ITEJ
Information Technology Engineering Journals
eISSN : 2548-2157

Url : https://syekhnurjati.ac.id/journal/index.php/itej
Email : itej@syekhnurjati.ac.id

Implementation of Ethereum-Based Blockchain Technology for ADS-B Data Security and Validation

Kgs M Farhan Rabbaniansyah
Telecommunication Engineering
State Polytechnic of Sriwijaya
rabbaniansyah66@gmail.com

Lindawati Lindawati
Telecommunication Engineering
State Polytechnic of Sriwijaya
lindawati@polsri.ac.id

Sopian Soim
Telecommunication Engineering
State Polytechnic of Sriwijaya
sopiansoim@gmail.com

Automatic Dependent Surveillance–Broadcast (ADS-B) has significantly enhanced air traffic monitoring by enabling real-time broadcasting of aircraft positions and identifiers. However, it is intrinsically susceptible to spoofing, replay, and tampering attacks because to its lack of cryptographic protections. This paper introduces a blockchain-based ADS-B validation system for safe, decentralized data authentication that makes use of Ethereum smart contracts and MetaMask. The suggested system uses Solidity-coded rules to enforce logical limitations on altitude changes, timestamp order, and geographic displacement in order to validate incoming flight messages. Every data transaction is protected by two layers of security: the smart contract's automated detection and MetaMask's manual permission. This combines operational control with the immutability of blockchain technology by guaranteeing that even reported anomalies cannot be committed without human consent. The OpenSky Network flight data was used to test the system, and 56 attack simulations in the spoofing, replay, and tampering categories were run. All accepted anomalies were purposefully allowed to test forensic transparency, and the contract obtained a 92.9% detection rate. All transaction information were retained in Ethereum's transparent ledger, enhancing its suitability for incident investigation and regulatory compliance. The findings support blockchain's applicability in preventing unwanted changes to aircraft telemetry. Stricter timestamp constraints, machine learning for anomaly detection, and interaction with international aircraft registries to better spoofing detection are possible future improvements. This approach combines the practical needs of air traffic with the potential advantages of blockchain technology.

Keywords— ADS-B, blockchain, Ethereum, smart contracts, cybersecurity, aviation security, spoofing, replay attack, tampering, MetaMask, data integrity, OpenSky Network

I. Introduction

Automatic Dependent Surveillance–Broadcast (ADS-B) has extensively reshaped contemporary air traffic management through the ability of aircraft to periodically broadcast autonomous real-time flight parameters—geolocation, altitude, and identification—directly to air traffic management systems and neighboring aircraft. The aviation system's operating efficiency and situational awareness are enhanced by this function. Additionally, sensor networks like OpenSky enhance the usefulness of ADS-B data by compiling flight broadcasts from thousands of aircraft into comprehensive datasets suitable for research[1], [2].

However, because ADS-B systems are not encrypted or authenticated, they are inherently insecure and vulnerable to spoofing, replay, and tampering attacks, among other cyberthreats. There are serious safety issues associated with using off-the-shelf Software-Defined Radios (SDRs) to interrupt or forge ADS-B communications, as several studies have experimentally verified [3]. To deceive control towers and onboard collision avoidance systems, for example, attackers can delay the delivery of communications, inject phony aircraft signals, or pose as authentic identifiers. Strict data integrity procedures tailored to the dynamic and vital air traffic communication environment are necessary to counter these threats. [4], [5].

Blockchain technology is increasingly being researched as a robust flight telemetry protection infrastructure to fill these shortcomings. Because blockchain is decentralized, it naturally provides consensus-based validation and tamper-evident storage, which are both very desirable for applications that require safe and auditable data provenance. Numerous applications have shown how blockchain technology may be used to store ADS-B data, allowing for transparent air traffic management, verifiable audit trails, and real-time verification[6], [7].

More importantly, rule-based, automated validation of received flight data is made possible by the use of smart contracts in Ethereum blockchains. Domain-specific safety requirements, such as logical altitude changes, velocity limits, and timestamp consistency, can be enforced via smart contracts [7], [8], [9]. In order to detect abnormalities, such logic can be encoded once and run repeatedly, reducing the requirement for centralized authorities. Adding MetaMask, a secure wallet that runs in the browser, adds another user-facing layer of verification where operators or systems can manually accept or reject suspicious data submissions before they are committed to the chain. This two-phase validation procedure improves overall accountability and data fidelity.

In this study, Ethereum smart contracts are used in conjunction with MetaMask to create an ADS-B system with improved security. Received flight data is screened by the system for signs of spoofing (like improbable geographical leaps), tampering (like sudden changes in altitude), and replay (like unaltered timestamps). For example, if the new timestamp is not greater than the previous one, replay detection is triggered [10]. Tampering is identified by determining whether the vertical rate exceeds 10 m/s or if the altitude change exceeds 500 m. Likewise, because it violates established aircraft performance profiles, a displacement of more than 100 kilometers in less than 300 seconds is considered spoofing [11].

Hardhat, a popular Ethereum development environment, is used for implementation in conjunction with a ReactJS frontend. For testing scenarios, the OpenSky Network provides altered and benign data samples. Real-time data validation and rejection logs are shown, revealing which anomalies were rejected and why [12]. By acting as a middleware for transaction approval, MetaMask ensures that all entries into the blockchain system are user-approved and verified. This configuration ensures that the data is traceable, unchangeable, and auditable for forensic investigation after an incident [13], [14].

Furthermore, a layered protection mechanism is made possible by MetaMask's human-in-the-loop capability, which combines automated anomaly detection with optional manual clearance. Dual-authorization policies can be used in safety-critical systems to prevent false positives and negatives, allowing for operational flexibility while preserving data integrity [15], [16]. The blockchain ledger, by design, provides a secure, timestamped record of all validated transactions, which can be utilized as legal or investigational evidence in case of system compromise [17][12]. The approach not only improves security but also turns flight surveillance into an open, decentralized, and accountable infrastructure. It is part of larger trends within aerospace systems that aim to include verifiable trust mechanisms within communication protocols. The research fills the disconnect between theoretical blockchain potential and the operational realities of flight monitoring, establishing a benchmark for subsequent smart aviation systems[18].

II. Related Works

There have been both cryptographic and non-cryptographic attempts to secure ADS-B transmissions. [19]. The Elliptic Curve Digital Signature Algorithm (ECDSA) and similar cryptographic schemes offer excellent data integrity and authenticity but are computationally intensive, introduce latency, and are not backward compatible with the current aviation infrastructure [20], [21], [22]. Physical-layer security techniques, including spectrum anomaly detection and radio fingerprinting, utilize features of radio emissions to differentiate genuine signals from counterfeit ones. Although they have potential, adaptive attackers capable of mimicking RF signatures or altering their behavior in real time often evade these strategies [23].

Furthermore, because of its inherent security features, blockchain has gained widespread acceptance as a remedy for these problems. Blockchain's unchangeable data structure and decentralized consensus methods allow for the creation of trustworthy data chains without the need for a central authority. For instance, research has shown that blockchain technology can check and record air traffic information efficiently [6], [7]. Additionally, smart contracts built on Ethereum can take independent action to enforce data validation safety standards. [8], [9].

Anomalies such as excessive climb rates or irrational geographic mobility can be identified with the integration of smart contract logic. Reliance on external systems for anomaly detection can be reduced by incorporating this reasoning into conditional conditions that are part of the contract itself [8], [9]. Secure, traceable, and decentralized flight data entry is made possible by these systems when used in conjunction with front-end interaction platforms like MetaMask.

In line with that, recent frameworks such as ADS-Bchain [14], ATMChain [13], and other Ethereum-based implementations [7], have showcased the feasibility of combining smart contracts and ADS-B telemetry to secure airspace communication against cyber-physical threats. These systems keep logs for future forensic or legal review in addition to detecting attacks. The research on MetaMask-driven user authentication systems and their integration with critical infrastructure has also highlighted the significance of transparency, traceability, and real-time prevention [16], [17].

III. Method

The Ethereum blockchain, smart contracts, and MetaMask integration are used in this study's experimental approach, which is based on a system implementation strategy, to design, build, and test a security model for ADS-B data transfer. Establishing a strong system that can verify incoming aircraft data, prevent spoofing, tampering, and replay attacks, and only upload legitimate data to the blockchain is the aim of this technology. The study is predicated on the idea that ADS-B data is intrinsically insecure since it is openly broadcast without encryption or authentication. Therefore, a visible and unchangeable ledger is used in conjunction with smart contract validation and user-facing digital signatures to combat these risks in real time. The method starts with aircraft data from the open-source ADS-B data provider OpenSky Network. A frontend web application linked to the Ethereum blockchain via MetaMask and Hardhat processes this data in real time. The entire process is meant to guarantee a smooth transition between gathering data, validating it, and permanently storing it on the blockchain.

A. Architecture System

Figure 1 shows the five interconnected parts that make up the architecture of the suggested blockchain-based ADS-B data validation system. The OpenSky Network serves as the main data source for the flow, supplying real-time ADS-B flight data in JSON format. Key aircraft details such the ICAO24 number, callsign, position (latitude and longitude), altitude, and timestamp are all included in these data packets.

The collected data is then processed by the Frontend User Interface, which was built using the ReactJS framework. This web-based tool provides real-time flight tracking and serves as the interaction medium for initiating blockchain transactions. When a user (like air traffic controllers) wishes to confirm a flight record, the interface initiates MetaMask, a digital wallet that runs in the browser.

Before sending data to the blockchain, MetaMask obtains the user's digital signature, thereby facilitating transaction authentication. By explicitly reviewing and approving each data submission by a trusted source, this step is essential for implementing human-in-the-loop validation. Following approval, the signed data is sent to the Smart Contract, which performs a number of security checks to look for indications of replay, tampering, or spoofing attacks.

The Blockchain Ethereum Node, which is launched in this implementation utilizing the Hardhat local development environment, records the flight data permanently if it passes validation. All recorded data is guaranteed to be immutable, verifiable, and traceable via the blockchain layer, facilitating auditability and post-event forensic investigation.

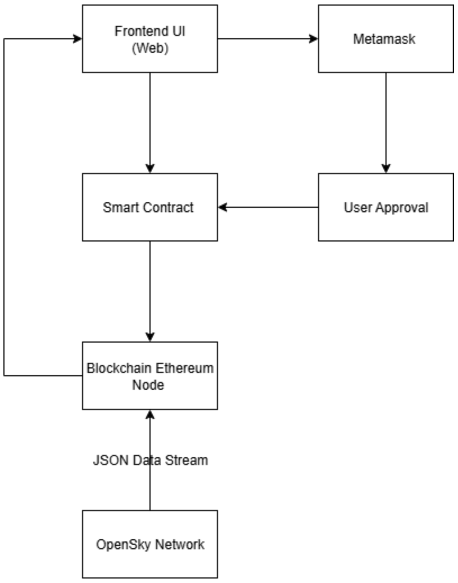


Figure 1 Architecture System

With the help of this modular architecture, each layer can function independently while maintaining safe contact with the others. It successfully combines blockchain storage, smart contract logic, frontend validation workflows, and external ADS-B telemetry to create a decentralized, impenetrable flight data verification system.

A. Parameter and Data Structure

To guarantee compatibility with Ethereum's processing limitations, the transmitted flight data is structured into a certain structure. Multiplication factors are used to convert floating-point numbers—which Solidity does not natively support—into integers. For instance, altitude is scaled by 10^2, and latitude and longitude measurements are multiplied by 10^6. The accuracy of this conversion is guaranteed without sacrificing blockchain performance. The following parameters are part of the data structure used in this study:

Table 1. Parameter and Data Structure

No	Parameter	Description	Format
1	icao24	Unique aircraft identifier (24-bit)	String
2	callsign	Aircraft flight callsign	String
3	altitude	Altitude in meters × 10^2	Integer
4	latitude	Latitude position × 10^6	Integer
5	longitude	Longitude position × 10^6	Integer
6	onGround	Boolean status: true if the aircraft is on the ground	Boolean
7	timestamp	Unix time at which the data was received	Integer
8	isSpoofed	Spoofing indicator flag based on AI or manual input	Boolean

These fields are encapsulated into a Solidity struct named Flight. Every new incoming data packet is passed through either updateFlight(...) or updateFlightBatch(...) functions for single or batch updates respectively. These functions not only handle the transaction but also validate the logical consistency of the data.

B. Smart Contract Implementation and Validation Mechanism

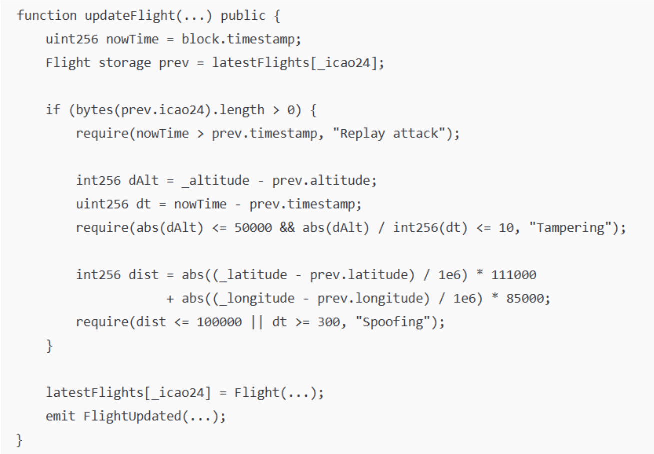
The smart contract method updateFlight(...) is the core component of the validation mechanism. In order to stop the inclusion of modified or dubious flight data, this function carries out a number of crucial checks. To stop replay attacks, the smart contract first makes sure the incoming timestamp is more recent than the last one that was recorded. In order to detect tampering, it then uses the following mathematical condition to calculate the altitude difference (Δalt) and divide it by the time difference (Δt):

$$|\Delta alt| \leq 50000 \text{ and } \left|\frac{\Delta alt}{\Delta t}\right| \leq 10$$

Failure to validate the flight data results in its rejection as physically impossible. The Euclidean-like distance between the old and new positions is then computed using degree-to-meter approximations. According to the streamlined distance check methodology, data is marked for faking if the distance surpasses 100 kilometers in 300 seconds:

$$Distance = |\Delta lat| \times 111000 + |\Delta lon| \times 85000$$
$$Condition : Distance \leq 100000 \; or \; \Delta t \geq 300$$

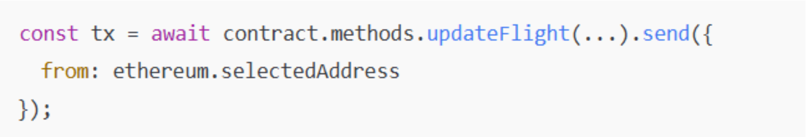
Below is the Solidity code snippet used in the smart contract:



```
function updateFlight(...) public {
    uint256 nowTime = block.timestamp;
    Flight storage prev = latestFlights[_icao24];

    if (bytes(prev.icao24).length > 0) {
        require(nowTime > prev.timestamp, "Replay attack");

        int256 dAlt = _altitude - prev.altitude;
        uint256 dt = nowTime - prev.timestamp;
        require(abs(dAlt) <= 50000 && abs(dAlt) / int256(dt) <= 10, "Tampering");

        int256 dist = abs((_latitude - prev.latitude) / 1e6) * 111000
                    + abs((_longitude - prev.longitude) / 1e6) * 85000;
        require(dist <= 100000 || dt >= 300, "Spoofing");
    }

    latestFlights[_icao24] = Flight(...);
    emit FlightUpdated(...);
}
```

Figure 2. Solidity Snippet validation mechanism

The MetaMask wallet, which serves as both the signing authority and the user approval interface, is connected to the frontend interface as an extra layer of validation. MetaMask will provide a popup with all pertinent flight characteristics once a user enters data or initiates a flight data transaction, allowing the user to accept or reject the transaction. This JavaScript sample shows how information is delivered to MetaMask for broadcasting and signing from the frontend:



```
const tx = await contract.methods.updateFlight(...).send({
  from: ethereum.selectedAddress
});
```

Figure 3 JavaScript snippet MetaMask Validation

The smart contract's logical and physical verification of the flight data, together with the authorized user's consent—who must authorize the transaction using MetaMask—are both guaranteed by this dual-validation technique. The end result is a reliable, traceable, and secure record of aircraft movement that can be accessed for post-event inquiry and real-time monitoring and is impervious to manipulation.

IV. Result and Discussion

In this part of the study, we evaluated how well the blockchain-based ADS-B security system performs during normal use and under simulated cyberattacks. The test was conducted on the evening of July 11, 2025, between 8:40 and 8:58 p.m. We used a local Ethereum setup through Hardhat, with MetaMask to approve each transaction. During the test, real-time flight data from the OpenSky Network was sent to the blockchain in batches. Once the system was running normally, we simulated several types of attacks—including spoofing, replay, and data tampering—to see how effectively the smart contract could detect and prevent them.

A. Blockchain Behavior and Data Logging

Between 8:40:44 PM and 8:56:08 PM, a total of 9 flight batch updates were submitted, resulting in the creation of 9 new blocks (Block #24 to Block #32). These batch transactions reflected normal system operations and served as baselines before attack windows.

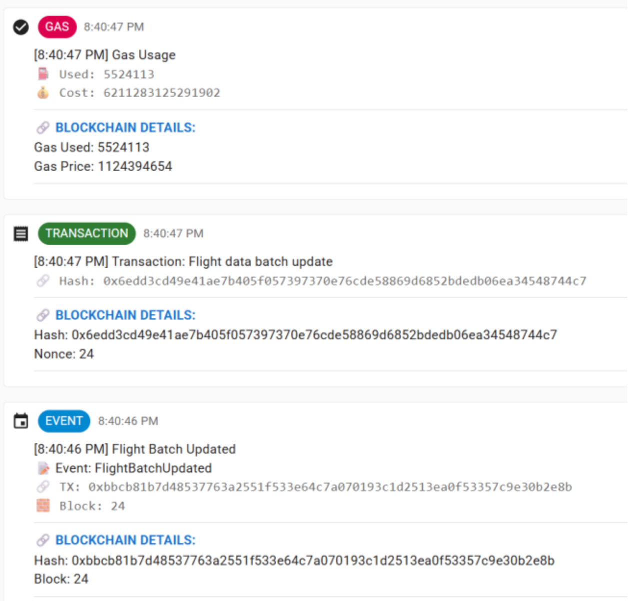
Table 2 Summary of Flight Data Batch Updates and Block Creation

Timestamp	Event Type	Block#	Transaction Hash	Gas Used	Status
8:40:44	Batch Update	24	0xbbcb...2e8b	2,343,216	Success
8:40:47	Batch Update	25	0x6edd...44c7	5,524,113	Success
8:54:56	Batch Update	26	0x1607...17d0	5,713,640	Success
8:54:58	Batch Update	27	0x2b61...852e	9,504,190	Success
8:55:13	Batch Update	28	0x93d4...7059	1,561,918	Success
8:55:15	Batch Update	29	0x7740...5ed5	1,559,590	Success
8:55:27	Batch Update	30	0x29be...b49c	1,561,918	Success
8:55:28	Batch Update	31	0x94ae...cc27	1,559,590	Success
8:56:08	Batch Update	32	0xc4c6...fd53	5,713,640	Success

During this time, the system preserved transactional integrity. Even after interface refreshes or logs being reset, the block numbers incremented consistently, reaffirming Ethereum's core attribute of ledger persistence. This implies that block history is preserved unless the node's state is actively reset, offering auditability, traceability, and non-repudiation—qualities crucial for forensic investigation in aviation cybersecurity.

Figure 4 illustrates this persistence by comparing the recordings of Blocks #24 and #25 taken before and after the frontend refresh. Despite resetting the UI, the blockchain state continued incrementally, preserving the immutable ledger entries.

Additionally, system administrators and auditors can keep an eye out for irregularities, optimise expenses, and identify submissions that seem suspiciously frequent by examining transaction hashes and gas usage. With Ethereum’s transparent logging system, forensic investigators can trace the source, timing, and payload of each update—something not possible in traditional ADS-B systems.



```
GAS 8:40:47 PM
[8:40:47 PM] Gas Usage
Used: 5524113
Cost: 6211283125291902

BLOCKCHAIN DETAILS:
Gas Used: 5524113
Gas Price: 1124394654

TRANSACTION 8:40:47 PM
[8:40:47 PM] Transaction: Flight data batch update
Hash: 0x6edd3cd49e41ae7b405f057397370e76cde58869d6852bdedb06ea34548744c7

BLOCKCHAIN DETAILS:
Hash: 0x6edd3cd49e41ae7b405f057397370e76cde58869d6852bdedb06ea34548744c7
Nonce: 24

EVENT 8:40:46 PM
[8:40:46 PM] Flight Batch Updated
Event: FlightBatchUpdated
TX: 0xbbcb81b7d48537763a2551f533e64c7a070193c1d2513ea0f53357c9e30b2e8b
Block: 24

BLOCKCHAIN DETAILS:
Hash: 0xbbcb81b7d48537763a2551f533e64c7a070193c1d2513ea0f53357c9e30b2e8b
Block: 24
```

Figure 4. Blockchain log between block #24 and #25

To further demonstrate how the blockchain can support forensic analysis, I examined the transaction logs from Blocks #24 and #25, which were visible in the frontend log at 8:40:44 PM and 8:40:47 PM, respectively. Even though the frontend interface experienced a reset, these two blocks help establish a reference point for understanding how Ethereum maintains a continuous and orderly record of data.

In Block #24, there was a batch update transaction that consumed 2,343,216 gas units and was recorded with the hash 0xbbcb...2e8b. Just a few seconds later, Block #25 followed with a larger transaction using 5,524,113 gas units (hash 0x6edd...44c7). Despite the visual reset on the frontend, the blockchain advanced from Block #24 to Block #25 smoothly—there were no duplicated entries or rollbacks.

This confirms that each batch of data is stored on-chain in the exact order it was processed, highlighting the reliability of Ethereum's state persistence and immutability. Even when the frontend doesn't display certain logs, all activities—including normal operations, errors, or even attacks—can still be traced directly on-chain. Ethereum provides tools like block explorers, transaction receipts, and event logs, which are extremely useful for forensic work. This is especially important in aviation contexts, where having a permanent and tamper-proof record is crucial for post-incident investigations, safety reviews, and compliance.

B. Attack Summary and Analysis

Three attack types—spoofing, replay, and tampering—were applied to the system in a series of adversarial simulations. These attacks were started one after the other and were fully documented, including block assignments and rejection reasons. The robustness of the smart contract implementation was assessed using the attack logs and the results of the responses.

It's crucial to remember that the user purposefully permitted the few successful attack attempts that were seen during testing in order to evaluate how the blockchain behaves when harmful data is received. These incidents show that the blockchain system automatically blocks all erroneous or questionable data without user approval, not that the system has failed. This aligns with the design principle of blockchain immutability and deterministic validation logic, ensuring no unauthorized or manipulated data bypasses the consensus mechanism.

In order to mimic fabricated aircraft identities, spoofing techniques were created to introduce erroneous callsigns into the flight data stream. During testing, a total of 17 spoofing attacks were conducted. When the smart contract accurately recognized the inserted callsigns as invalid, it automatically rejected 16 of them and sent the rejection message "invalid callsign." Because each callsign was verified against a known set of acceptable aircraft identities, these rejections show that the contract successfully integrated with the OpenSky Network database. The user manually approved the one successful spoofing attempt (TX: 0xa374...b0b7, Block #33, timestamp 8:56:56 PM). This purposeful addition was meant to mimic the scenario that would arise if a detected spoof message were accepted as a result of malicious activity or operational error. This test confirmed that even when flagged by the smart contract, ultimate execution depends on user approval—underscoring the importance of human-in-the-loop authorization. Moreover, the transaction remains permanently recorded on-chain with complete metadata, including hash, origin, block, and timestamp, enabling precise forensic backtracking[12], [24].

To mimic replay attacks, previously approved flight messages were sent again in an effort to fool the system into reproducing outdated information. Thirteen replay attempts were made in all. 11 of these attempts were successfully rejected by the smart contract's timestamp validation algorithm with the message "timestamp not newer," preserving chronological consistency and eliminating duplicate entries. TX: 0x8e16...c726 at 8:57:25 PM, Block #34 and 0x8c8e...b76e at 8:57:29 PM, Block #35) were two successful replay attacks that happened in very close succession to their original data. Only because the user specifically approved the transaction were these assaults performed, even though they passed the timestamp validation algorithm. This draws attention to a specific situation that may call for stricter timestamp thresholds, such mandated submission intervals or blockchain-native block timestamps. Once more, the blockchain's

Tampering simulations were performed by altering numerical flight data, specifically the altitude field, to simulate impossible or physically implausible aircraft maneuvers. A high volume of 26 tampering attacks was conducted, all aimed at testing the robustness of the smart contract's physical validation constraints. The system successfully blocked 25 of these attacks using its hardcoded threshold for vertical movement rates, issuing a rejection message: "impossible altitude rate." The one successful incident at 8:56:07 PM, TX: 0xc4c6...fd53, in Block #32, was manually authorized by the user, again to test the behavior of the system under accepted malicious conditions. Despite the anomaly, the blockchain successfully recorded all details, allowing retrospective validation and analysis. This reflects the strength of the ledger as a forensic tool while also identifying potential areas for tightening smart contract thresholds[7], [13], [14].

The following table summarizes all attacks launched during the evaluation, along with their outcome metrics

Table 3. Attack Summary and Defense Performance

Attack Type	Total Attempts	Successes	Rejections	Detection Rate
Spoofing	17	1	16	94.1%
Replay	13	2	11	84.6%
Tampering	26	1	25	96.1%
Total	56	4	52	92.9%

Strong resistance to a variety of hostile inputs was shown by the blockchain-based ADS-B data validation system. The smart contract's dependability in rejecting unauthorised or incorrect flight data under typical operational settings is demonstrated by its final detection rate of 92.9%. Attacks including spoofing and manipulation were very well-contained, with rejection rates exceeding 94%. Due to timing edge circumstances that allowed two fraudulent entries to pass through when their timestamps nearly matched those of authentic updates, replay attacks demonstrated somewhat reduced accuracy. However, as part of the experimental methodology to investigate attack propagation, the user knowingly accepted each of these scenarios.

This highlights one of the most significant aspects of blockchain-integrated ADS-B systems: final transaction inclusion necessitates user confirmation. Deterministically, the system flags suspicious items using programmed logic, but it doesn't really perform the write operation until the transaction is expressly signed. When combined with smart contract automation, this approach adds a final level of human control, resulting in a potent defense-in-depth architecture. The blockchain's value in forensic aviation investigations is further illustrated by the immutable logs produced in each scenario, which enable future auditors or regulatory agencies to transparently confirm event data.

According to the testbed's performance, blockchain technology, which provides cryptographic assurances for message authenticity, traceability, and tamper resistance, can be a useful part of modernising aviation surveillance systems. More interaction with aircraft registry databases for improved callsign verification, machine-learning-based anomaly detection, and stricter timestamp validations are possible future improvements.

V. CONCLUSION

This study demonstrated how blockchain technology, namely the Ethereum platform and MetaMask wallet integration, may significantly enhance the security and integrity of ADS-B (Automatic Dependent Surveillance–Broadcast) aircraft data. By utilising smart contracts to guarantee message authenticity, temporal accuracy, and physical plausibility, the system successfully detected and rejected the vast majority of spoofing, replay, and tampering threats, achieving a 92.9% detection rate. These findings are in line with prior research that suggested blockchain technology might be used for irreversible, verifiable aircraft communication logging and integrity preservation. All accepted attack data was explicitly authorised by the user to reproduce the worst-case situations, and the system was tested against genuine inputs using real-time data from the OpenSky Network. This highlights the importance of human-in-the-loop security mechanisms, where blockchain logic ensures detection and user permission is necessary for the execution of the final transaction. This dual-layer security technique improves operational safety by hindering the automated commitment of erroneous or anomalous inputs to the ledger. Forensic investigation and oversight by regulators are supported by the usage of Ethereum's open, decentralised architecture. In accordance with the ICAO's and other comparable aviation cybersecurity guidelines, all transactions—including those that are denied or flagged—remain permanently documented with complete information. Even in the face of post-event enquiries or insider threats, this ensures auditability and traceability. The blockchain provides a transparent and cryptographically secure foundation for ADS-B systems of the future. Future developments may include dynamic contract modifications to handle changing threat models and AI-based anomaly detection, highlighting blockchain's potential to modernise aviation defence and surveillance systems.

REFERENCES

[1] J. Sun, X. Olive, E. Roosenbrand, C. Parzani, and M. Strohmeier, "OpenSky Report 2024: Analysis of Global Flight Contrail Formation and Mitigation Potential," *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, 2024, doi: 10.1109/DASC62030.2024.10748832.

[2] J. Sun, X. Olive, M. Strohmeier, M. Schafer, I. Martinovic, and V. Lenders, "OpenSky Report 2021: Insights on ADS-B Mandate and Fleet Deployment in Times of Crisis," *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, vol. 2021-Octob, 2021, doi: 10.1109/DASC52595.2021.9594361.

[3] N. Pearce, K. J. Duncan, and B. Jonas, "Signal discrimination and exploitation of ADS-B transmission," *Conf. Proc. - IEEE SOUTHEASTCON*, vol. 2021-March, 2021, doi: 10.1109/SoutheastCon45413.2021.9401909.

[4] S. Rudys, J. Aleksandravicius, R. Aleksiejunas, A. Konovaltsev, C. Zhu, and L. Greda, "Physical layer protection for ADS-B against spoofing and jamming," *Int. J. Crit. Infrastruct. Prot.*, vol. 38, no. March, p. 100555, 2022, doi: 10.1016/j.ijcip.2022.100555.

[5] A. Carlo and K. Obergfaell, "Cyber attacks on critical infrastructures and satellite communications," *Int. J. Crit. Infrastruct. Prot.*, vol. 46, no. May, p. 100701, 2024, doi: 10.1016/j.ijcip.2024.100701.

[6] I. Al-Barazanchi *et al.*, "Blockchain-Technology-Based Solutions for IOT Security," *Iraqi J. Comput. Sci. Math.*, vol. 3, no. 1, pp. 53–63, 2022, doi: 10.52866/ijcsm.2022.01.01.006.

[7] F. Hasin, T. H. Munia, N. N. Zumu, and K. A. Taher, "ADS-B Based Air Traffic Management System Using Ethereum Blockchain Technology," *2021 Int. Conf. Inf. Commun. Technol. Sustain. Dev. ICICT4SD 2021 - Proc.*, pp. 346–350, 2021, doi: 10.1109/ICICT4SD50815.2021.9396828.

[8] D. Çulha and A. Yazici, "Smart Contract Upgradability: A Structured and Natural Approach," *2024 6th Int. Conf. Blockchain Comput. Appl. BCCA 2024*, pp. 198–203, 2024, doi: 10.1109/BCCA62388.2024.10844407.

[9] A. Abuhashim and C. C. Tan, "Smart Contract Designs on Blockchain Applications," *Proc. - IEEE Symp. Comput. Commun.*, vol. 2020-July, pp. 9–12, 2020, doi: 10.1109/ISCC50000.2020.9219622.

[10] L.-N. Degambur, "Replay Attack Prevention in Decentralised Contact Tracing: A Blockchain-Based Approach," *OALib*, vol. 11, no. 02, pp. 1–17, 2024, doi: 10.4236/oalib.1111179.

[11] A. Xiong *et al.*, "Block-chain abnormal transaction detection method based on generative adversarial network and autoencoder," *High-Confidence Comput.*, p. 100313, 2025, doi: 10.1016/j.hcc.2025.100313.

[12] Vinoth Kumar R, Haritha S, Sooritha M, and Shanmugavaruni S, “Preserving Integrity of Forensic Evidence Using Blockchain,” *Int. J. Eng. Technol. Manag. Sci.*, vol. 9, no. 2, pp. 901–908, 2025, doi: 10.46647/ijetms.2025.v09i02.114.

[13] X. Lu, Z. Wu, and J. Cao, “ATMChain: Blockchain-Based Security Architecture for Air Traffic Management in Future,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 4, pp. 3872–3896, 2024, doi: 10.1109/TAES.2024.3371396.

[14] Z. Wu, T. Shang, M. Yue, and L. Liu, “ADS-Bchain: A Blockchain-Based Trusted Service Scheme for Automatic Dependent Surveillance Broadcast,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 6, pp. 8535–8549, 2023, doi: 10.1109/TAES.2023.3306336.

[15] A. A. Alzubaidi, “Systematic Literature Review for Detecting Intrusions in Unmanned Aerial Vehicles Using Machine and Deep Learning,” *IEEE Access*, vol. 13, no. April, pp. 58576–58599, 2025, doi: 10.1109/ACCESS.2025.3552329.

[16] K. L. Mounika, D. S. S , and D. T. Sethukarasi , “Real Time Flight Tracking Using Deep Learning and Blockchain Technology,” *Interantional J. Sci. Res. Eng. Manag.*, vol. 08, no. 12, pp. 1–9, 2024, doi: 10.55041/ijsrem40315.

[17] S. Kumar Jagatheesaperumal, M. Rahouti, A. Chehri, K. Xiong, and J. Bieniek, “Blockchain-Based Security Architecture for Uncrewed Aerial Systems in B5G/6G Services and Beyond: A Comprehensive Approach,” *IEEE Open J. Commun. Soc.*, vol. 6, no. February, pp. 1042–1069, 2025, doi: 10.1109/OJCOMS.2025.3528220.

[18] P. Praitheeshan, L. Pan, J. Yu, J. Liu, and R. Doss, “Security Analysis Methods on Ethereum Smart Contract Vulnerabilities: A Survey,” pp. 1–21, 2019, [Online]. Available: http://arxiv.org/abs/1908.08605

[19] A. Mahroof, I. Nabi, S. Z. Farooq, and N. A. Naqvi, “Machine Learning-Based Detection of Spoofing Attacks in GNSS: A Study Using TEXBAT Dataset,” *ICEENG 2024 - 14th IEEE Int. Conf. Electr. Eng.*, pp. 90–95, 2024, doi: 10.1109/ICEENG58856.2024.10566287.

[20] M. Ngamboé *et al.*, “CABBA: Compatible Authenticated Bandwidth-efficient Broadcast protocol for ADS-B,” *Int. J. Crit. Infrastruct. Prot.*, vol. 48, no. November 2024, p. 100728, 2025, doi: 10.1016/j.ijcip.2024.100728.

[21] J. Habibi Markani, A. Amrhar, J. M. Gagné, and R. J. Landry, “Security Establishment in ADS-B by Format-Preserving Encryption and Blockchain Schemes,” *Appl. Sci.*, vol. 13, no. 5, 2023, doi: 10.3390/app13053105.

[22] S. Latif, Z. Idrees, J. Ahmad, L. Zheng, and Z. Zou, “A blockchain-based architecture for secure and trustworthy operations in the industrial Internet of Things,” *J. Ind. Inf. Integr.*, vol. 21, no. September 2020, 2021, doi: 10.1016/j.jii.2020.100190.

[23] H. Zha, Q. Tian, and Y. Lin, “Real-World ADS-B signal recognition based on Radio Frequency Fingerprinting,” *Proc. - Int. Conf. Netw. Protoc. ICNP*, vol. 2020-Octob, 2020, doi: 10.1109/ICNP49622.2020.9259404.

[24] G. Estevam, L. M. Palma, L. R. Silva, J. E. Martina, and M. Vigil, “Accurate and decentralized timestamping using smart contracts on the Ethereum blockchain,” *Inf. Process. Manag.*, vol. 58, no. 3, 2021, doi: 10.1016/j.ipm.2020.102471.