

Article

A Multi-Layer Secure Sharing Framework for Aviation Big Data Based on Blockchain

Qing Wang, Zhijun Wu *  and Yanrong Lu

School of Safety Science and Engineering, Civil Aviation University of China, Tianjin 300300, China;
2020095004@cauc.edu.cn (Q.W.); yr_lu@cauc.edu.cn (Y.L.)

* Correspondence: zjwu@cauc.edu.cn

Abstract

As a new type of production factor, data possesses multidimensional application value, and its pivotal role is becoming increasingly prominent in the aviation sector. Data sharing can significantly enhance the utilization efficiency of data resources and serves as one of the key tasks in building smart civil aviation. However, currently, data silos are pervasive, with vast amounts of data only being utilized and analyzed within limited scopes, leaving their full potential untapped. The challenges in data sharing primarily stem from three aspects: (1) Data owners harbor concerns regarding data security and privacy. (2) The highly dynamic and real-time nature of aviation operations imposes stringent requirements on the timeliness, stability, and reliability of data sharing, thereby constraining its scope and extent. (3) The lack of reasonable incentive mechanisms results in insufficient motivation for data owners to share. Consequently, addressing the issue of aviation big data sharing holds significant importance. Since the release of the Bitcoin whitepaper in 2008, blockchain technology has achieved continuous breakthroughs in the fields of data security and collaborative computing. Its unique characteristics—decentralization, tamper-proofing, traceability, and scalability—lay the foundation for its integration with aviation. Blockchain can deeply integrate with air traffic management (ATM) operations, effectively resolving trust, efficiency, and collaboration challenges in distributed scenarios for ATM data. To address the heterogeneous data usage requirements of different ATM stakeholders, this paper constructs a blockchain-based multi-level data security sharing architecture, enabling fine-grained management and secure collaboration. Furthermore, to meet the stringent timeliness demands of aviation operations and the storage pressure posed by massive data, this paper optimizes blockchain storage deployment and consensus mechanisms, thereby enhancing system scalability and processing efficiency. Additionally, a dual-mode data-sharing solution combining raw data sharing and model sharing is proposed, offering a novel approach to aviation big data sharing. Security and formal analyses demonstrate that the proposed solution is both secure and effective.

Keywords: aviation big data sharing; blockchain; model sharing; storage optimization; consensus mechanisms



Academic Editor: Paolo Bellavista

Received: 10 July 2025

Revised: 31 July 2025

Accepted: 4 August 2025

Published: 8 August 2025

Citation: Wang, Q.; Wu, Z.; Lu, Y. A Multi-Layer Secure Sharing Framework for Aviation Big Data Based on Blockchain. *Future Internet* **2025**, *17*, 361. <https://doi.org/10.3390/fi17080361>

Copyright: © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The aviation transportation industry is one of the core indicators of a nation's modernization level. With the global aviation industry accelerating its digital and intelligent transformation, smart civil aviation has become a concentrated manifestation of the new technological revolution and industrial transformation in the aviation sector, exhibiting

strong developmental momentum. Data, as a key element of production, is increasingly demonstrating its multidimensional value in aviation.

From an operational perspective, air traffic management (ATM) data reflects the real-time status of aviation activities, providing precise situational awareness and decision-making support for stakeholders (e.g., air traffic controllers, airlines, airports), such as real-time weather information sharing, flight conflict warnings, and dynamic traffic management. From a research perspective, ATM data serves as an empirical foundation for exploring aviation operational patterns, validating optimization strategies, and developing intelligent decision-making systems.

Data-driven aviation has become a consensus, and data sharing can significantly enhance data resource utilization, serving as a critical task in building smart civil aviation. However, data silos persist, with vast amounts of data only accessible and analyzable within limited scopes, leaving their potential underutilized. The challenges in data sharing stem from three main factors: (1) Security and privacy concerns: Aviation data, especially ATM data, contains sensitive operational information with high security requirements. Recent privacy breaches in aviation, such as the 21 May 2021, incident where Indian Airlines reported the theft of approximately 4.5 million customer records spanning a decade (August 2011 to February 2021), have heightened data owners' apprehensions. (2) Dynamic and real-time requirements: The highly dynamic nature of aviation operations imposes stringent demands on the timeliness, security, and reliability of data sharing, limiting its scope and extent. (3) Lack of incentives: Most data producers are reluctant to share data voluntarily, while data consumers remain skeptical of data utility, highlighting the absence of an effective fair incentive mechanism for ATM data sharing.

Blockchain, a specialized database technology, employs P2P networks for communication, cryptography for ownership and privacy assurance, and distributed consensus frameworks for consistency. Since the release of the Bitcoin whitepaper in 2008, blockchain technology has made continuous breakthroughs in data security and collaborative computing.

The Bitcoin network, with its SHA-256 encryption and distributed ledger architecture, pioneered tamper-proof storage without trusted third parties [1]. In 2015, Ethereum introduced smart contracts [2], enabling auditable trust mechanisms for automated program execution. For high-throughput processing, Solana's Proof-of-History consensus in 2021 achieved 710,000 transactions per second (TPS), 46 times faster than Ethereum 1.0 [3].

These advancements have transformed blockchain from a cryptocurrency tool into a universal trust infrastructure, laying the groundwork for its integration into aviation. Blockchain's decentralized, tamper-proof, traceable, and scalable features offer novel solutions for secure data architectures, attracting widespread attention. Organizations like Air France, ICAO, and NASA have explored blockchain applications in aviation maintenance and data tracking [4].

Blockchain can deeply integrate with ATM operations, effectively addressing trust, efficiency, and collaboration challenges in distributed data sharing. Aviation operations generate vast heterogeneous data with varying sensitivity and timeliness requirements. A blockchain-based multi-layer data-sharing mechanism can cater to diverse stakeholder needs. For real-time-dependent operators (e.g., airlines), encrypted blockchain data with access controls enables rapid airspace restriction updates and trajectory adjustments. For regulatory bodies focused on trend analysis, federated learning leveraging blockchain preserves privacy while enhancing decision-making models. To meet aviation's real-time demands and storage constraints, optimized blockchain deployment and consensus mechanisms can improve scalability and processing efficiency. This holistic approach balances security, timeliness, and consistency while maximizing data utility under multiple constraints.

This paper is structured as follows: Section 1 introduces the context; Section 2 covers background knowledge; Section 3 presents the mathematical model for secure data sharing; Section 4 details the multi-layer sharing framework; Section 5 analyzes security and formal verification; and Section 6 concludes this paper.

2. Related Work

As the value of aviation data resources becomes increasingly prominent, establishing secure, trustworthy, and efficient collaborative data-sharing mechanisms has become a research focus. In recent years, blockchain has gained widespread application in privacy-sensitive domains due to its decentralized, tamper-proof, and traceable characteristics. In the ATM field, existing research has attempted to overcome the trust deficiencies of centralized architectures. For example, ATMChain [4] integrates identity authentication, access control, and data sharing into a unified 4A security model (Authentication, Authorization, Accounting, Auditing). While it enhances trustworthiness and auditability, its consensus mechanism lacks real-time adaptability to high-frequency ATM data. Another blockchain-based distributed flight data platform [5] improves data consistency and availability under the SESAR and SWIM frameworks but remains largely conceptual, lacking in-depth technical implementation.

In healthcare [6], cloud storage [7], Internet of Things (IoT) [8], and intelligent transportation [9], blockchain technology has also been widely applied. For instance, HSHB [10] achieves secure and efficient data access by dividing private and consortium chain sharing, constructing access control policies and B+ tree indexing, and combining proxy re-encryption with smart contracts. Reference [8] proposes a multi-center IoT data-sharing framework that reduces computational overhead through improved attribute-based encryption and cloud-side obfuscation strategies.

The integration of blockchain and federated learning (FL) is building a new paradigm for trusted privacy-preserving computing. Reference [11] proposes a blockchain-FL energy trading framework based on edge computing in 5G environments, while [12] designs a medical privacy-preserving architecture based on FL and blockchain. However, existing research primarily focuses on horizontal federated learning, with limited support for vertical federated learning (VFL), which is crucial for cross-institutional ATM data sharing. Additionally, there is a lack of quantitative fairness evaluation mechanisms for multi-party collaboration contributions.

Consensus mechanisms are the core of blockchain-enabled trusted collaboration, with the key challenge being how to ensure system security and consistency in untrusted, heterogeneous, and dynamic network environments. Byzantine Fault Tolerance (BFT) protocols achieve fault tolerance and robustness in malicious node environments through cryptographic verification, redundant communication, and voting mechanisms. Existing solutions are mainly divided into proof-based (e.g., PoW, PoS) and election-based (e.g., PBFT). Proof-based schemes offer strong decentralization in open networks but struggle to meet ATM system requirements for strong consistency and low latency due to uncertain outcomes and high delays. PBFT, suitable for semi-trusted environments, enables deterministic consensus and is widely adopted. However, PBFT faces $O(n^2)$ communication complexity and a leader node bottleneck. To improve PBFT performance, various optimization schemes have been proposed: the work in [13] introduces a dynamic PBFT ordering service with bandwidth reservation, optimizing throughput and cluster scalability; reference [13] allows each ordering node to autonomously initiate consensus and employs a bandwidth reservation protocol to achieve efficient distributed ledger record insertion in permissioned blockchain. Reference [14] designs a scalable multi-layer PBFT consensus mechanism, reducing communication complexity through hierarchical node grouping to support large-scale systems. The

authors of [15] adopt threshold proxy signatures and a dual-layer reputation mechanism to enable the TP-PBFT protocol, which achieves consensus even when more than 1/3 of nodes fail. However, existing schemes mostly assume stable connectivity and fixed topologies, making them unsuitable for high-mobility, weak-link scenarios like ATM communications. Lightweight, strongly consistent, low-latency consensus protocols remain necessary.

Additionally, traditional blockchain's full ledger replication imposes heavy storage burdens on high-density data environments like ATM. Current mainstream solutions fall into two categories: off-chain and on-chain cooperative storage. Off-chain solutions, such as LightChain [16], IPFS [17], and LayerChain [18], store raw data externally while maintaining hash indices on-chain, reducing redundancy but introducing trust and availability challenges. On-chain cooperative storage reduces individual node loads through ledger sharding, such as BFT-Store [19] using erasure coding and VBG [20] using virtual block groups, but incurs high coordination costs in dynamic node environments. Light client solutions, such as FlyClient [21] based on Merkle Mountain Ranges, improve storage efficiency but heavily rely on full nodes and have limited autonomous validation capabilities.

In summary, ATM data sharing faces challenges such as unstable connectivity, high real-time demands, and massive data volumes. Neither purely on-chain nor off-chain storage can fully meet these needs. Constructing a multi-layer hybrid storage architecture that dynamically processes data based on priority is a critical path to enhancing system adaptability and scalability.

3. Analysis of ATM Big Data Sharing

To gain a deeper understanding of the fundamental characteristics and key challenges of ATM data sharing, this chapter first examines the functional objectives and structural composition of the ATM system from a systemic perspective. It then identifies the primary participants and data types involved in sharing, laying the groundwork for subsequent data modeling and mechanism design.

3.1. ATM System Concepts

ATM is a highly complex, dynamically evolving cyber-physical system (CPS). This section characterizes ATM from four perspectives: functions/objectives, structure, key elements (stakeholders), and constraints.

3.1.1. Functions and Objectives

ATM is an integrated management system leveraging Communication, Navigation, and Surveillance (CNS) technologies and information networks. Through "air-space-ground" coordination (e.g., satellite links, ground radar, onboard systems), it enables real-time data sharing and dynamic decision-making, with core objectives of ensuring safe, efficient, and orderly aircraft operations.

Per ICAO's Global ATM Operational Concept [22], ATM comprises seven key components, shown in Figure 1.

These components form ATM's foundational framework, enabling functions like traffic control, flight information dissemination, conflict resolution, and flow optimization to address complex airspace environments and diverse flight demands. The system's complexity and adaptability stem from its wide-area, multi-source heterogeneous data fusion and high real-time requirements [4].

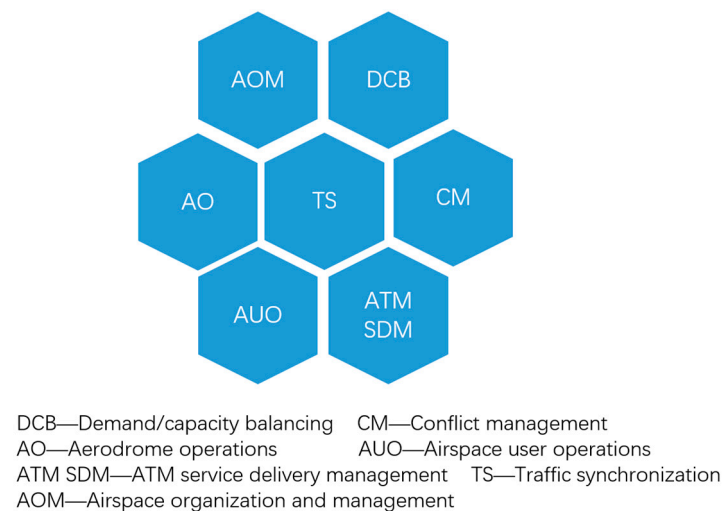


Figure 1. Seven components of ATM system.

3.1.2. ATM System Structure

ATM is analyzed temporally and spatially, with data-sharing implications.

- Temporal dimension.

The ATM system can be temporally divided into three distinct phases—strategic, pre-tactical, and tactical [23]—each exhibiting significant differences in functional implementation and data-sharing mechanisms.

The strategic phase spans months to years, with core functions including airspace design and traffic flow prediction. By analyzing historical data and macroeconomic trends, this phase forecasts future traffic distribution and optimizes air route networks and sector configurations to accommodate long-term traffic growth. A representative example is the Dynamic Airspace Configurations (DAC) tool developed by Europe’s SESAR project, which dynamically adjusts route networks and sectors by integrating multi-source data from various stakeholders including ANSPs, airline operators (AOs), airspace users (AUs), and network managers (NMs). Data shared during this phase encompasses airline flight plans, airport throughput capacity, airspace utilization metrics, and external socioeconomic data.

The pre-tactical phase operates on timescales ranging from hours to days, focusing on establishing flow management strategies and dynamic route adjustments to address real-time variables such as adverse weather and temporary airspace restrictions. This phase requires sharing real-time operational data including aircraft performance parameters, live weather conditions, and current airspace status [1]. The FAA’s System Wide Information Management (SWIM) exemplifies this through standardized interfaces enabling real-time data exchange between ATC, airlines, and airports, covering aircraft positions (ADS-B), meteorological radar data, and airspace status updates [23].

The tactical phase (operational layer) functions within minute-to-hour timeframes, emphasizing the execution of control instructions and conflict resolution under shared situational awareness. Controllers monitor operations in real time using surveillance technologies (ADS-B), issuing altitude and heading adjustments via voice communications (VHF) or data links (CPDLC), while coordinating with onboard collision avoidance systems (TCAS II) and the ground-based Short-Term Conflict Alert (STCA) to resolve aircraft conflicts. Tools like Electronic Flight Bags (EFBs) enhance situational awareness synchronization between pilots and controllers through features like route weather synchronization.

In summary, the strategic phase primarily relies on cross-institutional historical data sharing and multi-source data fusion modeling to improve prediction accuracy. The pre-tactical phase achieves dynamic resource allocation through real-time data synchro-

nization and promotes collaborative decision-making (CDM) through multi-party data exchange. The tactical phase accomplishes situational awareness synchronization between controllers and pilots via immediate data sharing. This hierarchical data-sharing model across different time dimensions reflects the evolving operational requirements of ATM from macro-planning to micro-execution.

- Spatial dimension.

In terms of spatial architecture, air traffic management (ATM) constitutes a three-dimensional system that deeply integrates space-based (space), aerial (air), and ground (ground) components through multimodal communication technologies including satellite data links and aeronautical data links.

The space-based system incorporates navigation satellites (GPS, GLONASS, Galileo, BeiDou, etc.), communication satellites (Inmarsat, Iridium, etc.), meteorological satellites (NOAA, Fengyun series), and space-based ADS-B (e.g., Aireon system). These collectively provide positioning/navigation, wide-area communication, weather monitoring, and transoceanic aircraft tracking capabilities.

The aerial layer employs onboard data links such as ACARS and CPDLC to enable real-time trajectory management and dynamic airspace allocation.

Ground layer: Built upon the Aeronautical Telecommunication Network (ATN) and international standardized protocols (e.g., ATN/IPS), it enables air-ground data interaction through VHF, HF, and satellite communications. Aircraft surveillance, identification, and interaction are achieved via ATC automation systems for flight plan processing, conflict detection, and traffic flow prediction, while airport departure sequencing is optimized using CDM tools.

This integrated spatial architecture forms a comprehensive framework for modern ATM operations, ensuring seamless coordination across all flight domains.

3.1.3. Key Element

ATM is a highly integrated complex system whose efficient operation relies on the coordinated collaboration of hardware infrastructure, software systems, multiple participating entities, and data resources. These key elements collectively constitute the core architecture of ATM, ensuring the safety, orderliness, and efficiency of air transportation.

As the physical foundation of ATM, hardware includes critical equipment such as surveillance, navigation, communication, and meteorological detection systems. Surveillance systems (e.g., radar, ADS-B, MLAT) track aircraft positions in real time; navigation systems (ground-based VOR/DME and space-based GNSS) provide precise route guidance; communication systems (VHF, CPDLC, SATCOM) enable air-ground information exchange; and meteorological equipment supports weather warnings and route optimization. These hardware components serve as the “sensors” of ATM, providing physical-layer perception and interaction capabilities.

Software systems act as the “nerve center” of ATM, responsible for data processing, decision support, and operational optimization. ATM automation systems (e.g., FDP, RDP) manage flight plans and trajectory tracking; decision support tools (STCA, MSAW) assist controllers in risk identification; and simulation systems are used for personnel training and contingency plan validation. These software components transform raw data collected by hardware devices into actionable control decisions, significantly enhancing system response speed and management precision.

ATM participants include air traffic service providers, airline operators, regulatory agencies, and technical support units. ATC units (area control, approach control, tower) are responsible for real-time command; airlines execute control instructions and provide operational feedback; international organizations (ICAO) and national agencies (FAA,

EASA) establish standards and regulations; and equipment manufacturers and technical service providers offer system support. This multi-party collaboration mechanism ensures the reliability and safety of ATM operations through clear division of responsibilities while also imposing higher requirements on the system's overall collaborative decision-making capability.

Data resources serve as the “lifeblood” connecting all elements, running through the entire ATM operation process. The sharing and exchange of data such as aircraft dynamic data (position, altitude, speed), airspace structure data (routes, sectors), meteorological information (real-time weather, forecasts), and operational management data (flight plans, delay statistics) provide a quantifiable decision-making basis for ATM operations. These data not only support daily operational management but also serve as important references for system performance evaluation and continuous optimization. These four elements do not exist in isolation but form an organic whole through continuous interaction: hardware collects data, software processes information, participants make decisions, and data flows both transmit information and feed back into system optimization.

3.1.4. Constraints

The International Civil Aviation Organization (ICAO) has explicitly identified the coordinated development of “physical safety—operational efficiency—environmental sustainability—information security” throughout the entire lifecycle as the overarching objective in the Global Air Traffic Management Operational Concept [22]. Under this macro-level guidance, these objectives can be further decomposed into specific constraints.

Safety prioritizes the security of aircraft and personnel above all else, with ATM implementing multidimensional constraints through systematic risk management. ICAO's Standards and Recommended Practices (SARPs) and Procedures for Air Navigation Services (PANS) provide theoretical guidance for safe operations. At the operational level, ATM adopts a layered defense strategy for conflict management, strictly enforcing horizontal, vertical, and temporal separation standards to maintain aircraft operational risks within acceptable limits while incorporating the Traffic Collision Avoidance System (TCAS) as a safeguard. Additionally, quantitative management of air traffic controller workload ensures it remains within safe thresholds.

Efficiency requirements primarily focus on optimizing airspace and airport resource allocation to reduce flight delays and lower operational costs. In practice, this involves multiple technical indicators and operational constraints such as airspace capacity limits, maximum acceptable delay thresholds, route optimization restrictions, and flow management constraints. To enhance efficiency, dynamic airspace management is achieved by balancing user preferences with airspace capacity limitations based on 4D trajectory requirements while mandating information sharing among all stakeholders including airlines, air traffic control, and airports to facilitate collaborative decision-making and improve airspace throughput.

Environmental sustainability aims to significantly reduce the environmental impact of aviation activities through optimized operational procedures, ultimately achieving ICAO's global goal of net-zero carbon emissions by 2050. This is manifested in three key areas: (i) carbon emission control: implementing trajectory optimization techniques, strictly enforcing phased carbon emission limits under ICAO's CORSIA mechanism [24], and adopting sustainable aviation fuels; (ii) noise management: maintaining strict compliance with ICAO's day-night noise limits [25] and optimizing flight procedures to reduce noise during approach and departure phases; and (iii) energy efficiency: integrating real-time meteorological data to minimize unnecessary detours and implementing flexible airspace management strategies.

Cybersecurity serves as a critical element in safeguarding global aviation safety, playing a vital role in ATM systems. ATM systems heavily depend on the availability of information and communication technologies (ICT), as well as the accuracy and confidentiality of data, all of which directly impact flight safety and efficiency [26]. As early as 1996, ICAO's ATNP Security Working Group identified that all ATM information transmitted via data links is vulnerable to tampering, replay, and forgery attacks, while the Communication, Navigation, and Surveillance (CNS) systems supporting ATM exhibit inherent vulnerabilities to Distributed Denial of Service (DDoS) attacks. With the continuous evolution of cyber threats, attacks targeting ATM systems are demonstrating broader impact and greater destructive potential.

To address these growing cybersecurity challenges, ICAO has established a systematic standard framework. The 2019 Aviation Cybersecurity Strategy [27] for the first time elevated cybersecurity to a strategic priority, requiring member states to incorporate cybersecurity into their national aviation safety and security oversight systems. Through measures such as building international cooperation mechanisms, improving information sharing platforms, and optimizing incident response procedures, the strategy aims to comprehensively enhance the cyber resilience of aviation systems. The 2022 revision to Annex 17—Aviation Security of the Convention on International Civil Aviation [28] further refined technical specifications, mandating that member states establish full lifecycle risk management mechanisms covering critical aviation ICT systems and data based on the Secure by Design principle, with emphasis on ensuring the confidentiality, integrity, and availability of critical systems and data.

Building on this foundation, the latest edition of the Global Aviation Security Plan (GASep) [29] proposed more actionable implementation measures, calling for the establishment of a cross-domain, multi-layered cybersecurity protection system encompassing key areas including air traffic management, communications, surveillance, and aircraft operations. The plan sets clear phased targets through detailed implementation roadmaps: by 2030, over 90% of member states must achieve effective implementation in aviation cybersecurity to ensure continuous availability of critical systems, thereby constructing a highly resilient global aviation cybersecurity protection framework.

Through coordinated management of these multidimensional constraints, global ATM systems can achieve efficient, sustainable, and interference-resistant operations while maintaining safety.

3.2. Foundation of Aviation Big Data Sharing

This chapter provides an in-depth overview of aviation big data sharing from three key perspectives: data classification and participants, sharing trends and characteristics, and demand analysis.

3.2.1. Data Classification and Participants

ICAO has systematically standardized global ATM data-sharing requirements through documents including Annex 11 "Air Traffic Services" and Doc 4444 "Procedures for Air Navigation Services—Air Traffic Management" (abbreviated as PANS-ATM) to ensure flight safety, efficient airspace utilization, and cross-regional coordination.

From the perspective of air traffic management operations, the main participants in data sharing include air traffic service providers (such as air traffic control centers and regional control units), airlines and operators (providing flight plans and dynamic data sharing), airport operators (providing takeoff/landing slots and ground operation data), aeronautical information service providers (supplying aeronautical charts and airspace restriction information), airspace users, meteorological service providers (offering real-time

weather and wind shear data), military aviation departments (for airspace coordination and temporary restricted area information), civil aviation regulatory authorities (conducting safety audits and operational compliance data), and international organizations (such as ICAO, Eurocontrol, and IATA).

Regarding shared content, this paper classifies seven categories of shared data based on application scenarios and functions: core operational data, airspace and traffic flow management data, communication–navigation–surveillance data, aeronautical information data, meteorological data, emergency data, and support data. Note that the aviation big data referred to in this paper specifically pertains to ATM data.

3.2.2. Aviation Big-Data-Sharing Trends and Characteristics

Under the impetus of digital transformation in the aviation industry, ATM data sharing exhibits three major trends [30].

(1) Exponential growth in data exchange volume.

The frequency of daily data interactions shows exponential growth, with ATM data evolving from “limited data flows between fixed stakeholders” to “massive data-sharing exchanges reaching tens of millions within specific timeframes.”

(2) Normalization of cross-domain sharing.

Data flow boundaries are breaking through traditional security perimeters, transitioning from “ATM data being used only within fixed, limited security domains” to “ATM data being utilized across different security domains.” This aligns with the “Internet of Aviation” concept proposed in ICAO’s Global Air Navigation Plan, which emphasizes network interconnectivity among all aviation nodes (aircraft, ANSPs, ground facilities) and achieves cross-domain information integration through large-scale data sharing across Flight Information Region (FIR) boundaries [31].

(3) Diversification of participants.

With third-party analysis institutions and research organizations emerging as new data users, the sharing participants are evolving from “data accessors being solely operational executors” to “including third-party analysts and researchers.”

3.2.3. Demand Analysis

These trends render traditional sharing mechanisms inadequate for new scenarios. While ICAO Annex 11 only specifies minimum security requirements for data sharing, new operational needs like multi-party situational awareness and collaborative decision-making have generated additional demands for ATM data sharing.

(1) Hierarchical sharing requirements.

Layered data sharing accommodates differentiated business needs among ATM participants, divided into raw data sharing and data–model sharing.

Participants can be categorized into three tiers, core operational entities, collaborative support entities, and regulatory (research) entities, forming a “decision–execution–supervision” tri-layer structure. This reflects ATM’s multi-level data-sharing requirements—different entities have varying demands for data granularity and timeliness. Core operators rely on high-precision real-time data to maintain operational safety (e.g., airlines needing second-level latency for airspace restriction updates), while regulators focus more on data trends, requiring aggregated historical data for decision models. The former necessitates direct raw data sharing, whereas the latter prefers model sharing, achieving information exchange through distributed model training while keeping raw data localized.

Raw data sharing ensures complete transmission of critical operational data to consumers, while model sharing minimizes unnecessary information leakage, adhering to the “data minimization principle” and transforming sharing from “ownership transfer” to “usage rights sharing.”

This hierarchical approach balances data openness with ownership control through data–model separation, offering new solutions to break “data silos” and laying groundwork for differentiated security strategies.

(2) Information security requirements.

The expanded sharing scope significantly increases attack surfaces, multiplying ATM system security risks. Recent years have seen growing information security incidents in global aviation, causing substantial economic losses and even threatening flight safety. In September 2016, DHS senior officials confirmed successful remote penetration testing on a Boeing 757 at Atlantic City International Airport via radio communications. In September 2018, Bristol Airport’s electronic displays were disabled for two days by ransomware attacks, forcing manual flight information updates. ATM data faces tampering, theft, and access failure risks regarding confidentiality, availability, and integrity. Unauthorized access and privacy breaches during sharing threaten ATM’s secure operation.

(3) Efficiency requirements.

Aviation operations’ characteristics—vast airspace networks, dense traffic flows, cross-regional coordination, and highly dynamic management—impose stringent timeliness demands on data sharing. As ATM systems transition from “human-led decision-making” to “data-driven decision-making,” exemplified by Trajectory-Based Operations (TBO) requiring real-time 4D trajectory data exchange among ATC, airlines, and airports [31], data timeliness becomes an increasingly critical constraint.

(4) Storage resource requirements.

There exists a contradiction between exponentially growing data volumes and limited storage resources. New-generation aircraft generate 5–8 TB data per flight (80 times more than older models), with global fleet data projected to reach 98 billion GB annually by 2026 [32]. This massive historical data imposes high storage demands on aviation participants, especially lightweight users like aircraft avionics systems with inherently limited storage capacity, making storage capacity adaptation a prerequisite for effective data sharing.

(5) Incentive requirements.

Due to operational inertia, most data producers are reluctant to share voluntarily. Meanwhile, data consumers remain skeptical about shared data’s utility, as current ATM operations lack an effective fair incentive mechanism for data sharing.

4. Data Security Sharing Modeling

This chapter constructs a behavioral model based on an extended infinite state machine (ISM) to formally characterize the key states, input events, and dynamic evolution mechanisms involved in the ATM data-sharing process, clearly expressing various secure sharing constraints. On this basis, a multi-layer secure sharing architecture for multi-source aviation data is proposed, followed by correctness analysis and formal verification. The proposed modeling approach and analytical framework provide a solid theoretical foundation for the subsequent design and implementation of secure sharing mechanisms.

4.1. Mathematical Model for Secure Data Sharing

This paper employs an extended infinite state machine (ISM) (hexatuple) to characterize the ATM data-sharing process:

$$M = (Q, \Sigma, \delta, q_0, F, C)$$

where Q is the set of system states (potentially infinite); Σ is the set of input events, such as data requests, data updates, access control modifications, etc.; $\delta : Q \times \Sigma \rightarrow Q$ is the state transition function, describing the dynamic changes in data sharing; q_0 is the initial state, representing the system state when no data sharing occurs; F is the set of accepting states, i.e., trusted sharing states that satisfy all constraints; and C is the set of constraints, including security, storage capacity, timeliness, etc., during the sharing process.

The description of system states can fully represent all information for a single instance of secure data sharing. Each state $q \in Q$ in the state set can be expressed as follows:

$$q = \langle \begin{array}{l} \text{Data_Provider, Data_Consumer, Data,} \\ \text{Share_Mode, Topology, Timeliness,} \\ \text{Security_Measure, Utility} \end{array} \rangle$$

The definitions of each element in the system state are shown in Table 1.

Table 1. The definitions of each element in the system state.

Tuple	Description	Example/Range
Data_Providers	Set of ATM units providing data/models (data owners O_k)	Airport towers, weather stations, etc.
Data_Consumers	Set of ATM units receiving data/models (data consumers A_j)	Airlines, regional ATC centers, etc.
Data	Shared data/model D_j	Data: flight trajectories, weather data Models: traffic prediction, conflict detection models
Sharing_Mode	Current sharing mode	Data_Sharing (raw data), Model_Sharing (parameters/gradients), Hybrid
Topology	Network topology determining data flow paths and fault tolerance	Centralized, Decentralized (P2P), Distributed, Hybrid
Timeliness	Maximum allowed latency for real-time data sharing	$\tau_{max} = 5s$
Security_Measures	Aggregated security mechanisms	Encryption, access control, integrity checks
Utility	Quantified value of data/model D_j to receiver	$U(D_j, A_i) \in [0, 1]$

4.2. Constraints for Secure Data Sharing

Traditional finite state machines (FSMs) or infinite state machines (ISMs) are typically defined as quintuples $(Q, \Sigma, \delta, q_0, F)$. However, the ATM data secure sharing scenario possesses unique characteristics. ATM data sharing must satisfy stringent security and timeliness constraints, which are fundamental to ATM operations and have irreplaceable impacts on data sharing. Therefore, this paper augments the quintuple with a constraint tuple. The constraints for secure data sharing include the following aspects.

4.2.1. Storage Space Constraints

Each participant has limited actual storage capacity, especially lightweight participants like aircraft with extremely constrained storage. The total stored data must be restricted:

$$\forall A_i \in \text{Participants}, \sum_{D_j \in \text{Stored}(A_i)} \text{Size}(D_j) \leq C_i$$

where C_i is the maximum storage capacity of participant A_i , and $\text{Size}(D_j)$ is the size of data D .

4.2.2. Efficiency (Real-Time) Constraints

The delay from data request to completed sharing must not exceed the maximum allowed value:

$$\forall D_j, \tau_j \leq \tau_{max}$$

where τ_j is the transmission delay of data D_j , and τ_{max} is the system's maximum allowed delay.

4.2.3. Security Constraints

(i) Integrity verification.

Define integrity verification function $\text{VeriInt}(x)$,

$$\text{VeriInt}(x) = \begin{cases} \text{True} & \text{if } x \text{ satisfies integrity} \\ \text{False} & \text{otherwise} \end{cases}$$

When x is shared data or a model, it must satisfy integrity requirements:

$$\forall x \in \text{SharedData}, \text{VeriInt}(x) = \text{True}$$

(ii) Confidentiality protection.

Define confidentiality verification function $\text{EnConfi}(x)$,

$$\text{EnConfi}(x) = \begin{cases} \text{True} & \text{if } x \text{ is encrypted during transmission/storage} \\ \text{False} & \text{otherwise} \end{cases}$$

When x is shared data or a model, it must satisfy confidentiality requirements:

$$\forall x \in \text{SharedData}, \text{EnConfi}(x) = \text{True}$$

(iii) Availability.

Defend against single points of failure through multi-replica storage:

$$\forall \text{SharedData}, \text{Replicas}(D_j) \geq 2$$

(iv) Privacy protection.

Define privacy protection function $\text{PrePrivacy}(m)$,

$$\text{PrePrivacy}(m) = \begin{cases} \text{True} & \text{if } \varphi(m, m') \leq \theta \\ \text{False} & \text{otherwise} \end{cases}$$

where φ is the privacy metric function, m is the original data, m' is the perturbed data, and θ is the security threshold

When m is a shared model, it must satisfy privacy protection requirements,

$$\forall m \in \text{SharedModel}, \text{PrePrivacy}(m) = \text{True}$$

(v) Access control.

$$\begin{aligned} \forall \text{Request}(D_j, A_i), \text{Consumer}(A_i) \in \text{PermittedList}(O_k) \\ \text{AccessControl}(A_i, D_j, O_k) = \text{True} \end{aligned}$$

4.2.4. Compliance Constraints

Sharing behaviors must comply with air traffic regulations:

$$\forall D_j, \text{Comply}(D_j, \text{Regulation}_k) = \text{True}$$

Here, Regulation_k refers to the k regulation (e.g., data retention period, privacy protection clauses, etc.).

4.2.5. Data Utility Constraints

Shared data must be valuable to the data consumer to avoid redundant transmission.

$$\forall D_j, A_i, U(D_j, A_i) \geq U_{\min}$$

Here, $U(D_j, A_i) \in [0, 1]$ represents the utility value of D_j for A_i , and $U_{\min}$ refers to the minimum utility value.

5. Aviation Big Data Multi-Layer Security Sharing Framework

According to the ATM data-sharing scenario and sharing requirements, the framework protects ATM data layer by layer through different protection strategies, achieving comprehensive security from data storage to sharing. This security architecture employs technologies such as blockchain, federated learning, attribute-based encryption, and multi-objective optimization to achieve secure data transmission, storage, and sharing in the ATM system. It provides a feasible solution for aviation big data security sharing, and its system architecture is shown in Figure 2.

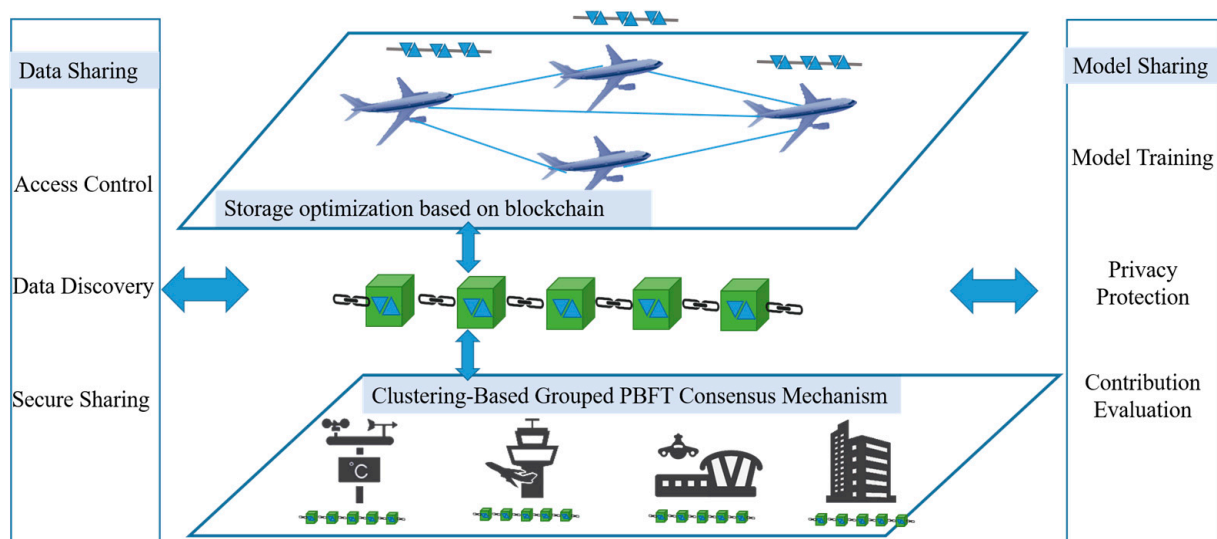


Figure 2. Multi-layer security sharing architecture for ATM data.

A trusted data-sharing mechanism is established using blockchain, leveraging its tamper-proof, traceable, decentralized, and single-point-failure-resistant characteristics to protect data security. By adopting a clustering-based grouped PBFT consensus mechanism, the consensus efficiency and throughput among ATM data participants are improved, ensuring efficient data sharing. Through a multi-objective optimization-based block deployment strategy, the storage occupancy ratio of participants is significantly reduced. The optimized block deployment and efficient consensus strategy address blockchain's storage and latency issues, laying the foundation for subsequent data sharing and model sharing.

In terms of data sharing, hash algorithms and encryption algorithms are used to ensure data integrity and confidentiality. To tackle the challenges of massive data retrieval and the over-/under-authorization issues in traditional access control, a data attribute-based retrieval method and a fine-grained attribute-based access control method are proposed.

To address privacy leakage risks in sharing, an ATM-oriented model-sharing method is introduced, leveraging blockchain's decentralization to provide a secure and reliable storage environment for federated learning. The Shapley value contribution assessment method is used to identify low-quality data and incentivize user sharing.

The combination of data sharing and model sharing covers the multi-level data sharing needs of current ATM participants.

5.1. Blockchain Optimization Deployment and Consensus Mechanism for ATM

As a data-sharing solution, blockchain is widely applied across various industries, such as the Internet of Things (IoT), vehicular networks, education, and healthcare. However, blockchain deployment and consensus models pose challenges for ATM participants. Since blockchain occupies significant storage space, resource-constrained terminals (e.g., aircraft) struggle to deploy a full blockchain. Additionally, in the Practical Byzantine Fault Tolerance (PBFT) consensus mechanism, each node must broadcast twice in a single consensus round, resulting in a network communication complexity of $O(N^2)$ for an N -node system. As the number of nodes increases, communication costs and consensus latency rise sharply, leading to rapid system performance degradation. This makes PBFT difficult to apply directly in dense, real-time ATM sharing systems.

To address these issues, an optimization algorithm is used to partition the full blockchain into multiple blocks and distribute them across different nodes, reducing storage pressure on resource-limited nodes. Additionally, a clustering-based grouped PBFT algorithm is employed to decompose consensus tasks, lowering communication overhead and consensus latency.

5.1.1. Blockchain Deployment Optimization Strategy

Blockchain provides secure protection for data storage and sharing, but as the chain grows, its storage demands increase, posing challenges for resource-constrained participants.

Time-based chain segmentation: The blockchain is divided into a real-time chain (containing real-time/near-real-time data, such as aircraft health status, flight trajectories, and weather updates) and a traceability chain (storing historical data).

Geographical network partitioning: The aviation network is divided into multiple consensus domains connected via heterogeneous networks. Each domain includes ATM participants such as air traffic control stations, weather stations, airports, airlines, and aircraft (both ground and airborne nodes). Airborne nodes within a domain jointly maintain a real-time chain instead of each storing a full blockchain (see Figure 3), which resolves the conflict between high storage demands and limited airborne node resources. Ground nodes, with sufficient computing and storage capacity, maintain full chains to ensure data security and traceability. If an airborne node lacks certain data, it sends a query to other

nodes in its domain. Nodes holding the requested block return it, and the requester verifies data authenticity and integrity by comparing multiple copies.

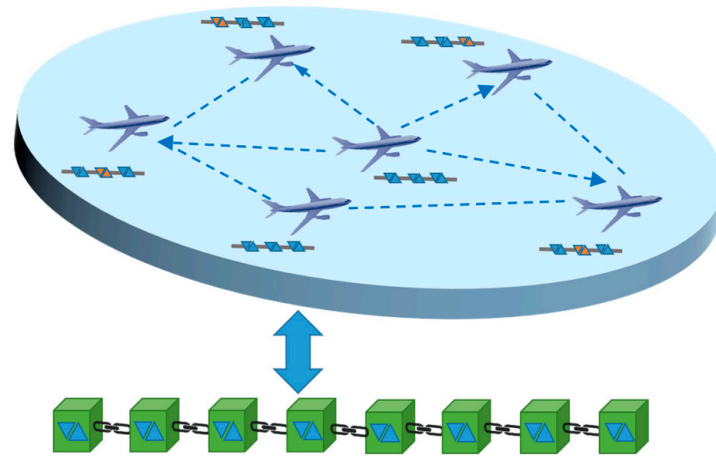


Figure 3. Airborne node block deployment diagram.

As ATM operations continue, blocks accumulate. When a node's storage exceeds a threshold, block redeployment optimization is triggered. By defining an objective function, the storage occupancy ratio, response latency, and block availability are balanced while ensuring security. A multi-objective optimization algorithm computes the optimal deployment scheme:

$$\begin{aligned} \min_{\mathbf{x}} \left\{ f_{AvgOccStoRatio}(\mathbf{x}), f_{AccRespLatency}(\mathbf{x}), f_{AvgUnAvailProb}(\mathbf{x}) \right\} \\ s.t. \quad g_{SecLevel}(\mathbf{x}) \leq 0 \\ g_{StoThres}(\mathbf{x}) \leq 0 \end{aligned}$$

where $\mathbf{x}$ is the block allocation scheme, $f_{AvgOccStoRatio}(\mathbf{x})$ denotes the average storage occupancy ratio, $f_{AccRespLatency}(\mathbf{x})$ is the access response latency, and $f_{AvgUnAvailProb}(\mathbf{x})$ is the average unavailability probability. And $g_{SecLevel}(\mathbf{x}) \leq 0$ and $g_{StoThres}(\mathbf{x}) \leq 0$ mean security level and storage threshold constraints, respectively.

The optimization process follows $\Pi_{opt} = (\text{Init}, \text{FactorizePop}, \text{SubPopEA}, \text{UpdateFullPop})$ to derive the Pareto-optimal solution for block deployment.

5.1.2. Clustering-Based Grouped PBFT Consensus Mechanism

Leveraging ATM's spatiotemporal characteristics, the K-means++ clustering algorithm groups nodes by spatial location. An intra-cluster to inter-cluster consensus strategy narrows the consensus scope, reducing per-round communication rounds, lowering overhead, and improving efficiency and throughput.

5.2. Publish–Subscribe-Based Secure Data-Sharing Scheme for ATM

To meet the demand for large-scale direct data sharing among ATM participants while ensuring secure sharing policies, this scheme addresses challenges such as difficult data sharing and inefficient access control. By leveraging blockchain for managing data service publishing, querying, subscribing, and deregistration, along with attribute-based encryption (ABE) and integrity verification, it enables secure ATM data sharing and fine-grained access control. These strategies enhance both data security and sharing efficiency.

The data provider (DP) encrypts data using attribute-based encryption (ABE) and stores the ciphertext in cloud storage. The DP generates metadata (recording data keys, verification credentials, etc.) and publishes it on the blockchain to complete data registration.

Beyond storing metadata, the blockchain also maintains data-sharing records and uses smart contracts to handle requests from participants, including data registration, data querying, data subscription, data updates, and data deregistration, while data consumers (DCs) search for target data via data query contracts. If a DC meets the DP's access control requirements, it retrieves the ciphertext from cloud storage and decrypts it. DCs can also subscribe to data via data subscription contracts. Data owners can push updates periodically to subscribers via data update contracts. Data owners can terminate data distribution via data deregistration contracts.

The specific process of the data-sharing scheme is illustrated in Figure 4.

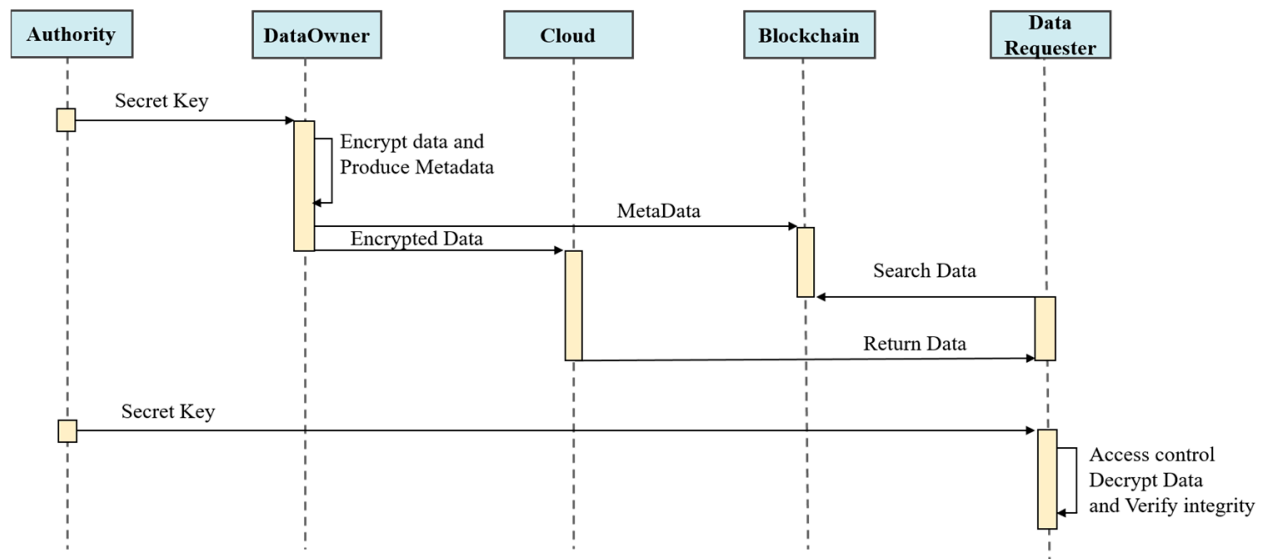


Figure 4. The specific process of the data-sharing scheme.

5.3. Privacy-Preserving and Evaluable Model-Sharing Scheme for ATM

With the advancement of ATM technology, ATM systems generate vast amounts of flight operation-related data, including flight data, flight plans, meteorological data, airport operation data, and intelligence data. Sharing and analyzing such data can facilitate activities like flight delay prediction and service assurance, thereby improving flight operation efficiency and service quality. However, concerns over data security and privacy may hinder necessary data-sharing activities.

To maximize the value of ATM data while ensuring security and privacy, it is essential to incentivize data sharing. By integrating blockchain and federated learning technologies, this scheme leverages blockchain's decentralized nature to eliminate reliance on third-party central servers, providing a secure, trustworthy, and reliable data storage environment for federated learning.

The Shapley value contribution assessment method is employed to fairly evaluate data-sharing participants, identifying low-quality data and equitably distributing rewards to incentivize high-quality data sharing. Furthermore, a reweighting strategy is applied to update local models, enhancing the performance of the global model even when some participants engage in passive training.

The specific process of model training and contribution assessment is illustrated in Figure 5.

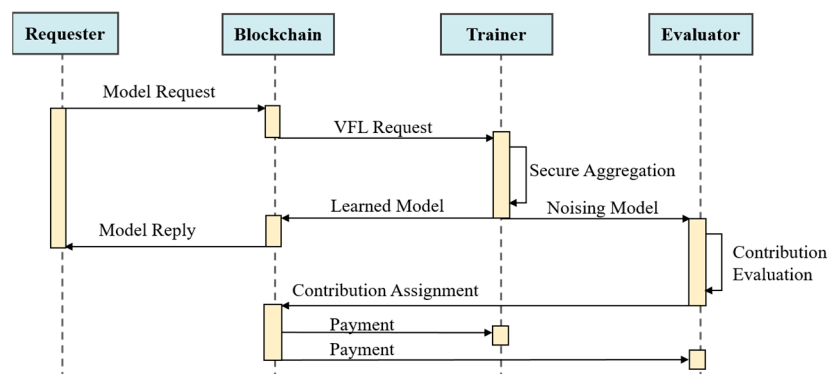


Figure 5. The specific process of model training and contribution assessment.

6. Security Analysis and Formal Verification

This chapter conducts a comprehensive security analysis and formal verification of the proposed architecture from four aspects: security analysis, TLA+ correctness verification, logical verification, and numerical experiments and validation.

6.1. Security Analysis

The advantages of the blockchain-based multi-layer aviation big data security sharing scheme are demonstrated as follows:

- (1) A blockchain-based data-sharing architecture eliminates security and privacy leakage risks associated with centralized trust mechanisms.
- (2) The optimized block deployment solution enhances both the security and efficiency of ATM data sharing in resource-constrained scenarios (e.g., airborne nodes) while achieving deep integration between blockchain and ATM operational workflows.
- (3) The publish–subscribe data security sharing scheme implements fine-grained access control and flexible, diversified data management, strengthening both security and convenience of data sharing in ATM environments.
- (4) The evaluable federated learning (FL) model sharing achieves dual privacy protection for both raw data and shared models, promotes high-quality data sharing, and prevents poisoning attacks through weighted aggregation of trained models, thereby ensuring secure data sharing.

6.2. TLA+ Correctness Verification

This section employs TLA+ formal specification to verify the correctness of the blockchain-based multi-layer aviation big data security sharing scheme. TLA+ is a formal specification language and toolset used for verifying the correctness of system designs. By defining states, system behaviors, and invariants, it helps identify potential design flaws and security issues, and it is widely applied in concurrent and distributed algorithms and systems.

When running the TLC model checker in the TLA+ Toolbox, it explores all user-defined finite states. If all generated states satisfy the specified invariants, it proves that the system executes correctly within that finite state space. This section decomposes the scheme’s optimal block deployment, searchable data sharing, and evaluable model sharing into discrete operations, defines corresponding state transitions for each operation, and uses TLA+ to verify the scheme’s correctness, ensuring all security properties are guaranteed.

6.2.1. Correctness of Optimal Block Deployment Model

The states of the optimal block deployment model are defined as shown in Table 2. All components start from the initial state.

Table 2. States of the optimal block deployment model.

Entity State	State	Description
AN_State	SBR	Node in “Storage-by-Request” phase.
	OptimizePrepared	Node prepared for block optimization.
	SOD	Node in “Storage-on-Demand” phase.
	PrunePrepared	Node prepared for block pruning.
MN_State	DPS	Node in block pruning phase.
	SBR	Master node in “Storage-by-Request” phase.
	SOD	Master node in “Storage-on-Demand” phase.
	DPS	Master node in block pruning phase.
AN_OptimizePrepared	AN	Candidate node set for block optimization.
AN_PrunePrepared	AN	Candidate node set for block pruning.
msgs	Message	Peer-to-peer communication message set.

The initial state is set with the aircraft node state AN_State as “SBR”, the master node state MN_State as SBR, and empty sets for the block optimization candidate node set as AN_OptimizePrepared, block pruning candidate node set as AN_PrunePrepared, and peer-to-peer message collection set as msgs.

Definition 5.1: System behavior of the optimal block deployment model: The system behavior is defined by the “TPNext” operation, which determines what operations are allowed to occur in the system and ensures they are executed in the correct order. The formal description of the optimal block deployment model’s system behavior is defined as:

TPNext ==

$\setminus / \setminus E \text{ an } \setminus \text{in AN} :$
 ANPrepareOptimize(an)
 $\setminus / \text{MNRcvOptimizePrepared(an)}$
 $\setminus / \text{MNCommitOptimize}$
 $\setminus / \text{ANRcvCommitOptimizeMsg(an)}$
 $\setminus / \text{ANPreparePrune(an)}$
 $\setminus / \text{MNRcvPrunePrepared(an)}$
 $\setminus / \text{MNCommitPrune}$
 $\setminus / \text{ANRcvCommitPrunedMsg(an)}$

where ANPrepareOptimize and MNRcvOptimizePrepared represent all nodes sending block optimization preparation messages to the master node and the master node receiving these messages, respectively. MNCommitOptimize and ANRcvCommitOptimizeMsg represent the master node committing block optimization deployment and other nodes receiving the optimization deployment messages, respectively. ANPreparePrune and MNRcvPrunePrepared represent all nodes sending block pruning preparation messages to the master node and the master node receiving these messages, respectively. MNCommitPrune and ANRcvCommitPrunedMsg represent the master node committing block pruning and other nodes receiving the pruning messages, respectively.

Definition 5.2: System specification of optimal block deployment model: The behavioral specification of the system includes the initial state and sequential execution of each operation, formally described as

```
TPSpec ==
TPInit[[]TPNext]_<<AN_State,MN_State,AN_OptimizePrepared,AN_PrunePrepared,
msgs>>
```

Definition 5.3: Invariants of optimal block deployment model:

(1) Before the master node commits block optimization deployment, it must receive block optimization preparation messages from all other nodes, and the master node state must be in the Block On-Demand Storage phase. The formal description is

$Inv_MNCommitOptimizeAfterPrepared == \wedge MN_State = "SBR" \wedge AN_OptimizePrepared = AN$

(2) Before the master node commits block pruning, it must receive block pruning preparation messages from all other nodes, and the master node state must be in the Block Optimized Storage phase. The formal description is

$Inv_MNCommitPruneAfterPrepared == \wedge MN_State = "SOD" \wedge AN_PrunePrepared = AN$

The TLA+ model checker was used to verify the formal specification of the optimal block deployment model. The verification results are shown in Figure 6. All invariant checks were passed, and no deadlocks were found in the entire state space. This indicates that every operation in the block optimal deployment executes correctly as designed, proving the correctness of the optimal block deployment model.

```
Starting... (2025-04-28 10:43:24)
Computing initial states...
"Optimized blockchain deployment model checking...."
Finished computing initial states: 1 distinct state generated at 2025-04-28 10:43:35.
Model checking completed. No error has been found.
```

Figure 6. Verification results of optimal block deployment model.

6.2.2. Correctness of Searchable Data-Sharing Model

The states of the searchable data-sharing model are defined as shown in Table 3. All components start from the initial state.

Table 3. States of the searchable data-sharing model.

Entity State	State	Description
SP_State	<i>Init</i>	Data provider (SP) in ready state.
	<i>Encryption</i>	SP in encryption state.
	<i>ServicePublish</i>	SP in service publishing state.
SC_State	<i>Init</i>	Data consumer (SC) in ready state.
	<i>ServiceQuery</i>	SC in service query state.
	<i>ServiceSubscribe</i>	SC in service subscription state.
AC_State	<i>Init</i>	Third-party authority (AC) in ready state.
	<i>KeyGeneration</i>	AC in key generation state.
BC_State	<i>Init</i>	Blockchain in initial state.
	<i>On-chain</i>	Blockchain in on-chain state.
SPPK_State	<i>False</i>	SP's public key not generated.
	<i>True</i>	SP's public key generated.
SPSK_State	<i>Unencrypted</i>	SP's private key unencrypted.
	<i>Encrypted</i>	SP's private key encrypted.
SCPK_State	<i>False</i>	SC's public key not generated.
	<i>True</i>	SC's public key generated.
SCSK_State	<i>False</i>	SC's private key not generated.
	<i>True</i>	SC's private key generated.
MT_State	<i>Unencrypted</i>	Shared data unencrypted.
	<i>Encrypted</i>	Shared data encrypted.
KEK_State	<i>Unencrypted</i>	Key-encrypting-key (KEK) unencrypted.
	<i>Encrypted</i>	KEK encrypted.

The initial state is configured with the data provider SP's state SP_State, data consumer SC's state SC_State, and third-party authority AC's state AC_State all set to "Ready"; the blockchain BC's state BC_State set to "Init"; the SP's public key state SPPK_State, SC's public key state SCPK_State, and SC's private key state SCSK_State all in "NotGenerated" status; and the SP's private key state SPSK_State, shared data state MT_State, and key encryption key state KEK_State for encrypting SP's private key all in "NotEncrypted" status.

Definition 5.4: System behavior of searchable data-sharing model:

The system behavior is defined by the TPNNext operation, which determines permitted operations and ensures correct execution order. The formal specification is as follows:

```
TPNNext ==
  \ / ACGenKEY
  \ / SPencMT
  \ / SPencSPSK
  \ / SPencKEK
  \ / SPPublishService
  \ / SCQueryService
  \ / SCSSubscribeService
```

In the specification, ACGenKEY denotes that AC generates a public key for SP and a private key for SC; SPencMT represents SP encrypting the shared data MT; SPencSPSK indicates SP encrypting its own private key; SPencKEK signifies SP encrypting the KEK; SPPublishService means SP publishing the service to the blockchain; SCQueryService refers to SC querying the service from the blockchain; and SCSSubscribeService describes SC subscribing to the service from the blockchain.

Definition 5.5: System specification:

The system specification includes initial state and sequential operation execution:

```
TPSpec ==
  TPInit[[TPNNext]_<<SP_State,SC_State,AC_State,BC_State,SPPK_State,SPSK_State,
  SCPK_State,SCSK_State,MT_State,KEK_State>>
```

Definition 5.6: Model invariants:

- (1) Service publishing invariant: SP must be in the encrypted state with both MT and KEK encrypted before publishing service:

$$\text{Inv_SPPublishServiceAfterEncrypted} == \backslash \text{SP_State} = \text{"Encryption"} / \backslash \text{MT_State} = \text{"Encrypted"} / \backslash \text{KEK_State} = \text{"Encrypted"}$$

- (2) Service query invariant: SP must have published service and SC must have generated keys before querying:

$$\text{Inv_SCQueryServiceAfterPublish} == \backslash \text{SP_State} = \text{"ServicePublish"} / \backslash \text{SCPK_State} = \text{"True"} / \backslash \text{SCSK_State} = \text{"True"}$$

- (3) Service subscription invariant: SP must have published service and SC must have queried it before subscribing:

$$\text{Inv_SCSubscribeServiceAfterQuery} == \backslash \text{SP_State} = \text{"ServicePublish"} / \backslash \text{SC_State} = \text{"ServiceQuery"}$$

Verification results: The TLA+ model checker validated the formal specification (see Figure 7). All invariants passed, and no deadlocks were found in the state space, confirming the correctness of the searchable data-sharing model.

```

Starting... (2025-04-28 14:32:06)
Computing initial states...
"Data sharing model based on CP-ABE checking...."
Finished computing initial states: 1 distinct state generated at 2025-04-28 14:32:08.
Model checking completed. No error has been found.

```

Figure 7. Model-checking results for searchable data sharing.

6.2.3. Correctness of Evaluable Model-Sharing Model

The states of the evaluable model-sharing model are defined as shown in Table 4. All components start from the initial state.

Table 4. States of the evaluable model-sharing model.

Entity State	State	Description
RN_State	<i>Init</i>	Data requester (RN) initialized.
	<i>Bid</i>	RN in bid-publishing phase.
	<i>Waited</i>	RN awaiting responses.
	<i>Obtained</i>	RN in data-obtained state.
PN_State	<i>Init</i>	Data provider (PN) initialized.
	<i>TrainPrepared</i>	PN prepared for FL model training.
	<i>Trained</i>	PN in FL training state.
	<i>EvaluatePrepared</i>	PN prepared for FL model evaluation.
PN_Prepared	<i>Evaluated</i>	PN in FL evaluation state.
	<i>AggregatePrepared</i>	PN prepared for FL model aggregation.
	<i>PN</i>	Set of nodes prepared for model sharing.
BC_State	<i>Init</i>	Blockchain initialized.
	<i>Trained</i>	Blockchain in FL training state.
	<i>Aggregated</i>	Blockchain in FL aggregation state.
Model_State	<i>False</i>	FL model initialized.
	<i>Trained</i>	FL model trained.
	<i>Evaluated</i>	FL model evaluated.
	<i>Aggregated</i>	FL model aggregated.
msgs	<i>Message</i>	Peer-to-peer message set.

The initial state is configured with the data requester node RN's state RN_State and data provider node PN's state PN_State both set to "Initialized", the set of all nodes prepared for model-sharing state PN_Prepared initialized as an empty set, the blockchain BC's state BC_State set to "Init", the FL model's state Model_State in initialized status, and the peer-to-peer communication message set msgs initialized as an empty collection.

Definition 5.7: System behavior of evaluable model-sharing model:

The system behavior is defined by the TPNNext operation, which determines permitted operations and ensures correct execution order. The formal specification is as follows:

```

TPNNext ==
\ / \E pn \in PN:
\ / RNBiddingPN
\ / PNRcvBiddingMsg(pn)
\ / RNRcvBidPreparedMsg(pn)
\ / RNCommitBidMsg
\ / PNTrainModel(pn)
\ / PNEvaluatePreparedMsg(pn)

```

\ / BCRcvEvaluatePreparedMsg(pn)
 \ / BCCCommitEvaluatedMsg
 \ / PNRcvCommitEvaluatedMsg(pn)
 \ / PNAggregatePreparedMsg(pn)
 \ / BCRcvAggregatePreparedMsg(pn)
 \ / BCCCommitAggregated

Herein, RNBiddingPN denotes that RN sends data request messages to all nodes on the blockchain; PNRcvBiddingMsg indicates that PN nodes receive the data request messages; RNRcvBidPreparedMsg represents RN receiving the complete set of PN nodes prepared to participate in model sharing; RNCommitBidMsg signifies RN submitting the message about participating model-sharing nodes; PNTrainModel shows that PN nodes receive the participation messages and subsequently initiate FL model training; PNEvaluatePreparedMsg demonstrates that PN nodes, upon completing model training, transmit model evaluation preparation messages to the blockchain; BCRcvEvaluatePreparedMsg reflects the blockchain receiving these model evaluation preparation messages from PN nodes; BCCCommitEvaluatedMsg embodies the blockchain committing the model evaluation messages; PNRcvCommitEvaluatedMsg portrays PN nodes receiving the model evaluation messages from the blockchain; PNAggregatePreparedMsg illustrates PN nodes sending model aggregation preparation messages to the blockchain after completing model evaluation; BCRcvAggregatePreparedMsg captures the blockchain receiving these model aggregation preparation messages from PN nodes; and BCCCommitAggregated ultimately represents the blockchain committing the final model aggregation message.

Definition 5.8: System specification:

The system specification includes initial state and sequential operation execution:

TPSpec ==

TPInit/\[[TPNext]_<<RN_State,PN_State,PN_Prepared,Modle_State,BC_State,msgs>>

Definition 5.9: Model invariants:

- (1) Model training invariant: PN must be in training-prepared state with FL model initialized before training:

Inv_PNTrainModelAfterPrepared==/\PN_State="TrainPrepared"/\Modle_State="False"

- (2) Model evaluation invariant: BC must receive messages from all evaluation-prepared nodes before committing:

Inv_BCCCommitEvaluatedMsgAfterPrepared==/\PN_Prepared=PN

- (3) Model aggregation invariant: BC must receive messages from all aggregation-prepared nodes before committing:

Inv_BCCCommitAggregatedMsgAfterPrepared==/\PN_Prepared=PN

Verification results: The TLA+ model checker validated the formal specification (see Figure 8). All invariants passed with no deadlocks in the state space, confirming the evaluable model-sharing model's correctness.

```

Starting... (2025-04-28 15:50:52)
Computing initial states...
"Decentralized and evaluable VFL model checking...."
Finished computing initial states: 1 distinct state generated at 2025-04-28 15:50:54.
Progress(20) at 2025-04-28 15:50:57: 290,791 states generated (290,791 s/min), 39,284 distinct states found
(39,284 ds/min), 14,536 states left on queue.
Model checking completed. No error has been found.

```

Figure 8. Model-checking results for evaluable model sharing.

6.3. Logical Verification

This section conducts formal definition and logical verification analysis of the blockchain-based multi-layer aviation big data security sharing scheme using Nash equilib-

rium from game theory. The core of game theory lies in how rational participants select strategies to optimally respond to others' decisions in interactive environments. This analytical method aims to ensure security, privacy, and effectiveness throughout the data-sharing process. Below are relevant definitions and theorems for this model.

Let $P = \{P_1, P_2, \dots, P_N\}$ represent the set of all nodes participating in the multi-layer aviation big data security sharing, $\chi = \{\chi_1, \chi_2, \dots, \chi_n\}$ denote the action strategy space of all participating nodes during sharing, and $U = \{u_1, u_2, \dots, u_Q\}$ represent the utility functions of all participating nodes under different game outcomes. The game payoff for node P_i under strategy combination $\{\chi_1, \chi_2, \dots, \chi_N\}$ can be expressed as $u_i : \chi \rightarrow V$.

Lemma 1. *The utility function $u_i(x), \forall x \in [0, 1]$ is monotonically increasing. By definition, $u_i(x)$ represents the probability that a participant can at least determine message reception/sending by others or at most obtain information with probability x . Since all participants act rationally in data sharing—meaning higher x yields greater game payoff and thus higher $u_i(x)$ for any $\forall x_1, x_2 \in [0, 1]$ where $x_1 \leq x_2$ —we have $u_1(x) \leq u_2(x)$, proving $u_i(x)$ is monotonically increasing.*

Definition 1. *Nash equilibrium. Let $r((\chi_i)_{i \in P})$ denote the game outcome when participants execute strategy combination $(\chi_i)_{i \in P}$. A strategy combination $(\chi_i^*)_{i \in P}$ constitutes a Nash equilibrium if and only if no participant can unilaterally change strategy to obtain better outcomes when others' strategies are known, i.e., $\forall i \in P, r(\chi_i, (\chi_j^*)_{j \in P \setminus i}) \leq r(\chi_i^*, (\chi_j^*)_{j \in P \setminus i})$. As all participants are rational, none will deviate from χ_i^* . Generally, a game may yield multiple Nash equilibria.*

Definition 2. *Rational security protocol. Let $\mu(M) = \{\mu_1(M_1), \mu_2(M_2), \dots, \mu_N(M_N)\}$ be the set of security communication protocols followed by all participants during data sharing. The communication protocol game is denoted as $G_{\alpha, \beta, \gamma}^{S, R, A, AviationChain}(M)$, where S denotes sender, R receiver, A attacker, and $AviationChain$ the sharing environment. Here, α represents the minimum probability that sender S confirms correct receipt by R , β the minimum probability that R confirms M 's origin from S , and γ the maximum probability that A obtains M . $\mu(M)$ is rationally secure if and only if the following holds:*

The strategy combination $(\chi_S^*, \chi_R^*, \chi_A^*)$ forms a Nash equilibrium in $G_{\alpha, \beta, \gamma}^{S, R, A, AviationChain}(M)$ producing optimal outcome $r = (\alpha, \beta, \gamma)$;

For any Nash equilibrium in this game, $u_i(\chi_S, \chi_R, \chi_A) \leq u_i(\chi_S^*, \chi_R^*, \chi_A^*)$.

Verification:

Per game definition, (α, β, γ) is the optimal outcome—no strategy change yields better results. Thus, if $(\chi_S^*, \chi_R^*, \chi_A^*)$ achieves (α, β, γ) , it constitutes a Nash equilibrium.

By Nash equilibrium definition, for any equilibrium (χ_S, χ_R, χ_A) with outcome $(\alpha_1, \beta_1, \gamma_1)$, the optimal (α, β, γ) dominates. From Lemma 1, $u_i(\chi_S, \chi_R, \chi_A) \leq u_i(\chi_S^*, \chi_R^*, \chi_A^*)$ holds for all participants.

Figure 9 presents the game tree of the proposed multi-level security sharing model for aviation big data, where the optimal game outcome is (α, β, γ) , generated by the Nash equilibrium action sequence. As shown in Table 5, individual security functions cannot guarantee comprehensive security throughout the aviation big data-sharing process. The proposed integrated solution—incorporating optimized block deployment, searchable data sharing, and evaluable model sharing—establishes multi-level security protection for aviation big data sharing. This achieves the game-theoretic mechanism's security assurance for rational protocols.

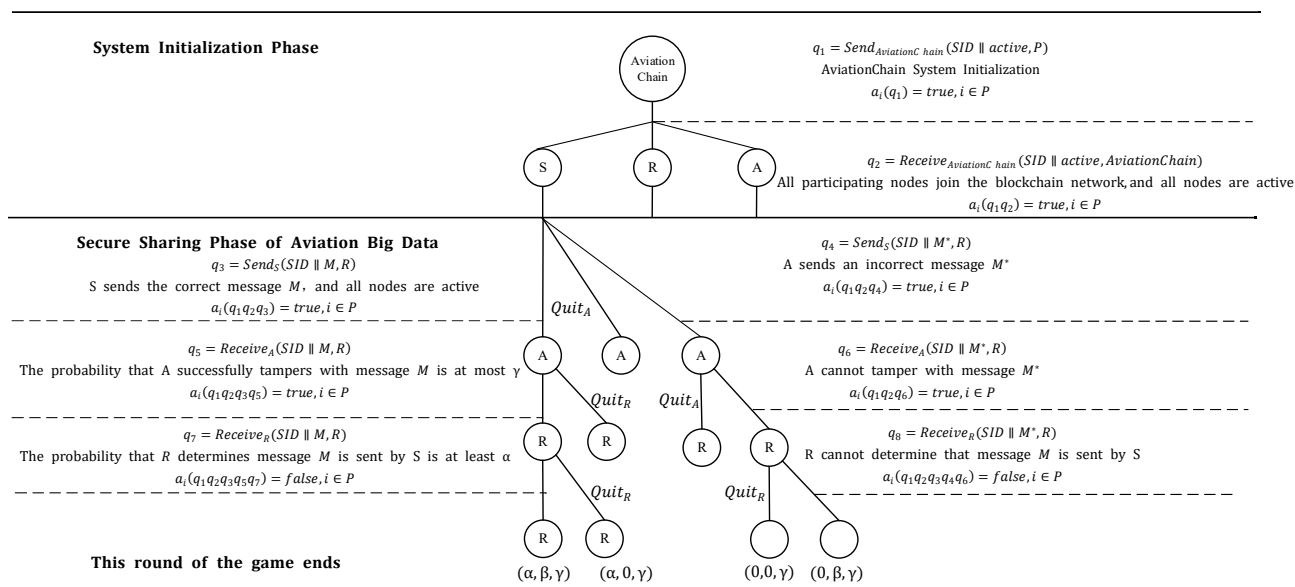


Figure 9. Game tree model analysis diagram.

Table 5. Security protocol game-theoretic analysis comparison.

Scheme	Block Optimization Deployment	Searchable Data Sharing	Evaluable Model Sharing	Multi-Level Data-Sharing Model
Authentication	×	×	✓	✓
Authorization	×	✓	×	✓
Privacy	×	×	✓	✓
Nash equilibrium	$q_1q_2q_4\text{Quit}_A\text{Quit}_R$	$q_1q_2q_4q_6\text{Quit}_R$	$q_1q_2\text{Quit}_S\text{Quit}_A\text{Quit}_R$	$q_1q_2q_3q_5q_7$
Security requirements	×	×	×	✓

6.4. Numerical Experiments and Validation

To validate the blockchain deployment optimization and consensus mechanism, we construct a realistic and reliable test scenario. Real-time aircraft position data from the OpenSky Network platform [33] is used to simulate a practical ATM data-sharing environment. For blockchain deployment optimization, we adopt our proposed IF-NSGA-II algorithm with the following parameter settings: the number of objective functions is 3, the number of constraints is 3, the crossover probability is 0.85, and the mutation probability is 0.15. For the consensus algorithm, we use our proposed C-GPBFT, where the total number of nodes is set to 100 and the number of sub-clusters is 4.

We compare IF-NSGA-II with the benchmark algorithm NSGA-II [34] in terms of solution quality and diversity. We randomly select one run result of two algorithms with a population size of 100 and visualize the non-dominated solution sets for the block allocation optimization problem. As shown in Figure 10, the x, y, and z axes represent the average weighted storage occupancy rate, average response time, and average unavailability probability, respectively. It can be seen that IF-NSGA-II achieves better coverage of the approximate non-dominated solution space than NSGA-II, with the approximate non-dominated solutions obtained by IF-NSGA-II exhibiting the widest distribution in the objective space. Furthermore, IF-NSGA-II contributes the most to the overall set of approximate non-dominated solutions, accounting for 92.24%, while NSGA-II contributes only 7.76%.

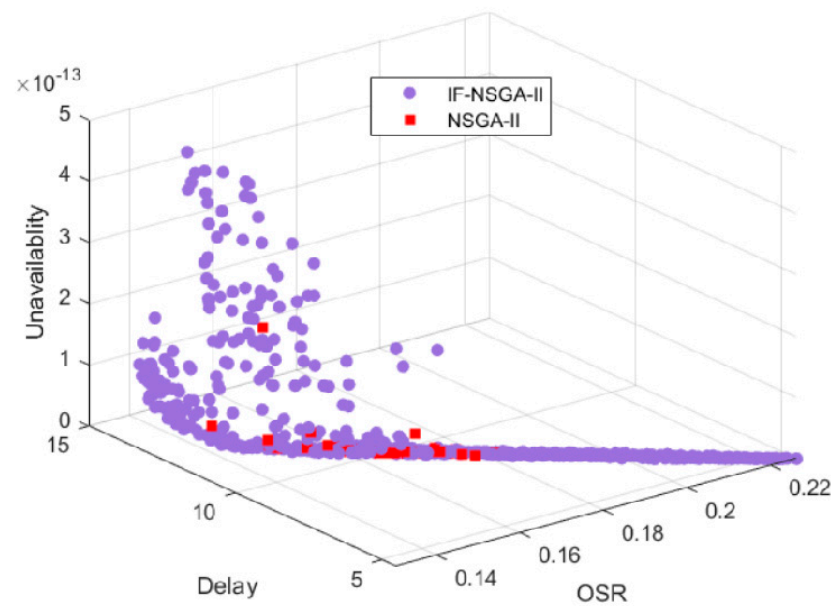


Figure 10. Optimization results of IF-NSGA-II and NSGA-II with population size 100.

To demonstrate the advantages of C-GPBFT in ATM data sharing, we conduct a comparison with the traditional PBFT [35] algorithm. First, the aircraft nodes are divided into four sub-clusters, and the clustering results are shown in Figure 11. Then, we evaluate the consensus latency and throughput (TPS) of both algorithms, with the results presented in Figures 12 and 13, respectively.

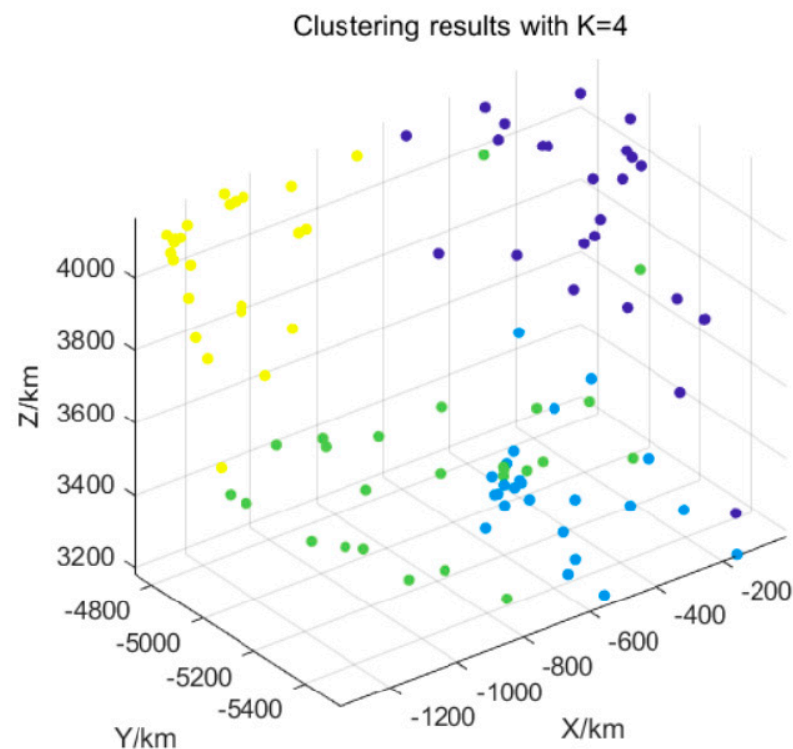


Figure 11. Clustering results with 4 sub-clusters (the total number of nodes is 100, and different colors represent different sub-clusters).

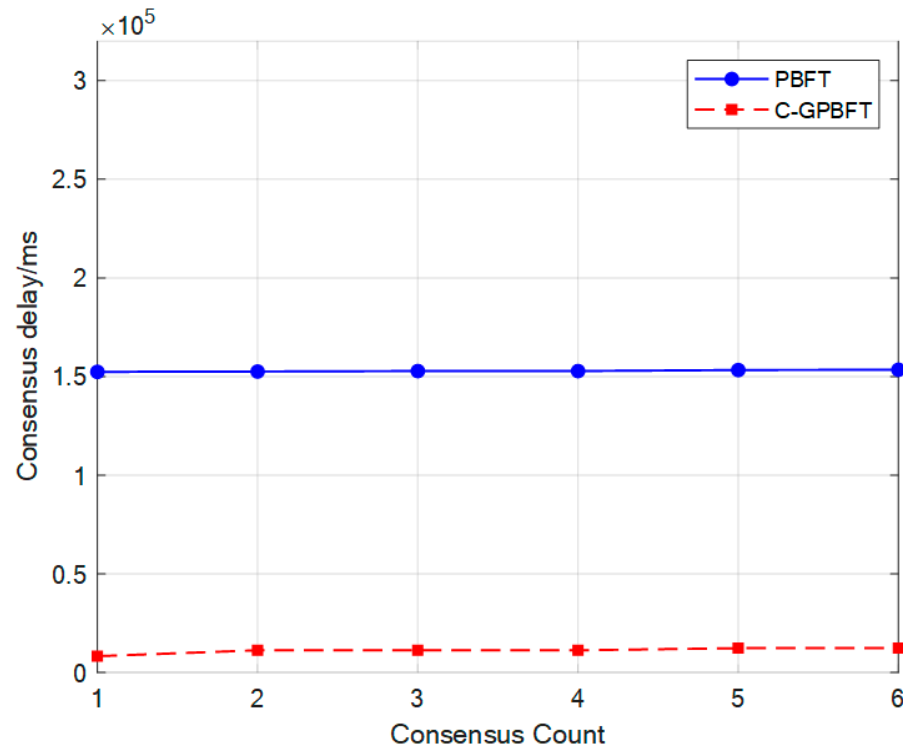


Figure 12. Consensus delay of C-GPBFT and PBFT in a dynamic cluster.

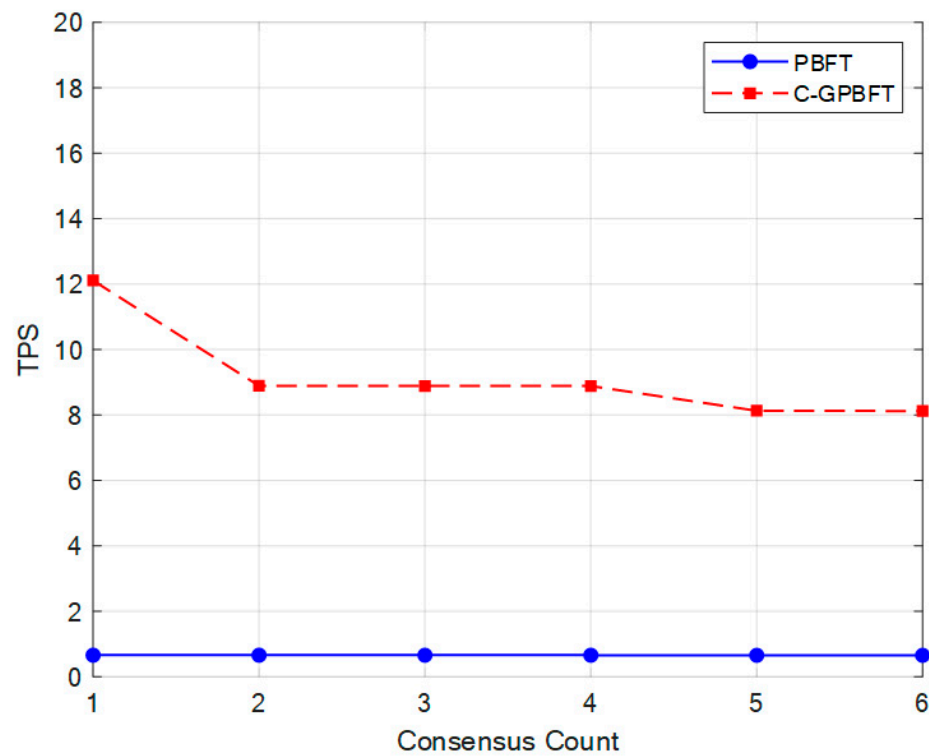


Figure 13. Throughput of C-GPBFT and PBFT in a dynamic cluster.

Figure 12 presents a comparison of consensus latency between the two algorithms. In the prepare and commit phases of PBFT, each aircraft node in the consensus domain broadcasts messages to all other nodes in the network, resulting in a large number of communications for each consensus and consequently higher latency. In contrast, C-GPBFT adopts a clustering strategy that divides nodes into multiple sub-clusters based on their spatial distribution, significantly narrowing the consensus domain and reducing the

number of required communications. Furthermore, as the number of consensus increases and node positions change in real time, C-GPBFT can dynamically update the clustering based on the node distribution, allowing the consensus latency to remain stable.

Figure 13 presents a comparison of throughput between the two algorithms. Since C-GPBFT has significantly lower consensus latency compared to PBFT, its throughput is considerably higher. Moreover, as the clusters are dynamically updated, the throughput exhibits only minor fluctuations across different consensus counts. This indicates that the proposed C-GPBFT algorithm has excellent adaptability to dynamic environments and holds promising practical potential in highly dynamic ATM scenarios.

7. Limitations

Although the proposed multi-layered secure sharing framework for aviation big data demonstrates promising theoretical performance in terms of security, privacy, and efficiency, several practical challenges and limitations remain, which require further exploration and refinement in future research.

First, the integration of blockchain with existing ATM data systems presents a significant challenge. Current aviation systems rely heavily on legacy infrastructures and standardized interfaces, yet no unified blockchain protocols or interfacing standards have been established for this domain. Large-scale system renovations may incur high costs and technical complexity, particularly within globally interconnected ATM networks. Future work may consider introducing middleware technologies to bridge blockchain with traditional infrastructures, thereby minimizing disruption and improving deployment feasibility.

Second, energy consumption remains an important concern. Although this work adopts a consortium blockchain to avoid the high energy costs of public chains, the collaborative learning scenario involving frequent interactions and multi-party participation still incurs non-negligible computational overhead. In light of global efforts toward green aviation and carbon reduction, subsequent research should explore more energy-efficient consensus mechanisms such as lightweight protocols and investigate edge–cloud collaborative architectures to optimize resource allocation.

Third, regulatory and legal adaptation for blockchain in aviation is still lacking. Currently, there is no well-established global regulatory framework for blockchain applications in civil aviation. Cross-border data sharing raises issues such as data sovereignty, accountability, and jurisdictional ambiguity. Future work could explore the design of auditable smart contracts, compliance-oriented templates, and participant authorization mechanisms. Collaboration with international bodies such as ICAO may be necessary to establish common standards and facilitate regulatory alignment.

Fourth, ethical issues must not be overlooked when deploying federated learning in critical infrastructures such as ATM systems. When federated learning models are involved in safety-critical decision-making, challenges may arise during model collaboration, including model bias, lack of decision transparency, and ambiguous responsibility attribution. If participants' data contain structural biases, it can adversely affect the overall model's performance and fairness. Moreover, since model updates and decision processes are mainly conducted locally without a unified auditing mechanism, future work should explore mechanisms such as explainable models and auditable smart contracts to enhance system trustworthiness.

Fifth, the deployment of blockchain-based data-sharing mechanisms on a global scale also faces significant data governance challenges. In the international aviation environment, cross-jurisdictional data flows may trigger issues related to compliance with data movement regulations, responsibility attribution, and the delineation of regulatory authority.

Currently, there is a lack of unified standards and protocols for cross-border inter-chain data governance. Subsequent research can integrate technologies such as verifiable privacy-preserving computation to build interoperable data governance frameworks, supporting secure and efficient international collaboration.

In summary, the architecture proposed in this paper provides a valuable exploratory path toward secure and trustworthy data sharing in ATM systems. However, further improvement and development are needed in areas including standardization, energy efficiency optimization, regulatory adaptation, ethical application, and cross-border data governance.

8. Conclusions

This paper analyzes the practical data-sharing requirements in aviation operations and establishes a mathematical model for secure aviation big data sharing. Building on this foundation, we propose a multi-layer security sharing architecture that addresses the storage and timeliness constraints in aviation data sharing through blockchain optimization deployment and consensus mechanisms. By implementing a dual-layer sharing framework combining data sharing and model sharing, we achieve comprehensive aviation big data sharing while providing targeted support for ATM operations.

Based on real-world aviation operational scenarios, this study constructs a mathematical model for secure aviation big data sharing and innovatively develops a multi-layer security architecture that effectively resolves the critical challenges of data security and real-time performance in aviation data sharing. At the technical implementation level, we employ a dynamic deployment strategy to significantly reduce storage pressure on lightweight nodes, design a clustering-based grouped PBFT consensus mechanism to meet the timeliness requirements of critical business data, and establish a dual-layer architecture integrating data sharing with model sharing to ensure security, efficiency, privacy protection, and sharing incentives for ATM data sharing. This solution provides secure and efficient data-sharing support for ATM operations while offering a viable technical pathway for the digital transformation of air traffic management systems.

The proposed approach demonstrates three key technical innovations: the dynamic deployment strategy effectively alleviates storage constraints for resource-limited nodes, the optimized consensus mechanism ensures timely processing of critical aviation data, and the integrated sharing framework simultaneously addresses security, privacy, and efficiency requirements. Together, these advancements establish a comprehensive solution for aviation big data sharing that meets the stringent demands of modern air traffic management systems while supporting their ongoing digital transformation.

Author Contributions: Conceptualization, Z.W. and Q.W.; Methodology, Q.W.; Software, Q.W.; Validation, Q.W., Z.W. and Y.L.; Formal analysis, Q.W.; Investigation, Q.W.; Resources, Z.W.; Data curation, Q.W.; Writing—original draft preparation, Q.W.; Writing—review and editing, Z.W. and Y.L.; Visualization, Q.W.; Supervision, Z.W.; Project administration, Zhijun Wu; Funding acquisition: Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key R&D Program of China (No. 2023YFB4302901), National Natural Science Foundation of China (No. 62172418), Tianjin Natural Science Foundation Project (No. 24JCZDJC01000), and Fundamental Research Funds for the Central Universities of China (No. 3122022081).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in this study is available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: <https://bitcoin.org/bitcoin.pdf> (accessed on 3 August 2025).
2. Buterin, V. A Next-Generation Smart Contract and Decentralized Application Platform. *White Pap.* **2014**, *3*, 1–36. Available online: <https://ethereum.org/en/whitepaper/> (accessed on 3 August 2025).
3. Yakovenko, A. Solana: A New Architecture for a High Performance Blockchain. Solana Whitepaper, v0.8.13. 2018. Available online: <https://solana.com/solana-whitepaper.pdf> (accessed on 3 August 2025).
4. Lu, X.; Wu, Z.; Cao, J. ATMChain: Blockchain-Based Security Architecture for Air Traffic Management in Future. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 3872–3896. [[CrossRef](#)]
5. Clementi, M.D.; Larrieu, N.; Lochin, E.; Kaafar, M.A.; Asghar, H. When Air Traffic Management Meets Blockchain Technology: A Blockchain-Based Concept for Securing the Sharing of Flight Data. In Proceedings of the 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 8–12 September 2019; pp. 1–10. [[CrossRef](#)]
6. Zhang, L.; Zheng, Z.Y.; Yuan, Y. A Controllable Sharing Model for Electronic Health Records Based on Blockchain. *Acta Autom. Sin.* **2021**, *47*, 2143–2153. [[CrossRef](#)]
7. Tong, F.; Shao, R.R. Study on Blockchain Based Access Control Model for Cloud Data. *Comput. Sci.* **2023**, *50*, 16–25. [[CrossRef](#)]
8. Wei, X.; Yan, Y.; Guo, S.; Qiu, X.; Qi, F. Secure Data Sharing: Blockchain-Enabled Data Access Control Framework for IoT. *IEEE Internet Things J.* **2021**, *9*, 8143–8153. [[CrossRef](#)]
9. Jiang, S.; Cao, J.; Wu, H.; Chen, K.; Liu, X. Privacy-preserving and efficient data sharing for blockchain-based intelligent transportation systems. *Inf. Sci.* **2023**, *635*, 72–85. [[CrossRef](#)]
10. Wang, T.; Wu, Q.; Chen, J.; Chen, F.; Xie, D.; Shen, H. Health data security sharing method based on hybrid blockchain. *Futur. Gener. Comput. Syst.* **2023**, *153*, 251–261. [[CrossRef](#)]
11. Otoum, S.; Ridhawi, I.A.; Mouftah, H. A Federated Learning and Blockchain-Enabled Sustainable Energy Trade at the Edge: A Framework for Industry 4.0. *IEEE Internet Things J.* **2023**, *10*, 3018–3026. [[CrossRef](#)]
12. Singh, S.; Rathore, S.; Alfarraj, O.; Tolba, A.; Yoon, B. A Framework for Privacy-Preservation of IoT Healthcare Data Using Federated Learning and Blockchain Technology. *Future Gener. Comput. Syst.* **2022**, *129*, 380–388. [[CrossRef](#)]
13. Misic, J.; Misic, V.B.; Chang, X.; Qushtom, H. Adapting PBFT for Use With Blockchain-Enabled IoT Systems. *IEEE Trans. Veh. Technol.* **2021**, *70*, 33–48. [[CrossRef](#)]
14. Li, W.; Feng, C.; Zhang, L.; Xu, H.; Cao, B.; Imran, M.A. A Scalable Multi-Layer PBFT Consensus for Blockchain. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 1146–1160. [[CrossRef](#)]
15. Tang, F.; Xu, T.; Peng, J.; Gan, N. TP-PBFT: A Scalable PBFT Based on Threshold Proxy Signature for IoT-Blockchain Applications. *IEEE Internet Things J.* **2024**, *11*, 15434–15449. [[CrossRef](#)]
16. Hasanzadeh-Nazarabadi, Y.; Ku, A. LightChain: Scalable DHT-Based Blockchain. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 2582–2593. [[CrossRef](#)]
17. She, W.; Huo, L.; Tian, Z.; Zhuang, Y.; Niu, C.; Liu, W. A Double Steganography Model Combining Blockchain and Interplanetary File System. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 3029–3042. [[CrossRef](#)]
18. Yu, Y.; Liu, S.; Yeoh, P.L.; Vucetic, B.; Li, Y. LayerChain: A Hierarchical Edge-Cloud Blockchain for Large-Scale Low-Delay Industrial Internet of Things Applications. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5077–5086. [[CrossRef](#)]
19. Qi, X.; Zhang, Z.; Jin, C.; Zhou, A. A Reliable Storage Partition for Permissioned Blockchain. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 14–27. [[CrossRef](#)]
20. Yu, B.; Li, X.; Zhao, H. Virtual Block Group: A Scalable Blockchain Model with Partial Node Storage and Distributed Hash Table. *Comput. J.* **2020**, *63*, 1524–1536. [[CrossRef](#)]
21. Büinz, B.; Kiffer, L.; Luu, L.; Zamani, M. FlyClient: Super-Light Clients for Cryptocurrencies. In Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 18–20 May 2020; pp. 928–946. [[CrossRef](#)]
22. Organisation de L’Aviation Civile Internationale. Global Air Traffic Management Operational Concept. ICAO. 2005. Available online: https://www2023.icao.int/Meetings/anconf12/Document%20Archive/9854_cons_en%5B1%5D.pdf (accessed on 3 August 2025).
23. International Civil Aviation Organization (ICAO). 10039—Manual on System Wide Information Management (SWIM) Concept. International Civil Aviation Organization. 1999. Available online: https://www2023.icao.int/safety/acp/ACPWGF/CP%20WG-I%2019/10039_SWIM%20Manual.pdf (accessed on 3 August 2025).
24. Scheelhaase, J.; Maertens, S. How to improve the global ‘carbon offsetting and reduction scheme for international aviation’ (CORSIA)? *Transp. Res. Procedia* **2020**, *51*, 108–117. Available online: <https://www.icao.int/environmental-protection/CORSIA/Pages/default.aspx> (accessed on 3 August 2025). [[CrossRef](#)]

25. Zaporozhets, O. Balanced approach to aircraft noise management. In *Aviation Noise Impact Management: Technologies, Regulations, and Societal Well-being in Europe*; Springer International Publishing: Cham, Switzerland, 2022; pp. 29–56. Available online: <https://www.icao.int/environmental-protection/Pages/noise.aspx> (accessed on 3 August 2025).
26. International Civil Aviation Organization (ICAO). Global Aviation Security Plan (GASeP). 2022. Available online: <https://www2023.icao.int/Security/Documents/GLOBAL%20AVIATION%20SECURITY%20PLAN%20EN.pdf> (accessed on 3 August 2025).
27. International Civil Aviation Organization (ICAO). *Aviation Cybersecurity Strategy*; ICAO: Montreal, QC, Canada, 2019. Available online: <https://www2023.icao.int/aviationcybersecurity/Documents/AVIATION%20CYBERSECURITY%20STRATEGY.EN.pdf> (accessed on 3 August 2025).
28. International Civil Aviation Organization (ICAO). *Annex 17 to the Convention on International Civil Aviation: Safeguarding International Civil Aviation Against Acts of Unlawful Interference*, 12th ed.; ICAO: Montreal, QC, Canada, 2022. Available online: <https://ffac.ch/wp-content/uploads/2020/10/ICAO-Annex-17-Security.pdf> (accessed on 3 August 2025).
29. International Civil Aviation Organization (ICAO). *Global Aviation Security Plan*, 2nd ed.; ICAO: Montreal, QC, Canada, 2024. Available online: <https://www2023.icao.int/Security/Documents/GLOBAL%20AVIATION%20SECURITY%20PLAN%202nd%20Ed.EN.pdf> (accessed on 3 August 2025).
30. Wang, Q.; Lu, X.; Wu, Z. A Secure Sharing Framework Based on Cloud-Chain Fusion for SWIM. In *Proceedings of the International Conference, DataCom 2022, Denarau Island, Fiji, 8–10 December 2022*; Springer: Singapore, 2022; pp. 127–141.
31. International Civil Aviation Organization (ICAO). *2013–2028 Global Air Navigation Plan*, 4th ed.; Doc 9750; ICAO: Montreal, QC, Canada, 2013. Available online: <https://www2023.icao.int/ESAF/Documents/meetings/2013/ASBU/en/WP04.GANP-4th%20edition.pdf> (accessed on 3 August 2025).
32. Wyman, O. *Aviation’s Data Science Revolution: Optimizing Operations Through Connectivity*. White Paper, New York. 2017. Available online: <https://www.oliverwyman.com/our-expertise/insights/2017/jun/aviation-s-data-science-revolution.html> (accessed on 3 August 2025).
33. Schafer, M.; Strohmeier, M.; Lenders, V.; Martinovic, I.; Wilhelm, M. Bringing up OpenSky: A Large-Scale ADS-B Sensor Network for Research. In *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks (IPSN-14)*, Berlin, Germany, 15–17 April 2014; pp. 83–94. [CrossRef]
34. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
35. Castro, M. Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM Trans. Comput. Syst.* **2002**, *20*, 398–461. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.