Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

A Secure Boot and Tamper-Resistant Firmware Framework for UAV Systems using Cryptographic Authentication and Runtime Access Control

Archana Kokatkar
Department of Computer Engineering
JSPM's JSCOE, Pune

Vishal Gajanan Zalake
Department of Computer Engineering
JSPM's JSCOE, Pune

Vaishnavi Vijay Saraf
Department of Computer Engineering
JSPM's JSCOE, Pune

Radhey Parikshit Bildikar
Department of Computer Engineering
JSPM's JSCOE, Pune

Smitesh Vivek Shinde
Department of Computer Engineering
JSPM's JSCOE, Pune

Abstract - Unmanned Aerial Vehicles (UAVs) are increasingly deployed in safety-critical domains such as defense, surveillance, and disaster management, making them attractive targets for cyberattacks. One of the most critical vulnerabilities lies in UAV firmware, which can be tampered with to gain unauthorized control or disrupt operations. This paper presents a secure UAV framework that ensures firmware integrity and prevents unauthorized parameter modification using a combination of secure boot and cryptographic authentication mechanisms. The proposed system implements a secure boot process using SHA-256 hashing and asymmetric cryptography to verify firmware authenticity before execution. Additionally, a novel runtime authentication mechanism based on a challenge-response protocol using nonce and digital signatures is introduced to restrict unauthorized Ground Control Station (GCS) access. The system is implemented on the ArduPilot platform using Pixhawk hardware and MAVLink communication. The framework is lightweight, scalable, and suitable for embedded UAV systems. While experimental evaluation is ongoing, the system successfully demonstrates prevention of unauthorized firmware execution and secure parameter access control.

Keywords — cyberattacks, defense, UAV firmware, SHA-256, asymmetric cyptography, GCS, Pixhawk, ArduPilot, MAVLink.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs), commonly known as drones, have revolutionized the fields of defense, disaster management, agriculture, logistics, and surveillance. These systems are increasingly being integrated into mission-critical applications where accuracy, reliability, and safety are of utmost importance. UAVs operate autonomously or semi-autonomously, controlled by onboard microcontrollers and specialized firmware that governs flight behavior, communication, navigation, and payload control. However, with the rapid growth of UAV technology comes an equally rising concern for security threats. UAV firmware — the core software that runs the flight controller — is vulnerable to tampering, unauthorized modification, or malicious code injection. Such compromises can lead to severe consequences such as unauthorized access, control loss, privacy breaches, or even physical damage in safety-critical environments. Ensuring firmware authenticity and integrity has therefore become a key requirement in modern UAV systems. Tampering with UAV firmware could allow attackers to alter flight parameters, disable safety protocols, or introduce backdoors for remote exploitation. These risks emphasize the need for secure firmware design and validation mechanisms capable of ensuring that only trusted, verified code executes on UAV hardware. The proposed project addresses these challenges by designing and implementing a secure UAV system with tamper-resistant firmware using asymmetric cryptographic techniques. The system utilizes a secure boot mechanism wherein the firmware is digitally signed with a private key during compilation. Upon startup, the UAV verifies the firmware signature using the corresponding public key embedded within the controller's memory. This ensures that only authenticated and untampered firmware can be executed. Furthermore, a key-based access control mechanism is integrated into the ArduPilot platform to prevent unauthorized configuration changes during runtime. Ground Control Stations (GCS) like Mission Planner often allow parameter modifications for flight tuning. If communication is compromised, attackers could misuse this feature. Hence, the project introduces an authentication key layer, allowing only verified users to modify UAV

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

parameters. By combining secure boot validation and parameter access control, the project strengthens the UAV's resilience against cyberattacks and unauthorized operations, ensuring the trustworthiness and safety of drone deployments in critical environments.

Software Requirement and Specification (SRS) defines the functionalities, features, and constraints of the proposed system. It serves as the foundation for development, testing, and deployment. The primary goal of this project is to design and implement a secure UAV firmware that can resist tampering and unauthorized access using cryptographic validation. This chapter specifies the algorithms, hardware-software requirements, and quality parameters that define the behavior of the system. The system is developed using ArduPilot firmware integrated with cryptographic modules written in C++ and Python, running on a Pixhawk flight controller with a Mission Planner Ground Control Station for operation and testing.

A. DEMAND FORECASTING

With the expanding adoption of UAVs across various sectors, the demand for secure and intelligent drone systems has surged. Governments, research institutions, and industries are actively seeking UAVs that not only perform efficiently but also maintain robust protection against firmware manipulation, data leakage, and unauthorized access. In particular, defense, surveillance, and critical infrastructure operations demand high-assurance drones that cannot be easily compromised or hijacked. The forecasted demand for secure UAV platforms indicates a significant market shift towards cyber-resilient systems. Incorporating firmware security mechanisms will be a key differentiator in next-generation UAVs. Thus, the integration of cryptographic firmware validation, as proposed in this project, aligns with future trends in autonomous aerial technology and national security requirements.

II. RELATED WORK

The paper "Electronic Warfare Cyberattacks, Countermeasures, and Modern Defensive Strategies of UAV Avionics" by Aaron Yu, Iuliia Kolotylo, Hashim A. Hashim, and Abdelrahman E. E. Eltoukhy, published in IEEE Access (2025) with the DOI 10.1109/ACCESS.2025.3561068, provides a comprehensive overview of modern cyberattacks targeting UAV avionics systems. The authors emphasize that firmware-level vulnerabilities significantly expose UAVs to exploitation, making them susceptible to a range of advanced electronic warfare tactics. [1] The study categorizes cyberattacks into three main types: communication interception, firmware injection, and hardware manipulation. Each category represents a different layer of vulnerability within UAV systems, illustrating how attackers can compromise data transmission, alter embedded software, or directly manipulate physical components. [1] To address these vulnerabilities, the authors propose several defensive strategies, including the integration of cryptographic validation mechanisms, secure boot sequences, and intrusion detection systems. These countermeasures aim to enhance UAV resilience by ensuring that only authenticated and verified code can be executed, while also providing real-time monitoring against unauthorized access or tampering attempts. [1]

The paper "Understanding and Securing the Risks of Uncrewed Aerial Vehicle Services" by Ioannis Anagnostis, Panayiotis Kotzanikolaou, and Christos Douligeris, published in IEEE Access (2025) with the DOI 10.1109/ACCESS.2025.3549861, explores the risk factors associated with UAV systems operating in networked environments. The authors analyse how wireless communication links and cloud-based control systems increase the exposure of UAVs to various cyber vulnerabilities, making them potential targets for exploitation. [2] The study presents a risk quantification framework for UAV services, which classifies threats according to the confidentiality, integrity, and availability (CIA) triad. This framework enables systematic evaluation of potential security risks, helping identify areas where UAV systems are most susceptible to compromise. [2] Furthermore, the authors emphasize the importance of implementing both policy-based and technical safeguards to mitigate these risks. Key measures discussed include firmware verification and the use of encrypted communication channels to protect against unauthorized access and data manipulation. [2]

The paper "Sensor Deprivation Attacks for Stealthy UAV Manipulation" by Alessandro Erba, John H. Castellanos, Sahil Sihag, Saman Zonouz, and Nils Ole Tippenhauer, published on arXiv (2024) with the identifier 2410.11131v1, investigates stealthy cyberattacks that target UAV sensor systems without directly modifying the firmware. The authors demonstrate how manipulation of sensor input data can mislead UAV control logic, causing the system to perform unsafe or unintended behaviors while avoiding conventional detection mechanisms. [3] Although the study primarily examines sensor-level attack vectors, it highlights the crucial role of firmware-level integrity as a defense mechanism. The authors emphasize that if UAV firmware incorporates the ability to detect anomalies or verify the authenticity of input signals, it can effectively mitigate such stealthy manipulations before they result in operational compromise. [3] The insights from this research directly influenced our project's approach by motivating the extension of firmware integrity checks beyond boot-time validation. Building on these findings, our project proposes the future integration of real-time firmware-based anomaly detection modules to strengthen UAV resilience against sensor-level stealth attacks. [3]

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

The paper "A Security Review in the UAVNet Era: Threats, Countermeasures, and Gap Analysis" by A. Rugo, C. A. Ardagna, and N. E. Ioini, published in ACM Computing Surveys, Vol. 55, No. 1, January 2023, presents a comprehensive review of modern UAV networking (UAVNet) security challenges. The authors examine a wide range of vulnerabilities, including communication protocol weaknesses, firmware vulnerabilities, and network intrusion risks, outlining how these issues collectively undermine UAV system resilience. [4] The paper categorizes UAV security threats into three hierarchical levels — firmware-level, communication-level, and network-level — emphasizing that firmware integrity serves as the first and most critical layer of defense against cyberattacks. The authors discuss emerging trends such as the integration of cryptographic bootloaders, secure firmware update mechanisms, and remote attestation techniques designed to detect and prevent firmware tampering in UAV systems. [4] Additionally, the survey identifies a significant research gap in the development of lightweight firmware authentication methods that can be effectively implemented on resource-constrained flight controllers. Addressing this limitation is essential for achieving practical and scalable UAV security solutions. [4] This research is highly relevant to our project, as it provides a theoretical foundation for implementing asymmetric cryptography and key-based control mechanisms to ensure firmware authenticity and maintain operational trustworthiness in UAV systems. [4]

III. METHODOLOGY

The proposed secure UAV system is developed using a multi-stage methodology that ensures firmware integrity, prevents unauthorized access, and enhances overall system security. The methodology is divided into three major phases: **Firmware Signing (Offline Stage), Secure Boot Verification (Boot Stage), and Runtime Authentication (Operational Stage)**. These phases work together to provide a comprehensive security framework for UAV firmware and communication.

To achieve firmware integrity and authentication, the system uses asymmetric cryptography combined with hashing and key verification mechanisms.

SHA-256 is used in the proposed system to generate a fixed-length cryptographic digest of the firmware binary, ensuring data integrity. It processes the firmware file by dividing it into data blocks and applying multiple hashing rounds to produce a 256-bit (64-character hexadecimal) hash value. Even a single-bit modification in the firmware results in a completely different hash, making any tampering easily detectable. For example, an input firmware file (e.g., *Firmware.bin*) produces a unique hash such as *6A2D7EAB5F....* SHA-256 is highly collision-resistant, meaning no two different inputs generate the same hash, and it is computationally efficient, making it well-suited for resource-constrained embedded systems like UAV controllers.

A. FIRMWARE SIGNING PHASE(OFFLINE PHASE)

The firmware signing phase is performed during the development stage prior to deployment onto the UAV. In this phase, the firmware is first compiled using the ArduPilot framework to generate a binary file. To ensure data integrity, a cryptographic hash of the firmware is computed using the SHA-256 hashing algorithm, represented as:

$$H=SHA256(Firmware)$$

This hash uniquely represents the firmware content, ensuring that any modification in the firmware results in a completely different hash value. The generated hash is then encrypted using the developer's private key to produce a digital signature:

$$S=Encrypt(H,Kpriv)$$

This digital signature guarantees both authenticity and non-repudiation of the firmware. The final deployment package consists of the firmware binary, the generated signature, and the corresponding public key. The public key is securely embedded into the UAV's memory and is later used for verification during the boot process.

B. SECURE BOOT VERIFICATION PHASE

The secure boot phase is executed every time the UAV is powered on. Its primary objective is to verify the authenticity and integrity of the firmware before execution. During this stage, the bootloader initializes and loads the firmware, its associated digital signature, and the stored public key. The system recomputes the firmware hash using the same SHA-256 algorithm:

$$H'=SHA256(Firmware)$$

Simultaneously, the stored digital signature is decrypted using the public key to retrieve the original hash:

$$H=Decrypt(S,Kpub)$$

A comparison is then performed between the computed hash H′H'H′ and the decrypted hash HHH. If both values match, the firmware is considered authentic and execution proceeds normally. However, if a mismatch is detected, it indicates possible tampering, and the system immediately halts the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

boot process. This prevents execution of unauthorized or malicious firmware and ensures that only trusted code runs on the UAV system. Additionally, any verification failure is logged for further analysis.

C. RUNTIME AUTHENTICATION PHASE

To enhance security during UAV operation, a runtime authentication mechanism is implemented to restrict unauthorized parameter modifications. This phase introduces a **challenge-response authentication protocol** based on nonce generation and digital signatures.

When a Ground Control Station (GCS) attempts to modify UAV parameters, an authentication request is initiated. In response, the UAV generates a random 32-byte nonce, which acts as a unique challenge value. This nonce is transmitted to the GCS using MAVLink communication. Upon receiving the nonce, the GCS signs it using its private key to generate a digital signature:

$$Sig=Sign(Nonce,Kpriv)$$

The signed nonce is then transmitted back to the UAV. The UAV verifies this signature using the stored public key:

$$Verify(Sig,Nonce,Kpub)$$

If the verification is successful, the user is authenticated and granted permission to modify UAV parameters. Otherwise, access is denied, and the attempt is logged as an unauthorized access attempt. This mechanism ensures that only legitimate users possessing the correct private key can interact with critical system parameters.

D. MAVLINK-BASED COMMUNICATION INTEGRATION

The authentication protocol is seamlessly integrated into the UAV system using MAVLink communication commands. Custom commands are used to facilitate secure interaction, where one command is used to request the nonce and another to confirm authentication. This approach ensures compatibility with existing Ground Control Stations without requiring significant modifications to the communication protocol.

E. LOGGING AND SECURITY ENFORCEMENT

To maintain traceability and support forensic analysis, the system incorporates a logging mechanism that records all critical events. These include firmware verification results, authentication attempts, and unauthorized access incidents. Logs are stored in onboard memory and can be retrieved via the Ground Control Station. This feature enhances system transparency and aids in identifying potential security threats.

IV. SYSTEM DESIGN

System design serves as the blueprint of the entire project, translating theoretical concepts into a practical and implementable model. The proposed UAV system integrates firmware integrity verification and key-based access control to ensure that only authenticated software runs on the UAV flight controller and that only authorized personnel can modify its configuration. The implementation is carried out on the ArduPilot open-source firmware platform using the Pixhawk flight controller and Mission Planner ground control software. The security framework operates in two main stages:

1. Secure Boot Stage – Verifies the firmware integrity before allowing the UAV to boot.

2. Secure Runtime Stage – Controls access to UAV parameter modification using authentication keys.

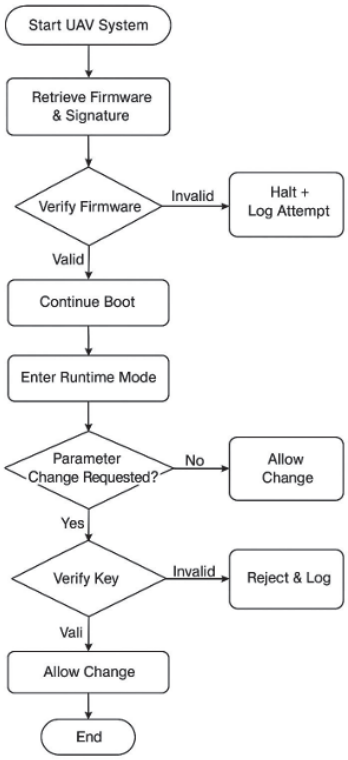

Fig. 1. Flowchart of Secure UAV

Both stages together form a tamper-resistant architecture that ensures the UAV operates only under legitimate firmware and verified control commands.

The proposed system design follows a modular and layered architecture to ensure robust security and efficient operation of the UAV system.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

The first layer, known as the firmware signing layer, is executed during the firmware compilation stage by the developer. In this layer, the firmware binary is processed using a cryptographic hashing algorithm such as SHA-256 to generate a unique hash value, which is then encrypted using a private key to produce a digital signature. The signed firmware, along with the corresponding public key, is securely stored and prepared for deployment onto the UAV.

The second layer, referred to as the secure boot layer, operates during system startup. In this stage, the UAV's bootloader retrieves the firmware and its associated signature and verifies its authenticity using the embedded public key stored in secure flash memory. If the verification is successful, the firmware is allowed to execute; otherwise, the system halts the boot process and logs an alert to prevent execution of potentially tampered code.

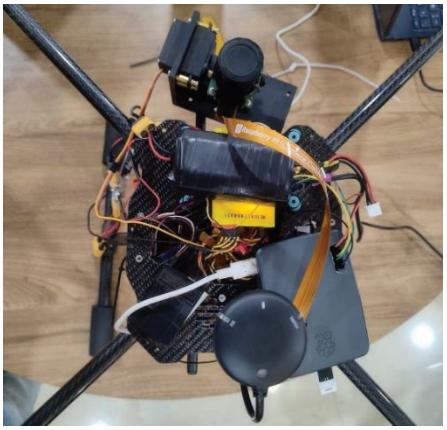

Fig. 2. Hardware of UAV

The third layer, the parameter access control layer, functions during runtime and is responsible for restricting unauthorized configuration changes. During flight or system configuration, any user attempting to modify UAV parameters must provide a valid access key, which is verified by the system before accepting any changes. Unauthorized access attempts are rejected and recorded for traceability.

Finally, the logging and monitoring layer ensures continuous tracking of system activities by maintaining detailed logs of firmware verification results, authentication attempts, and system integrity checks. These logs can be accessed through the Ground Control Station, such as Mission Planner, or exported for forensic analysis, thereby enhancing system transparency and security auditing capabilities.

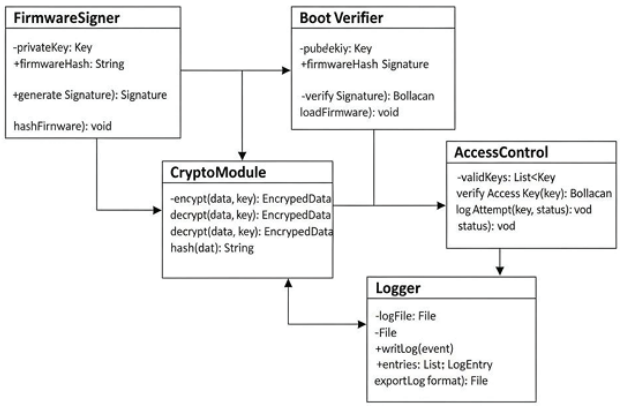

Fig. 3. Class Diagram of Secure UAV

This UML class diagram depicts the architecture of a secure boot and access control system, where responsibilities are clearly separated into five interconnected modules. The central CryptoModule handles all encryption, decryption, and hashing tasks. The FirmwareSigner uses cryptographic primitives to generate a digital signature for the firmware, which the BootVerifier then checks against a public key before deciding to load the firmware. System security is managed by the AccessControl module, which maintains a list of valid keys to verify user access. All critical events, including access attempts and cryptographic operations, are recorded by the Logger module, ensuring a complete audit trail. The overall structure emphasizes modularity and a secure, verifiable process for managing firmware integrity and system access.

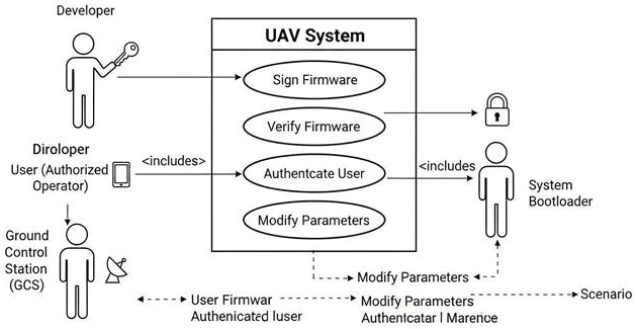

Fig. 4. Use Case Diagram of Secure UAV

The Use Case Diagram shows the functionality and actors: a Developer signs the firmware, the System Bootloader verifies it, and the Authorized Operator must Authenticate User before they are allowed to Modify Parameters.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

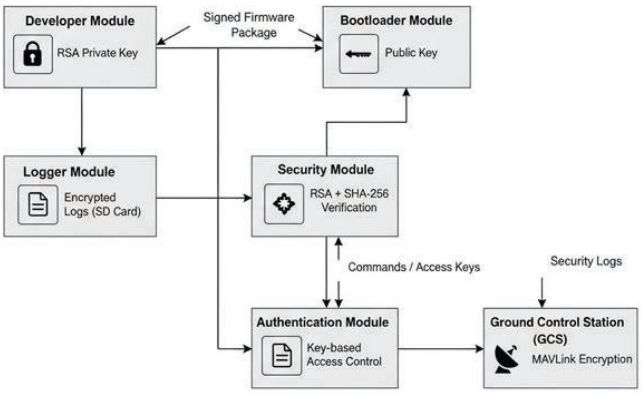

Fig. 5. Components of Secure UAV

The Component Diagram illustrates the high-level architecture of the Secure UAV System, showing how software and hardware modules work together to ensure robust security. The Developer Module signs firmware using an RSA private key, which the Bootloader verifies with a public key via the Security Module (RSA + SHA-256). The Authentication Module controls user access through key-based verification, while the Ground Control Station (GCS) communicates securely using MAVLink encryption. All critical events, including firmware checks and access attempts, are logged by the Logger Module on an encrypted SD card. This modular design ensures firmware integrity, secure access, and traceable UAV operations.

IV. RESULTS

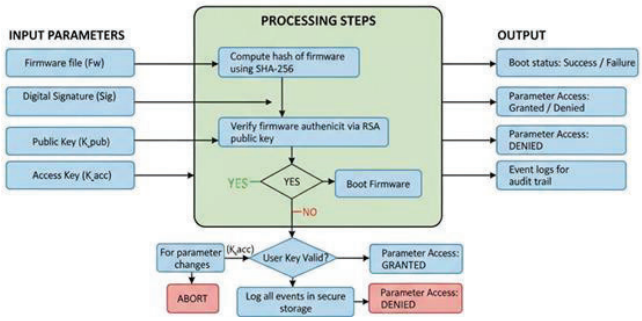

Fig. 5. Analysis Model of Secure UAV

This diagram represents the secure firmware validation and access control workflow implemented in the proposed UAV framework. The process begins with a set of input parameters, including the firmware image (Fw), digital signature (Sig), public key (Kpub), and access authentication key (Kacc). Initially, the firmware undergoes SHA-256 hashing to generate a cryptographic digest, which is then validated against the digital signature using the RSA public key. If the verification process succeeds, the system reports a successful boot status and proceeds with firmware execution. If verification fails, the framework checks whether the request involves modification of UAV parameters. In such cases, the validity of the user's authentication key is examined. Authorized users are granted parameter access, whereas invalid authentication attempts are denied. All critical events, including boot outcomes and access requests, are recorded to maintain a complete audit trail of system activities.

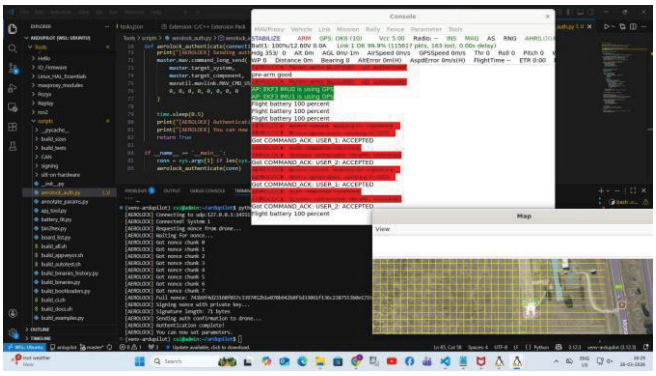

Fig. 6. Implementation of Securing firmware

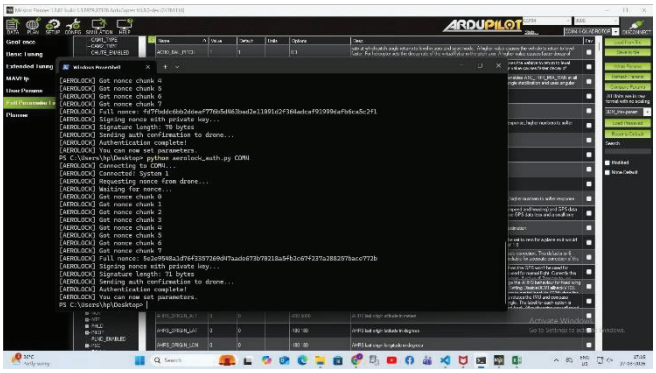

Fig. 7. Implementation of Securing firmware(The nonce chunks shows the key)

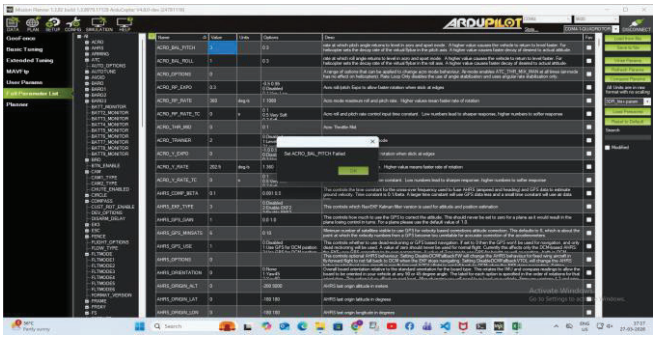

Fig. 8. Implementation of Securing firmware(Parameter changing failed due to unauthorised Access)

The primary objective of this work was to develop a secure UAV platform capable of resisting firmware tampering and preventing unauthorized configuration changes through the use of cryptographic verification and access control techniques. The proposed framework was designed to authenticate firmware during system startup by validating digitally signed firmware images, thereby

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

blocking the execution of altered or unsigned code. In addition, a runtime authentication mechanism was incorporated to ensure that UAV parameters could only be modified after successful key-based verification. Any unauthorized attempts were automatically rejected and documented. The framework also maintains detailed logs of firmware validation results and user authentication activities to support system auditing and post-flight investigations. Furthermore, the security mechanisms were intentionally designed to be lightweight, introducing only minimal performance overhead, with boot verification delays remaining within acceptable operational limits, typically below two seconds. The solution was also developed with scalability in mind, ensuring compatibility with open-source autopilot platforms such as ArduPilot and facilitating adaptation to related flight controllers, including PX4.

V. DISCUSSION

The implementation results demonstrate that the integration of cryptographic techniques can significantly improve firmware security and operational access control in UAV environments. The proposed framework successfully combines secure boot validation with runtime authentication, enhancing the system's ability to withstand firmware-related attacks and unauthorized user interactions.

Experimental observations confirmed that the secure boot mechanism effectively preserves firmware integrity by verifying the digital signature before execution. Any intentional modification to the firmware produced discrepancies between the generated hash and the decrypted signature, causing the boot process to terminate immediately. This behavior verifies the effectiveness of the SHA-256 hashing and asymmetric cryptography approach in identifying tampered or untrusted firmware and preventing the execution of malicious code. Beyond startup protection, the runtime authentication mechanism provides an additional layer of defense by limiting parameter modifications exclusively to authenticated users. The nonce-based challenge-response protocol ensures that every authentication session is unique, thereby reducing the risk of replay attacks. By validating digitally signed responses received through MAVLink communication, the UAV can reliably confirm the identity of the Ground Control Station and reject unauthorized command requests. This enhancement addresses a major weakness found in many conventional UAV systems, where configuration parameters often lack sufficient protection.

From a performance standpoint, the introduced security measures impose only a minor computational burden. The additional startup delay observed during testing was approximately one to two seconds, which had no noticeable impact on the normal operation of the UAV. These findings indicate that the proposed framework is lightweight enough for deployment on embedded platforms with limited computational resources, such as Pixhawk flight controllers. Likewise, the runtime authentication process introduced negligible latency during parameter updates, preserving a smooth user experience. The integrated logging mechanism further contributes to the overall reliability of the system by maintaining records of firmware validation outcomes, authentication requests, and unauthorized access attempts. Such records improve transparency and support post-mission investigations, enabling operators to identify security incidents and evaluate system behavior more effectively.

Overall, the results confirm that the combination of secure boot validation and runtime authentication provides a comprehensive and practical security solution for UAV applications. Unlike approaches that focus solely on firmware verification or communication security in isolation, the proposed framework delivers a unified defense strategy. Its successful implementation on the ArduPilot platform further demonstrates its practicality and suitability for real-world UAV deployments.

VI. CONCLUSION

This project successfully designed and implemented a secure, tamper-resistant firmware framework to address the growing cybersecurity concerns associated with UAV operations. The proposed solution integrates secure boot functionality using RSA-based asymmetric cryptography and SHA-256 hashing to validate firmware integrity before execution. In addition, a key-based authentication mechanism was introduced to restrict unauthorized parameter modifications during runtime.

Testing performed using the Pixhawk flight controller and Mission Planner environment demonstrated that the framework effectively prevents the execution of altered firmware while maintaining stable UAV operation. The system achieved reliable performance with only minimal computational overhead, introducing an average boot delay of approximately 1.6 seconds.

The outcomes of this work illustrate how cybersecurity mechanisms can be effectively embedded within UAV firmware to enhance trust, safety, and operational reliability in mission-critical domains such as defense, surveillance, agriculture, and emergency response. The scope of the proposed framework focuses on strengthening firmware protection and improving runtime security through cryptographic validation and controlled access mechanisms. Future enhancements may extend these capabilities by incorporating advanced anomaly detection and broader support for additional UAV platforms.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to their project guide and faculty members for their valuable guidance, support, and encouragement throughout the development of this project. We also thank our institution for providing the necessary resources and environment to carry out this work successfully. Finally, we appreciate the contributions of our team members for their cooperation and dedication in completing this project.

REFERENCES

[1] Aaron Yu, Iuliia Kolotylo, Hashim A. Hashim and Abdelrahman E. E. Eltoukhy, " Electronic Warfare Cyberattacks, Countermeasures, and Modern Defensive Strategies of UAV Avionics," IEEE Access, 2025.

[2] Ioannis Anagnostis, Panayiotis Kotzanikolaou, Christos Douligeris, " Understanding and Securing the Risks of Uncrewed Aerial Vehicle Services," IEEE Access, 2025.

[3] Yuanxu Zhu, Tianze Zhang, Aiying Wu, and Gang Shi, "PISCFF-LNet: A Method for Autonomous Flight of UAVs Based on Lightweight Road Extraction," 10.3390/drrones9030226, 20 March 2025.

[4] Alessandro Erba, John H. Castellanos, Sahil Sihag, Saman Zonouz and Nils Ole Tippenhauer, "Sensor Deprivation Attacks for Stealthy UAV Manipulation," arXiv, 2024.

[5] D. Wanner, H. A. Hashim, S. Srivastava, and A. Steinhauer, "UAV avionics safety, certification, accidents, redundancy, integrity, and reliability: A comprehensive review and future trends," Drone Syst. Appl., vol. 12, pp. 1-23, Jan 2024.

[6] A. Rugo, C. A. Ardagna, N. E. Ioini — A Security Review in the UAVNet Era: Threats, Countermeasures, and Gap Analysis, ACM Computing Surveys, 2023.

[7] Z. Lv, Y. Li, J. Wu, H. Lv — Securing the Internet of Drones Against Cyber-Physical Attacks, IEEE Internet of Things Magazine, 2021.

[8] Khamvilai, J. Dunham, E. Feron, and E. N. Johnson, "Avionics of aerial robots," Current Robot. Rep., vol. 2, no. 2, pp. 113-124, June 2021.

[9] Z. Lv, Y. Li, J. Wu, and H. Lv, "Securing the Internet of Drones against cyber Physical attacks," IEEE Internet Things Mag., vol. 4, no. 4, pp. 74-78, Dec 2021.

[10] M. Leonardi, M. Strohmeier, and V. Lenders, "On jamming attacks in crowdsourced Air traffic surveillance," IEEE Aerosp. Electron. Syst. Mag., vol. 36, no. 6, pp. 44-54, June 2021.

[11] V. P. Riabukha, "Radar surveillance of unmanned aerial vehicles," Radioelectronics Commun. Syst., vol. 63, no. 11, pp. 561-573, 2020.

[12] Dong Yang, Wei Dong, Wei Lu, Yanqi Dong, and Sirui Liu – Vulnerabilities Analysis and Secure Controlling for Unmanned Aerial System Based on Reactive Synthesis, VOL. 14 NO. 8, August 2015.

[13] J. Gu, T. Su, Q. Wang, X. Du, and M. Guizani, " Multiple moving targets surveillance Based on a cooperative network for multi-UAV," IEEE Commun. Mag., vol. 56, no. 4, pp. 82-89, April 2018.

[14] K. Hartmann, C. Steup — The Vulnerability of UAVs to Cyberattacks — A Risk Assessment Approach, CYCON Conference, 2013.

Author Biographies

ARCHANA KOKATKAR has completed her Bachelor's degree in Electronics and Telecommunication Engineering from Savitribai Phule Pune University and her Master's degree in VLSI System Design from Jawaharlal Nehru Technological University (JNTU). She is currently pursuing her Ph.D. at Vel Tech University. She is presently working as an Assistant Professor at JSCOE, Pune.

VISHAL GAJANAN ZALAKE is currently in the final year of his Bachelor of Engineering degree in Computer Engineering at Savitribai Phule Pune University, Pune, India, expected to graduate in 2026. He is currently undertaking a full-time internship at CerebroSpark Innovations Pvt. Ltd. His research interests include hardware and firmware design and optimization, authorization, and implementation.

VAISHNAVI VIJAY SARAF is currently in the final year of her Bachelor of Engineering degree in Computer Engineering at Savitribai Phule Pune University, Pune, India, expected to graduate in 2026. She is currently undertaking a full-time internship at CerebroSpark Innovations Pvt. Ltd. Her research interests include firmware design and optimization, cryptographic encryption, and implementation.

RADHEY PARIKSHIT BILDIKAR is currently in the final year of his Bachelor of Engineering degree in Computer Engineering at Savitribai Phule Pune University, Pune, India, expected to graduate in 2026. His research interests include firmware design and optimization, cryptographic encryption, and implementation.

SMITESH VIVEK SHINDE is currently in the final year of his Bachelor of Engineering degree in Computer Engineering at Savitribai Phule Pune University, Pune, India, expected to graduate in 2026. His reaerch interests include hardware design and optimization and implementation.