International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

A Survey of Secure and Reconfigurable System-on-Chip Architectures for UAV Flight Control Applications

Kalakurasa Rakesh
Associate Professor: ECE Department,
MVGR College of Engineering(A)
Vizianagaram, India

Moturi Satyanarayana
Professor: ECE Department,
MVGR College of Engineering(A)
Vizianagaram, India

***Abstract* - Safety-critical and mission-oriented applications have seen an increase in the use of Unmanned Aerial Vehicles (UAVs). These require reliable real-time flight control and robust security guarantees. Modern UAV platforms are integrated with sensing, computation, communication, and control capabilities utilizing embedded System-on-Chip (SoC) architectures. However, with increasing levels of autonomy, connectivity, and computational complexity, UAVs are all exposed to many types of cyber-physical security threats, both in dual-use and the civilian sector. Examples of threat actors to a UAV include communication spoofing, sensor manipulation, firmware tampering, and hardware Trojan attacks.**

This paper is a comprehensive survey of the state-of-the-art secure and reconfigurable System-on-Chips (SoCs) and SoC architectures for UAV flight control applications. It provides a systematic review of UAV flight control system architectures, embedded computing platforms, and UAV cyber-physical system-specific security challenges. Additionally, it defines a UAV flight control-specific threat model and attack taxonomy and discusses vulnerabilities associated with the communication, sensing, control, software, and hardware levels of a UAV. It also discusses the role of FPGA-based and FPGA-SoC platforms as a means of augmenting hardware-assisted security and runtime adaptability. Finally, the authors identify key research gaps and propose control-aware, dynamically reconfigurable secure SoC architectures for the next generation of UAV flight controllers.

I. INTRODUCTION

This Over the past few years, Unmanned Aerial Vehicles (UAV) have transitioned from being simply "remotely piloted" systems into autonomous computer-controlled systems that can perform complex tasks like surveillance, disaster relief, precision farming, and military operations [1], [2]. Central to each UAV's operation is their Flight Control System (FCS), which is responsible for real-time stabilization, navigation, and synchronous operation between actuators, while adhering to strict timing and safety constraints [3]. Increasingly, there are new UAV technologies that use Embedded Systems-on-Chip (SoC) that integrate all three functionality areas of flight control, sensor processing, communication and mission management in a one device due to weight and power limitations [4], [5]. However, as UAVs become more autonomous and connected to networks, the potential impact of cyber-physical attacks will increase significantly. Several research papers document instances of communication spoofing, sensor manipulation, firmware tampering, and other hardware-level attacks that expose flight control systems. [6]-[8] shows that most UAV flight controllers have been built around microcontrollers, or CPU-RTOS platforms powered primarily by software-based security mechanisms.[9] Because their architecture relies heavily on software-based security, they cannot provide adequate protection from sophisticated adversaries who have the ability to exploit a common memory, gain access to an update path for firmware, or have physical access to an embedded device to exploit it in some way.[10] Recent work indicates that the current security mechanisms available do not adequately protect the flight control loops of UAVs from attacks such as GPS spoofing, maliciously injecting commands into a UAV's flight control, and manipulating control signals sent to the UAV's flight control that will cause the UAV to either not complete its mission or completely lose the UAV.[11],[12] Because of the inadequate protection afforded by current software-only security mechanisms, researchers are turning to hardware-assisted security mechanisms and adaptive system architectures for UAV control. Therefore, a promising area of study will be the use of FPGA-based and FPGA-SoC platforms as alternative solutions because they provide extensive parallelism, offer fine-grain hardware isolation, and provide the ability for system adaptivity at runtime using Dynamic Partial Reconfiguration (DPR)[13]-[15]. Therefore, these capabilities provide avenues to conduct security, computation, and control co-designs, which ultimately suggest a need to perform a survey of the literature for secure and reconfigurable SoC architectures for UAV flight control applications.

II. UAV FLIGHT CONTROL SYSTEMS AND EMBEDDED SOC ARCHITECTURES

2.1 Overview of UAV Flight Control Systems A UAV Flight control system is an embedded safety-related subsystem that ensures that the UAV maintains its desired stability and the accuracy of its trajectory While executing both the proper trajectory and expected flight path on a timely basis [toggle Ref: not needed]. The Flight Controller continuously receives real time data from various Sensors, Makes an estimate of the Aircrafts State (where the Aircraft is momentarily relative to its intended Path) makes Control Commands (i.e., Move or turn), and provides that Control command to the Motors and/or Control Surfaces [16]. The Sensor Types typically found on UAV`s are: IMU, GNSS, Magnetometer, Pressure Sensor (Barometric Altimeter), Vision Sensors, & LiDAR (Laser Radar) [16]. UAV Flight Control algorithms are typically composed of a series of Hierarchical Feedback Control Loops - Inner loop to stabilize the Attitude of UAV; Mid-level Loop to control the Velocity of the UAV; and Outer Loop to regulate the Position of the UAV on the Ground or in the Air [3] [17].

The Feedback Control Loops typically Require Control Updates at Rates of 100Hz to 1Khz [17]. This results in requiring Deterministic Control Execution and Low Latency between Control Commands when controlling the UAV during Operations. As such, The embedded Platform that implements the Various Flight Controller Functions Must Meet the Required Define of Real-Time Performance, as well as High Reliability & Fault Tolerance.

2.2 Embedded Computing Platforms for UAV Flight Control

The choice of embedded computing platform significantly influences the performance, security, and adaptability of UAV flight control systems [4], [5]. Current UAV platforms predominantly employ microcontroller-based systems, CPU–RTOS-based platforms, or reconfigurable hardware architectures.

2.2.1 Microcontroller-Based Flight Controllers

Microcontroller-based flight controllers are widely used in small and medium-sized UAVs due to their low cost, compact form factor, and energy efficiency [9]. These platforms typically integrate a single processor core with limited memory and peripheral interfaces. While microcontroller-based controllers provide adequate real-time performance for basic stabilization and navigation tasks, they rely heavily on software-based security mechanisms [10]. Such platforms lack hardware-enforced isolation between control, communication, and auxiliary tasks, making them vulnerable to firmware tampering, memory corruption, and privilege escalation attacks [6], [8]. Moreover, the static nature of microcontroller architectures limits their ability to adapt to evolving security threats during flight [21].

2.2.2 CPU and RTOS-Based Systems

Real-time operating systems running on CPU-based platforms enhance modulatory, task scheduling and determinism in relation to bare metal microcontroller systems. This type of operating system allows priority driven tasks (such as task execution), inter process communication and improved fault mitigations (i.e. error detection) via a real-time operating system for UAV controllers [18]. Despite many of the advantages of a real-time operating system, these types of systems continue to be vulnerable to a variety of cybersecurity threats, including kernel-level exploits, shared memory vulnerabilities and malicious code inserted through a task [19]. Security protocols, such as secure boot, memory protection units and encrypted firmware updates, provide some degree of mitigation against risks but do not adequately protect the hardware level from being compromised or adjusted post deployment [10],[24].

2.2.3 FPGA and FPGA-SoC-Based Architectures

Cited as sources: References [13] and [14] Hardware abstractions can provide equivalent levels of parallelism and determinism at the hardware level; therefore, FPGAs can implement function-specific hardware driving sensor fusion, signal processing, and control law executions.

Reference(s): [20] The use of an initialized piece of hardware, i.e., FPGA-SoC may enable separation of critical hardware from noncritical hardware, while also reducing the execution time/cycles of a function because no time is spent programming control logic programming. Hence, implementations in FPGAs reduce the time taken to complete, and they improve functional and performance isolation between functionally independent functions (or modules) [20] FPGA-SoCs allow the interoperability of logic-based accelerated software (i.e., programming of a software-based application) with hardware (i.e., FPGAs) and allow for the creation of mixed-criticality systems where there are life-safety critical functions (e.g., flight control) that are isolated from non-critical functions (e.g., communication) while simultaneously providing hardware-accelerated security features such as hardware-assisted cryptographic capabilities and secure boot processes [21][25].

2.3 Comparative Analysis of Embedded Platforms

A comparative analysis of embedded platforms reveals distinct trade-offs between flexibility, security, and real-time performance. Microcontroller-based systems excel in simplicity and energy efficiency but provide limited hardware isolation and security extensibility [9], [10]. CPU–RTOS platforms improve modularity and scheduling determinism but remain constrained by software-centric protection mechanisms [18], [19]. In contrast, FPGA and FPGA-SoC architectures offer strong spatial and temporal isolation, deterministic execution, and the potential for runtime adaptability through reconfiguration [13], [15]. However, these platforms introduce increased design complexity and require specialized development expertise. Despite these challenges, FPGA-SoCs represent a promising foundation for secure and adaptive UAV flight control systems, particularly in safety- and mission-critical applications [20].

2.4 Security Implications of SoC-Based UAV Controllers

Many UAV implementations use static hardware configurations where the original SoC features, including the advantage of FPGA-SoC’s reconfigurability, are not being fully utilized for security enforcement [21]. The static nature of these designs means they have been designed with a specific threat environment in mind, which restricts their ability to adjust to unforeseen threats, e.g. attacks that happen after the mission has started.

The challenges of static hardware designs indicate the need for architectures that combine real-time control with adaptive and hardware-based security mechanisms. Reconfigurable SoC platforms are one option for providing dynamic security policy adjustments, isolation of attacked modules, and resilient UAV flight control under conditions of hostility [14], [15], [25].

Table 1: Comparison of Embedded Platforms for UAV Flight Control

Platform	**Reconfigurability**	**Security Support**	**Real-Time Capability**	**Limitations**
MCU-based	No	Low	High	Static security
CPU + RTOS	No	Medium	High	Software-centric

Platform	**Reconfigurability**	**Security Support**	**Real-Time Capability**	**Limitations**
FPGA	Yes	Medium	Very High	Design complexity
FPGA-SoC	Yes (DPR)	High	Very High	Limited adoption

III. UAV SECURITY THREAT MODEL AND ATTACK TAXONOMY

3.1 UAV Security Threat Model

Unmanned Aerial Vehicles operate as distributed cyber–physical systems that tightly couple sensing, computation, communication, and control. Unlike traditional embedded systems, UAVs are exposed to both cyber and physical adversaries due to their wireless connectivity, mobility, and deployment in untrusted environments [6], [22]. A comprehensive security threat model for UAVs must therefore consider attacks targeting multiple layers of the system stack.

In this work, the threat model assumes adversaries with varying capabilities, including remote attackers with access to wireless communication channels, insider attackers capable of exploiting firmware update mechanisms, and physical attackers with limited or full access to onboard hardware [7], [21]. The attacker's objectives may include loss of control, mission disruption, data exfiltration, or persistent compromise of the flight controller.

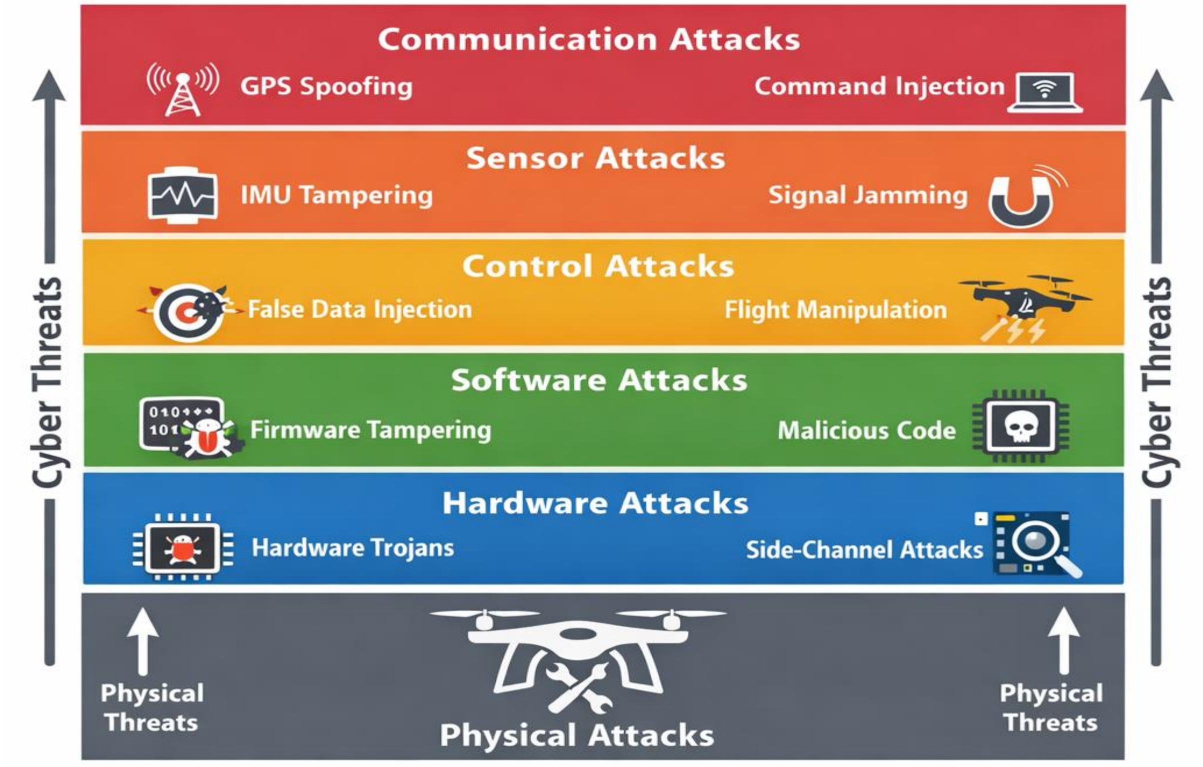


Figure 3.1. Layered UAV security threat model illustrating attack surfaces across communication, sensor, control, software, and hardware layers of the flight control system.

This figure highlights the cyber–physical nature of UAV security threats, where attacks originating in cyber layers can propagate to physical flight dynamics and safety-critical control loops [6], [11], [21].

3.2 Attack Surface in UAV Flight Control Systems

The UAV flight control system exposes a broad attack surface spanning communication interfaces, sensor inputs, control logic, software components, and hardware infrastructure. Attacks targeting any of these layers can propagate across the cyber–physical boundary, directly affecting flight stability and safety [3], [11]. The real-time and safety-critical nature of flight control loops amplifies the impact of security breaches. Even short-lived or low-rate attacks can destabilize feedback control systems, resulting in degraded performance or catastrophic failure [17].

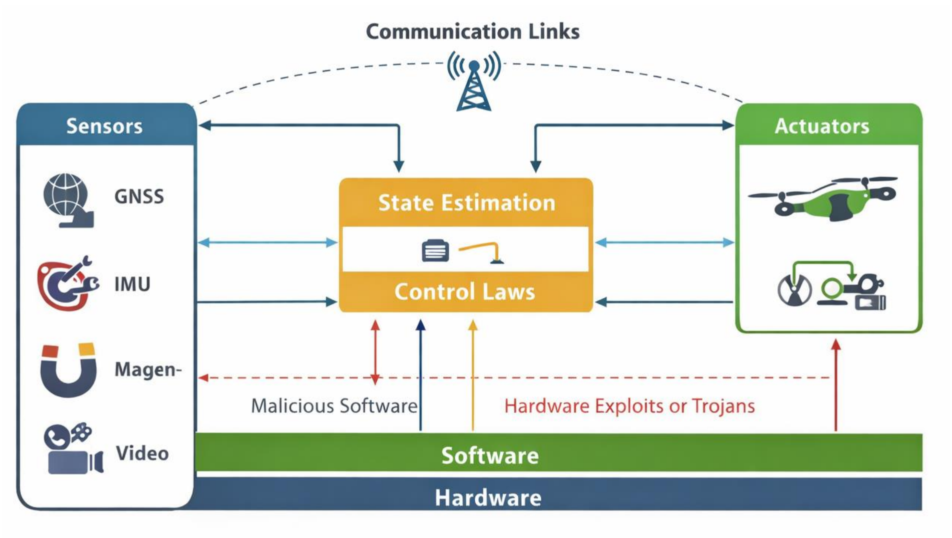


Figure 3.2. Attack surface of a UAV flight control system showing interactions between sensing, computation, communication, and actuation.

The figure illustrates how vulnerabilities at different layers can directly affect real-time control loops and system stability [3], [17].

3.3 Communication Layer Attacks

Communication layer attacks exploit wireless links used for command and control, navigation, and telemetry. Global Navigation Satellite System (GNSS) spoofing and jamming are among the most widely reported threats, enabling attackers to manipulate position estimates and mislead navigation algorithms [11], [12]. Command injection and telemetry hijacking attacks target unencrypted or weakly authenticated communication channels, allowing adversaries to issue malicious commands or suppress critical system status information [6]. Such attacks pose significant risks to beyond-visual-line-of-sight (BVLOS) and autonomous UAV operations.

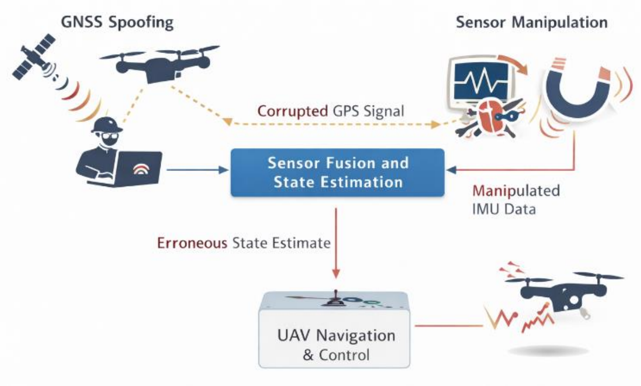


Figure 3.3. Communication and sensor layer attack scenarios in UAVs, including GNSS spoofing and inertial sensor manipulation

3.4 Sensor Layer Attacks

Sensor layer attacks aim to corrupt the integrity of measurements used for state estimation and control. Inertial sensor manipulation, magnetic interference, and GNSS signal distortion can introduce erroneous data into sensor fusion algorithms, leading to incorrect attitude or position estimates [23].

Fault injection attacks, including electromagnetic or power-based disturbances, further threaten sensor reliability and can induce transient or permanent faults in sensing components [21]. Due to the reliance of flight control algorithms on accurate sensor feedback, sensor-layer attacks can rapidly propagate to control-layer failures.

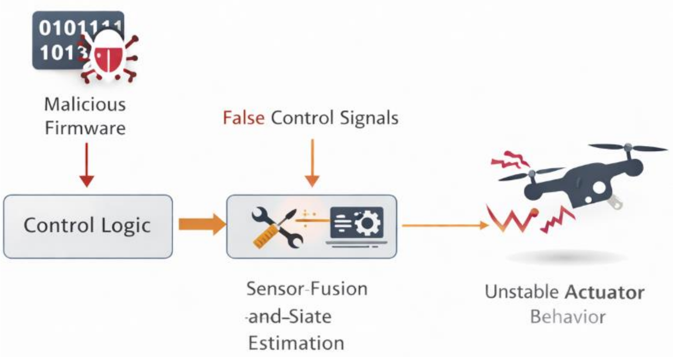


Figure 3.4. Propagation of control-layer and software-based attacks in UAV flight control systems.

3.5 Control Layer Attacks

Control layer attacks directly target the feedback control algorithms or actuator command paths. These attacks may involve modifying control gains, injecting false reference signals, or corrupting actuator outputs [3]. In tightly coupled control systems, even minor perturbations can result in oscillations, instability, or loss of control authority [17]. Unlike communication or software attacks, control-layer attacks exploit the physical dynamics of the UAV, making detection and mitigation particularly challenging.

3.6Software and Firmware Attacks

The software and firmware components play an essential role in the UAV systems as they are a primary attack vector for UAV systems. Many of the ways in which adversaries can gain persistent access to the flight controller include: firmware tampering, using malicious updates (e.g., to the firmware), and conducting code injection attacks. Vulnerabilities in bootloaders, update mechanisms, and memory exploit these bootloaders to increase exposure to exploitation. Software-based security techniques (e.g., secure boot and firmware encryption) provide for improved resilience against runtime attacks and execution environments that have been compromised, but still do not provide full protection from these attacks [10].

The figure emphasizes how firmware tampering and malicious control inputs can directly impact actuator commands and flight stability [3], [19], [24].

3.7 Hardware-Level Attacks

Attacks against hardware represent one of the largest classes of threats to UAV flight control systems. Hardware Trojans, side channel attacks, and manipulation of configuration bitstreams have the potential to bypass software-based defenses and affect the integrity of systems at their core [21], [25].

In FPGA-based UAV systems, unauthorized modification of the configuration bitstreams is an especially significant threat that can change control logic or disable the UAV's security features. Physical attacks on UAV systems, including probing of circuit boards, reverse engineering of the flight controller, and fault injection will compound these threats and make them much more significant in hostile environments.

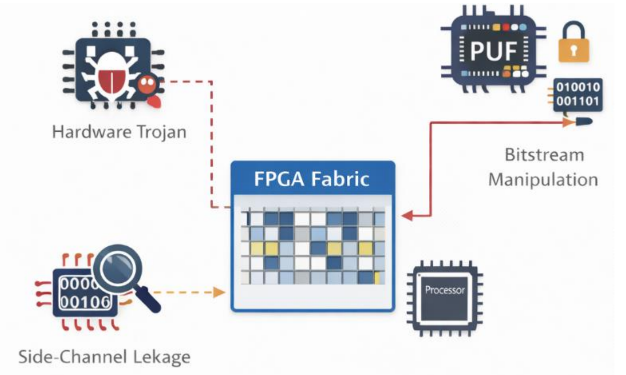
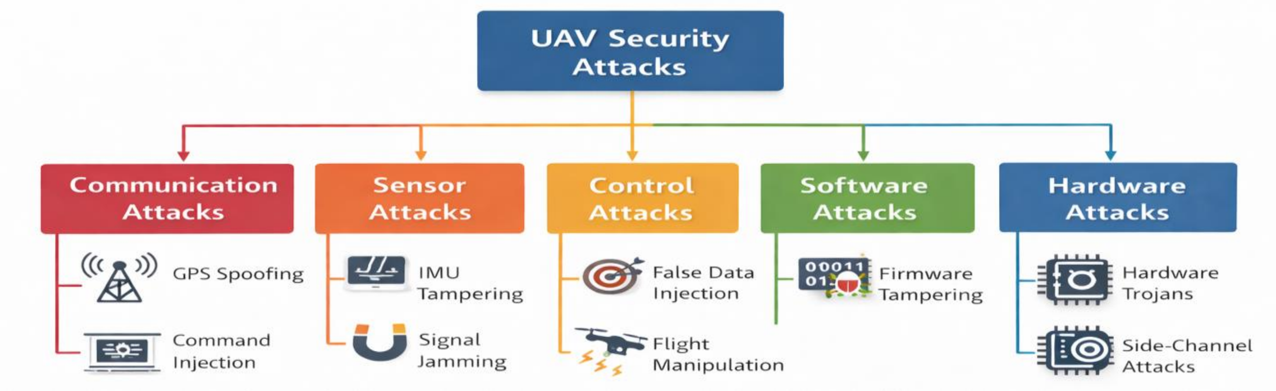


Figure 3.5. Hardware-level attack vectors targeting FPGA-SoC-based UAV flight controllers.

These attacks bypass software defenses and pose significant risks to system integrity, confidentiality, and availability [21], [25].

3.8 Attack Taxonomy Summary



3 .6. Taxonomy of security attacks targeting UAV flight control systems.

The taxonomy categorizes attacks based on their origin and impact on cyber–physical flight control operations, providing a structured basis for security analysis and mitigation design [6], [11], [21].
Table 2: UAV Attack–Impact–Mitigation Overview

Attack Layer	Example Attack	Impact	Typical Mitigation
Communication	GPS spoofing	Navigation failure	Sensor fusion

Table 3 summarizes the primary attack categories, representative threats, and their potential impact on UAV flight control systems.

Table 3: UAV Attack Taxonomy and Impact

Attack Layer	Representative Attacks	Impact on Flight Control
Communication	GNSS spoofing, command injection	Navigation failure
Sensor	IMU manipulation, fault injection	State estimation errors
Control	False control signals	Instability, loss of control
Software	Firmware tampering	Persistent compromise
Hardware	Trojans, side-channel attacks	Complete system breach

3.9 Discussion and Research Gaps

The attack taxonomy and threat model that is shown provide insight into the limitations of current UAV security solutions that are primarily focused on software-based or static architectures. These solutions do not provide adequate protection against an adaptive adversary who is able to exploit weaknesses in both the control level and the hardware level [6] and [21].

The analysis highlighted the need for adaptive and hardware-based or assisted security mechanisms that can respond to the changing threat landscape without delaying or degrading real-time flight control. This will drive future efforts to develop dynamically reconfi-gurable SoC architectures that contain all three elements (control, computation, security) within a single hardware implementation.

Attack Layer	**Example Attack**	**Impact**	**Typical Mitigation**
Sensor	IMU manipulation	Attitude instability	Redundancy
Control	False commands	Loss of control	Control validation
Software	Firmware tampering	System compromise	Secure boot
Hardware	Trojan insertion	Data leakage	PUF, DPR

IV.HARDWARE SECURITY IN SYSTEM ON CHIP (SOC) PLATFORM FOR UAV APPLICATIONS

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

4.1 Role of Hardware Security in UAV SoCs

In an environment that may be dangerous or may even contain hostile elements, the only method that can be used to protect Unmanned Vehicles (UAVs) from being compromised, manipulated or tampered with is through the inclusion of built-in Hardware Security in a System on Chip (SoC). The use of built-in Hardware Security will provide a second level of security beyond that provided by a Software-based security system and help ensure that when a UAV is in the operation stage, it has a secure boot sequence, a trusted execution, hardware-based Cryptographic Isolation and Tamper Resistance—necessary for the safe execution of Flight Controllers and other UAV systems involved in a critical flight operation [1], [14], [18].

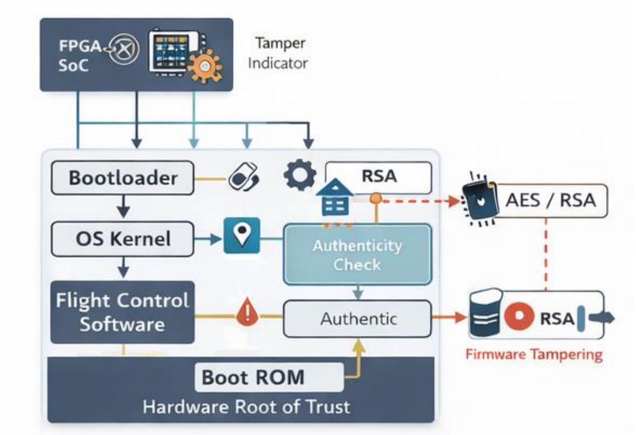


Figure 4.1: Secure boot and hardware root of trust architecture in UAV SoCs

4.2 Secure Boot and Hardware Root of Trust

Secure boot is the main line of defense against executing malicious code during the boot process. In UAVs, secure boot relies primarily on hardware RoT through immutable boot ROM, eFuses, or PUFs to verify that the bootloader, operating system, and flight control software are cryptographically signed prior to executing them [3], [11]. SoCs, such as Xilinx Zynq UltraScale+ and Intel SoC FPGAs, provide multi-stage secure boot protcols that include RSA/ECC based authentication and AES encrypted bitstreams for each boot stage. The use of secure boot prevents unauthorized firmware injection and mitigates the risk of supply chain attacks, which are particularly problematic for both defense and surveillance UAVs [7], [15].

4.3 Physically Unclonable Functions (PUFs)

To generate a unique fingerprint for each device, Physically Unclonable Functions (PUF) take advantage of uncontrolled variations during manufacture of silicon. In Unmanned Aerial Vehicle (UAV) System on Chips (SoC) PUFs are heavily relied upon for secure key generation, device authentication and protection against counterfeiting without needing non peripheral key storage [9][16].

Some of the types of PUF’s include SRAM PUF’s, Ring Oscillator PUF’s, etc. Among them, SRAM PUF’s appear to be the preferred implementation on a UAV platform because of their low resource footprint and speed; however, the variations in the environmental conditions that the UAV will experience and the degradation of the device over time will require robust error correcting algorithms and helper data algorithms to maintain reliability over time [12][21].

4.4 Hardware Cryptographic Accelerators

To meet the stringent real-time and power constraints of UAV systems, SoCs integrate dedicated hardware cryptographic accelerators. These accelerators offload computationally intensive tasks such as encryption, decryption, hashing, and authentication from the main processor, enabling secure communication without compromising flight control deadlines [4], [19].

Typical cryptographic blocks include:

- AES engines for secure data storage and communication
- SHA-2/SHA-3 modules for integrity verification
- RSA and ECC engines for key exchange and authentication

Hardware acceleration not only improves performance but also reduces exposure to timing-based side-channel attacks compared to software implementations [8], [22].

4.5 Protection Against Side-Channel and Physical Attacks

UAV SoCs are vulnerable to physical attacks such as power analysis, electromagnetic (EM) leakage, fault injection, and hardware Trojan insertion. Side-channel attacks can reveal cryptographic keys by analyzing power consumption or EM emissions, while fault injection attacks can bypass authentication checks or corrupt control logic [5], [17].

Countermeasures include:

- Power and clock randomization
- Dual-rail logic and masking techniques
- Sensor-based tamper detection
- On-chip voltage and temperature monitors

These defenses are increasingly integrated into secure SoC designs to protect UAV platforms deployed in adversarial environments [10], [23].

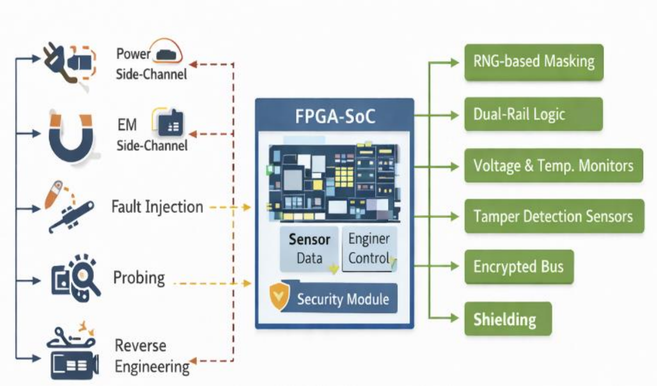


Figure 4.4: Threats and countermeasures against side-channel and physical attacks in UAV SoCs

4.6 FPGA Security Primitives and Bitstream Protection

FPGA-based UAV SoCs provide flexibility through reconfigurable logic but introduce additional security challenges related to bitstream confidentiality and integrity. Encrypted bitstreams, authenticated configuration, and secure key storage mechanisms are essential to prevent reverse engineering and malicious reconfiguration [13], [24]. Advanced SoCs support partial reconfiguration with security controls, allowing trusted modules to be dynamically loaded or updated during operation. When combined with hardware firewalls and isolation regions, this capability enables adaptive security architectures for UAV flight control systems [18], [25].

CONCLUSION

This paper presented a comprehensive survey of secure and the focus of this research is on reconfigurable System-on-Chip (SoC) architectures for UAV flight control applications. These platforms will be investigated with a strong emphasis on the incorporation of hardware-assisted security mechanisms and dynamically adaptive capabilities. As UAVs have evolved from being remotely piloted to becoming autonomous, networked and mission-critical cyber-physical systems, there are now many more stringent requirements with respect to security for their embedded control architectures than ever before. The use of traditional software-based protective measures is no longer adequate for defending against sophisticated cyber, physical or hardware-based attacks.

This study used a survey methodology to methodically assess current UAV flight control architectures and their limitations with respect to isolation, resilience, and adaptability, when compared to traditional microcontroller & RTOS-based platforms. The use of a detailed threat model and attack taxonomy allowed for the capture of the multi-layered nature of UAV vulnerabilities that exist within the communication, sensor, control, software, and hardware layers. This analysis identified that for attacks against lower layers (typically the hardware and/or the firmware), flight safety and mission integrity can be compromised through direct attack.

An extensive examination of modern SoC hardware security functions was conducted, which included secure boot chains, hardware root of trust, physically unclonable function, cryptography accelerator and side-channel and physical attack protections. FPGA-SoC technology is considered to be very viable because of the combination of strong hardware isolation and deterministic real-time control. Furthermore, the need for bitstream protection and secure configuration management for UAV-based reconfigurable systems was also noted.

One of the major findings of this study is that static security architectures do not fit well into the dynamic, adversarial environment of the UAV. Dynamic partial reconfiguration has the potential to allow UAV flight control systems, to adjust their security policy according to changing conditions; isolate compromised components; and reconfigure the use of trusted hardware, while preserving the safety and reliability of the UAV flight control system. Hence, the potential for adaptive rather than preventive hardware security.

Although many advancements have taken place in this area, there are still several challenges in this area for further development and deployment. Some examples of challenges that will need to be resolved and developed for further implementation include: assuring predictable reconfiguration latency; minimizing power and area overhead; and providing formal verification of dynamically reconfigurable security architectures. Additionally, there are many design challenges associated with coordinating security adaptations with flight control time constraints and certification requirements.

In conclusion, Secure, Dynamically reconfigurable SoCs represent an excellent foundation for next-generation UAV flight control systems (and are very much forward-looking). Combining hardware-rooted trusts with the design flexibility provided by reconfigurable hardware (thereby creating "adaptable security") will allow these systems to grow in severity to any level needed due to evolving threats while providing the same functionality as current UAVs/flight control systems. This survey provides researchers/system designers with a foundation to begin further research into autonomous, trustworthy, and mission-resilient UAVs.

REFERENCES

[1] A. Perrig, J. Stankovic, and D. Wagner, “Security in wireless sensor networks,” *Communications of the ACM*, vol. 47, no. 6, pp. 53–57, Jun. 2004.

[2] R. Austin, *Unmanned Aircraft Systems: UAVS Design, Development and Deployment*, 2nd ed. Chichester, U.K.: Wiley, 2010.

[3] R. Beard and T. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton, NJ, USA: Princeton Univ. Press, 2012.

[4] S. Trimberger, “Three ages of FPGAs: A retrospective on the first thirty years of FPGA technology,” *Proc. IEEE*, vol. 103, no. 3, pp. 318–331, Mar. 2015.

[5] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in *Proc. CRYPTO*, Santa Barbara, CA, USA, 1999, pp. 388–397.

[6] Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava, “Non-invasive spoofing attacks for anti-lock braking systems,” in *Proc. CHES*, Santa Barbara, CA, USA, 2013, pp. 55–72.

[7] M. Tehranipoor and F. Koushanfar, “A survey of hardware Trojan taxonomy and detection,” *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 10–25, Jan.–Feb. 2010.

[8] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. New York, NY, USA: Springer, 2007.

[9] G. E. Suh and S. Devadas, “Physical unclonable functions for device authentication and secret key generation,” in *Proc. DAC*, San Diego, CA, USA, 2007, pp. 9–14.

[10] N. Rostami, F. Koushanfar, and R. Karri, “A primer on hardware security: Models, methods, and metrics,” *Proc. IEEE*, vol. 102, no. 8, pp. 1283–1295, Aug. 2014.

[11] T. E. Humphreys *et al.*, “Assessing the spoofing threat: Development of a portable GPS civilian spoofer,” in *Proc. ION GNSS*, Savannah, GA, USA, 2008, pp. 2314–2325.

[12] M. Psiaki and T. Humphreys, “GNSS spoofing and detection,” *Proc. IEEE*, vol. 104, no. 6, pp. 1258–1270, Jun. 2016.

[13] K. Vipin and S. Fahmy, “A survey of partial reconfiguration in FPGAs,” *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–39, Aug. 2018.

[14] A. Waksman and S. Sethumadhavan, “Silicon secure: Hardware-based computer security,” *IEEE Security & Privacy*, vol. 8, no. 5, pp. 28–37, Sep.–Oct. 2010.

[15] Xilinx Inc., *Zynq UltraScale+ MPSoC Technical Reference Manual*, San Jose, CA, USA, 2022.

[16] D. Lim *et al.*, “Extracting secret keys from integrated circuits,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 10, pp. 1200–1205, Oct. 2005.

[17] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton, NJ, USA: Princeton Univ. Press, 2008.

[18] H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, 2nd ed. New York, NY, USA: Springer, 2011.

[19] E. Rescorla, *SSL and TLS: Designing and Building Secure Systems*. Boston, MA, USA: Addison-Wesley, 2001.

[20] S. S. Shende and M. B. Patil, “FPGA-based real-time embedded control systems,” *IEEE Embedded Systems Letters*, vol. 6, no. 3, pp. 53–56, Sep. 2014.

[21] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, “Trustworthy hardware: Identifying and classifying hardware Trojans,” *Computer*, vol. 43, no. 10, pp. 39–46, Oct. 2010.

[22] D. Dolev and A. Yao, “On the security of public key protocols,” *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[23] J. Clark and J. Jacob, “Fault injection attacks,” *IEEE Security & Privacy*, vol. 5, no. 2, pp. 30–37, Mar.–Apr. 2007.

[24] S. Skorobogatov, “Semi-invasive attacks—A new approach to hardware security analysis,” *University of Cambridge*, Tech. Rep. UCAM-CL-TR-630, 2005.

[25] M. Peiffer, A. DeHon, and S. Devadas, “Towards practical secure FPGA reconfiguration,” in *Proc. FCCM*, Napa, CA, USA, 2014, pp. 129–136.