satellites employ continuous authentication to prevent unauthorized access [37,38]. Yue et al.'s [39] integration of physical layer security, blockchain and cognitive radio in LEO satellites emphasizes the need for real-time adaptation and layered security. Chistousov et al. [40] employed zero-knowledge proofs and modular codes for fast, secure satellite authentication, enhancing secrecy albeit high computational complexity. Yang et al.'s AI-based two-phase multifactor authentication for integrated space networks achieved over 92% accuracy but faces scaling challenges in resource-limited environments [41]. These diverse approaches aim to balance adaptive security with the unique constraints of satellite systems, highlighting the ongoing challenges in space-based cybersecurity.

The implementation of Zero Trust authentication in satellite systems faces challenges, particularly scalability. The distributed nature and global reach of satellite networks make centralized authentication impractical. Decentralized structures like blockchain or peer-to-peer protocols have been suggested [10,30], but they introduce complexities such as computational overhead, latency, and the need for efficient consensus mechanisms. Balancing these factors is necessary to ensure security adapts to the dynamic and resource-constrained satellite network environment [14].

3. Network model

The network model, shown in Fig. 2, consists of four main segments: Network Control Centre, User Devices, LEO Satellites, and the Communication Links. This work focuses on secure communication between ground stations and neighboring satellites via ISL.

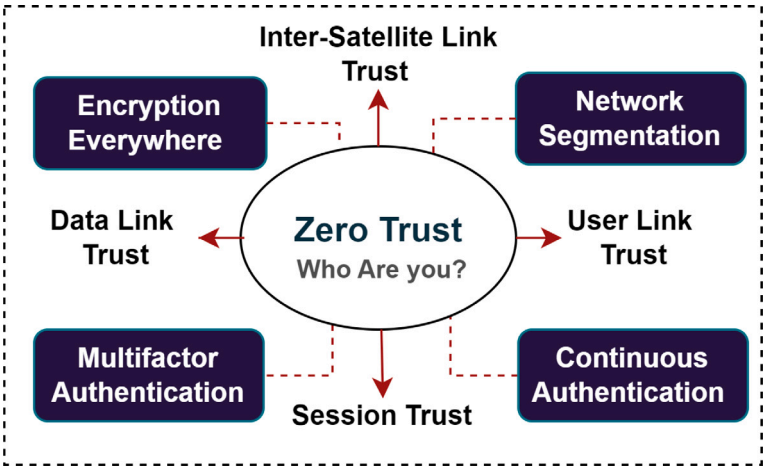


Fig. 1. Zero trust principles for satellite communication.

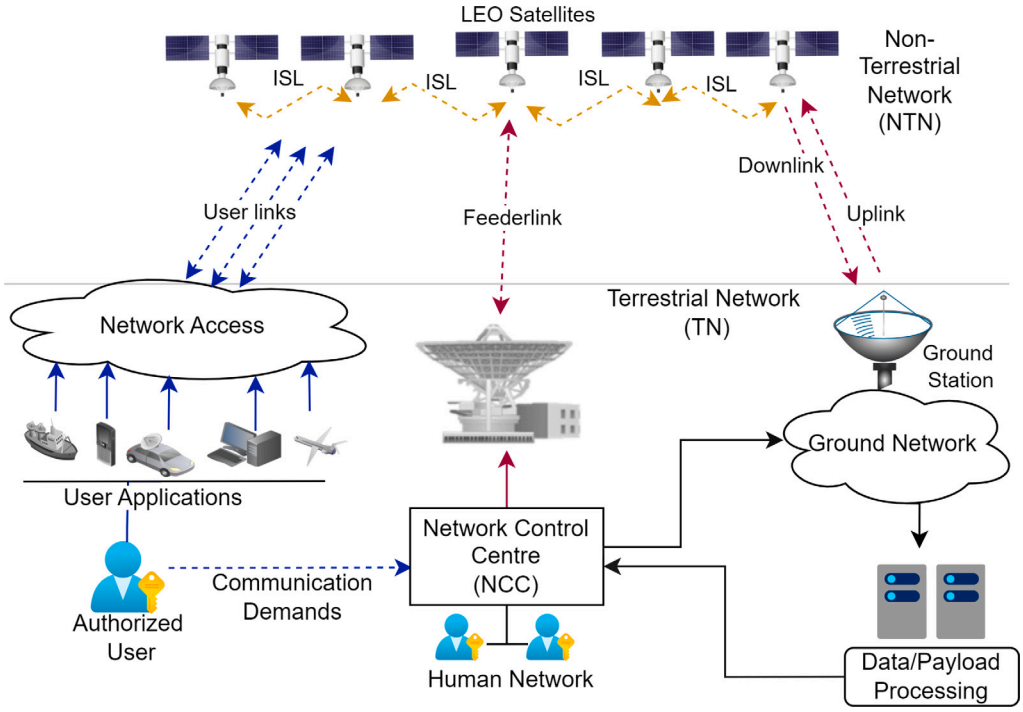


Fig. 2. Higher level network overview.

1. *Network Control Center (NCC):* The NCC encompasses all infrastructure for communication with satellites, including ground stations typically equipped with large dish antennas and communication equipment. It handles user registration, system parameters, key distribution, and initial satellite configuration.
2. *User Segment:* The user segment includes diverse end users who connect and utilize satellite services. Typical use cases are internet access, voice calls, navigation, weather monitoring, and television broadcasts, but are now expanding to include analytics in diverse vertical domains and IoT applications [5]. User equipment can include satellite receivers, mobile phones, very small aperture terminals (VSATs), and GPS receivers.
3. *Satellite Nodes:* The network comprises multiple satellites, including neighboring NextGen LEO satellites (*LEOA* and *LEOB*), using ISLs. Advanced computing capabilities enable complex operations [42]. Satellites maintain real-time orbital parameters via their bus subsystems (AD&C, TT&C, CDH, power) without significant additional resource use.
4. *Communication Links:* Three main link types exist: ISL, uplink, and downlink. ISLs enable direct communication between neighboring satellites, often using high-speed laser links for their greater bandwidth and security. The narrow beam of laser links minimizes risks of eavesdropping and jamming, as interception requires precise alignment. Uplink and downlink support satellite-to-ground/user communications but are more susceptible to threats like eavesdropping and replay attacks unless secure channels are used.

4. Threats and adversarial model

This section categorizes security threats to satellite communications based on attackers' objectives: unauthorized access and service disruption. Following RFC 4949 [43], attack methods are classified into four actionable threat events: exposure, interception, intrusion, and interference. These are further detailed in an attack tree. Additionally,

Table 1
Description of threat events.

Threat event	Description
Exposure	Unauthorized entities access sensitive data due to leaks or protocol failures.
Interception	Data in transit is compromised, altered, or redirected.
Intrusion	Attackers bypass security to access or manipulate data.
Interference	Deliberate disruptions degrade or halt system operations.

Table 2
Components involved in authentication.

Component	Description
Satellites (S1, S2..Sn)	Interconnected nodes responsible for data transmission and reception. An attack on a single satellite could compromise the entire network.
Network Control Center (NCC)	Attacks against the NCC, typically offline, can disrupt command and control operations for the entire satellite network.
Communication Channel (CC)	Includes Satellite-to-Ground (S2G), Satellite-to-User (S2U), and Satellite-to-Satellite (S2S) communications, which are vulnerable to attacks.
User Terminal/User (UT/U)	Comprises user endpoint equipment (devices, smartcards, password generators) and actions. Attacks may target equipment vulnerabilities or manipulate user behavior.

the adversarial model outlines potential attackers' capabilities and assumptions.

4.1. Threat event classification

Attackers typically aim to gain unauthorized access or control over satellite networks, compromising integrity and confidentiality. They may also target service availability through disruption. These objectives are achieved through various threat events, detailed in Table 1.

4.2. Attack tree

The attack tree provides a hierarchical visualization of potential satellite network threats. Based on [44], Fig. 3 presents our modified attack tree focused on satellite authentication vulnerabilities. The tree's root splits into two primary attack objectives: gaining unauthorized access/control and disrupting service availability. Blue shaded leaf nodes represent broader attack types applicable across all components, while green nodes categorize specific attack types. All non-leaf nodes function as OR-nodes, where achieving any child node's condition satisfies the parent condition. This structure aids in visualizing the various paths an attacker could take to achieve their objectives.

The analysis considered the four main components involved in the authentication process detailed in Table 2.

4.3. Attack methods

Exposure Attacks: These passive attacks extract data through eavesdropping, traffic analysis, and side-channel vulnerabilities. Attackers exploit weak authentication by monitoring credential entry or compromising physical tokens, potentially exposing sensitive data during transmission [45].

Interception Attacks: Active attacks that compromise communication integrity through man-in-the-middle (MiTM) manipulation, replay of captured data, session hijacking, and signal spoofing [15].

Intrusion Attacks: Direct system compromises through command injection, data modification, or malware deployment, enabling persistent unauthorized access and operational control.

Interference Attacks: Service disruption through signal jamming, Denial of Service (DoS), or Distributed Denial of Service (DDoS) attacks [22]. These attacks target system availability by overwhelming network resources or degrading communication channels.

Table 3
Adversarial model assumptions.

Assumption	Description
Passive attack	Adversaries can eavesdrop on public channels, extract secret keys, and perform cryptanalysis. It is assumed that all cryptographic keys are known to the adversary.
Active attacks	Adversaries can intercept and modify data, impersonate entities, resend or redirect signals, exhaust resources, or disrupt communication between *LEOA* and *LEOB*.
Insider or outsider	The adversary can be either a valid user (insider) or an outsider to the system.
Protocol awareness	Adversaries can guess low-entropy secrets or identities individually but cannot deduce both secret parameters simultaneously within polynomial time.

Table 4
Core security properties for satellite communication.

Property	Description
Mutual authentication	Verifies that both satellites authenticate each other, ensuring authorized connections.
Key confidentiality	Protects cryptographic keys from unauthorized access.
Unforgeability	Prevents message and credential forgery.
Satellite anonymity	Maintains the confidentiality of satellite identities.
Lightweight design	Ensures efficient use of computational, energy, and bandwidth resources.
Withstands exposure	Safeguards against eavesdropping by maintaining communication confidentiality.
Interception resilience	Prevents signal interception and attacks like MiTM and replay, ensuring signals are secure.
Intrusion resilience	Protects against unauthorized access and data modification, including spoofing and impersonation.
Withstands interference	Defends against disruptions like DoS attacks that aim to degrade performance.

4.4. Adversarial model

In this work, we define the threat as a malicious attack on the satellite communication network. Communication between *LEOA* and *LEOB* occurs over an open wireless channel, making it susceptible to various threats and attack methods as shown in Fig. 3. Our threat model follows the Canetti–Krawczyk (CK) model [46]. The attacker's capabilities are outlined in Table 3.

5. Security requirements

To ensure the integrity, confidentiality, and availability of satellite communication, we outline the following security properties in Table 4:

Additionally, our protocol embraces the fundamental tenets of Zero Trust, which include the principles outlined in Table 5.

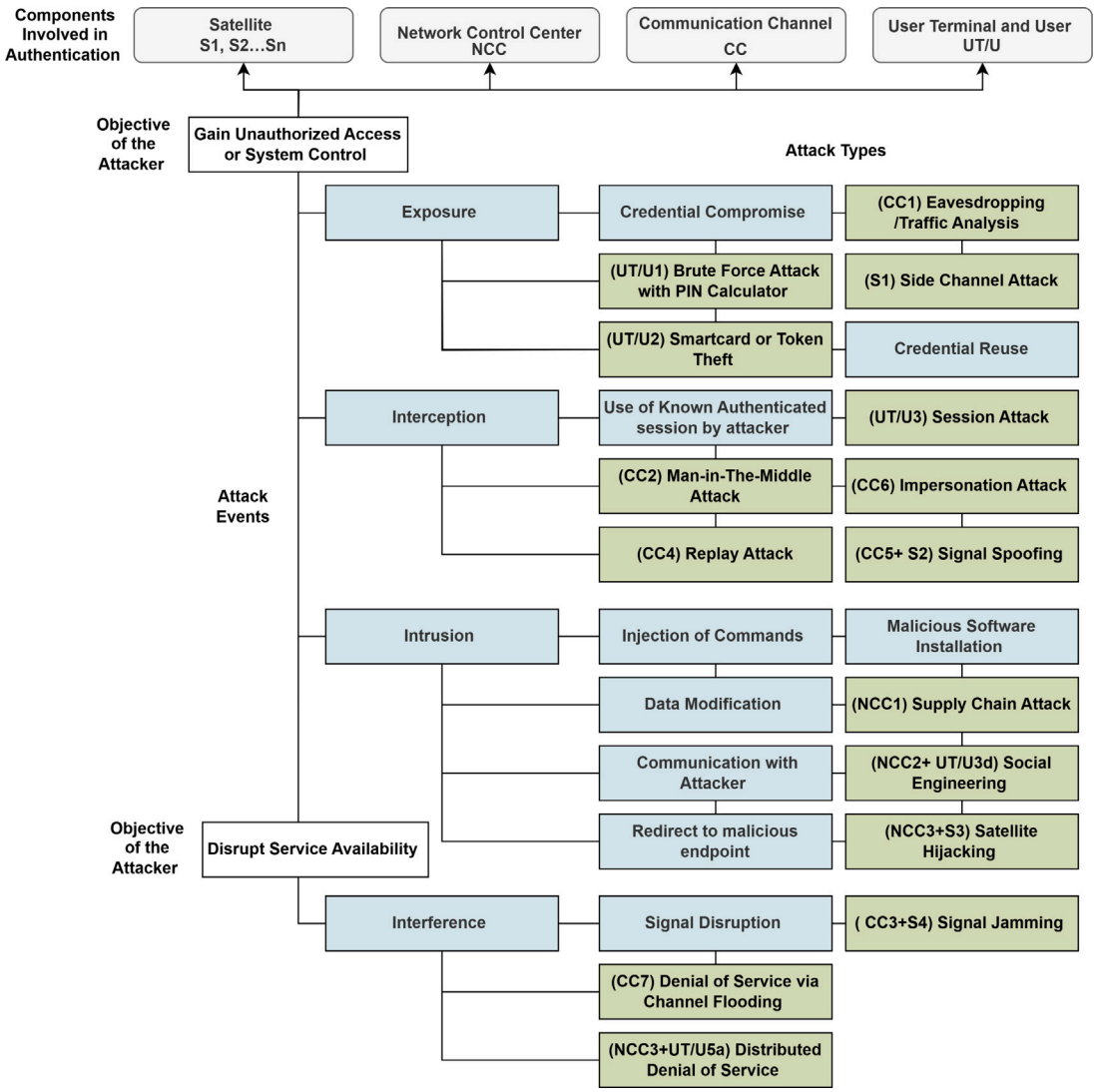


Fig. 3. Attack tree: Blue boxes represent generic attack types applicable across all components, while green boxes indicate specific attack types that may target one or more components, such as Satellites, NCC, Communication Channels, or User Terminals.

Table 5
Zero trust principles in satellite communication.

Zero trust principle	Description
Continuous authentication	Verifies satellite credentials continuously during communication sessions.
Multi-factor authentication	Combines orbit authentication with cryptographic keys to enhance verification processes.
Encryption everywhere	Applies encryption to all data transmissions to protect against network attacks.
Network segmentation	Divides the network into secure segments, limiting lateral movement of threats and isolating critical components.

6. Design concepts

6.1. Dynamic source authentication using orbital vectors

Our protocol leverages satellites' physical state for authentication by integrating orbital parameters with cryptographic verification. Each satellite validates incoming connections by comparing received orbital vectors (position, velocity, and time) against independently calculated trajectories. These dynamically changing parameters—including azimuth, inclination, and eccentricity—serve as physical trust anchors that complement traditional cryptographic methods. By utilizing orbital dynamics, the system naturally resists replay attacks while maintaining minimal computational overhead due to existing trajectory calculations. This multifactor approach aligns with Zero Trust principles, authenticating every interaction without burdening resources or relying heavily on cryptographic operations or external entities.

6.2. Hyperelliptic curve cryptography (HECC)

Hyperelliptic Curve Cryptography (HECC) offers significant advantages over Elliptic Curve Cryptography (ECC) in resource-constrained environments [47–49]. Operating over Galois Field $GF(2^{83})$ with genus 2 curves, HECC achieves equivalent security to ECC implementations that require larger $GF(2^{167})$ operations with genus 1 curves [50]. This reduction in field size and operand length results in more efficient encryption and decryption processes. While ECC is more widely deployed, recent advancements in HECC have enhanced its practicality through specialized curve designs and optimization techniques [51]. Implementation efficiency has been significantly improved through fragmentation algorithms and Complex Multiplication (CM) methods for secure curve selection. Hardware architectures have demonstrated that HECC's

Table 6
Phases of protocol and zero trust principles.

Phase	Description
Phase 1	Establishes mutually authenticated channels between LEOA, LEOB, and NCC, distributing cryptographic parameters (shared keys, dynamic identities). Implements MFA and network segmentation for secure communication.
Phase 2	Ensures secure exchange of dynamic identities and cryptographic data, with minimized trust radius and encryption of all data.
Phase 3	Satellites authenticate using pre-established keys and dynamic orbital parameters, employing MFA, encryption, and continuous authentication.
Phase 4	Ongoing verification of orbital parameters and performs periodic re-authentication with HECC, updating keys and encrypting all communications for ongoing security.

group operations can be optimized through parallel processing and efficient scheduling of finite field arithmetic [48].

The lightweight nature of HECC makes it suitable for FPGA implementation in satellite systems. Modern satellites, such as ESA's OPS-SAT, demonstrate the viability of FPGA-based cryptographic processing through their use of the Xilinx Spartan-6 [52]. HECC's efficient resource utilization aligns well with such space-qualified hardware platforms [53], as documented in comprehensive implementations of both HECC [48] and comparative ECC systems [49].

Consider a finite field denoted as $\mathbb{F}_q$ with an order of q. When the genus g exceeds 1, the equation defining the hyperelliptic curve (HEC) takes the form [54]:

$$\mathcal{HEC} : y^2 + h(x)y = f(x) \tag{1}$$

In this equation, $(x, y) \in \mathbb{F}_q$, $h(x)$ is a polynomial of degree at most g, and $f(x)$ is a monic polynomial of degree $2g + 1$ [54].

Complexity premise

Divisor

The key distinction between hyperelliptic curves and elliptic curves lies in the characterization of HEC points as divisors, which are formal sums of points on the curve [55]. These divisors collectively form a group known as the Jacobian Group [54]:

$$\mathfrak{D} = \sum_i n_i P_i \quad \text{where } n_i \in \mathbb{Z} \text{ and } P_i \in \mathcal{HEC} \tag{2}$$

Hyperelliptic curve discrete logarithm problem (HECDLP)

The discrete logarithm problem forms the basis for many cryptographic systems built upon trapdoor functions. A trapdoor function is a mathematical operation that is efficiently computable in one direction but computationally difficult to reverse without specific additional information [56]. The security of HEC cryptosystems relies entirely on the difficulty of solving the HECDLP. Given a divisor $\mathfrak{D}$ and its scalar multiple $X' = n\mathfrak{D}$, where n is an integer, the HECDLP involves determining the value of n [54,56]. This can be expressed as:

$$X' = n \cdot \mathfrak{D} \tag{3}$$

6.3. Phases aligned with zero trust

Our protocol consist of four phases: (1) Initial Configuration, (2) Pre-Negotiation, (3) Mutual Authentication, and (4) Continuous Authentication with Key Update. Table 6 and Fig. 4 summarizes how each phase aligns with ZT principles of continuous verification, MFA, encryption, and network segmentation.

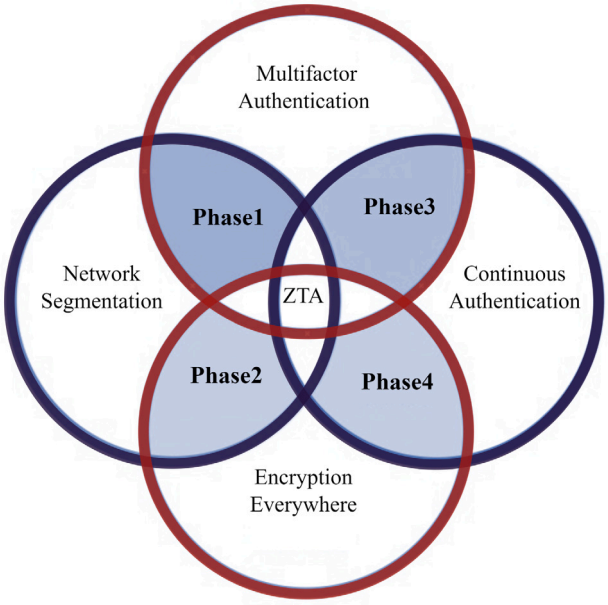


Fig. 4. Our protocol aligned with ZT approach.

Table 7
Notations in the proposed scheme.

Symbol	Description
ID_X	True identity of the satellite
$dTID_X$	Dynamic Temporary Identity of the satellite
OP_X	Orbital Parameters of the satellite
XYZ_X	Coordinates of the satellite
PHK_X	Physical key of the satellite
MAC_X	Message Authentication Code of the satellite
KC_X	Key Confirmation of the satellite
AV_X	Authentication Vector of the satellite
CAT_X	Continuous Authentication Token of the satellite
CAV_X	Continuous Authentication Vector
TS_1, TS_2, TS_3	Timestamps
SK, SK'	Shared key and updated shared key
IK, CK	Integrity key and confidentiality key
EnK, $DecK$	Encryption and decryption operations
N_X	Nonce of the satellite
R_1, R_2	Encrypted messages sent to satellites
M_p, M_k	Public key and private key
$\mathcal{H}$	Hyperelliptic curve
RNG	Random number generator
h	Hash operation
Y_4	Point on $\mathcal{H}$ used as a divisor

Note: X represents either LEOA or LEOB throughout the protocol.

6.4. Notations

The notations used in this protocol are summarized in Table 7.

7. Proposed protocol

Phase 1: Initial configuration

In the initial phase, NCC and the satellites set up secure communication parameters through the following steps:

1. *Secure Channel Establishment*: The *NCC, LEOA*, and *LEOB* establish secure channels using TLS, which supports HECC.
2. *Parameter Selection and Transmission*: The NCC selects a genus 2 hyperelliptic curve ($\mathcal{H}$EC) over $\mathbb{F}_z$ ($z \geq 2^{280}$), a hash function h from $\mathcal{H}$EC, and generates a private key M_k. The public key M_p is computed as $M_p = M_k \cdot Y$ (Y being the base point on $\mathcal{H}$EC).

The NCC also defines the true IDs (ID_A and ID_B) for LEOA and LEOB, as points on $\mathcal{H}$EC, and generates long-term shared keys K_A and K_B using HECC, along with a long-term common secret S.

3. *Temporary Identity Generation*:
   (a) The NCC generates a temporary identity $dTID_n$ using a secure RNG and computes $A_{NCC} = M_p + dTID_n$.
   (b) The NCC sends this encrypted message M_0 to both satellites:

$$M_0 = E_{TLS}(A_{NCC}, \mathcal{H}\text{EC}, h, ID_A, K_A, S, M_p, dTID_n, TS_{NCC})$$

4. *Satellite Authentication and Response*: Upon receiving M_0, each satellite verifies the authenticity of the NCC by checking A_{NCC}.
   (a) If successful, each satellite generates its own temporary identity:

$$dTID_X = RNG_X(K_X \parallel TS_{NCC} \parallel ID_X)$$

   (b) The satellite then computes:

$$A_X = ID_X + dTID_X$$

   (c) Each satellite sends the following message M_1 back to the NCC:

$$M_1 = E_{K_X}(ID_X \parallel dTID_X \parallel A_X \parallel TS_X)$$

5. *NCC Confirmation*:
   (a) The NCC verifies the information from each satellite by ensuring that the timestamps are fresh and that $A_X = ID_X + dTID_X$ is valid.
   (b) If the verification passes, the NCC computes a confirmation value:

$$C_X = h(K_X \parallel dTID_X \parallel TS_{NCC} \parallel \text{“Confirmed”})$$

   (c) The NCC then sends the following confirmation message M_2 to the satellite:

$$M_2 = E_{K_X}(C_X \parallel TS_{NCC\_new})$$

6. *Final Satellite Verification*: Each satellite decrypts and verifies the confirmation value C_X. If C_X is correct, mutual authentication is established between the NCC and the satellites.

Phase 2: Pre-negotiation

After initial configuration phase, the satellites need to share their temporary identities and orbit parameters for secure LEO-to-LEO satellite communication. This phase aims to reduce the authentication delay and computing overhead of the access authentication process for satellite handover, ensuring real-time requirements for latent sensitive applications.

1. LEOB sends a handover pre-negotiation request when it senses LEOA in range. This request is influenced by routing protocols that determine the optimal timing based on network conditions, such as changes in routing paths or load balancing needs. For example, a request may be triggered when a shift in the optimal communication path is predicted due to LEO movement for handover. The request message (M_1) includes the dynamic temporary identity of LEOB (dT IDB) timestamp encrypted with the shared key.:

$$M_1 = E_{SK}(dTID_B \parallel h(TS_2))$$

2. LEOA verifies the timestamp and decrypts the message M_1. Upon successful verification, LEOA sends a message (M_2) to NCC:

$$M_2 = E_{K_A}(dTID_B \parallel dTID_A \parallel TS_1)$$

3. *NCC verifies and generates parameters*:
   (a) NCC verifies the timestamp TS_1 in M_2.
   (b) NCC generates the current timestamp TC^{NCC}.
   (c) NCC generates a random number from the hyperelliptic curve $\mathcal{H}$EC.
   (d) NCC retrieves the orbit parameters.

4. The NCC sends encrypted messages (M_3) to both satellites:

$$M_3 = E_{K_A}\{(OP_A, OP_B, T^{NCC}, dTID_A) \parallel E_{K_B}(OP_B, OP_A, dTID_B)\}$$

5. The satellites decrypt the messages, verify the timestamps T^{NCC}, and store the relevant information.

Phase 3: Mutual authentication

In the mutual authentication phase, *LEOA* and *LEOB* negotiate a pair of session keys for a secure channel based on the shared value and the physical state of their orbit shown in Fig. 5. In scenarios where the NCC is not visible to both LEOA and LEOB simultaneously due to satellite movement, the protocol supports decentralized authentication between LEOA and LEOB. Using previously exchanged keys and dynamic time-based parameters, LEOA and LEOB can maintain secure communication directly during periods of intermittent visibility with the NCC.

LEOA authentication request for LEOB

1. *LEOA* calculates its current position at time TS_1, given its orbital parameters OP_A. This calculated position is then used in the generation of a physical state key, which in turn is used in the derivation of cryptographic keys used in the secure communication:

$$XYZ_A = f(OP_A, TS_1)$$

2. *LEOA* generates the physical state key based on the physical characteristics of its orbit and a point y over Hyperelliptic Curve Cryptography (HECC) called a divisor:

$$PHK_A = h(XYZ_A) \oplus y$$

3. *LEOA* calculates the Confidentiality Key (CK) and Integrity Key (IK) using the shared value between the satellites, PHK_A, and a randomly generated number selected from $\mathcal{H}$EC:

$$CK_A = PHK_A \oplus SK, \quad IK_A = h(PHK_A \parallel SK)$$

4. *LEOA* generates the Message Authentication Code (MAC) using the Integrity Key, Physical Key, dynamic temporary ID, timestamp, and a nonce:

$$MAC_A = h(IK_A \parallel PHK_A \parallel dTID_A \parallel TS_1 \parallel Nonce_A)$$

5. *LEOA* generates the Key Confirmation by combining the Physical Key, Confidentiality Key, Integrity Key, and the timestamp:

$$KC_A = h(PHK_A \parallel CK_A \parallel IK_A \parallel TS_1)$$

6. *LEOA* creates the Authentication Vector, which encapsulates the MAC, dynamic temporary ID, timestamp, nonce, and Key Confirmation for transmission:

$$AV_A = (MAC_A \parallel dTID_A \parallel TS_1 \parallel Nonce_A \parallel KC_A)$$

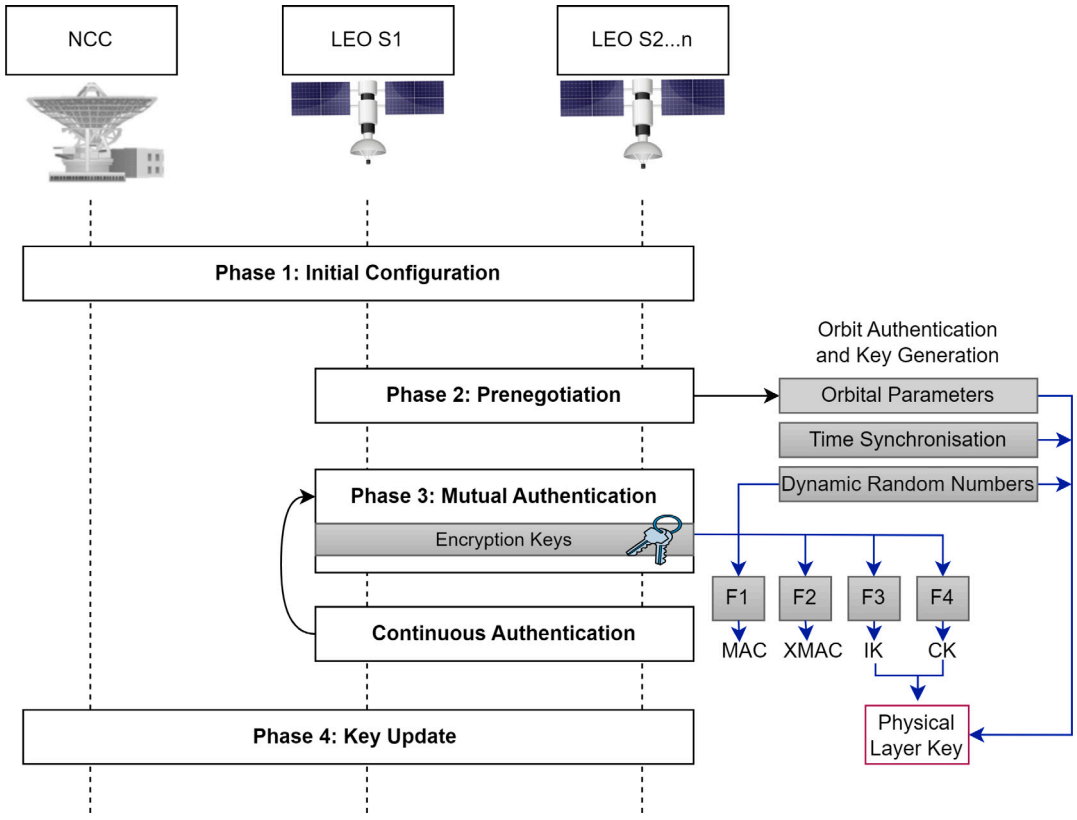


Fig. 5. Overview of authentication protocol.

LEOB authenticating LEOA

1. *LEOB* verifies the freshness of the timestamp by checking $|TS_1 - \text{current\_time}| < \text{threshold}$. It also ensures that $dTID_A$ matches the one provided by NCC. If both checks pass, the connection continues; otherwise, it is terminated.
2. *LEOB* calculates the current coordinates of *LEOA* using its orbital parameters at TS_1:

   $$XYZ'_A = f(OP_A, TS_1)$$

   Then, using these coordinates, *LEOB* generates the Physical Key PHK'_A:

   $$PHK'_A = h(XYZ'_A) \oplus y$$

   Finally, *LEOB* derives the Confidentiality Key CK'_A and Integrity Key IK'_A by combining PHK'_A with the shared secret SK:

   $$CK'_A = PHK'_A \oplus SK$$

   $$IK'_A = h(PHK'_A \parallel SK)$$
3. *LEOB* computes the expected Message Authentication Code (MAC):

   $$XMAC_A = h(IK'_A \parallel PHK'_A \parallel dTID_A \parallel TS_1 \parallel Nonce_A)$$

   LEOB then verifies that $MAC_A == XMAC_A$.
4. *LEOB* verifies the Key Confirmation:

   $$XKC_A = h(PHK'_A \parallel CK'_A \parallel IK'_A \parallel TS_1)$$

   LEOB then verifies that $KC_A == XKC_A$.
5. If all verifications succeed, *LEOB* authenticates *LEOA* and the process proceeds; otherwise, the connection is aborted.

The mutual authentication between *LEOA* and *LEOB* is symmetrical. After *LEOA* initiates, *LEOB* mirrors the steps, generating and verifying Authentication Vectors to securely establish a connection. The steps are summarized in Table 8 due to page constraints.

Continuous authentication

1. *Continuous Monitoring*: LEOA and LEOB continuously monitor changes in their operational environment, such as physical location, internal system status, and external security threats. This monitoring is facilitated by sensors and system health checks that report back to the satellite's main control system.
2. *Dynamic Identity Verification*: At regular intervals, or upon detecting significant changes, LEOA and LEOB generate new dynamic temporary identities ($dTID'_A$ and $dTID'_B$) using an HECC-based random number generator. These new dTIDs are hashed and combined with a timestamp to create a Continuous Authentication Token (CAT):

   $$CAT_A = h(dTID'_A \parallel TS_3 \parallel N_A), \quad CAT_B = h(dTID'_B \parallel TS_3 \parallel N_B)$$

   Note: If a satellite fails to receive an acknowledgment for its dTID, it can initiate a retransmission to ensure that the identity update is successfully processed by the receiving satellite. If repeated dTID conflicts or synchronization issues occur, the protocol can switch to alternative authentication paths, such as leveraging the NCC as a trusted intermediary for identity verification and synchronization.
3. *Authentication Token Exchange*: LEOA encrypts CAT_A along with its new dynamic temporary identity and nonce, then sends it to LEOB. LEOB does the same with CAT_B and its new $dTID'_B$. The exchange uses the previously established shared secret key (SK') for encryption to ensure confidentiality.
4. *Token Verification and Identity Update*: Upon receiving the new CAT and dTID, each satellite verifies the authenticity of the received token by recalculating the hash using the received dTID

Table 8
Phase 3: Mutual authentication between LEOA and LEOB.

LEOA	LEOB
Authentication request	
$\longrightarrow$ **M1: Sends AV_A to LEOB** $XYZ_A = f(OP_A, TS_1)$ $PHK_A = h(XYZ_A) \oplus y$ $CK_A = PHK_A \oplus SK$ $IK_A = h(PHK_A \| SK)$ $MAC_A = h(IK_A \| PHK_A \| dTID_A \| TS_1 \| N_A)$ $KC_A = h(PHK_A \| CK_A \| IK_A \| TS_1)$ $AV_A = (MAC_A \| dTID_A \| TS_1 \| N_A \| KC_A)$	**M1: Receives AV_A from LEOA**
Authentication	
	$XYZ'_A = f(OP_A, TS_1)$ $PHK'_A = h(XYZ'_A) \oplus y$ $CK'_A = PHK'_A \oplus SK$ $IK'_A = h(PHK'_A \| SK)$ $XMAC_A = h(IK'_A \| PHK'_A \| dTID_A \| TS_1 \| N_A)$ $XKC_A = h(PHK'_A \| CK'_A \| IK'_A \| TS_1)$ **M2: Authenticates LEOA if all verifications succeed**
Authentication request	
M3: Receives AV_B from LEOB	$\longleftarrow$ **M3: Sends AV_B to LEOA** $XYZ_B = f(OP_B, TS_2)$ $PHK_B = h(XYZ_B) \oplus y$ $CK_B = PHK_B \oplus SK$ $IK_B = h(PHK_B \| SK)$ $MAC_B = h(IK_B \| PHK_B \| dTID_B \| TS_2 \| N_B)$ $KC_B = h(PHK_B \| CK_B \| IK_B \| TS_2)$ $AV_B = (MAC_B \| dTID_B \| TS_2 \| N_B \| KC_B)$
Mutual authentication	
$XYZ'_B = f(OP_B, TS_2)$ $PHK'_B = h(XYZ'_B) \oplus y$ $CK'_B = PHK'_B \oplus SK$ $IK'_B = h(PHK'_B \| SK)$ $XMAC_B = h(IK'_B \| PHK'_B \| dTID_B \| TS_2 \| N_B)$ $XKC_B = h(PHK'_B \| CK'_B \| IK'_B \| TS_2)$ **M4: Authenticates LEOB if all verifications succeed**	

Table 9
Continuous authentication phase.

LEOA	LEOB
Continuous monitoring and identity verification	
Continuously monitors environment, generates new $dTID'_A$ Computes $CAT_A = h(dTID'_A \| TS_3 \| N_A)$ $\longrightarrow$ Sends $M_1 : CAT_A$ and $dTID'_A$ to LEOB	Continuously monitors environment, generates new $dTID'_B$ Computes $CAT_B = h(dTID'_B \| TS_3 \| N_B)$ $\longleftarrow$ Receives $M_1 : CAT_A$ and $dTID'_A$ from LEOA
Token verification and acknowledgment	
Verifies CAT_B, updates $dTID'_B$ if valid $\longrightarrow$ Sends $M_2 : CAT_B$ and $dTID'_B$ to LEOA Sends acknowledgment encrypted with SK'	Verifies CAT_A, updates $dTID'_A$ if valid $\longleftarrow$ Receives $M_2 : CAT_B$ and $dTID'_B$ from LEOB Sends acknowledgment encrypted with SK'

and comparing it against the received CAT. If the verification is successful, each satellite updates its record of the other's dynamic temporary identity, ensuring that any further communication uses the latest and most authentic identity markers.

5. *Continuous Authentication Acknowledgment*: Following successful verification and any necessary updates, LEOA and LEOB send acknowledgments to each other, confirming the successful continuation of their authenticated session. These acknowledgments are encrypted with the updated shared secret key (SK') to maintain confidentiality (see Table 9).

Phase 4: Key update phase

The satellites and the NCC securely update their encryption keys to maintain the confidentiality and integrity of their ongoing communication, ensuring that the security of the communication channel remains robust over time. The NCC determines the update frequency based on satellite velocity, orbital changes, and communication latency. It uses shorter cycles during rapid changes to enhance security with frequent key refreshes, while longer cycles are applied during stable periods to balance security with resource efficiency. Shorter cycles reduce vulnerability but increase resource use, whereas longer cycles conserve resources but slightly widen the window for potential attacks.

1. The NCC monitors a specific time threshold that indicates the need to update the confidentiality and integrity keys. When the threshold is reached, the NCC generates a new shared secret key Sk' and an update key request Uk:

   $$Uk = h(T'_{NCC} \parallel Mp)$$

   The NCC then sends encrypted messages $R1$ to LEOA and $R2$ to LEOB:

   $$R1 = Encrypt_{K_A}(Uk \parallel Sk'), \quad R2 = Encrypt_{K_B}(Uk \parallel Sk')$$

2. Upon receiving the message, both satellites verify that the timestamp T'_{NCC} falls within the allowed threshold. If verified, they decrypt the message to retrieve Sk' and Uk.

3. LEOA then derives new Confidentiality and Integrity keys using Sk' and its physical key:

   $$CK \parallel IK = RNG1(Sk' \oplus PHK_A)$$

4. LEOA generates an authentication token At based on a hash of $dTID_A$ and a divisor point denoted as Y_4 from the HECC:

   $$At = h(dTID_A) \cdot Y_4$$

5. LEOA sends an update key message, including CK, IK, At, and T_{S2}, encrypted with Sk', to LEOB:

   $$Initiate\,Uk = Encrypt_{Sk'}(CK \parallel IK \parallel At \parallel T_{S2})$$

6. Upon receiving the message, LEOB verifies the freshness of T_{S2} and checks if $h(dTID_A) \cdot Y_4 = At$. If valid, LEOB decrypts the message and derives its new keys:

   $$CK' \parallel IK' = (PHK'_A \parallel Sk' \oplus RNG1)$$

7. LEOB then generates an encrypted confirmation response $Ursp$ and sends it to LEOA:

   $$Ursp = Encrypt_{Sk'}(CK' \parallel IK' \parallel T_{S2})$$

8. LEOA verifies the integrity of T_{S2} upon receiving the response. If valid, LEOA decrypts $Ursp$ and updates its keys to CK' and IK'.

This key update phase ensures that LEOA and LEOB regularly renew their confidentiality and integrity keys while maintaining the security of their long-term keys. It guarantees that communication remains secure, even after prolonged session key usage.

8. Heuristic security analysis

Satellite-to-satellite mutual authentication

Mutual authentication between satellites is achieved by exchanging Authentication Vectors (AVs), where $AV_A = (\text{MAC}_A \parallel dTID_A \parallel TS_1 \parallel \text{Nonce}_A)$. Each satellite verifies the other's AV by computing a Physical Key (PHK) from orbital data, $\text{PHK}'_X = h(f(\text{OP}_X, \text{TS}_X)) \oplus y$, deriving session keys CK'_X and IK'_X, ensuring authenticity of all parties.

NCC-satellite mutual authentication

NCC-Satellite authentication follows a similar process, with satellites sending $AV_A = (MAC_A \parallel dTID_A \parallel TS_1 \parallel \text{Nonce}_A)$, where $MAC_A = h(IK_A \parallel PHK_A)$. The NCC confirms the satellite's identity using an encrypted message, ensuring mutual authentication between the satellite and NCC.

Confidential communication

Confidentiality is ensured using Hyper-Elliptic Curve Cryptography (HECC) over $\mathbb{F}_z$, where $z \geq 2^{280}$, protecting long-term shared keys for NCC-Satellite communication and session keys for satellite-to-satellite communication. Session keys are derived as $CK_A = PHK_A \oplus SK$ and $IK_A = h(PHK_A \parallel SK)$, ensuring robust confidentiality.

Integrity

Message integrity is maintained through MACs generated from integrity keys (IKs), with $MAC_A = h(IK_A \parallel PHK_A \parallel dTID_A \parallel TS_1 \parallel \text{Nonce}_A)$. Binding the MAC to PHK, timestamps, and nonces provides strong protection against tampering.

Forward secrecy

Partial forward secrecy is ensured by deriving session keys from dynamic PHKs based on satellite positions. Long-term keys K_A and K_B remain static, while time-variant orbital parameters secure past session keys against compromise.

Anonymity

Anonymity is maintained by periodically refreshing dynamic temporary identities (dTID), generated as $dTID_X = RNG_X(K_X \parallel TS_{NCC} \parallel ID_X)$, ensuring frequent updates and unpredictability.

Replay attack resistance

Replay attacks are mitigated by using fresh timestamps and nonces in authentication vectors. The condition $|TS_1 - \text{current\_time}| <$ threshold ensures replayed messages are detected and rejected, making successful replay attacks infeasible.

Eavesdropping resistance

An eavesdropper must intercept encrypted communications and decrypt them to access content. The protocol prevents this by encrypting NCC-Satellite communications as $E_{K_X}(ID_X \parallel dTID_X \parallel A_X \parallel TS_X)$, with dynamic session keys for inter-satellite communication. Frequent PHK updates ensure forward secrecy, complicating successful eavesdropping without solving HECDHP and predicting satellite positions.

Spoofing resistance

A spoofing attacker would need to forge valid dynamic temporary IDs, authentication vectors, and pass NCC verification. The protocol resists this through dynamic temporary IDs $dTID_X = RNG_X(K_X \parallel TS_{NCC} \parallel ID_X)$ and authentication vectors $AV_A = (MAC_A \parallel dTID_A \parallel TS_1 \parallel \text{Nonce}_A)$, along with NCC verification of $A_X = ID_X + dTID_X$. Forging these elements is computationally infeasible without knowledge of multiple secret keys and current orbital parameters.

Impersonation resistance

Impersonation requires forging valid Authentication Vectors (AV). The protocol prevents this using PHKs derived from orbital vectors, $PHK_A = h(XYZ_A) \oplus y$ where $XYZ_A = f(OP_A, TS_1)$. Multi-layer key derivation, such as $CK_A = PHK_A \oplus SK$ and $IK_A = h(PHK_A \parallel SK)$, makes impersonation unlikely without compromising multiple system components and knowing precise orbital parameters.

Resistance to satellite hijacking

An attacker hijacking LEOA must generate valid dynamic temporary IDs $dTID_X = RNG_X(K_X, \|, TS_{NCC}, \|, ID_X)$, timestamps, and physical state keys based on real-time orbital parameters to maintain authentication with LEOB. The protocol's time-sensitive updates and continuous verification make this impractical, as each session has a strict validity period, requiring ongoing recalculations.

Denial of service (DoS) resistance

A DoS attacker would need to flood the system with numerous authentication requests. The protocol mitigates this through the computational cost of generating valid AVs, requiring HECC operations and hash functions. Timestamp verification $|TS_1 - \text{current\_time}| <$ threshold allows quick discarding of invalid requests, significantly reducing DoS impacts.

Table 10
Protocol variables and cryptographic primitives.

Component	Description
Variables	TimeStamp, h, ka, kb, y, sk, seed, OPA, OPB, IDa, IDb, Ts, Tsa, Tsb, TIDA, TIDB, dTIDa, dTIDb
Primitives	h, hec, F, Mult, Concat, XOR, RNG1, RNG2
Macros	XYZA = Concat(OPa, Tsa), Z = h(XYZA), PHkA = Mult(Z, y), val = XOR(sk, PHkA)

Table 11
Protocol messages.

Step	Sender	Receiver	Message
1	NCC	LEOA	$\{C\}_{k(\text{NCC,LEOA})}$
2	NCC	LEOB	$\{D\}_{k(\text{NCC,LEOB})}$
3	LEOA	NCC	$\{\text{TIDA, TIDB, Ts}\}_{ka(\text{LEOA,NCC})}$
4	NCC	LEOA	$\{\text{OPA, OPB, Ts, TIDA}\}_{ka(\text{NCC,LEOA})}$
5	NCC	LEOB	$\{\text{OPB, OPA, TIDB}\}_{kb(\text{NCC,LEOB})}$
6	LEOA	LEOB	$\{\text{AVA}\}_{sk(\text{LEOA,LEOB})}$
7	LEOB	LEOA	$\{\text{AVB}\}_{sk(\text{LEOB,LEOA})}$

Man-in-the-middle (MitM) resistance

A MitM attacker would intercept and modify communications without detection. The protocol thwarts this using shared secrets (SK, K_A, K_B) combined with PHKs derived from orbital positions. Key confirmation adds a verification layer, requiring knowledge of all component keys, making successful MitM attacks highly impractical.

Quantum attack difficulty

The dynamic use of orbital vectors in generating Physical Keys complicates quantum attacks, as the keys are not solely based on static constructs. Physical Keys $PHK_A = h(f(OP_A, TS_1)) \oplus y$ and session keys $(CK_A = PHK_A \oplus SK, IK_A = h(PHK_A \parallel SK))$ require solving complex problems over the hyperelliptic curve for each session. This dynamic approach significantly increases the difficulty of quantum attacks compared to static key systems.

9. Formal security analysis using Scyther

9.1. Overview and methodology

The security analysis employs Scyther [57], a symbolic analysis tool, for protocol verification. Scyther's Security Protocol Description Language (SPDL) models protocol roles and interactions, enabling efficient simulation and vulnerability detection through symbolic and operational semantics.

9.2. Protocol specification

The protocol defines three roles: NCC, LEOA, and LEOB. The main variables and cryptographic primitives are outlined in Table 10.

The protocol messages are summarized in Table 11.

9.3. Security claims and verification

The verification process establishes security properties including confidentiality, data integrity, freshness, and resistance to spoofing, Man-in-The-Middle (MiTM), and replay attacks. Table 12 presents the Scyther claims and corresponding verification results for each security property. The analysis confirms the absence of potential attacks within the bounds of maximum verification rounds, validating the protocol's security. The protocol maintains satellite anonymity through hash values combined with orbit identity during mutual authentication. Appendices A and B contain the detailed Scyther script and complete verification results.

10. BAN logic analysis

Burrows–Abadi–Needham (BAN) logic [58] provides formal verification of the authentication protocol's security properties. This methodology, founded on knowledge and belief systems, validates the protocol's security objectives.

10.1. Notation and rules

Tables 13 and 14 define the BAN Logic notation and security rules applied in this analysis.

10.2. Protocol messages in idealized form

1. M1: NCC → LEOA, LEOB: $\{\mathcal{H}\text{EC}, \text{h}, ID_A, ID_B, K_A, K_B, \text{S}, M_p, TS_{NCC}\}$
2. M2: LEOA → NCC: $\{\text{Enc}K_A\ (ID_A, dTID_A, TS_A)\}$
3. M3: LEOB → NCC: $\{\text{Enc}K_B\ (ID_B, dTID_B, TS_B)\}$
4. M4: NCC → LEOA: $\{\text{Enc}K_A\ (C_A, TS_{NCC\_new})\}$
5. M5: NCC → LEOB: $\{\text{Enc}K_B\ (C_B, TS_{NCC\_new})\}$
6. M6: LEOB → LEOA: $\text{Enc}SK\ (dTID_B \parallel \text{h}(TS_2))$
7. M7: LEOA → NCC: $\text{Enc}K_A\ (dTID_B \parallel dTID_A \parallel TS_1)$
8. M8: NCC → LEOA: $\text{Enc}K_A\ (OP_A, OP_B, T_{NCC}, dTID_A)$
   NCC → LEOB: $\text{Enc}K_B\ (OP_B, OP_A, T_{NCC}, dTID_B)$
9. M9: LEOA → LEOB: $AV_A = \{MAC_A, dTID_A, TS_1, Nonce_A, KC_A\}$
10. M10: LEOB → LEOA: $AV_B = \{MAC_B, dTID_B, TS_2, Nonce_B, KC_B\}$

10.3. Security goals and assumptions

Goals:

G1: NCC believes LEOA and LEOB acknowledge their true identities
$NCC \equiv (LEOA \parallel\sim IDA) \wedge (NCC \equiv (LEOB \parallel\sim IDB))$

G2: LEOA and LEOB believe their temporary identities are fresh
$LEOA \equiv \#(dTIDA) \wedge LEOB \equiv \#(dTIDB)$

G3: LEOA and LEOB believe in a shared key for secure communication
$LEOA \equiv (LEOB \equiv K_{SK})$

G4: LEOA and LEOB believe NCC sent their respective orbit parameters
$LEOA \equiv (NCC \parallel\sim OP_A) \wedge LEOB \equiv (NCC \parallel\sim OP_B)$

G5: LEOA and LEOB believe in agreed-upon shared confidentiality and integrity keys
$LEOA \equiv (LEOB \equiv (CK, IK)) \wedge LEOB \equiv (LEOA \equiv (CK', IK'))$

G6: LEOA and LEOB believe the timestamps used are fresh
$LEOA \equiv \#(TS) \wedge LEOB \equiv \#(TS)$

G7: LEOA and LEOB mutually acknowledge the authentication vectors
$LEOA \equiv (LEOB \equiv AV_A) \wedge LEOB \equiv (LEOA \equiv AV_B)$

Assumptions.

A1: NCC believes in secure channels with LEOA and LEOB using shared keys
$NCC \equiv (LEOA \leftrightarrow_K NCC) \wedge (NCC \equiv (LEOB \leftrightarrow_K NCC))$

A2: NCC believes in the keys and identities it generated for LEOA and LEOB
$NCC \equiv (K_A, K_B, ID_A, ID_B)$

A3: NCC, LEOA, and LEOB believe the timestamps used are fresh
$NCC \equiv \#(TS), LEOA \equiv \#(TS_{NCC}), LEOB \equiv \#(TS_{NCC})$

A4: LEOA and LEOB once used a shared secret key K_{SK}
$LEOA \parallel\sim K_{SK} \wedge LEOB \parallel\sim K_{SK}$

A5: LEOA and LEOB once used their respective encryption keys
$LEOA \parallel\sim Enc_{K_A} \wedge LEOB \parallel\sim Enc_{K_B}$

Table 12
SPDL definitions and formal verification of the protocol.

Security property	Scyther claim	Scyther results
Confidentiality	Achieved when messages are encrypted with a secure key known only to the sender and receiver. Formal claims: `Secret AV`, `Secret AV B`.	No attacks detected; secret values are protected.
Data integrity	Ensured when the sender and receiver agree on the exchanged data. Formal claims: `claim(sender, Commit, receiver, data)`, `claim(sender, Running, receiver, data)`.	No attacks detected.
Freshness	Achieved when the sender and receiver are synchronized and agree on exchanged variables. Formal claims: `claim(LEOA, LEOB, Nisynch)`, `claim(LEOA, LEOB, Niagree)`.	No attacks detected; freshness maintained.
Spoofing attack	Ensured when devices confirm they are communicating with each other and not an attacker. Formal claim: `claim(LEOA, L2, Weakagree)`.	No attacks detected; successfully resisted spoofing.
MiTM attack	Achieved when sender and receiver confirm identities, including hashed identities in messages. Formal claims: `claim(LEOA, LEOB, Weakagree)`, `claim(LEOA, LEOB, Nisynch)`.	No attacks detected within bounds.
Replay attack	Prevented when the protocol ensures both agents are alive and not replaying old messages. Formal claims: `claim(LEOA, LEOB, Alive)`, `claim(LEOA, LEOB, Nisynch)`.	No attacks detected; effective resistance to replay attacks.

Table 13
BAN logic notation.

Syntax	Semantics
$Y\vert\equiv X$	Y believes X is true
$Y \overset{K}{\leftrightarrow} Z$	Y and Z share a secret key K
$\{S\}_K$	S is encrypted with key K
$K \rightarrow Z$	K is the public key of Z
$\#(X)$	X is fresh (not replayed from an earlier run)
$Y \lhd S$	Y has received a message containing S
$Y\vert\sim K$	Y once said K (Y sent or used K at some point)
$Y \Longrightarrow S$	Y has jurisdiction over S (Y has full control of S)
YX	Y sees message X

Table 14
BAN logic security rules.

Rule	Definition
Message-Meaning (MMR)	$\frac{Y\vert\equiv(Y\overset{K}{\leftrightarrow}Z),Y\lhd\{S\}_K}{Y\vert\equiv(Z\vert\sim S)}$
Belief (BR)	$\frac{Y\vert\equiv X,Y\vert\equiv U}{Y\vert\equiv(X,U)}$
Seeing (SR)	$\frac{Y\vert\equiv(Y\overset{K}{\leftrightarrow}Z),Y\lhd\{S\}_K}{Y\lhd S}$
Nonce-Verification (NVR)	$\frac{Y\vert\equiv\#(S),Y\vert\equiv(Z\vert\sim S)}{Y\vert\equiv(Z\vert\equiv S)}$
Freshness (FR)	$\frac{Y\vert\equiv\#(S)}{Y\vert\equiv\#(S,Z)}$
Jurisdiction (JR)	$\frac{Y\vert\equiv(Z\Longrightarrow S),Y\vert\equiv(Z\vert\equiv S)}{Y\vert\equiv S}$

A6: LEOA and LEOB believe in the authenticity of shared confidentiality and integrity keys
$LEOA \equiv (LEOB \parallel\sim (CK, IK)) \wedge LEOB \equiv (LEOA \parallel\sim (CK', IK'))$

A7: LEOA and LEOB once used their respective Authentication Vectors
$LEOA \parallel\sim AV_A \wedge LEOB \parallel\sim AV_B$

10.4. Security proofs

Table 15 presents the security proofs, demonstrating how the protocol achieves its security goals. Each proof statement corresponds to one or more security goals, showing how the protocol's design ensures confidentiality, integrity, authentication, and freshness of communications between the NCC and LEO satellites.

Textual explanation of the BAN logic proof

The BAN logic proof demonstrates systematic achievement of security goals (G1–G7) based on initial assumptions (A1–A7). Applying the *message-meaning rule (MMR)* to messages (M1), (M2), and (M3) establishes NCC confirmation of LEOA and LEOB encrypted message authenticity, fulfilling Goals (G1) (NCC satellite identity acknowledgment) and (G3) (satellite shared key beliefs).

Application of the *nonce-verification rule (NVR)* to messages (M9) and (M10) validates LEOA and LEOB trust in authentication vector freshness (AV_A and AV_B), supporting Goal (G6) (timestamp and nonce freshness). Additional rule applications validate mutual beliefs regarding shared keys, temporary identities, and timestamps. Derived statements (S12) and (S13) confirm mutual authentication vector acknowledgment between LEOA and LEOB, achieving Goal (G7). Statements (S8) and (S9) establish shared confidentiality and integrity key agreement, fulfilling Goal (G5). The proof culminates in statements (S1)–(S19), demonstrating comprehensive security goal satisfaction and protocol threat resilience under the specified assumptions.

11. Comparative analysis

The evaluation compares this protocol against existing satellite authentication solutions based on the security properties defined in Table 16. Table 17 presents a comparative overview demonstrating the protocol's security features.

The analysis reveals several limitations in existing protocols. Xu et al. [59] employs ECC and El Gamal for direct authentication but lacks data integrity and eavesdropping resistance [28]. Kong et al. [60] uses identity-based and proxy re-encryption for Beyond 5G networks but omits satellite anonymity and formal security evaluation. Yang et al. [61] supports direct authentication but relies on resource-intensive q-SDH and ECDSA schemes, requiring a global offline Trusted Third Party (TTP) while lacking data integrity [63]. Liu et al. [62] provides mutual authentication for Satellite–Terrestrial Networks through hybrid cryptography but introduces high computational overhead and inadequate user-side key confidentiality. The proposed protocol addresses these limitations through efficient HECC and Physical State Keys, enhancing both security and performance. It implements dedicated integrity mechanisms, secure key management, and satellite anonymity via dynamic temporary identities. Protection against various attacks is achieved through encryption and Message Authentication Codes (MACs). Rigorous Scyther tool evaluation confirms achievement of all security properties (P1–P10).

12. Security and performance modeling

This section presents security and performance analysis through MATLAB and Systems Tool Kit (STK) simulations. The evaluation examines three key aspects. First, performance metrics measure authentication success rates under varying conditions (normal and high load). The analysis demonstrates how Continuous Authentication (CAT) effectively mitigates spoofing attacks, while also evaluating system performance metrics including computational overhead and communication costs. Second, comparative analysis against existing protocols demonstrates the protocol's efficiency in LEO environments, focusing

Table 15
Security proofs of the protocol.

Proof	Statement	Goals achieved
S1	$NCC \equiv (LEOA \leftrightarrow_K NCC) \wedge (LEOB \leftrightarrow_K NCC)$	G1, G3
S2	$LEOA \equiv (NCC \equiv IDA)$	G1
S3	$LEOB \equiv (NCC \equiv IDB)$	G1
S4	$LEOA \equiv \#(dTID_A) \wedge LEOB \equiv \#(dTID_B)$	G2
S5	$LEOA \equiv (LEOB \equiv dTID_B)$	G2
S6	$LEOB \equiv (LEOA \equiv dTID_A)$	G2
S7	$LEOA \equiv (NCC \mid OPA) \wedge LEOB \equiv (NCC \mid OPB)$	G4
S8	$LEOA \equiv (LEOB \equiv K_{SK})$	G3, G5
S9	$LEOA \equiv (LEOB \equiv (CK, IK)) \wedge LEOB \equiv LEOA \equiv KCA$	G5
S10	$LEOB \equiv (LEOA \equiv (CK', IK')) \wedge LEOB \equiv LEOA \equiv KCB$	G5
S11	$LEOA, LEOB \equiv \#(TS1, TS2)$	G6
S12	$LEOA \equiv (LEOB \equiv AV_A)$	G7
S13	$LEOB \equiv (LEOA \equiv AV_B)$	G7
S14	$LEOA \equiv \#(Nonce_A) \wedge LEOB \equiv \#(Nonce_B)$	G6
S15	$LEOA \equiv (LEOA \leftrightarrow_S LEOB)$	G3
S16	$LEOA \equiv (M_p \rightarrow NCC) \wedge LEOB \equiv (M_p \rightarrow NCC)$	G3
S17	$LEOA \equiv \#(PHK_A) \wedge LEOB \equiv \#(PHK_B)$	G2
S18	$NCC \equiv (LEOA \equiv dTID_A) \wedge NCC \equiv (LEOB \equiv dTID_B)$	G1, G2
S19	$LEOA \equiv LEOB \equiv (LEOA \leftrightarrow_{SK} LEOB)$	G3, G5, G7

Table 16
Security properties.

Property	Description
P1	Mutual authentication: Direct satellite authentication
P2	Lightweightness: Minimal resource overhead
P3	Data integrity: Protection against data tampering
P4	Satellite anonymity: Identity protection
P5	Key confidentiality: Cryptographic key secrecy
P6	Eavesdropping resistance: Protection against unauthorized listening
P7	Impersonation resistance: Prevention of entity spoofing
P8	Data modification resistance: Prevention of unauthorized changes
P9	Replay resistance: Protection against reuse of old communications
P10	Formal evaluation: Rigorous security analysis

Table 17
Security comparison with existing literature.

Scheme	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Xu et al. [59]	✓	✓	✗	✓	✓	✗	✓	✗	✓	✓
Kong et al. [60]	✓	✗	✓	✗	✗	✗	✓	✓	✓	✗
Yang et al. [61]	✓	✗	✗	✓	✗	✓	✓	✗	✓	✓
Liu et al. [62]	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗
Our protocol	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

on latency and bandwidth overhead. Third, scalability validation uses two configurations: a Walker 120/10/1 baseline configuration with 120 satellites across 10 planes, and a Walker 1200/30/1 mega-scale deployment with 1200 satellites across 30 planes.

12.1. Simulation environment

Simulations were conducted on a Lenovo ThinkPad Gen2i with an Intel® Core™ i5-1145G7 processor and 16 GB of RAM. The software environment included MATLAB (R2024a) and ANSYS Systems Tool Kit (STK) v.11.2, with Visual Studio Code as the development environment on Windows 10 Enterprise. STK was used to simulate the lightweight processing requirements of satellites, ensuring an accurate representation of satellite operations independent of laptop hardware capabilities.

12.2. Simulation components and parameters

The components of the simulation environment are shown in Fig. 6 and outlined Table 18.

The STK simulation parameters are summarized in Table 19 from parameters obtained from [64,65]

12.3. Main processes

The `leoSatelliteAuthentication` function the programming environment simulates satellite-to-satellite communication, beginning with an authentication request from LEOA and concluding with mutual authentication with LEOB. Predefined orbit parameters and shared keys to mirror the operational environment of LEO satellites, ensuring the scheme's efficacy for authorized entities. STK's Scheduler sets up discrete events for authentication attempts and implements logic for successful and failed authentication scenarios.

Orbit parameter calculation

The process dynamically calculates orbit parameters relevant to communication timing. The Skyfield library generates realistic Two-Line Element (TLE) sets for each simulated satellite, enabling accurate position calculations and unique orbital elements for the authentication process. Key simulation steps include:

1. **Position Vector:** We used STK's High Precision Orbit Propagator (HPOP) to determine each satellite's position vector (x, y, z coordinates). This calculation incorporates Keplerian elements and accounts for factors such as Earth's gravitational field, atmospheric drag, and third-body effects, ensuring high precision in modeling each satellite's trajectory.
2. **Velocity Vector:** Calculated using the equation $v = \sqrt{\frac{GM}{r}}$, where G is the gravitational constant, M is Earth's mass, and r is the distance from Earth's center to the satellite. For a typical LEO satellite at an altitude of 500 km, this yields an approximate velocity of 7.6 km/s (27,400 km/h), with variations depending on the specific orbital parameters.
3. **Time of Ephemeris (ToE):** STK automates the generation of ephemeris data, providing time-tagged position and velocity vectors, which ensure accurate satellite motion modeling over time.

The simulation established two satellite instances with distinct orbital characteristics, detailed in Table 20 and visualized in Fig. 6. LEOA occupied a position of (−38.97°, −167.57°, 426.70 km), with an inclination of 39.93° and an eccentricity of 0.0011. LEOB operated at (79.97°, 121.96°, 552.85 km), with an inclination of 82.27° and an eccentricity of 0.0006. These diverse parameters demonstrate the capability to model satellites in different orbits, essential for validation of the authentication scheme. While orbital parameters remain stable, each satellite's position and velocity vary continuously along its path, providing dynamic authentication parameters.

Table 18
Simulation environment components.

Component	Description
`NCC`	Central hub for communications and control
`DLRcvr`	Downlink signal receiver
`NCC_Ant`	Ground station-satellite communication antenna
`LEOA, LEOB`	Peer satellites for mutual authentication
`LEOA_Ant, LEOB_Ant`	Satellite communication antennas
`ULRcvr`	Uplink signal receiver
`Servomotor`	Used for steering satellite for precise alignment and orientation.

Table 19
STK simulation parameters.

Model	Specs	Model	Specs
Ground station model			
NCC-DLRcvr	12.1 GHz	Power	10 dBW
NCC-ULRcvr	Uplink 6 GHz	Data rate	10 Mbps
Antenna design frequency	12.1 GHz	Antenna type	Gaussian
Beamwidth	26.30 deg	Diameter	5.3 m
Main-lobe gain	45.02 dB	Efficiency	55%
Transmit pointing loss	−1.74 dB	Polarization loss	−0.5 dB
Atmospheric loss	−2.1 dB	Ionospheric loss	−0.1 dB
LEOA-B receiver model			
Frequency	12.1 GHz	LNA gain	0 dB
Antenna design frequency	12.1 GHz	Antenna to LNA line loss	−2 dB
Rain model outage percent	5%	Receive pointing error	0.2 deg
System noise temperature	522 K	Antenna efficiency	55%
LEOA-B transmitter model			
Frequency	12.1 GHz	Power	10 dBW
Antenna design frequency	12.1 GHz	Data rate	10 Mbps
Beamwidth	26.30 deg	Diameter	5.3 m
Main-lobe gain	45.02 dB	Efficiency	55%
Back-lobe gain	−30 dB	Path length	500 km

Table 20
Orbital parameters and positions of LEOA and LEOB.

Parameter	LEOA	LEOB
Position (lat, lon, alt)	(−38.97°, −167.57°, 426.70 km)	(79.97°, 121.96°, 552.85 km)
Semi-major axis (km)	6789.68	6905.92
Eccentricity	0.0011	0.0006
Inclination (°)	39.93	82.27
RAAN (°)	274.37	0.74
Arg. of perigee (°)	76.23	262.51
True anomaly (°)	207.01	193.85

Key generation and exchange

The key generation and exchange process involves both physical state and cryptographic keys, along with the creation of message authentication codes (MACs):

1. **Physical State Keys (PHK):** Generated based on the actual satellite positions, implemented using the Skyfield Library's `calculate_position` method. This method is used during communication to validate the authenticity of the position claims in the authentication process
2. **Cryptographic Operations:** Implementation includes Hyperelliptic curve multiplication, Bilinear pairing operations, and Hash functions. Table 21 presents computational costs derived from the MIRACL Library and literature [66,67].
3. **Authentication Vector Generation:** The derived keys create Authentication Vectors (AV) containing Message Authentication Codes (MAC), dynamic Temporary IDs (dTID), timestamps, and nonces.

12.4. Performance results

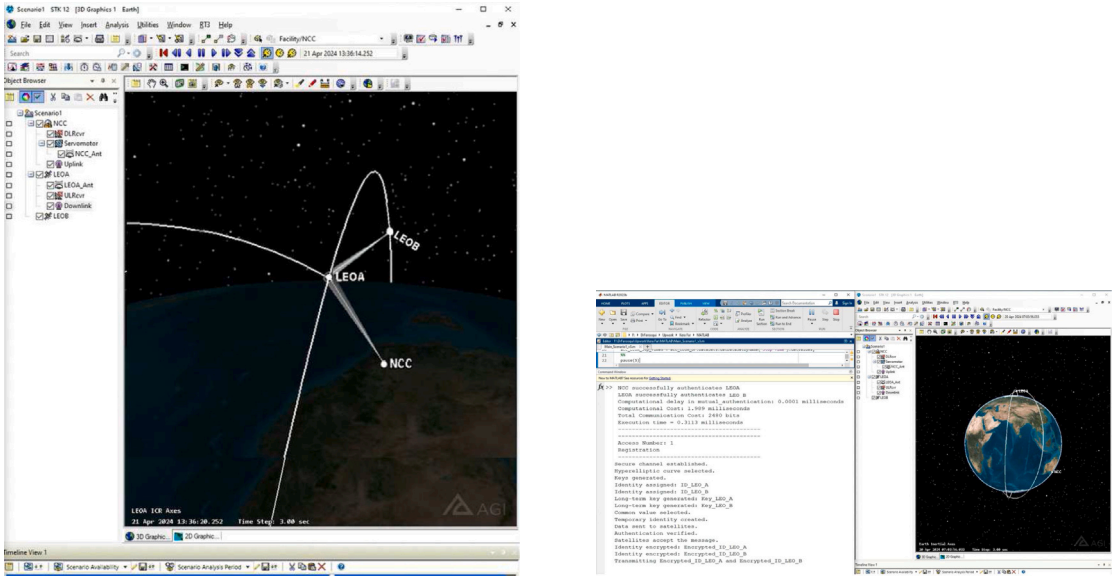
Testing results show communication completion at Epoch 56630.9518281369 ms. The mutual authentication phase achieved a computational delay of 0.0001 ms, computational cost of 1.989 ms, and communication cost of 2480 bits, resulting in an execution time of 0.3113 ms. The key update phase increased the computational cost to 2.9904 ms with a total communication cost of 2880 bits.

Table 21
Computation descriptions.

Computation	Time (ms)	Bit-length
Hyperelliptic curve multiplication (HECM)	0.48	80 bits
Elliptic curve multiplication (ECM)	0.97	160 bits
Modular multiplication (MM)	4.31	1024–2048 bits
Bilinear pairing operation (bp)	14.90	512–1024 bits
Hashing operation (h)	0.0023	Negligible
Encryption/Decryption	0.0046	–

12.5. Link budget analysis with clock drift considerations

STK-generated link budget analysis evaluated signal quality and strength shown in Table 22. The analysis includes Effective Isotropic Radiated Power (EIRP), received frequency and power, flux density, signal-to-noise ratios (C/N_0, C/N, E_b/N_0), bandwidth, Bit Error Rate (BER), and timing offset due to clock drift. This analysis validates the



(a) Overview of STK Environment (b) Mutual Authentication in STK

Fig. 6. Combined overview of mutual authentication STK simulation.

Table 22
Link budget report: Facility-ComFac Receiver-DLRcvr-To-Satellite-SATCOM-Transmitter.

Time (UTC)	EIRP (dBW)	Recv. Freq. (GHz)	Recv. Power (dBW)	Flux Dens. (dBW/m^2)	G/T (dB/K)	C/No (dB*Hz)	BW (kHz)	C/N (dB)	Eb/No (dB)	Clock Drift (ppb)	Timing Offset (ps)	BER
11 Apr 2024 18:00:00	50.0	0.137	−141.87	−108.40	20.0	83.0	34.0	49.0	41.0	0.010	0.6	$<10^{-5}$
11 Apr 2024 18:01:00	50.1	0.137	−141.85	−108.38	20.1	83.2	34.0	49.1	41.1	0.010	1.2	$<10^{-5}$
11 Apr 2024 18:02:00	50.2	0.137	−141.84	−108.37	20.2	83.4	34.0	49.2	41.2	0.010	1.8	$<10^{-5}$
11 Apr 2024 18:03:00	50.3	0.137	−141.82	−108.35	20.3	83.6	34.0	49.3	41.3	0.010	2.4	$<10^{-5}$
11 Apr 2024 18:04:00	50.4	0.137	−141.81	−108.34	20.4	83.8	34.0	49.4	41.4	0.010	3.0	$<10^{-5}$
11 Apr 2024 18:05:00	50.5	0.137	−141.80	−108.33	20.5	84.0	34.0	49.5	41.5	0.010	3.6	$<10^{-5}$
11 Apr 2024 18:06:00	50.6	0.137	−141.79	−108.32	20.6	84.2	34.0	49.6	41.6	0.010	4.2	$<10^{-5}$
11 Apr 2024 18:07:00	50.7	0.137	−141.78	−108.31	20.7	84.4	34.0	49.7	41.7	0.010	4.8	$<10^{-5}$
11 Apr 2024 18:08:00	50.8	0.137	−141.77	−108.30	20.8	84.6	34.0	49.8	41.8	0.010	5.4	$<10^{-5}$
11 Apr 2024 18:09:00	50.9	0.137	−141.76	−108.29	20.9	84.8	34.0	49.9	41.9	0.010	6.0	$<10^{-5}$

EIRP: Effective Isotropic Radiated Power, Recv. Freq.: Received Frequency, Recv. Power: Received Power
Flux Dens.: Flux Density, G/T: Gain-to-Noise-Temperature, C/No: Carrier-to-Noise-Density Ratio
BW: Bandwidth, C/N: Carrier-to-Noise Ratio, Eb/No: Energy per Bit to Noise Power Spectral Density Ratio
Clock Drift: Instantaneous frequency drift rate in parts per billion
Timing Offset: Accumulated timing error due to clock drift in picoseconds
BER: Bit Error Rate.

authentication scheme's effectiveness under various LEO communication scenarios while accounting for timing impairments.

The link budget analysis incorporates clock drift effects, addressing critical timing considerations in satellite communications. Clock drift values derive from validated BDS-3 satellite onboard atomic clock specifications [68], with PHM exhibiting drift rates of $< 1 \times 10^{-14}$/s. High-stability atomic frequency standards and the two-way measurement approach achieve this minimal drift impact by canceling common-mode drift effects. The results demonstrate negligible clock drift contribution to the overall link budget, with timing offsets well within operational margins for reliable inter-satellite communications.

12.6. Authentication success rates under varied conditions

Performance analysis examines protocol resilience across three operational states: baseline operation, peak load conditions (80%–90% ISL utilization), and link interruption scenarios. The high load threshold, derived from LEO broadband network studies [69], represents critical stress points where Inter-Satellite Links (ISLs) and processing capabilities approach maximum capacity.

Peak communication periods and emergency situations typically trigger high load conditions when multiple satellites handle large-scale data transmissions simultaneously. Under these circumstances, ISLs normally supporting 25 Mbps experience significant performance degradation. Latency increases from 50 ms under normal conditions to 150 ms at 90% utilization, while packet drop rates escalate from less than 2% to 10%–15%. Temporary interruptions, occurring during 10%–20% of communication time, further impact performance by delaying token synchronization and reducing authentication success rates [69].

12.7. Authentication success rate

The Performance analysis involved 30 authentication attempts between satellites LEOA and LEOB, with each attempt logged for statistical evaluation. The protocol demonstrated 96%–100% success rates under normal conditions for authorized users. In contrast, unauthorized access attempts, including spoofing attacks, achieved success rates below 10%, indicating effective threat detection and rejection. The

Table 23
Simulation of compromised variables and functions.

Variable/Function	Description
compromisedOP	Potentially compromised orbital parameters
compromisedKey	Potentially compromised shared key
generateDynamicIdentity	Generates new dynamic identity based on compromised parameters
encrypt	Encrypts CAT and dynamic identity using compromised key
transmitToken	Sends forged authentication token
verifyToken	Checks token validity; simulates potential attack success

Table 24
Performance metrics under various conditions.

Condition	Latency (ms)	Processing time (ms)	Data size (bits)
Normal load (50% Utilization)	0.3113	1.989	2480
High load (80% Utilization)	1.2	2.5	2700
With key update phase	0.5	2.9904	2880
High load + Key update	1.5	3.5	3100

non-zero unauthorized access rate, however, indicates a potential vulnerability window. Protocol resilience testing under high load and temporary interruptions focused on attack scenarios involving forged authentication tokens using compromised orbital parameters and shared keys. Table 23 details the simulation variables and functions.

With Continuous Authentication (CAT) disabled, unauthorized access success rates ranged between 2%–8%. CAT implementation reduced this vulnerability by periodically recalculating Continuously Authenticated Tokens, effectively limiting compromised key validity periods, shown in Fig. 9a. The protocol maintains communication integrity through inter-satellite acknowledgments that confirm authentication cycle completion. Session updates and encrypted timestamps further minimize the attack window.

Feasibility under high load conditions

The analysis examines protocol performance under high network load conditions affecting real-time orbital data exchange. The simulation used an 80% link utilization scenario with parameters from [69]. The network configuration established ISL capacity at 25 Mbps with 20 ms base delay. Traffic modeling combined 15 Mbps constant bit rate (CBR) flows with Pareto-distributed On/Off traffic bursts. Testing encompassed 30 authentication attempts under both normal (50%) and high (80%) utilization.

The results, summarized in Table 24 and Fig. 10. show increased latency from 0.3113 ms to approximately 1.2 ms under high load due to queuing delays. The mutual authentication phase maintained a computational delay of 0.0001 ms, with computational costs rising from 1.989 ms to 2.5 ms during peak load. Communication costs increased from the baseline 2480 bits to approximately 2700 bits due to retransmissions. The key update phase showed similar scaling, with computational costs increasing from 2.9904 ms to 3.5 ms and communication costs from 2880 to 3100 bits under high load conditions. During communication interruptions, occurring 20% of the time, the protocol achieved state resynchronization within 0.5 ms of reconnection in 98% of cases, maintaining success rates above 98%. Failures primarily resulted from token expiration during extended interruptions. Mitigation strategies for extreme conditions include implementing a traffic reduction ratio ($\chi = 30\%$) for load burst optimization, redistributing traffic from congested ISL links to reduce packet drop rates below 5%, and caching authentication tokens for immediate post-interruption retransmission [69–71], restoring success rates to 97%–99%.

Table 24 presents comprehensive authentication success rates under various conditions.

12.8. performance comparison

The protocol's computational and communication efficiency was evaluated against four existing satellite authentication schemes [59–62]. Execution time measurements using the MIRACL library shown in Table 21 measured cryptographic operations: bilinear pairing ($\mathcal{BP}$), scalar multiplication ($\mathcal{SM}$), exponentiation ($\mathcal{EX}$), elliptic curve multiplication ($\mathcal{EC}_M$) and addition ($\mathcal{ECA}$), hyperelliptic curve multiplication ($\mathcal{HECM}$), and hash operations ($\hbar$). Table 24 presents the comparative computational costs for encryption and decryption across all schemes.

For communication efficiency, the bit-transfer overhead comparison used a standardized message length of $|\mathbb{m}| = 100$ bits. Our scheme showed a total communication cost of 2880 bits, using 6 $\mathcal{HECM}$ and 7 $\hbar$. This represents a reduction of 72.03% compared to [62], which required 3040 bits, and a 64.71% reduction relative to [59], which used 3520 bits. [61] exhibited a higher cost of 4160 bits, yielding a 96.68% reduction with our method, while [60] incurred the highest cost at 22528 bits, resulting in a substantial 98.15% reduction when using our protocol. These results highlight the superior communication efficiency of our HECC-based scheme compared to the alternatives, while maintaining strong cryptographic security. Total computation and communication cost with reduction percentages are shown (Table 25, Figures 11–14).

12.9. Scalability analysis

The scalability of the proposed authentication protocol is validated through analysis of two satellite configurations: a baseline Walker configuration with 10 planes of 120 satellites and a mega-scale Walker configuration with 30 planes of 1200 satellites. This evaluation addresses modern mega-constellations comprising thousands of satellites. The baseline configuration derives from findings for ISL network design in LEO satellite networks [72], demonstrating efficient global coverage while maintaining continuous inter-satellite links. The mega-scale configuration extends this model by a factor of 10 to validate practical scalability.

For both configurations, the protocol maintains fixed local authentication boundaries. Each satellite manages exactly four secure neighbor connections:

$$\text{Links per Satellite} = k_{\text{intra}} + k_{\text{inter}} \tag{4}$$

where k_{intra} represents two intra-plane links (forward/backward neighbors) and k_{inter} represents two inter-plane links (adjacent planes), as illustrated in Fig. 7.

Table 25
Total computational and communication costs with reduction percentages.

Scheme	Computational steps	Comm. cost (bits)	Comm. reduction (%)	Comp. cost (ms)	Comp. reduction (%)
Ref. [59]	$6\mathcal{BP} + 2\mathcal{SM} + 4\mathcal{EX} + 7\hbar$	3520	64.71	8.473	18.18
Ref. [60]	$11\mathcal{BP} + 9\mathcal{SM} + 7\mathcal{EX} + 10\hbar$	22 528	98.15	162.323	87.21
Ref. [61]	$14\mathcal{EC}_M + 8\mathcal{ECA} + 28\hbar$	4160	96.68	90.114	30.76
Ref. [62]	$4\mathcal{BP} + 4\mathcal{SM} + 1\mathcal{EX} + 6\hbar$	3040	72.03	10.693	5.26
Ours	$6\mathcal{HECM} + 7\hbar$	2880	–	2.990	–

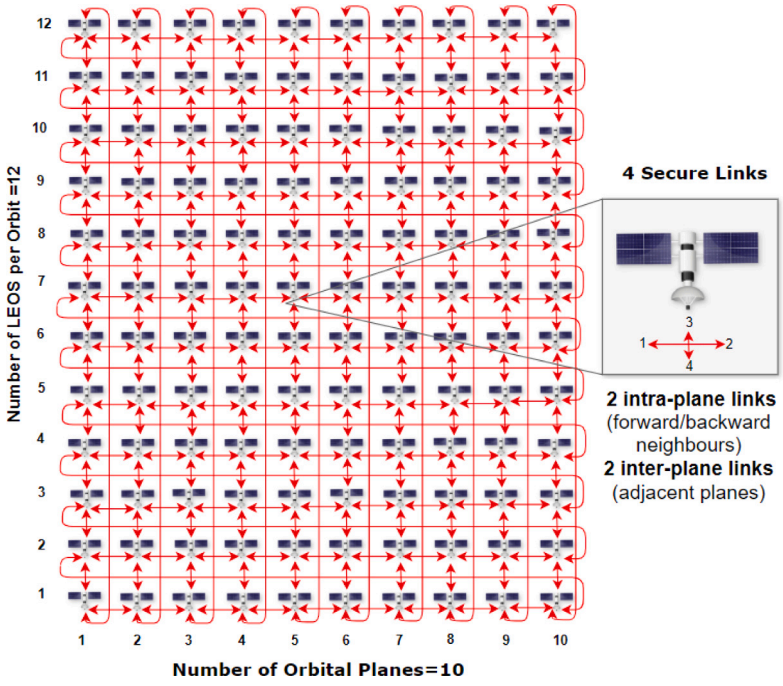


Fig. 7. Link topology of 10 planes of 122 LEOs.

Table 26
Protocol performance across constellation sizes.

Configuration	(N) LEOS	Links per LEO	Latency (ms)	Bandwidth (bits)
Baseline Walker 120/10/1	$N = 120$	$n = 4$	$T_{\text{auth}} = 954.72$	$T_{\text{comm}} = 1,190,400$
Mega-Scale Walker 1200/30/1	$N = 1200$	$n = 4$	$T_{\text{auth}} = 9547.2$	$T_{\text{comm}} = 11,904,000$

Per-satellite performance

The baseline Walker 120/10/1 configuration establishes core per-satellite performance metrics. Each authentication link requires 1.989 ms computation time and 2480 bits communication overhead. For a satellite with four secure links, the total computational overhead follows:

$$T_{\text{total}} = n \times (T_{\text{comp}} + T_{\text{comm}}) \tag{5}$$

where T_{comp} is the computational time per link (1.989 ms), T_{comm} is the communication processing time, and n is the number of links (4). This results in a fixed per-satellite overhead of 7.956 ms computation and 9920 bits communication. These metrics remain constant when scaling to the Walker 1200/30/1 mega-constellation, as each satellite maintains exactly four neighbor connections regardless of total constellation size shown in Fig. 16. (see Fig. 8).

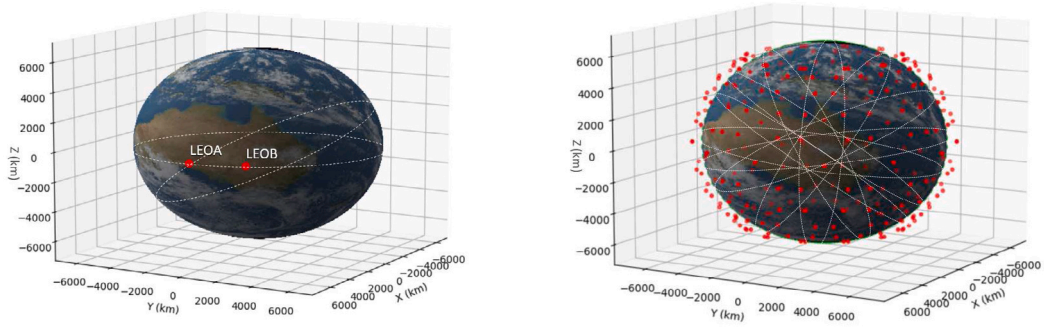
Network-wide scaling

The protocol achieves linear scaling across constellation sizes is depicted in Table 26.

The protocol achieves linear complexity $O(4N)$ rather than exponential $O(N^2)$ through localized authentication patterns shown in Fig. 15. Authentication occurs strictly between immediate neighbors, while additional planes or satellites replicate the same local pattern. This architecture ensures authentication chains remain bounded by plane size rather than total constellation size.

13. Conclusion

This work presents an efficient Zero Trust authentication protocol for LEO satellite networks that leverages orbital dynamics with Hyperelliptic Curve Cryptography. The protocol's security properties are formally verified through comprehensive analysis using Scyther, BAN Logic, and heuristic techniques, confirming resistance against known attacks while ensuring mutual authentication, key confidentiality, unforgeability, and satellite anonymity. The performance evaluation demonstrates significant improvements over existing schemes, reducing communication overhead by 18.18% to 87.21% and computational costs by 5.26% to 98.15%. Scalability analysis validates the protocol's efficiency from a baseline Walker 120/10/1 configuration through to a mega-scale Walker 1200/30/1 deployment, achieving linear growth in network-wide resources while maintaining constant per-satellite overhead. This architectural approach, combined with formal security guarantees, establishes a robust foundation for secure authentication in next-generation LEO satellite networks.



(a) 2-Satellite Scenario (b) 120-Satellite Scenario

Fig. 8. Scalability evaluation.

CRediT authorship contribution statement

Kerry Anne Farrea: Writing – original draft, Visualization, Validation, Methodology, Formal analysis. **Zubair Baig:** Writing – review & editing, Supervision. **Robin Doss:** Writing – review & editing, Supervision. **Dongxi Liu:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: All coauthors report financial support was provided Cyber Security Research Centre Limited whose activities are partially funded by the Australian Government's Cooperative Research Centres Programme.

Acknowledgments

The work has been supported by the Cyber Security Research Centre Limited whose activities are partially funded by the Australian Government's Cooperative Research Centres Programme.

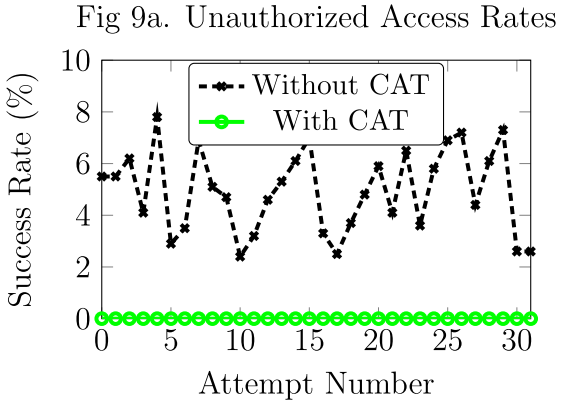
Appendix A. Verified results in Scyther

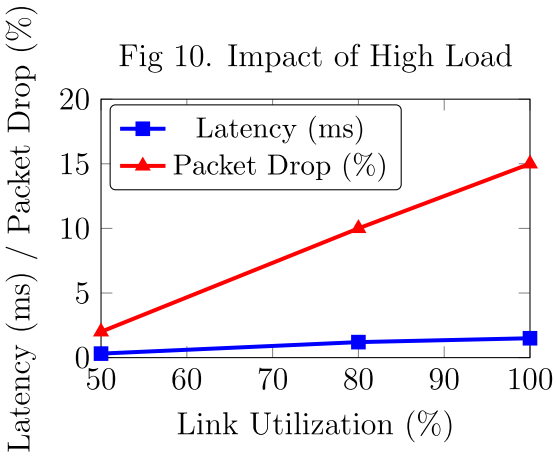
See Fig. A.17.

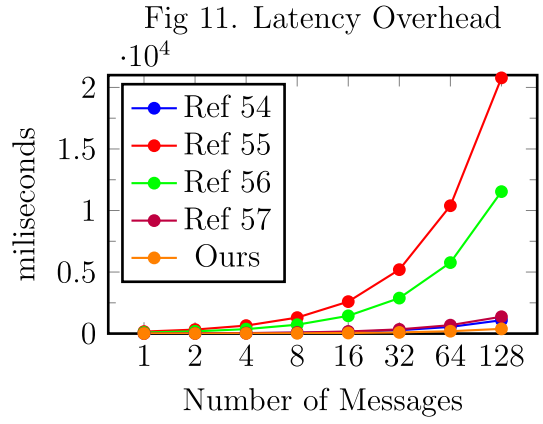
Appendix B. SPDL code for Scyther evaluation

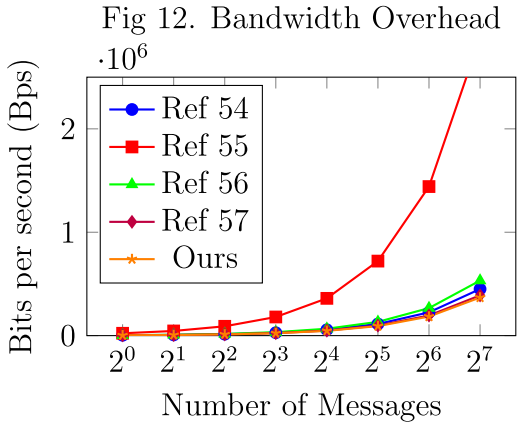
```
usertype NCC, LEOA, LEOB;
usertype TimeStamp;
hashfunction h;
secret ka: Function;
secret kb: Function;
secret hec: Function;
secret F: Function;
secret Mult: Function;
secret Concat: Function;
secret XOR: Function;
secret RNG1: Function;
secret RNG2: Function;
secret y;
secret sk;
secret seed: Function;
const OPA, OPB, IDa, IDb, Ts, Tsa, Tsb, TIDA, TIDB, dTIDa
    , dTIDb;

protocol MutualAuthentication(NCC, LEOA, LEOB) {
    role NCC {
        fresh ni: Nonce;
        var Ts;
        macro C = (h, hec, F, IDa, ka);
        macro D = (h, hec, F, IDb, kb);
        send_1(NCC, LEOA, {C}k(NCC, LEOA));
        send_2(NCC, LEOB, {D}k(NCC, LEOB));
        recv_3(LEOA, NCC, {TIDA, TIDB, Ts}ka(LEOA, NCC));
        send_4(NCC, LEOA, {OPA, OPB, Ts, TIDA}ka(NCC,
            LEOA));
        send_5(NCC, LEOB, {OPB, OPA, TIDB}kb(NCC, LEOB));
    }
    role LEOA {
        fresh Ts: TimeStamp;
        fresh Tsa: TimeStamp;
        fresh AVA;
        var AVB;
        recv_1(NCC, LEOA, {C}k(NCC, LEOA));
        send_3(LEOA, NCC, {TIDA, TIDB, Ts}ka(LEOA, NCC));
        recv_4(NCC, LEOA, {OPA, OPB, Ts, TIDA}ka(NCC,
            LEOA));
        macro OPa = seed(OPA);
        macro OPb = seed(OPB);
        macro XYZA = Concat(OPa, Tsa);
        macro Z = h(XYZA);
        macro PHkA = Mult(Z, y);
        macro val = XOR(sk, PHkA);
        macro Ck1 = RNG1(val);
        macro IK1 = Ck1;
        macro vala = Concat(IK1, PHkA, dTIDa, Tsa);
        macro MACA = RNG2(vala);
        macro AVa = Concat(MACA, dTIDa, Tsa);
        match(AVa, AVA);
        send_6(LEOA, LEOB, {AVA}sk(LEOA, LEOB));
        recv_7(LEOB, LEOA, {AVB}sk(LEOB, LEOA));
        macro XYZB = Concat(OPb, Tsb);
        macro q = h(XYZB);
        macro PHkB = Mult(q, y);
        macro vale = XOR(sk, PHkB);
        macro Ck2 = RNG1(vale);
        macro IK2 = Ck2;
        macro valu = Concat(IK2, PHkB, dTIDb, Tsb);
        macro MACB = RNG2(valu);
        match (MACB, MACB);
        claim(LEOA, Secret, AVA);
        claim(LEOA, Secret, AVB);
        claim(LEOA, Secret, ka(LEOA, LEOB));
        claim(LEOA, Alive);
        claim(LEOA, Nisynch);
        claim(LEOA, Niagree);
        claim(LEOA, Weakagree);
        claim(LEOA, Secret, sk(LEOA, LEOB));
    }
    role LEOB {
        fresh Tsb: TimeStamp;
        fresh Tsa: TimeStamp;
        fresh AVB;
        var AVA;
        recv_2(NCC, LEOB, {D}k(NCC, LEOB));
        recv_5(NCC, LEOB, {OPB, OPA, TIDB}kb(NCC, LEOB));
        recv_6(LEOA, LEOB, {AVA}sk(LEOA, LEOB));
        macro OPA = seed(OPa);
        macro OPB = seed(OPb);
        macro XYZA = Concat(OPa, Tsa);
        macro j = h(XYZA);
        macro PHkA = Mult(j, y);
        macro valz = XOR(sk, PHkA);
        macro Ck1 = RNG1(valz);
        macro IK1 = Ck1;
        macro valx = Concat(IK1, PHkA, dTIDa, Tsa);
        macro MACAX = RNG2(valx);
        match (MACA, MACAX);
```









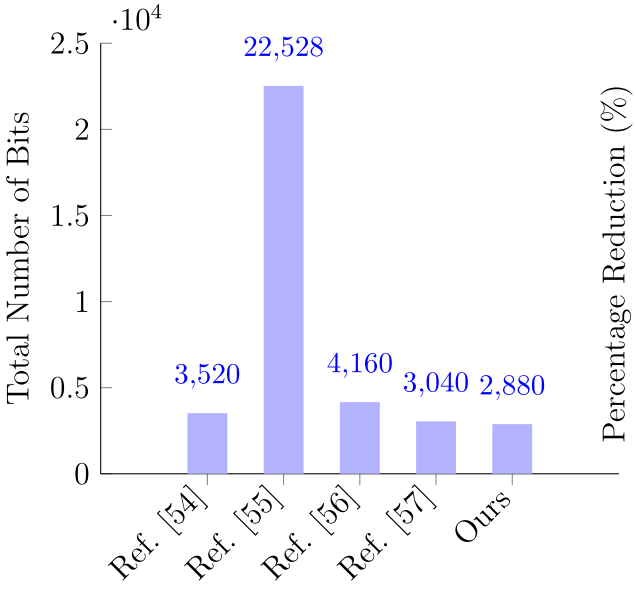


Fig 13: Computational Cost.

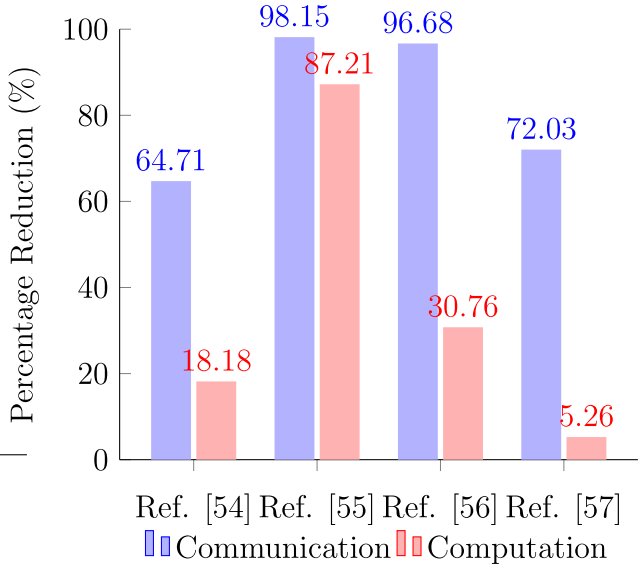


Fig 14: Total Overhead Cost.

Fig. 9. Performance simulation results (Figs. 9a–14).

```
macro XYZB = Concat(OPb, Tsb);
macro p = h(XYZB);
macro PHkB = Mult(p, y);
macro valv = XOR(sk, PHkB);
macro Ck2 = RNG1(valv);
macro IK2 = Ck2;
macro valt = Concat(IK2, PHkB, dTIDb, Tsb);
macro MACB = RNG2(valt);
macro AVb = Concat(MACB, dTIDb, Tsb);
```

```
match(AVb, AVB);
send_7(LEOB, LEOA, {AVB}sk(LEOB, LEOA));
claim(LEOB, Secret, AVA);
claim(LEOB, Secret, AVB);
claim(LEOB, Secret, ka(LEOA, LEOB));
claim(LEOB, Secret, kb(LEOA, LEOB));
claim(LEOB, Alive);
claim(LEOB, Nisynch);
claim(LEOB, Niagree);
```

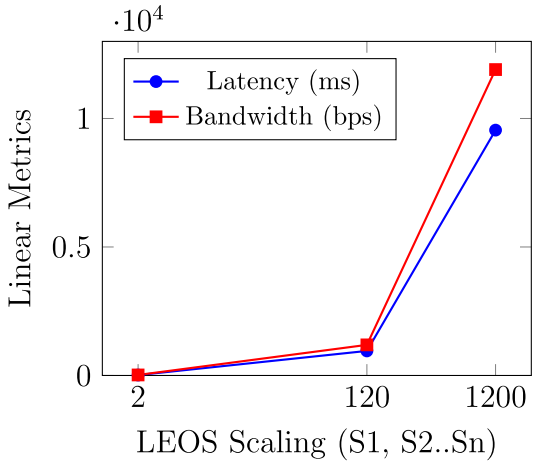


Fig. 15. Network-wide protocol scalability: linear scaling of latency and bandwidth.

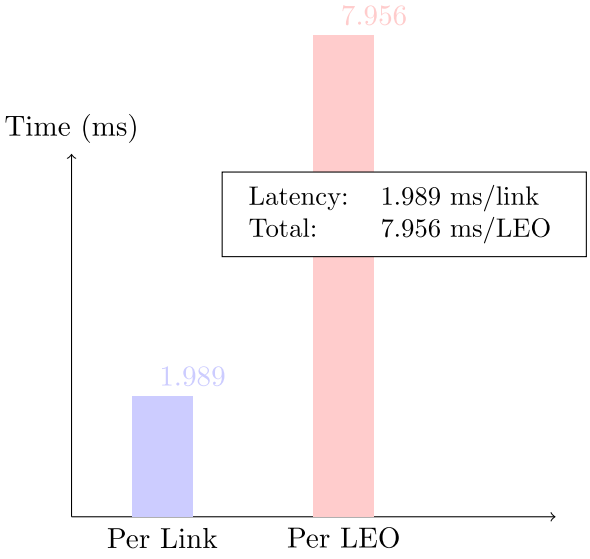


Fig. 16. Total satellite overhead.

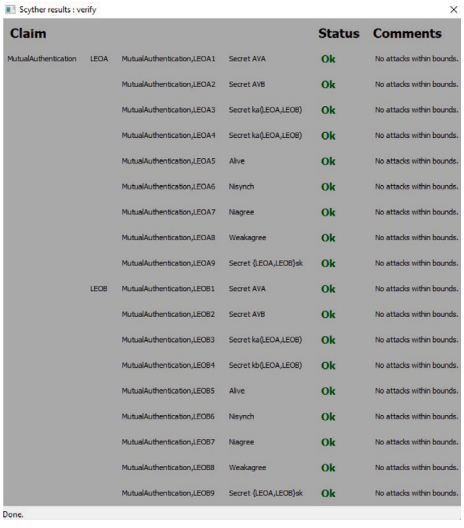


Fig. A.17. Verified Scyther results.

```
        claim(LEOB, Weakagree);
        claim(LEOB, Secret, sk(LEOA, LEOB));
    }
}
```

Data availability

No data was used for the research described in the article.

References

[1] T. Taleb, et al., On-demand media streaming to hybrid networks over quasi-geostationary satellite systems, Comput. Netw. 47 (2) (2005) 287–306, http://dx.doi.org/10.1016/j.comnet.2004.07.010.

[2] S. Vishwakarma, et al., A comparative study of satellite orbits: LEO and GEO, SAMRIDDHI 6 (2) (2015) http://dx.doi.org/10.18090/samriddhi.v6i2.1559.

[3] C. Carrizo, et al., Optical inter-satellite link terminals for next-gen satellite constellations, 2020, http://dx.doi.org/10.1117/12.2545629.

[4] M. Handley, Delay is not an option: Low latency routing in space, in: Proc. HotNets, 2018.

[5] K.A. Farrea, Z.A. Baig, R. Doss, D. Liu, Provably secure optimal homomorphic signcryption for satellite-based internet of things, Comput. Netw. 250 (2024) 110516.

[6] X. Fang, et al., 5G embraces satellites for 6G ubiquitous IoT, IEEE IoT J. 8 (18) (2021) 14399–14417, http://dx.doi.org/10.1109/JIOT.2021.3068596.

[7] S. Lohani, R. Joshi, Satellite network security, in: Proc. ICONC3, 2020, pp. 1–5, http://dx.doi.org/10.1109/ICONC345789.2020.9117553.

[8] M.Z. Chowdhury, et al., 6G wireless communication systems: Applications, requirements, Technol. IEEE Open J. Commun. Soc. 1 (2020) 957–975, http://dx.doi.org/10.1109/OJCOMS.2020.3010270.

[9] L.-J. Wang, et al., Experimental authentication of quantum key distribution with post-quantum cryptography, Npj Quantum Inform. 7 (67) 2021.

[10] X. Deng, et al., A blockchain-based privacy protection protocol using smart contracts in LEO satellite networks, Peer- To- Peer Netw. Appl. 17 (2024) 800–818, http://dx.doi.org/10.1007/s12083-023-01614-6.

[11] S.-K. Liao, et al., Satellite-to-ground quantum key distribution, Nature 549 (2017) 43–47.

[12] T. Muhammad, et al., Integrative cybersecurity: Zero trust, layered defense, global standards, Int. J. Comp. Sci. Tech. 6 (4) (2022) 99–135.

[13] G. Falco, N.G. Gordon, Zero-trust satellite services marketplace, IEEE Access 12 (2024) 71066–71075, http://dx.doi.org/10.1109/ACCESS.2024.3403483.

[14] C. Shepherd, Zero trust architecture: Framework and case study, 2022.

[15] P. Tedeschi, et al., Satellite-based communications security: Threats, solutions, and challenges, Comp. Netw. 216 (2022).

[16] M.S. Hwang, et al., An authentication scheme for mobile satellite communication systems, ACM Oper. Syst. Rev. 37 (4) (2003) 42–47.

[17] T.Z. Chen, et al., A self-verification authentication mechanism for mobile satellite systems, Comp. Elect. Eng. 35 (1) (2009) 41–48.

[18] M. Qi, et al., A secure authentication with key agreement scheme using ECC for satellite communication systems, Int. J. Satell. Commun. Netw. 37 (2019) 234–244.

[19] Y. Zhong, J. Ma, A highly secure identity-based authenticated key-exchange protocol for satellite communication, J. Commun. Netw. 12 (6) (2010) 592–599.

[20] Z. Yantao, M. Jianfeng, Identity-based authenticated key-exchange protocol for satellite communication, J. Commun. Netw. 12 (2010) 592–599.

[21] O.A. Topal, G. Karabulut Kurt, Physical layer authentication for LEO satellite constellations, in: Proc. IEEE WCNC, 2022, pp. 1952–1957, http://dx.doi.org/10.1109/WCNC51071.2022.9771727.

[22] M. Abdrabou, T.A. Gulliver, Authentication for satellite communication systems using physical characteristics, IEEE Open J. Veh. Tech. 4 (2023) 48–60, http://dx.doi.org/10.1109/OJVT.2022.3218609.

[23] A. Murtaza, et al., Lightweight authentication and key sharing protocol for satellite communication, Int. J. Comp. Commun. Eng. 9 (2020) 46–53.

[24] C. Huang, et al., A mutual authentication and key update protocol in satellite communication network, Automatika 61 (3) (2020) 334–344.

[25] P. Zuo, et al., Intelligent encryption decision for multilayer satellite networks, Alex. Eng. J. 81 (2023) 337–346, http://dx.doi.org/10.1016/j.aej.2023.08.072.

[26] W. Lan, et al., Deep reinforcement learning for privacy-preserving task offloading in satellite networks, IEEE Trans. Mob. Comput. (2024) http://dx.doi.org/10.1109/TMC.2024.3366928.

[27] U. Kumar, M. Garg, Learning with error-based key agreement and authentication scheme for satellite communication, Int. J. Satell. Commun. Netw. 40 (2) (2022) 83–95, http://dx.doi.org/10.1002/sat.1417.

[28] Z. Yi, et al., Access authentication algorithm based on hierarchical identity-based signature over lattice for space-ground integrated networks, in: Proc. Int. Conf. Adv. Commun. Technol. Netw, 2019, pp. 1–9.

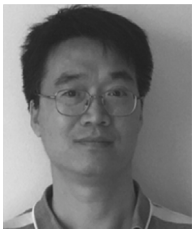
[29] E. Gilman, D. Barth, Zero Trust Networks: Building Secure Systems in Untrusted Networks, first ed., O'Reilly Media, 2017.
[30] P. Fu, et al., ZTEI: Zero-trust and edge intelligence empowered continuous authentication for satellite networks, in: Proc. IEEE GLOBECOM, 2022, pp. 2376–2381, http://dx.doi.org/10.1109/GLOBECOM48099.2022.10000958.
[31] Y. Liu, et al., Anonymous distributed key management for space information network, in: Proc. IEEE ICC, 2016, pp. 1–7, http://dx.doi.org/10.1109/ICC.2016.7510841.
[32] N.T. Nguyen, C.C. Chang, Biometric-based authenticated key agreement for mobile satellite networks, Wirel. Pers. Commun. 107 (2019) 1727–1758, http://dx.doi.org/10.1007/s11277-019-06354-6.
[33] A. Ostad-Sharif, et al., Efficient elliptic curve cryptography for three-factor authentication in satellite communications, Comp. Commun. 147 (2019) 85–97, http://dx.doi.org/10.1016/j.comcom.2019.08.018.
[34] Y. Chen, J. Chen, Robust three-factor authentication protocol for satellite communication systems, Int. J. Commun. Syst. 33 (15) (2020) 1–16, http://dx.doi.org/10.1002/dac.4508.
[35] A.A. Khan, et al., RAKS: Robust authentication and key agreement scheme for satellite infrastructure, Telecommun. Syst. (2022) 83–98.
[36] J. Guan, et al., BSLA: Blockchain-assisted secure lightweight authentication for SGIN, Comp. Commun. 176 (2021) 46–55, http://dx.doi.org/10.1016/j.comcom.2021.05.015.
[37] D. Selva, et al., Distributed earth satellite systems: Moving forward, J. Aerosp. Inform. Syst. 14 (8) (2017) 412–438.
[38] S. Fox, Tracking satellite navigation in Europe: Galileo 2020, Commun. Law 25 (4) (2020) 191–208.
[39] P. Yue, et al., LEO satellite communication systems: Vulnerabilities and countermeasures, IEEE Consum. Electron. Mag. 13 (1) (2024) 79–90, http://dx.doi.org/10.1109/MCE.2023.3262904.
[40] N. Chistousov, et al., Algorithms for increasing information secrecy in satellite communication, CEUR Work. Proc. 3094 (2022) 59–64.
[41] B. Yang, et al., AI-based two-phase multifactor authentication in SAGINs, IEEE Consum. Electron. Mag. 13 (1) (2024) 79–90, http://dx.doi.org/10.1109/MCE.2023.3262904.
[42] D. Bhattacherjee, et al., In-orbit computing: An outlandish thought experiment, in: Proc. ACM HotNets-XIX, 2020, pp. 197–204, http://dx.doi.org/10.1145/3422604.3425937.
[43] RFC 4949. Available: https://www.rfc-editor.org/info/rfc4949.
[44] C. Dimitriadis, Analyzing internet banking authentication mechanisms, IS Control. J. 3 (2007).
[45] I. Altaf, et al., Lightweight key agreement and authentication for satellite systems, IEEE Access 8 (2020) 46278–46287.
[46] Q. Do, et al., The role of the adversary model in applied security research, Comp. Secur. 81 (2019) 156–181, http://dx.doi.org/10.1016/j.cose.2018.12.002.
[47] S. Ullah, N. Din, Blind signcryption with hyperelliptic curves, Peer- To- Peer Netw. Appl. 14 (2) (2021) 1–16.
[48] B.S. Kaliski, C.K. Koç, C. Paar, J. Webster, Cryptographic Hardware and Embedded Systems—CHES 2002 4th International Workshop, 2002, pp. 13–15.
[49] N. Alimi, A. Sghaier, M. Machhout, R. Tourki, Exploring the design space of curve-based cryptographic accelerators, in: 2017 International Conference on Engineering & MIS, ICEMIS, Monastir, Tunisia, 2017, pp. 1–5, http://dx.doi.org/10.1109/ICEMIS.2017.8273045.
[50] G. Routis, et al., Enhancing privacy in internet of vehicles with hyperelliptic cryptography, Electronics 13 (730) (2024) http://dx.doi.org/10.3390/electronics13040730.
[51] J. Pelzl, et al., Hyperelliptic curve cryptosystems: Closing the performance gap to elliptic curves, in: CHES, vol. 2779, Springer Berlin Heidelberg, 2003, pp. 351–365.
[52] D. Evans, et al., OPS-SAT: An ESA nanosatellite for accelerating innovation in satellite control, in: Proc. SpaceOps 2016 Conf, 2016.
[53] C. Araguz, M. Marí, E. Bou-Balust, E. Alarcon, D. Selva, Design guidelines for general-purpose payload-oriented nanosatellite software architectures, J. Aerosp. Inf. Syst. 15 (3) (2018) 107–119.
[54] D. Jian-zhi, et al., Design of hyper elliptic curve digital signature, in: Proc. Int. Conf. Info. Tech. Comp. Sci, 2009, pp. 45–47, http://dx.doi.org/10.1109/ITCS.2009.146.
[55] T.S. Messerges, et al., Smart-card security under power analysis attacks, IEEE Trans. Comput. 51 (5) (2002) 541–552.
[56] K. Chatterjee, et al., Mutual authentication with hyperelliptic curve cryptosystem in constrained devices, Int. J. Netw. Secur. 15 (1) (2013) 9–15.
[57] C.J.F. Cremers, The scyther tool: Verification of security protocols, in: Proc. Int. Conf. Comp. Aided Verification, Springer, 2008, pp. 414–418.
[58] M. Burrows, et al., A logic of authentication, ACM Trans. Comput. Syst. 8 (1) (1990) 18–36.
[59] K. Xu, et al., A secure and efficient access and handover authentication protocol for the IoT in space information networks, IEEE Internet Things J. 6 (3) (2019) 5485–5499, http://dx.doi.org/10.1109/JIOT.2019.2902907.
[60] Q. Kong, et al., Efficient and secure handover in LEO constellation-assisted beyond 5G networks, IEEE Open J. Commun. Soc. 3 (2022) 641–653, http://dx.doi.org/10.1109/OJCOMS.2021.3139462.
[61] Q. Yang, et al., AnFRA: Anonymous and fast roaming authentication for space information networks, IEEE Trans. Inf. Forensics Secur. 14 (2) (2019) 486–497, http://dx.doi.org/10.1109/TIFS.2018.2854740.
[62] Y. Liu, et al., A secure and efficient authentication protocol for satellite-terrestrial networks, IEEE Internet Things J. 10 (7) (2023) 5810–5822, http://dx.doi.org/10.1109/JIOT.2022.3152900.
[63] J. Guo, Y. Du, Secure three-factor anonymous roaming authentication using ECC for space networks, Peer- To- Peer Netw. Appl. 14 (2) (2021) 898–916.
[64] AGI, Satellite Classical Coordinate Types, Available: https://help.agi.com/stk/index.htm#stk/vehSat_coordType_classical.htm, (Accessed 06 October 2024).
[65] ADDVulcan, Hack-a-sat 2021: Guardians of the linky writeup, GitHub repository, 2021, [Online]. Available: https://github.com/ADDVulcan/Writeups/tree/main/Hack-A-Sat. [Accessed 6 April 2024].
[66] C. Zhou, et al., Certificateless key-insulated encryption without bilinear pairings, Secur. Commun. Netw. 2017 (2017) 1–17, http://dx.doi.org/10.1155/2017/8405879.
[67] A.D.E. Berini, et al., HCALA: Hyperelliptic curve-based anonymous lightweight authentication for internet of drones, Pervasive Mob. Comput. 92 (2023) 1–10, http://dx.doi.org/10.1016/j.pmcj.2023.101798.
[68] J. Pan, X. Hu, S. Zhou, C. Tang, R. Guo, L. Zhu, G. Tang, G. Hu, Time synchronization of new-generation BDS satellites using inter-satellite link measurements, Adv. Space Res. 61 (1) (2018) 145–153.
[69] T.M. Tirmizi, N. Kato, A. Jamalipour, IP traffic load distribution in NGEO broadband satellite networks, Comput. Netw. 45 (3) (2022) 235–250.
[70] S. Geng, S. Liu, Z. Fang, Resilient communication model for satellite networks using clustering technique, Reliab. Eng. Syst. Saf. 215 (2021) 107850, http://dx.doi.org/10.1016/j.ress.2021.107850.
[71] W. Lan, K. Chen, Y. Li, J. Cao, Y. Sahni, Deep reinforcement learning for privacy-preserving task offloading in integrated satellite-terrestrial networks, IEEE Trans. Mob. Comput. 23 (10) (2024) 9678–9691, http://dx.doi.org/10.1109/TMC.2024.3366928.
[72] R. Suzuki, Y. Yasuda, Study on ISL network structure in LEO satellite communication systems, Acta Astronaut. 61 (7) (2007) 648–658, http://dx.doi.org/10.1016/j.actaastro.2006.11.015.

Kerry Anne Farrea holds a Master of Science in Homeland Security and Cybersecurity. She has worked as an Information Security Engineer contracting with the U.S. Department of Defense (DoD), where she gained experience in network design and system configurations for risk detection and cyber threat mitigation. She is a Cyber Security Cooperative Research Centre (CSCRC) Scholarship recipient and is currently pursuing a Ph.D. at Deakin University, Victoria, Australia. Her research focuses on defining and validating a novel mutual authentication scheme for space-based networks. Her interests include Security in Space-Terrestrial Integrated Networks and Lightweight Dynamic Authentication Methods.

Zubair Baig (Senior Member, IEEE) is currently an Associate Professor at Deakin University's School of Information Technology and also the Head of Research Translation (Cyber Security). He has authored over 110 journal articles, conference papers, and book chapters, and five white papers. He is also the Inventor of two Cyber Security Technologies granted patents by the U.S. Patent and Trade Office. He is currently serving as an Editor for three international journals: IET Wireless Sensor Systems (the Institute of Engineering and Technology in partnership with Wiley), PSU Research Review (Prince Sultan University in partnership with Emerald Publishing), and the Journal of Information and Telecommunication (Taylor & Francis). He has served on the technical program committees of numerous international conferences and has delivered over 30 keynote talks on cyber security. His research interests include cyber security, artificial intelligence, critical infrastructure (CI), and the IoT. He has a broad risk assessment skill set which extends to the IoT, CI, and other sensor network contexts.

Robin Ram Mohan Doss (Senior Member, IEEE) is currently the Director of the Centre for Cyber Resilience and Trust (CREST), Deakin University. He also leads the "Development of next-generation authentication technologies" theme within the National Cyber Security Cooperative Research Centre (CSCRC). He has an extensive research publication portfolio. His research interests include system security, protocol design, and security analysis, with a

focus on smart, cyber–physical, and critical infrastructures. His research program has been funded by the Australian Research Council (ARC) and government agencies, such as the Defence Signals Directorate (DSD), the Department of Industry, Innovation and Science (DIIS), and industry partners. He is a member of the Research Council of the Oceania Cyber Security Centre (OCSC) and the Executive Council of the IoT Alliance Australia (IoTAA). He was a recipient of the Cyber Security Researcher of the Year Award from the Australian Information Security Association (AISA) in 2019.



Donxi Liu is a senior research scientist at the Commonwealth Scientific and Industrial Research Organisation (CSIRO) since 2008. His research interests include applied cryptography, lightweight encryption, and system security. His recent work aims to design public key encryption based on checkable hardness facts and design new proof-of-work blockchain protocol for crowdmining.