Space Network Attack and Defense Simulation Platform and Its Application in Space NDN Networks

Hui Qi[1], Yu Zhao[1], Xu Liu[1], Jinhui Cao[1], Yu Wang[1], Pei Xiao[2], Pengfei Hu[3], and Chunbo Wang[1]
[1]School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
[2]Institute for Communication Systems, University of Surrey, Guildford, UK
[3]School of Computer Science and Technology, Shandong University, Qingdao, Shandong, 266237, China

With the rapid development of space networks, their high dynamics and open links pose severe challenges to network security management. Traditional network simulation platforms can hardly meet the needs of space network attack-defense simulation and visualization, especially in supporting highly dynamic topologies and verifying data integrity in deep space environments. This paper proposes a modular space network attack-defense simulation and visualization platform, which consists of four modules: satellite motion simulation, network simulation, network attack-defense, and visualization, thereby enabling dynamic network topology simulation and attack-defense scenario verification. Based on this platform, an efficient integrity verification method for cached network data is proposed to address the problem of data corruption in caches caused by cosmic rays in deep space environments. By integrating a dual-signature audit mechanism and a Content Store Naming Audit Tree (CSNAT), the approach ensures the reliability of cached data and low-latency access. Experimental results show that in a simulated Earth-Moon communication scenario, the proposed scheme can reduce the average end-to-end delay by about 83.6%, while maintaining low computational and storage overhead. This research provides efficient and reliable technical support for space network security verification and the application of NDN in deep space communications, with significant theoretical and practical value.

***Index Terms*—Space Network, NDN Network, Data Security, Integrity Verification.**

I. Introduction

SPACE networks, as an extension of terrestrial Internet, demonstrate great potential in global communication, deep space exploration, and integrated space–ground networking due to their wide coverage and dynamic characteristics [1]. However, the open links, complex topology, and high dynamics of space networks pose severe security challenges [2]. Extreme environmental factors such as cosmic rays may cause cache data corruption, while the high latency of long-distance communication further amplifies the cost of data retransmission [3]. Traditional connection-oriented network architectures struggle to address these challenges, whereas Named Data Networking (NDN), as a content-centric architecture, can effectively reduce data retrieval latency through distributed caching and content naming mechanisms, making it a promising solution for deep space communication. Nevertheless, cached NDN data in deep space environments is prone to bit-flip errors caused by radiation, leading to integrity damage and triggering costly end-to-end retransmissions, which offset the advantages of caching [4] [5] [6].

Existing network simulation platforms, such as NS3 [7] and Mininet [8], have limitations in simulating dynamic topologies of space networks and supporting attack-defense scenarios, making them ineffective in verifying the security and reliability of NDN in deep space environments [9] [10]. To address this, this paper proposes a modular space network attack-defense simulation and visualization platform. Through the collaborative operation of four subsystems—satellite mobility simulation, network simulation, attack-defense, and visualization—the platform achieves simulation and security verification of highly dynamic space networks. The platform supports NDN deployment and, targeting the cache integrity problem in deep space NDN networks, introduces an efficient verification scheme that combines a dual-signature audit mechanism and a CSNAT(A multi-layer tree structure for quickly verifying cached data and locating damages, with high efficiency and adaptability to dynamic NDN naming). This approach ensures data reliability with low computation and storage overhead, significantly reducing retransmission delays.

The main contributions of this paper include:

1) Designing a highly cohesive and loosely coupled space network attack-defense simulation and visualization platform that supports dynamic topology and attack-defense scenario visualization;
2) Proposing a cache data integrity verification scheme for deep space NDN networks, improving verification efficiency through the CSNAT structure;
3) Verifying the effectiveness of the platform and scheme through Earth–Moon communication ex-

Manuscript received June 16, 2025; revised September 18, 2025. Corresponding author: Chunbo Wang (email: wangcb1213@126.com).

periments, with results showing an average end-to-end delay reduction of approximately 83.6%.

The rest of this paper is organized as follows: Section II reviews related work; Section III introduces the platform architecture; Section IV presents the NDN cache verification scheme; Section V provides experimental simulations; Section VI concludes the paper.

II. Related Work

The high dynamics and openness of space networks pose unique challenges in terms of communication efficiency and security verification. Existing research mainly focuses on network simulation, attack-defense simulation platforms, and the application of Named Data Networking (NDN) in space networks.

Network simulation technology: Traditional network simulation tools such as NS3, OMNeT++, and Mininet are widely used for terrestrial network protocol verification, but they are insufficient in space networks. NS3 supports protocol development through a discrete event model, but additional extensions are required to simulate highly dynamic topologies and satellite movement [11]. OMNeT++ provides good extensibility with its modular design and supports wireless communication simulation, but it has limited support for attack-defense scenarios in space networks [12]. Mininet, based on the real network protocol stack, is suitable for static topology simulation, but it cannot easily adapt to the dynamics and complexity of space networks [13]. Some studies have attempted to combine NS2 and STK to simulate multi-layer satellite networks, verifying the throughput advantages of DTN protocols in long-delay environments [14], or to use OPNET to evaluate QoS routing performance in satellite networks [15]. However, these approaches mainly focus on protocol performance and lack integrated support for security attack-defense scenarios, making them insufficient for the dynamic simulation needs of deep space networks.

Attack-defense simulation platforms: Cyber ranges provide important tools for attack-defense verification. The U.S. National Cyber Range (NCR) and the U.K. Federal Cyber Range (FCR) support attack-defense exercises in complex network scenarios, but they mainly target terrestrial networks and lack dedicated support for space networks [16] [17]. In China, the National Big Data Security Integrated Cyber Range combines virtual and real environments to verify network security technologies, but it does not address the dynamics and radiation environment of space networks [18]. Research on attack-defense simulation specifically for space network security remains limited. Existing solutions mostly focus on terrestrial attack models, such as DDoS detection [19], which are difficult to directly apply in highly dynamic, resource-constrained space environments.

NDN in space networks: Named Data Networking (NDN), due to its content-driven and distributed caching features, is regarded as an ideal architecture for deep space communications [20]. Studies have shown that NDN reduces data retrieval latency through in-network caching, making it suitable for high-latency and unstable deep space links [21] [22]. However, single-event upsets caused by space radiation may corrupt cached data, leading to increased retransmission delays [23] [24]. Existing NDN integrity verification schemes are mostly based on cloud storage environments, such as PDP and PoR [25] [26], or use Merkle trees and blockchain to optimize auditing efficiency [27] [28]. Yet, these schemes do not consider the resource constraints and radiation effects in deep space, and their verification overhead is high, making them impractical for direct deployment.

In summary, this paper constructs a modular attack-defense simulation platform and proposes an NDN cache integrity verification scheme that incorporates an efficient auditing tree structure, filling this gap and providing a new approach for space network security verification.

III. Platform Architecture Design

This paper proposes a modular space network attack-defense simulation and visualization platform, aiming to support the simulation and security verification of highly dynamic space networks. The platform consists of four subsystems: the satellite motion simulation subsystem, the satellite network simulation subsystem, the attack-defense subsystem, and the visualization subsystem. These subsystems collaborate through a high-cohesion and low-coupling design, ensuring the scalability and flexibility of the platform. The platform can efficiently simulate the dynamic topology of space networks and support the verification of attack-defense scenarios. The following sections provide a detailed description of the functions and interaction processes of each subsystem.

A. Architectural Design

Satellite Motion Simulation Subsystem: Simulate the dynamic motion characteristics of nodes in a space network, with inputs including satellite orbit parameters (such as orbit inclination, ascending node right ascension, orbit eccentricity, perigee angle, etc.) and ground node position parameters. Based on these parameters, calculate the coordinates of nodes in space and generate a distance matrix sequence of time series. The time interval is seconds, and the matrix elements represent the straight-line distance between nodes. Calculate the visibility matrix sequence through the distance matrix to verify the communication reachability between nodes. These matrix sequences provide a dynamic topological foundation for subsequent network simulations.

Satellite Network Simulation Subsystem: Construct a virtual space network based on the distance

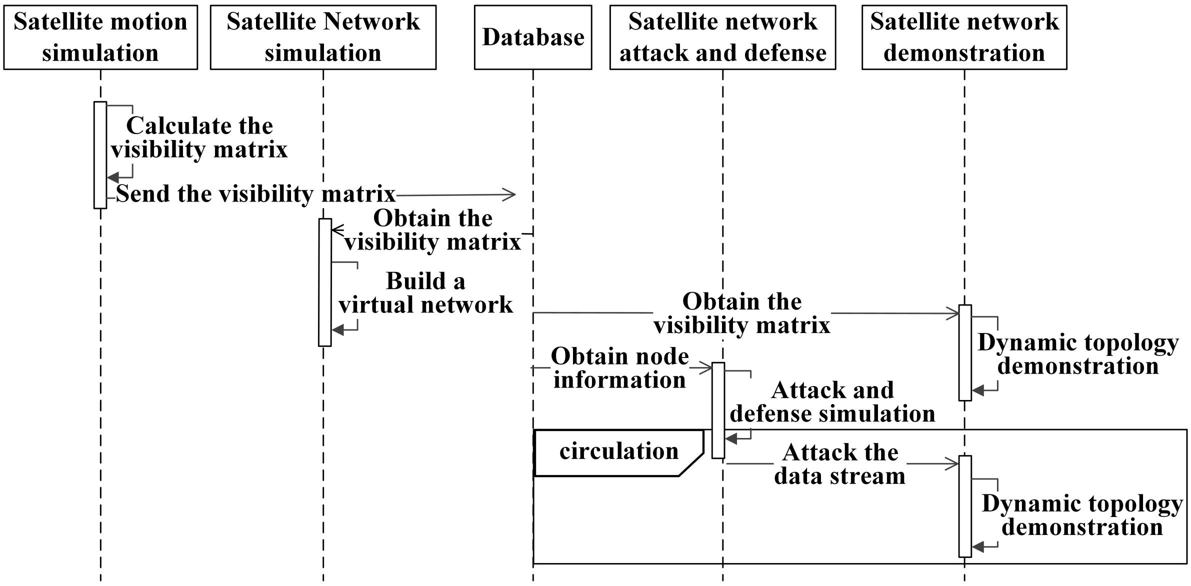
Satellite motion simulation
Satellite Network simulation
Database
Satellite network attack and defense
Satellite network demonstration
Calculate the visibility matrix
Send the visibility matrix
Obtain the visibility matrix
Build a virtual network
Obtain the visibility matrix
Dynamic topology demonstration
Obtain node information
Attack and defense simulation
circulation
Attack the data stream
Dynamic topology demonstration

Fig. 1. Interaction between Subsystems of the Space Network Attack and Defense Platform, detailing the flow of the Visibility Matrix and the Dynamic Topology Demonstration

matrix sequence and visibility matrix sequence provided by the satellite motion simulation subsystem. The input also includes configuration parameters of network nodes, such as node type (switching node or application node), hardware configuration, and the running container or virtual machine image. The subsystem determines the number and connection relationships of network nodes based on the matrix sequence, implements node representation through containers or virtual switches, and dynamically adjusts the links between nodes to reflect topology changes. Output as a virtual space network, including network parameters such as node network card names and addresses, supporting application running and dynamic link management.

Attack-Defense Subsystem: Build attack-defense scenarios using virtual networks generated by satellite network simulation subsystems. The input is network parameters, and the output is a description of the attack-defense scenario, including the attack source, target, and attack behavior. The subsystem supports deploying attack or defense mirrors on selected nodes, simulating secure interactions (such as data tampering, integrity verification), and generating attack path information (including attack source, target, and data flow attributes) at time intervals. This subsystem provides a flexible experimental environment for space network security verification, especially suitable for testing NDN cache integrity and other scenarios.

Visualization Subsystem: Dynamically display spatial network topology and attack-defense processes. The input includes distance matrix sequence, visibility matrix sequence, network parameters, and attack path information generated by the attack-defense subsystem. Output as a 3D network topology demonstration and visualization of attack-defense behavior. By analyzing the visibility matrix sequence, the subsystem displays the motion trajectories and dynamic connection relationships of satellites and ground nodes in three-dimensional form; Based on attack path information, real-time drawing of attack paths and data streams enhances users' intuitive understanding of the attack-defense process.

This platform adopts a modular architecture with high cohesion and low coupling, consisting of four interdependent subsystems. Each subsystem works together through standardized data interfaces. The satellite motion simulation subsystem serves as a data source and generates distance and visibility matrices for time series by calculating orbital dynamics. These matrices are passed to the satellite network simulation subsystem, which serves as the infrastructure layer and constructs a dynamic virtual network using containers and virtual switches. On the basis of this virtual environment, the satellite network attack and defense subsystem executes security scenarios and generates real-time attack path data. Finally, the visualization demonstration subsystem integrates the matrix and attack logs, providing three-dimensional visualization of satellite trajectories, dynamic topology changes, and attack and defense processes. This logical process ensures a smooth transition from physical motion modeling to high fidelity network security verification.

B. Attack-defense demonstration plan

The attack-defense demonstration scheme of the platform aims to showcase the operational state and security verification process of space networks through dynamic network simulation and visualization technologies. Based on satellite network simulation, com-

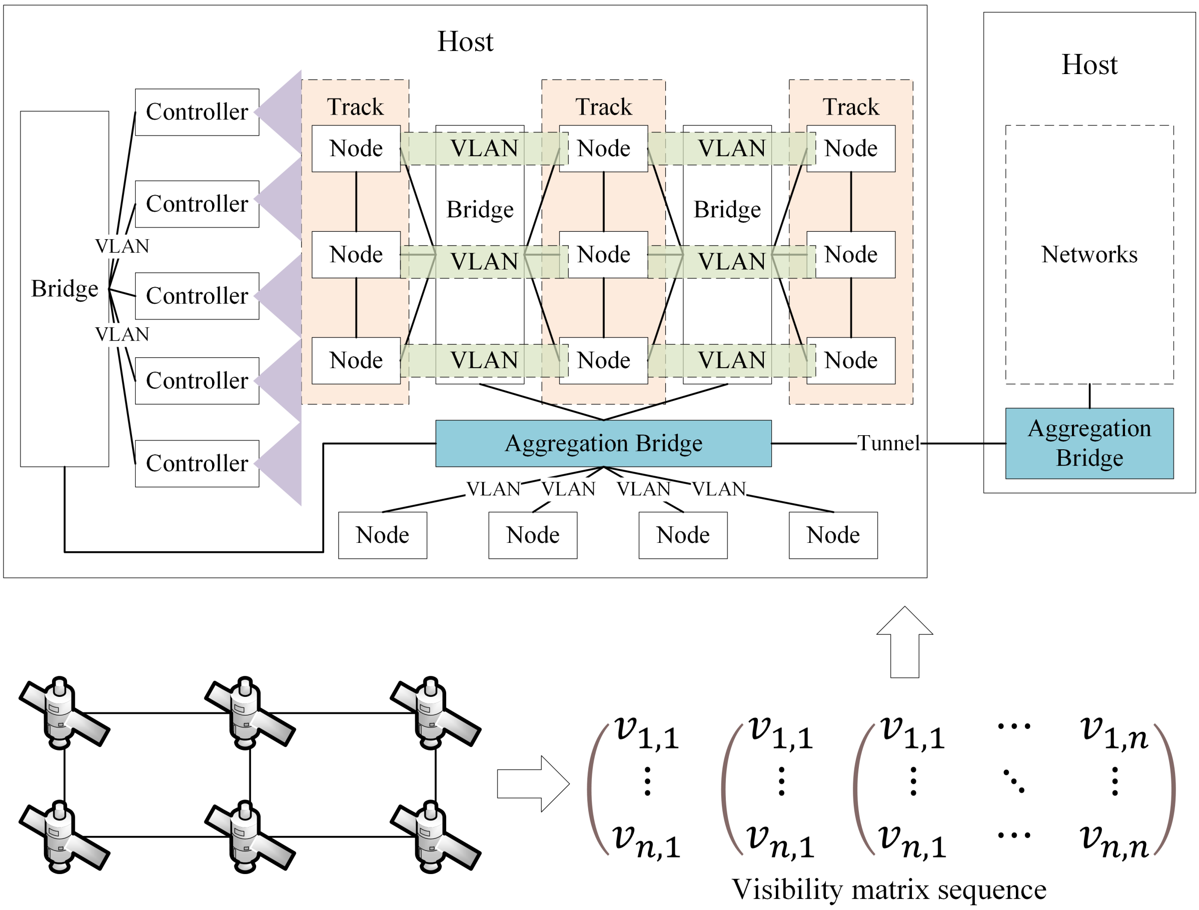

Fig. 2. Satellite Network Simulation Framework and the Corresponding Visibility Matrix Sequence ($v_{i,j}$ indicating inter-staellite visibility at each time step)

bined with the attack-defense subsystem and the visualization subsystem, the scheme realizes the simulation of dynamic topologies and the intuitive presentation of attack-defense scenarios, thereby supporting space network security verification such as NDN cache integrity verification. The core processes of the attack-defense demonstration are described as follows.

Satellite Network Simulation: The first stage of the demonstration is the construction of a dynamic virtual space network. The satellite motion simulation subsystem, based on satellite orbital parameters and ground node positions, calculates the sequence of distance matrices and visibility matrices with second-level intervals, reflecting the dynamic connectivity between nodes. The satellite network simulation subsystem reads these sequences, parses the network topology, and determines the number and type of nodes (satellite or ground). To simulate a highly dynamic space network, the subsystem employs virtual bridges and VLAN technology to manage links between nodes: satellite nodes within the same orbital plane form a ring connection, inter-plane connections are established via virtual bridges, and ground nodes communicate with satellites through aggregation bridges. VLAN technology supports the dynamic switching of links, simulating inter-satellite and satellite-ground handovers without the need for frequent creation or deletion of connections, thereby reducing computational overhead. Figure 2 illustrates the principle of satellite network simulation.

Attack-Defense Scenario Demonstration: The attack-defense subsystem deploys attack-defense scenarios on the virtual network. For example, in the NDN cache integrity verification scenario, the subsystem can deploy NDN Content Store (CS), producers, and auditors on selected nodes to simulate bit-flip effects caused by cosmic rays. By configuring attack images (e.g., simulating data tampering) or defense images (e.g., integrity verification algorithms), the subsystem generates attack path information, including attack source, target, and data flow attributes. This information is transmitted in real time to the visualization subsystem. The visualization subsystem utilizes distance and visibility matrix sequences to dynamically draw the three-dimensional network topology, showing the motion trajectories of satellites and ground nodes as well as link changes; at the same time, it renders attack behaviors or defense effects in real time, intuitively presenting the attack-defense interaction process.

This demonstration scheme achieves efficient sim-

TABLE I
COMPARISON OF EXISTING SIMULATION PLATFORMS BASED ON DYNAMIC NETWORK CAPABILITIES, ATTACK-DEFENSE SUPPORT, VISUALIZATION, SPACE-NETWORK SUITABILITY AND EXTENSIBILITY

Platform	Dynamic Network Support	Attack-Defense Integration	Visualization Support	Space Network Applicability	Extensibility
NS3	Medium (discrete event model, requires dynamic extensions)	Low (no dedicated attack-defense modules)	Medium (built-in analysis tools, requires 3D extensions)	Medium (requires satellite modules and dynamic extensions)	High (open source, modular extension)
OMNeT++	High (modular design, supports dynamic extension)	Medium (limited support for attack-defense scenarios)	High (supports graphical user interface)	Medium (supports inter-satellite links, requires customization for space scenarios)	High (modular, easy to extend)
Mininet	Low (mainly static topology extension)	Medium (supports basic security testing)	Low (no dynamic visualization)	Low (focused on terrestrial networks, insufficient for dynamics)	Medium (relies on real protocol stack)
OPNET	High (commercial tool, supports dynamic simulation)	Medium (requires customized attack-defense modules)	High (supports wireless and satellite visualization)	High (built-in satellite modules)	Medium (commercial software, limited extensibility)
Proposed Platform	High (matrix sequence-based dynamic update)	High (dedicated attack-defense subsystem)	High (3D topology and attack-defense visualization)	High (dedicated to space network design)	High (modular, easy integration)

ulation and visualization through modular design. The satellite network simulation subsystem supports dynamic topology adjustment, adapting to the high dynamics of space networks; the attack-defense subsystem provides flexible scenario configuration, supporting various security verification experiments; the visualization subsystem enhances users' understanding of the network state and the attack-defense process through three-dimensional representation. Compared with traditional static simulation, this scheme significantly reduces link management overhead by leveraging dynamic matrix sequences and VLAN technology. It provides a simulation and verification environment for NDN integrity verification and offers an efficient tool for space network security research.

C. Comparison with existing network simulation platforms

To evaluate the advantages of the proposed space network attack-defense simulation and visualization platform, this paper compares the characteristics of existing mainstream network simulation platforms (NS3, OMNeT++, Mininet, and OPNET), focusing on five key dimensions: dynamic network support, attack-defense integration, visualization support, applicability to space networks, and extensibility.

NS3 and OMNeT++ perform well in protocol development and modular design, but their support for highly dynamic space network topologies and attack-defense scenarios requires additional customization, limiting their direct applicability. Mininet, relying on a real protocol stack, is suitable for terrestrial network testing but is insufficient in supporting dynamics and space scenarios. OPNET, as a commercial tool, provides strong satellite simulation and visualization functions, but its extensibility is limited and its cost is high. In contrast, the proposed platform efficiently supports dynamic topologies through the matrix sequence mechanism of the satellite motion simulation and network simulation subsystems. The attack-defense subsystem is specifically designed for security verification, supporting complex scenarios such as NDN cache integrity verification. The visualization subsystem offers three-dimensional dynamic presentations, enhancing user comprehension. The modular design ensures high extensibility, facilitating the integration of new functions or technologies, and is particularly well suited for the high dynamics and security verification requirements of space networks.

IV. NDN CACHE DATA INTEGRITY VERIFICATION SCHEME

A. System Model

Space Named Data Networking (NDN) is a content-centric network architecture. In deep space networks, NDN enables communication through Interest Packets and Data Packets: consumers send Interest Packets to request content with specific names, and network nodes respond with Data Packets according to their caching and forwarding strategies. Figure 2 illustrates the architecture of a deep space NDN network, where the CS, Forwarding Information Base (FIB), and Pending Interest Table (PIT) are the core components. The CS caches Data Packets that have passed through a node to satisfy subsequent identical requests; the FIB is used for routing decisions; and the PIT records outstanding Interest Packets, ensuring that Data Packets return along the reverse path.

However, the strong cosmic radiation in deep space may induce Single-Event Upsets (SEU), causing bit flips in cached data or signatures and thus threatening data integrity. Once a consumer receives corrupted Data Packets, signature verification fails, triggering a re-fetch from the original producer (e.g., a lunar probe) and leading to significant end-to-end delay. To address this challenge, this paper deploys NDN nodes on the space network attack-defense simulation and visualization platform, simulating the deep space radiation environment to validate cache data integrity schemes. The details are as follows.

The platform constructs a virtual NDN network through the satellite network simulation subsystem, simulating CS, producer, consumer, and auditor nodes. Nodes are deployed as containers in the virtual network, each container running NDN-related software (e.g., the NDN Forwarding Daemon), thereby forming a complete virtual NDN network. Producer containers generate Data Packets with dual signatures (content signature and audit tag), which are cached in CS containers. Auditor containers are deployed on selected satellite or ground nodes, responsible for building CSNAT and performing integrity verification. The attack-defense subsystem simulates cosmic-ray-induced bit flips by configuring attack images, generating corrupted data or signatures to test the robustness of the verification scheme.

The platform leverages the attack-defense subsystem to emulate the deep space radiation environment by randomly introducing bit flips in CS containers, simulating SEU. For instance, based on the error rates in GEO orbits reported in [29] (proton SEU $\approx 1.51 \times 10^{-7}$, heavy-ion SEU $\approx 4.21 \times 10^{-8}$), the subsystem modifies cached data or signatures according to predefined probabilities, generating attack path information. These records include the location and type of corrupted data, which are then used by the auditor for verification and localization.

The visualization subsystem presents the operation of the NDN network and the verification process in real time. It renders the dynamic topology of NDN nodes in three dimensions, reflecting changes in inter-satellite and satellite-ground links. Based on attack path information generated by the attack-defense subsystem, it dynamically illustrates the locations of bit flips (e.g., corrupted CS nodes) and the verification results (e.g., successful or failed audit paths), thereby providing users with an intuitive understanding of the effectiveness of the verification scheme.

By deploying NDN nodes as containers, the platform supports dynamic network simulation and ensures that verification schemes are feasible in realistic deep space scenarios. The attack-defense subsystem flexibly simulates radiation interference, validating the robustness of the dual-signature and CSNAT mechanisms, while the visualization subsystem enhances the interpretability of the verification process. The modular design of the platform further enables its extension to other NDN security scenarios, offering a powerful tool for deep space communication research.

B. Validation Plan

To address the problem of cache data corruption caused by cosmic rays in deep space environments, this paper proposes an efficient NDN cache data integrity verification scheme. By combining a dual-signature auditing mechanism and the CSNAT, the scheme ensures reliable and low-latency access to cached data. The proposed method is implemented on the space network attack-defense simulation and visualization platform, where its effectiveness is validated by simulating NDN nodes and radiation environments.

Dual-Signature Auditing Mechanism: The dual signature audit mechanism we propose enhances data integrity and non repudiation by requiring two different encrypted signatures for critical operations. This mechanism generates two types of signatures for each NDN packet: content signatures and audit tags. Content signatures are generated by producers based on data blocks to ensure the authenticity and tamper resistance of the data source; Audit tags are used to quickly verify the integrity of cached data and reduce verification overhead. After the data packet is generated, it is distributed through the NDN network and cached in the Content Storage (CS). Auditors regularly challenge the cached data to provide storage proof, and CS verifies its integrity upon receiving the proof. If packet damage is detected, the auditor triggers early retransmission to avoid high latency caused by failed consumer requests and ensure the integrity of cached data in NDN.

DataGen$(ContentName, sk, pp) \rightarrow (DataPacket)$: This algorithm takes as input the private key sk, the content name $(ContentName)$, and the public parameters pp, and outputs the Data Packet $(DataPacket)$ corresponding to the given content name.

DataGen$(ContentName, sk, pp) \rightarrow (DataPacket)$: This algorithm takes as input the private key sk, the content name $(ContentName)$, and the public parameters pp, and outputs the Data Packet $(DataPacket)$ corresponding to the given content name. First, the producer splits the original data M into multiple data blocks $F = (M_1, M_2, \ldots, M_n)$, where each block $M_i \in \mathbb{Z}_q^*$, $i \in [1, n]$, is assigned a unique content name $ContentName = (ContentName_1, ContentName_2, \ldots, ContentName_n)$. Next, the producer uses the signing key ssk to compute the digital signature of each data block M_i: $sig_i = Sign_{ECDSA}(ssk, M_i)$, where sig_i ensures the integrity and authenticity of the data packet content, preventing unauthorized tampering or forgery. Subsequently, the producer computes the audit tag σ_i based on the data block M_i, $\ldots$

The audit tag is computed as follows:

$$\sigma_i = \alpha \cdot (H_1(ContentName_i \,||\, i) + M_i \cdot g_1) \quad (1)$$

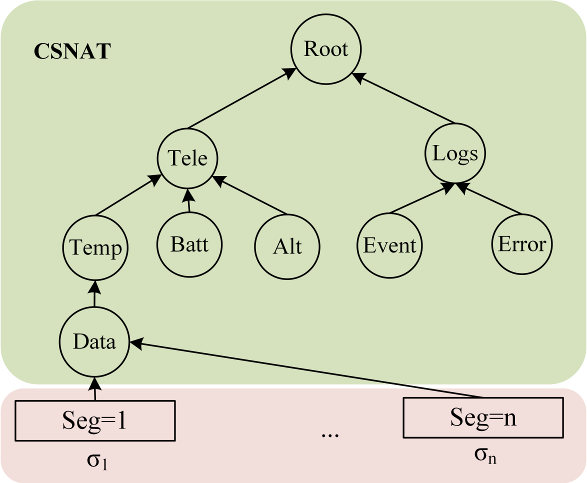

Fig. 3. Audit tree based on content storage naming

This tag is used for the integrity verification of cached data in NDN nodes, improving verification efficiency. Finally, the audit tag σ_i, together with the data block M_i and its signature sig_i, is encapsulated into the NDN Data Packet:

$$DataPacket_i = (ContentName_i, M_i, sig_i, \sigma_i, metadata) \tag{2}$$

where $metadata$ includes additional information of the NDN Data Packet, such as timestamp, freshness, and priority. The producer publishes Data Packets through the NDN network, where each packet is indexed by its content name and cached in the CS of NDN nodes.

CSNAT Audit Tree: To efficiently manage and validate cached data, a Content Storage Named Audit Tree (CSNAT) was designed that combines NDN naming structure and data integrity verification requirements, as shown in Figure 3. CSNAT adopts a multi fork hierarchical structure, organizing nodes with NDN content naming prefixes (e.g., `/Root/Tele/Temp/Data`). Leaf nodes store audit labels, which are generated by producers based on their data blocks and signatures. The parent node generates values through aggregation of child node labels and supports recursive verification. When a new data packet is cached in CS, the auditor extracts its content name and label, inserts CSNAT along the path from the leaf node to the root node, and updates the ancestor node value. The original value of the parent node is added to the current inserted label. CSNAT supports fast localization of damaged data, optimizes verification efficiency, and adapts to the dynamic nature of the NDN naming system.

During the construction of the tree structure, whenever a cache node receives a new Data Packet and its corresponding tag, the auditor extracts the content name and tag of the packet and inserts them into the audit tree.

$$Parent.Value \leftarrow Parent.Value + \sigma_i \tag{3}$$

The audit tree can maintain the stability of the tree structure while enabling the aggregation of tags from lower-level data by upper-level nodes. This facilitates rapid judgment during the auditing phase on whether data corruption has occurred and allows precise localization of the corrupted data item. Moreover, it adapts to the dynamic growth of the NDN namespace, thereby achieving auditing and integrity management for cached NDN content.

Verification Process: As shown in Figure 4, the verification process consists of the following steps: 1) The producer generates Data Packets with dual signatures and caches them in the NDN node's CS. 2) The auditor constructs the CSNAT and periodically initiates random challenges to the CS, requesting storage proofs. 3) The CS generates proofs based on the challenges, including aggregated signatures. 4) The auditor verifies whether the proofs match the tags in the CSNAT; if verification fails, the auditor recursively locates the corrupted data and triggers retransmission.

This scheme ensures the immediate availability of cached data and significantly reduces end-to-end latency by performing efficient auditing and early retransmission.

The detailed method is as follows:

ChallengeGen$(ContentName) \rightarrow (Q)$: The auditor randomly generates a challenge set. Specifically, it randomly selects c name prefixes $(Prefix)$, and for each prefix generates a random number $\eta \in \mathbb{Z}_q^*$, thereby forming the challenge set Q.

$$Q = \{(Prefix_1, \eta_1), (Prefix_2, \eta_2), \ldots, (Prefix_j, \eta_j)\} \tag{4}$$

Then, the auditor sends a challenge Q to the CS. **ProofGen**$(Q) \rightarrow (Proof)$: When the CS receives the challenge request from the auditor, it needs to compute the storage proof $Proof$ and return it to the auditor. The CS locates the cached content m according to each $ContentName$ in the challenge set, and then constructs the storage proof.

$$proof = \{(ContentName_1, m_1, sig_1, pk_1), (ContentName_2, m_2, sig_2, pk_2), \ldots, (ContentName_i, m_i, sig_i, pk_i)\} \tag{5}$$

Here, pk denotes the producer's public key of the Data Packet. Finally, the CS sends the storage proof $proof$ to the auditor.

Verify$(Proof, CSNAT.$prefix$) \rightarrow \{0, 1\}$: Upon receiving the storage proof, the auditor verifies it. If the verification succeeds, it indicates that the cached data and the signatures have not been corrupted, and the output is 1. If verification fails, the auditor recursively executes the verification algorithm to further locate

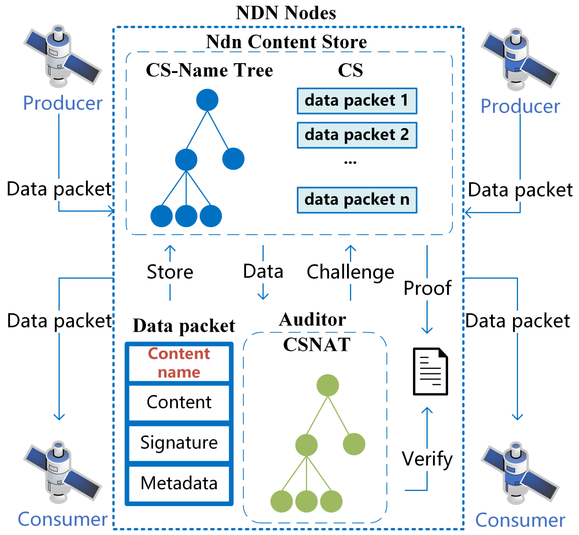

Fig. 4. Cache verification mechanism for deep space NDN nodes

the data packets that fail integrity verification and requests the CS to retransmit. The verification proceeds as follows: the auditor computes the aggregated tag $\sigma_{\text{agg}} = \sum_{j=1}^{c} \sigma_{\text{prefix}_j} \cdot \eta_j$. The auditor then checks whether the following equality holds (omitted here for brevity) and outputs the verification result accordingly.

$$e(\sigma_{\text{agg}}, g_2) = e\Bigg(\sum_{j=1}^{c} \sum_{i \in prefix_j} \eta_j \Big(H_1(ContentName_i \\ \|i) + m_i g_1\Big),\ \beta_i\Bigg) \tag{6}$$

If the verification succeeds, the data and signature are intact, and the output is 1. If the verification fails, the algorithm recursively executes $Verify(Proof, CSNAT.prefix.child)$ to verify the integrity of the child nodes of the challenged prefix in the audit tree, as well as the corresponding data and signatures, until the corrupted Data Packet m_j is precisely located. The auditor then re-verifies the located Data Packet using $Verify_{ECDSA}(spk_j, m_j, sig_j) = True$. If this verification succeeds, it is determined that the cached data has not been corrupted, and the output is 1. If the verification fails, it is determined that the cached content is corrupted, and the output is 0, after which the node is required to retransmit the Data Packet.

V. EXPERIMENT AND EVALUATION

To verify the effectiveness of the NDN cache data integrity verification scheme, this section conducts experimental evaluation by simulating the Earth Moon communication scenario, focusing on the dynamic simulation capability of the platform and the performance of the verification scheme in deep space radiation environment. Experiment using two open-source software to build NDN container images: ndnd (https://github.com/named-data/ndnd)Used to implement NDN forwarding daemon, supporting core NDN protocol functions; python-ndn (https://github.com/named-data/python-ndn)Provide Python interface for easy development and integration of validation algorithms.

A. Experimental setup

The experiment is based on the platform proposed in this article, simulating the communication scenario between the Earth and the Moon, including ground consumers, GEO orbit cache nodes, and lunar producers. The average distance between the moon and GEO nodes is about 384400 kilometers, with a one-way propagation delay of about 1284 milliseconds; The distance between the ground and GEO nodes is about 42460 kilometers, with a one-way delay of about 142 milliseconds. The cache node is affected by cosmic rays, and based on reference [29], the proton single particle flipping effect (SEU) error rate is set to about 1.51×10^{-7}, and the heavy ion SEU error rate is 4.21×10^{-8} . NDN nodes are deployed in container form, and the core image implements forwarding function based on NDND. Python NDN supports dual signature and CSNAT verification logic. The attack and defense subsystem randomly introduces bit flips to simulate data or signature corruption. Consumers initiate 1000 interest package requests per hour, and the experiment lasts for 10 hours. The average end-to-end latency, damage detection rate, retransmission data volume, and verification time cost are calculated.

B. Experimental results

End-to-End Latency: Figure 5 compares the average end-to-end latency between deployment validation schemes (including audit mechanisms) and those without audit mechanisms. Without an audit mechanism, the accumulation of cache data corruption results in interest packets needing to be traced back to the lunar producer. In the first hour, the average end-to-end delay is about 821 milliseconds. As time goes on, the damaged packets gradually accumulate. Once valid packets are not obtained, the interest packets will be transmitted to the producer and then retransmitted. Therefore, when the simulation reaches the 10th hour, the average end-to-end delay increases to 1587 milliseconds. This scheme actively detects and retransmits damaged data through CSNAT and dual signature auditing. Through this mechanism, the vast majority of interest packets can hit the cache and obtain valid data packets, resulting in an average end-to-end delay of 360-500 milliseconds in the 10 hour simulation, with an average reduction of about 83.6%. This indicates that the platform supports efficient verification and significantly reduces costly retransmissions.

Corruption Detection Efficiency: Figure 6 shows the detection performance of damaged entries under

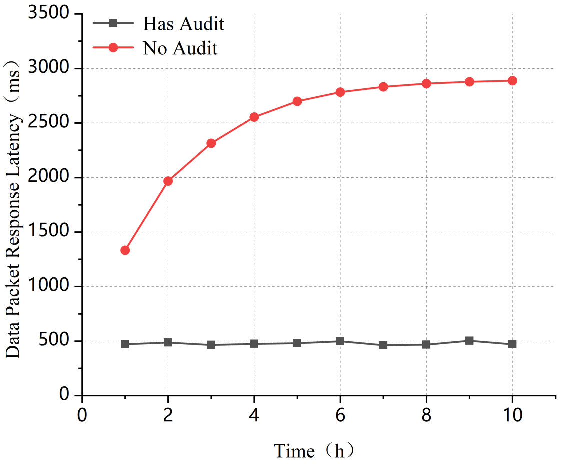

Fig. 5. Comparison trend of end-to-end transmission delay over time

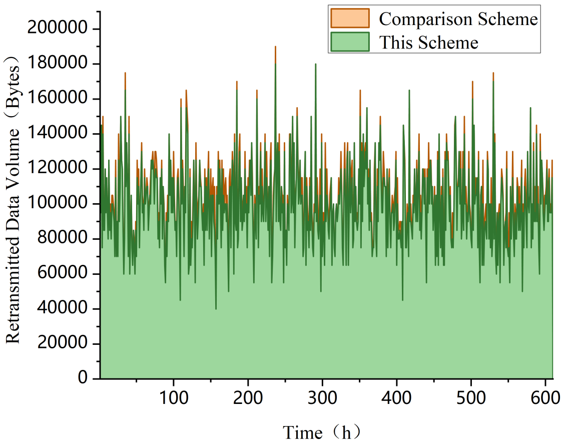

Fig. 7. Comparison of retransmission data volume between traditional single signature and the dual signature scheme proposed in this paper in deep space environment

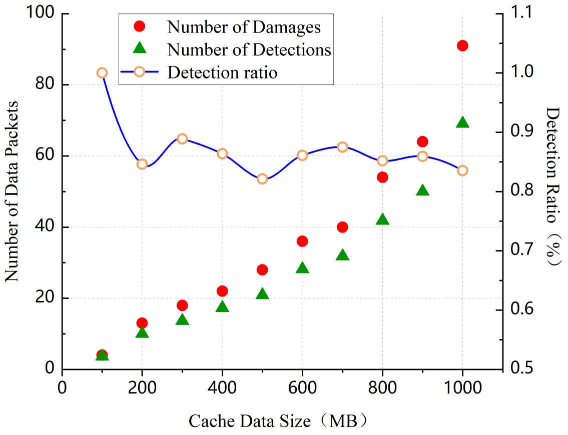

Fig. 6. Trend chart of damage detection with changes in cache size

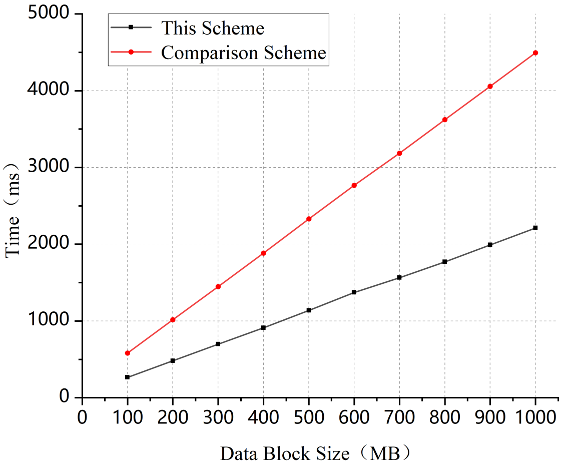

Fig. 8. Comparison of Verification Time Cost under Different Data Block Sizes

different cache sizes (100MB to 1000MB). As the cache size increases, the detection ratio gradually decreases. This is because the prefix space grows exponentially, and the auditor's random challenge mechanism cannot cover all prefixes, resulting in a gradual decrease in the detection ratio. However, as the number of damaged entries increases, the detection ratio stabilizes at 0.8-0.9. This indicates that the CSNAT structure, supported by this platform, can efficiently locate damaged data and maintain validation robustness when facing large-scale data.

Retransmission Data Volume: Figure 7 compares the retransmission data volume triggered within 200 hours between the traditional single signature scheme and this scheme. The dual signature audit scheme proposed in this scheme distinguishes between content and signature errors, retransmits only necessary data in the vast majority of time periods, and significantly reduces the amount of retransmitted data. This platform can also perform unnecessary retransmissions on erroneous data blocks, thereby significantly reducing communication overhead. From the simulation situation, the amount of retransmitted data is better than that of the single signature scheme.

Verification Time Overhead: Figure 8 shows the validation time of this scheme and the comparative scheme [29] under different data block sizes (100MB to 1000MB). As the size of the data block increases, the time of both methods shows a linear upward trend. This scheme benefits from the simplified design of pairing verification, which reduces the computational cost before pairing in the spatial network, significantly reduces the computational cost, lowers the frequency of pairing and computation, reduces resource pressure, and saves about 48% of time, adapting to the limited computing power environment of deep space nodes.

CSNAT Update Efficiency: Figure 9 compares the number of update nodes between CSNAT and traditional Merkle trees at different cache sizes. As the number of data packets increases, in traditional Merkle trees, each update requires recursive recalcu-

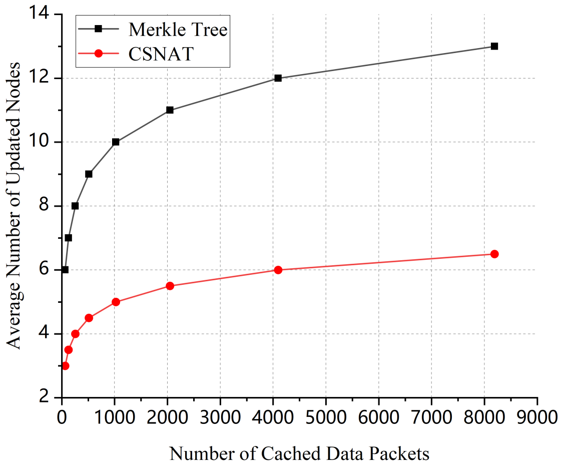

Fig. 9. Comparison of the average number of update nodes for different audit trees as cache size increases

lation of all hash values from the modified node to the root node, and the computational cost also increases accordingly. Compared with the CSNAT proposed in this paper, which has a smaller depth and uses named prefix indexing, the number of nodes updated each time is also reduced. The number of updated nodes is about half of the Merkle tree (about 6.5 vs. 13 for 8000 data points), and the update cost is reduced by about 50%, significantly reducing overhead.

Experimental results show that the platform, through dynamic topology simulation and the attack-defense subsystem, accurately simulates the radiation environment of deep space NDN networks and supports the efficient operation of the verification scheme. The integration of `ndnd` and `python-ndn` ensures rapid deployment of NDN nodes and flexible implementation of verification logic. The dual-signature and CSNAT mechanisms effectively guarantee the integrity of cached data, while early retransmission reduces latency. Optimization of verification time and update overhead makes the scheme well-suited for resource-constrained deep space nodes. The visualization subsystem intuitively presents the verification process, enhancing the interpretability of the scheme. Overall, the platform's dynamic support and modular design effectively validate the performance of the NDN scheme and provide a reliable tool for space network security research.

VI. Conclusion

This paper proposes a modular space network attack-defense simulation and visualization platform. Through the collaboration of four subsystems—satellite motion simulation, network simulation, attack-defense, and visualization—the platform achieves the simulation of highly dynamic space networks and the integrity verification of NDN cached data. It addresses the problem of cache data corruption caused by cosmic rays in deep space environments, significantly reducing end-to-end latency. Experimental results show that in simulated Earth–Moon communication scenarios, the proposed scheme reduces the average delay by approximately 83.6% while maintaining low computational and storage overhead, thus demonstrating the efficiency of the platform and the robustness of the scheme. Moreover, the visualization subsystem enhances the interpretability of the attack-defense process through three-dimensional dynamic presentations, providing intuitive support for security research in deep space NDN networks.

Looking forward, the platform can be further extended to other attack-defense scenarios, such as integrating artificial intelligence to optimize radiation environment simulation or supporting more sophisticated NDN security mechanisms (e.g., post-quantum signature algorithms). By incorporating more advanced dynamic topology algorithms and multi-scenario verification modules, the platform is expected to provide more comprehensive security verification support for integrated space–ground networks and deep space communications, thereby laying a solid foundation for the reliable operation of space networks.

Acknowledgment

This study is supported by 2025 Jilin Province Science and Technology Development Plan Project (20250102226JC), the National Natural Science Foundation of China (Grant 62422208, 62232010), Ministry of Industry and Information Technology of China (Grant TC240A9ED-70), Shandong Science Fund (Grant ZR2025LZH006), Research Project of Quancheng Laboratory, China (Grant QCL20250106), Project of the Major Innovation Project of Key Laboratory of Computing Power Network and Information Security, Ministry of Education (Grant 2024ZD012).

References

[1] M. Al Mamun, M. Li, and BK. Pramanik, "Development of Delay-Tolerant Networking Protocols for Reliable Data Transmission in Space Networks: A Simulation-Based Approach," *IEEE Access*, vol. 12, pp. 178642–178658, 2024.

[2] J. Yu, D. Huang, W. Li, *et al.*, "Adaptive Network Routing Technology for Near-Moon Space Cross-Domain Transmission," *Applied Sciences*, vol. 14, no. 22, 2024.

[3] IA. Lagoida, I. Astapov, and PS. Kuzmenkova, "Reconstruction of Near-Earth Cosmic Ray Fluxes from Ground-Based Neutron Monitors," *Physics of Atomic Nuclei*, vol. 87, no. 12, pp. 1912–1917, 2024.

[4] XG. Long, K. Huang, RW. Yang, *et al.*, "Pegasus: A Practical High-Speed Cross-Platform NDN Forwarder," *Computer Communications*, vol. 269, p. 111474, 2025.

[5] Y. Fei, JQ. Yin, and LJ. Yan, "Security Verification Framework for NDN Access Control," *Scientific Reports*, vol. 15, no. 1, p. 5479, 2025.

[6] SP. Devi and K. Dhanalakshmi, "MoDT: Interest Forwarding in Named Data Networking Based Vehicular Ad Hoc Networks by Predicting the Mobility Using Direction and Timer," *Ad Hoc & Sensor Wireless Networks*, vol. 58, nos. 1–2, pp. 53–77, 2024.

[7] KHM. Gularte, JPJ. da Costa, JAR. Vargas, *et al.*, "Integrating Cybersecurity in V2X: A Review of Simulation Environments," *IEEE Access*, vol. 12, pp. 177946–177985, 2024.

[8] A. Mardaus, E. Biernacka, and R. Wójcik, "Open Source Software-Defined Networking Controllers—Operational and Security Issues," *Electronics*, vol. 13, no. 12, p. 2329, 2024.

[9] M. Tropea, CEQ. Aldana, EP. de Freitas, and F. De Rango, "SFEM3: A Performance Comparative Analysis for SDN-Based FANET Emulation Using Mininet-WiFi and ns-3," *Ad Hoc Networks*, vol. 177, p. 103859, 2025.

[10] PA. Pan, WL. Chin, YC. Huang, *et al.*, "Dynamic RSVP in Modern Networks for Advanced Resource Control with P4 Data Plane," *Sensors*, vol. 25, no. 7, p. 2244, 2025.

[11] B. Kempton and A. Riedl, "Network Simulator for Large Low Earth Orbit Satellite Networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2021, pp. 1–6. doi:10.1109/ICC42927.2021.9500439.

[12] F. G. Lavacca, P. Salvo, L. Ferranti, *et al.*, "Performance Evaluation of 5G Access Technologies and SDN Transport Network on an NS3 Simulator," *Computers*, vol. 9, no. 2, p. 43, 2020.

[13] JL. Xu, WS. Pan, HB. Tan, *et al.*, "An Adaptive Congestion Control Optimization Strategy in SDN-Based Data Centers," *Computers, Materials & Continua*, vol. 81, no. 2, pp. 2709–2726, 2024.

[14] G. Li, H. Zhou, B. Feng, *et al.*, "Multi-Layer Satellite Network and Earth–Moon Satellite Network Simulation," *Journal of the China Railway Society*, vol. 39, no. 234, pp. 76–87, 2017.

[15] Heyu Liu, Fuchun [author], *et al.*, "Virtual Strategy QoS Routing in Satellite Networks," *Science China: Information Sciences*, vol. 59, no. 9, 2016.

[16] M. Karjalainen and T. Kokkonen, "Comprehensive Cyber Arena; The Next Generation Cyber Range," in *Proc. IEEE EuroS&PW*, 2020, pp. 11–16. doi:10.1109/EuroSPW51379.2020.00011.

[17] A. Peratikou, C. Louca, S. Shiaeles, *et al.*, "On Federated Cyber Range Network Interconnection," in *Proc. Int. Networking Conf.*, Cham, Switzerland: Springer, 2020, pp. 117–128.

[18] B. Fang, Y. Jia, A. Li, *et al.*, "Research on Cyberspace Range Technology," *Journal of Information Security*, vol. 3, p. 9, 2016.

[19] E. Mousavinejad, Eman, Yang, *et al.*, "A Novel Cyber Attack Detection Method in Networked Control Systems," *IEEE Trans. Cybernetics*, vol. 48, no. 11, pp. 3254–3264, 2018.

[20] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.

[21] T. Koponen, *et al.*, "A Data-Oriented (and Beyond) Network Architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 181–192, 2007.

[22] S. Lederer, C. Mueller, C. Timmerer, and H. Hellwagner, "Adaptive Multimedia Streaming in Information-Centric Networks," *IEEE Network*, vol. 28, no. 6, pp. 91–96, 2014.

[23] A. Santin, M. Bagatin, R. Harboe-Sorensen, *et al.*, "The Deep Space Radiation Probe: Development of a First Lunar Science Payload for Space Environment Studies and Capacity Building," in *Proc. IEEE Radiation Effects Data Workshop (REDW)*, 2022, pp. 1–6. doi:10.1109/REDW55356.2022.9917463.

[24] KL. Ryder and MJ. Campola, "The Lunar Radiation Environment," NASA Goddard Space Flight Center, Greenbelt, MD, Tech. Rep. NASA/GSFC Code 561, Apr. 27, 2022.

[25] G. Ateniese, R. Di Pietro, L. V. Mancini, *et al.*, "Scalable and Efficient Provable Data Possession," in *Proc. 4th Int. Conf. Security and Privacy in Communication Networks*, 2008, pp. 1–10.

[26] R. Kumari, K. Kaur, and A. Almogren, "C-BIVM: A Cognitive-Based Integrity Verification Model for IoT-Driven Smart Cities," *Computers, Materials & Continua*, vol. 84, no. 3, pp. 5509–5525, 2025.

[27] T. Li and L. Hu, "Audit as You Go: A Smart Contract-Based Outsourced Data Integrity Auditing Scheme for Multiauditor Scenarios with One Person, One Vote," *Security and Communication Networks*, 2022. doi:10.1155/2022/8783952.

[28] D. Yue, R. Li, Y. Zhang, W. Tian, and Y. Huang, "Blockchain-based verification framework for data integrity in edge-cloud storage," *J. Parallel Distrib. Comput.*, vol. 146, pp. 1–14, 2020.

[29] F. Q. Zhang, G. Guo, Y. C. Qin, and Q. M. Chen, "Prediction of proton-induced single event effect on SRAM's in-orbit soft error rate on typical satellite orbit," *Spacecraft Environment Engineering*, vol. 35, no. 4, pp. 365–370, Aug. 2018.

[30] C. Zhang, H. Xuan, T. Wu, X. Liu, G. Yang, and L. Zhu, "Blockchain-Based Dynamic Time-Encapsulated Data Auditing for Outsourcing Storage," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 1979–1993, 2024.

[31] S. Kumari, M. Singh, R. Singh, and H. Tewari, "Signature based Merkle Hash Multiplication algorithm to secure the communication in IoT devices," *Knowledge-Based Systems*, vol. 253, p. 109543, 2022.