2026 Volume: 02 Issue: 06 www.ijrpa.com ISSN 2456-9995 Research Article

International Journal Research Publication Analysis

Page: 01-35

# HARDWARE SECURITY FRAMEWORK FOR FPGA-SOC BASED UAV FLIGHT CONTROL SYSTEMS USING PUF-ASSISTED SECURE DYNAMIC RECONFIGURATION

**[*1]Kalakurasa Rakesh, [2]Dr. Moturi Satyanarayana**

[1]Associate Professor, [2]Professor

ECE Department, MVGR College of Engineering(A) Vizianagaram, India.

**Article Received: 29 April 2026**

**Article Revised: 19 May 2026**

**Published on: 09 June 2026**

***Corresponding Author: Kalakurasa Rakesh**

Ass,ociate Professor, ECE Department, MVGR College of Engineering(A) Vizianagaram, India.

DOI: https://doi-doi.org/101555/ijrpa.3944

## ABSTRACT

FPGA-SoC platforms are becoming more popular for Unmanned Aerial Vehicle (UAV) systems due to their deterministic real-time processing capacity, parallel processing ability, and ability to provide hardware acceleration. Inputting a bitstream, altering the hardware, cloning, and side channel leakage are new attack vectors on the programmable logic and dynamic partial reconfiguration (DPR) at the hardware level. The current solutions are mostly based on a single security solution or on a static FPGA design and offer limited support for secure runtime reconfiguration.

In this paper, a DPR-aware secure FPGA-SoC architecture for UAV flight control systems is proposed, which is integrated from secure boot, hardware root of trust, device authentication using PUF, hardware cryptographic acceleration, and hardware-enforced isolation in a single runtime security framework. The introduction of a Secure Reconfiguration Manager (SRM) enables runtime (DPR) operations to verify, decrypt, and authenticate partial bitstreams.

It has been implemented on a Xilinx Zynq-7000 FPGA-SoC platform, and its performance has been analyzed and compared based on the utilization of hardware resources, latency, power overhead, and security effectiveness during runtime. The results of the experiments demonstrate that the introduced control-loop latency is only 2.5% larger than the baseline, while the runtime security is significantly enhanced. Optimization of cryptographic functions in FPGA resulted in latency 24 times lower than software execution and throughput 15 times higher. The security validation experiments also validated the ability to prevent injection of

malicious bitstreams, to resist hardware cloning by binding to the PUF, and to decrease the side-channel leakage in timing experiments.

The proposed architecture permits authenticated DPR at runtime with integrated hardware-rooted trust and real-time validation, which is different from existing FPGA security approaches that are based on static configurations. The result highlights that secure runtime DPR can be realized while meeting the real-time UAV control constraints without breaking them, thus offering a secure hardware-based security solution for adaptive FPGA-SoC UAV platforms.

**KEYWORDS:** UAV security, FPGA-SoC, hardware root of trust, PUF, cryptographic acceleration, dynamic partial reconfiguration.

## 1. INTRODUCTION

The use of FPGA-SoC platforms in UAV systems has been growing in recent years, as they feature deterministic execution and hardware acceleration, critical properties for mission-critical applications, and runtime adaptability [1], [2], [34], [35]. But the programmable logic and embedded processors provide hardware-based attack surfaces that have to be considered if they are used in conjunction with Dynamic Partial Reconfiguration (DPR). An attacker may exploit the system configuration interface, shared interconnect, and runtime reconfiguration to compromise the integrity of the system and its operational safety [1], [2], [4], [34], [35].

The primary ways to secure UAVs are mainly through the software level and secure communication protocols. These mechanisms address network-related attacks but are not enough to protect against attacks on the hardware level such as bitstream injection, hardware cloning, side channel leakage, and modifying the runtime without authorization. Previous research on FPGA security has focused on the issues of secure boot, hardware roots of trust, physically unclonable functions (PUFs), and cryptographic accelerators, but few studies have explored their application on FPGAs in dynamically reconfigurable UAV systems.

Dynamic Partial Reconfiguration (DPR) enables to exchange FPGA hardware modules at runtime without affecting the critical system performance and offers flexibility and efficient utilization of resources. However, other attacks can be made against DPR with some bitstream manipulation and unauthorized hardware deployment, such as in [15]-[16], [18], [36], and [37]. Existing solutions consider static configurations of systems and cannot enforce security at runtime for dynamically reconfigured UAV architectures.

In order to overcome these drawbacks, this paper suggests an architecture that combines FPGA and SoC into the secure UAV flight control system aware of the DPR. The proposed architecture integrates secure boot, hardware root of trust, PUF device authentication, hardware cryptographic acceleration, and hardware-enforced isolation within the same runtime security architecture. To provide authentication and control of runtime partial reconfiguration operations, a Secure Reconfiguration Manager (SRM) is added.

The novelty of this work lies in the fusion of DPR-aware runtime security measures with the real-time constraints of UAVs and, thereby, the secure adaptation of the hardware without disturbing the deterministic operation of UAVs.

The key achievements of this work are outlined as follows:

- Secure aware architecture for UAV systems, based on DPR, designed using an FPGA-SoC.
- Securing the deployment of a runtime bitstream using an authenticated Secure Reconfiguration Manager.
- Use a PUF to prevent partial bitstream binding and cloning.
- Low latency cryptographic protection (hardware accelerated).

Experimental verification of the ability to reconfigure the FPGA for security at run-time with minimal real-time performance impact on an FPGA platform based on the Zynq-7000 family of FPGAs is conducted.

## 2. RELATED WORK

### 2.1 UAV Security Challenges

The wireless connectivity, autonomous operation, and the deployment in adversarial environments expose UAV systems to a variety of cyber-physical threats. Those studies in the past revealed vulnerabilities of spoofing, command injection, denial of service, and navigation manipulation [3] [31]. These works provide additional insights into UAVs' threat models but are primarily focused on software and communication layers' protection.

The threats at the hardware level are still a big problem in FPGA-based UAV platforms. Changes to system behavior or to subvert mission-critical system control operations might be achieved via configuration bitstreams, debug interfaces, and run-time reconfiguration mechanisms.

### 2.2 FPGA Hardware Security

For edge intelligence, autonomous UAV operation, and hardware-accelerated security services [10]–[14], [34], [35], and [40], FPGA-SoC devices have emerged as important computing platforms. Programmability can have drawbacks, such as bitstream exposure, unauthorized hardware modification, and hardware Trojans, however.

There are some existing FPGA security approaches that look at secure boot, FPGA hardware roots of trust, and trusted execution. Most of these methods are geared towards static architecture and don't take into account threats during runtime; they merely guarantee the security of initialization.

### 2.3 PUF-Based Authentication

Physically Unclonable Functions (PUFs) are hardware-based identities based on variations that are fundamental to manufacturing and are emerging as an essential primitive for lightweight device authentication and secure key generation for FPGA-based systems [14]–[16], [41], [42]. The device-specific PUF responses are used to generate runtime cryptographic keys that do not need to be stored in a non-volatile memory.

A few studies have been recently conducted on software authentication and key management for reconfigurable embedded systems using PUFs. A combination of PUF keys with runtime DPR security enforcement has not been widely explored; however, such a combination is crucial for UAV-related FPGA-SoC architectures [41] and [42].

### 2.4 Hardware Cryptographic Acceleration

FPGA-based hardware cryptographic accelerators offer deterministic execution and are much faster than software implementations and are well suited for real-time UAV security applications [18], [19], and [40].

The hardware acceleration of AES execution can easily achieve hardware acceleration of DPR and meet the timing demands in the UAV control loop, which supports the use of authenticated DPR.

While these benefits are very positive, most of the currently available work is focused on performance optimization, and there is no coordinated integration with the mechanisms for enforcing runtime security.

### 2.5 Isolation and Access Control

At runtime, AXI firewall policies provide a group of dynamically reconfigurable regions isolated from mission-critical control paths and cryptographic resources [28, 29].

The implementations boost system integrity in mixed FPGA-SoC designs. There are, however, difficulties for maintaining secure isolation in runtime reconfigurations, especially if the dynamically loaded modules are involved in important UAV control functions.

**2.6 DPR Security Challenges**

While DPR offers flexibility and better use of resources, recent research has shown that runtime reconfiguration creates extra attack surfaces, such as bitstream tampering, malicious module insertion, and hardware reverse engineering [15, 16, 18, 36, 37]. In the current state of the art, most DPR approaches emphasize mainly performance and efficiency in reconfiguration without much attention to integrated runtime security enforcement.

The research gaps and motivation are described. The research gaps and motivations are presented.

While security research of FPGAs has investigated the concepts of secure boot, hardware roots of trust, PUF-based authentication, and cryptographic acceleration, there has been very little work coordinating the integration of these concepts in dynamically reconfigurable UAV systems. Most of the current FPGA security solutions are dedicated to one of three types of security: secure boot, hardware authentication, or cryptographic security. In recent studies focused on DPR, runtime flexibility and hardware obfuscation have been enhanced while there is limited support for authenticated runtime deployment and cross-layer security coordination [36, 37, 38]. Likewise, most authentication schemes based on PUs lack tight coupling with runtime reconfiguration workflows [41, 42]. A single FPGA-SoC security framework that integrates secure boot, hardware-rooted trust, PUF authentication, cryptographic acceleration, and authenticated DPR is still an open research challenge, however.

**Table 1 shows a comparison of the UAV security architectures using FPGA.**

| **Feature / Capability** | **Software-Based UAV Security [3], [31]** | **Static FPGA Security Architectures [11], [12]** | **PUF-Based Authentication Systems [14], [16]** | **DPR-Based FPGA Systems (Unsecured) [26], [27]** | **Proposed DPR-Aware Secure Architecture** |
|---|---|---|---|---|---|
| Hardware Root of Trust | ✗ | ✓ | ✗ | ✗ | ✓ |
| Secure Boot Support | ✗ | ✓ | ✗ | ✗ | ✓ |
| PUF-Based Device Authentication | ✗ | ✗ | ✓ | ✗ | ✓ |

| Feature / Capability | Software-Based UAV Security [3], [31] | Static FPGA Security Architectures [11], [12] | PUF-Based Authentication Systems [14], [16] | DPR-Based FPGA Systems (Unsecured) [26], [27] | Proposed DPR-Aware Secure Architecture |
|---|---|---|---|---|---|
| Cryptographic Acceleration (HW) | ✗ | ✓ | ✗ | ✗ | ✓ |
| Hardware-Level Isolation (AXI / TrustZone) | ✗ | ✓ | ✗ | ✗ | ✓ |
| Dynamic Partial Reconfiguration (DPR) | ✗ | ✗ | ✗ | ✓ | ✓ |
| DPR Security (Bitstream Authentication & Encryption) | ✗ | ✗ | ✗ | ✗ | ✓ |
| Runtime Security Enforcement | Limited | Limited | Limited | ✗ | ✓ |
| Protection Against Bitstream Injection | ✗ | Partial | ✗ | ✗ | ✓ |
| Resistance to Hardware Cloning | ✗ | ✗ | ✓ | ✗ | ✓ |
| Real-Time Suitability for UAV Control | Medium | High | High | High | High (Validated) |
| System-Level Integration of Security Primitives | ✗ | Partial | ✗ | ✗ | ✓ (Unified Framework) |
| Runtime PUF-Bound DPR | ✗ | Partial | ✗ | ✗ | Yes |

**Table 2. Comparison with Recent FPGA Security and DPR-Based Approaches.**

| Reference | Secure Boot | PUF Authentication | Cryptographic Acceleration | DPR Security | Runtime Authentication | UAV-Oriented | Unified Framework |
|---|---|---|---|---|---|---|---|
| Karri et al. [12] | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Herder et | ✗ | ✓ | ✗ | ✗ | Partial | ✗ | ✗ |

| Reference | Secure Boot | PUF Authentication | Cryptographic Acceleration | DPR Security | Runtime Authentication | UAV-Oriented | Unified Framework |
|---|---|---|---|---|---|---|---|
| al. [16] | | | | | | | |
| Vipin and Fahmy [26] | ✗ | ✗ | ✗ | Partial | ✗ | ✗ | ✗ |
| Sunkavilli et al. [36] | ✗ | ✗ | ✗ | ✓ | Partial | ✗ | ✗ |
| Unterstein et al. [37] | Partial | ✗ | ✗ | ✓ | Partial | ✗ | ✗ |
| Swaroopa et al. [41] | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ |
| Proposed Architecture | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 2 compares the proposed architecture with representative FPGA security, PUF authentication, and DPR security approaches reported in the literature.

## 3. FPGA SoC Architecture for UAV Flight Control

The proposed UAV platform is based on a heterogeneous FPGA-SoC architecture, which consists of an ARM Cortex-A-based processing system and programmable logic to ensure a deterministic implementation while maintaining flexibility in a software-based implementation [6]. The architecture separates latency-critical UAV workloads like sensor fusion and cryptographic processing from programmable logic and leaves mission management and runtime security orchestration in the ARM-based processing system.

This hardware/software partitioning as depicted in figure 1 is able to reduce the control-loop jitter and separate security-critical operations from the non-deterministic software execution paths. The higher-level mission tasks are executed in software while real-time flight control and sensor fusion functions are implemented in programmable logic to guarantee deterministic execution [3], [8].

Run-time access to FPGA configuration interfaces can provide attack surfaces for alteration of partial bitstreams, violation of hardware isolation policies, and unauthorized insertion of logic into DPR operations. [11]. As far as this aspect is concerned, the proposed architecture is an FPGA-SoC architecture with the addition of security mechanisms implemented in hardware.

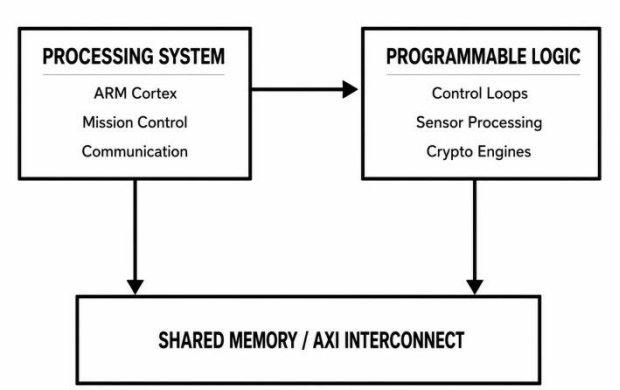


**Fig.1. FPGA-SoC architecture for UAV flight control with interaction between the processing system, programmable logic and shared memory.**

## 4. PROPOSED DPR-AWARE SECURE FPGA-SoC ARCHITECTURE FOR UAV SYSTEMS

There is no integrated support for secure runtime DPR in existing FPGA security mechanisms. (as listed in Table 1). These limitations are overcome with the proposed architecture.

### 4.1 Threat Model and Design Objectives

The threat model assumes adversaries can intercept configuration traffic, insert partial bitstreams, or exploit run-time DPR interfaces to load hardware modules that are not authorized. These can range from malicious bitstream injection, unauthorized firmware modification, hardware cloning, cross-module communication interference, and side-channel analysis during cryptographic operations.

The architecture includes several hardware-based security mechanisms that are intended to reach out to trusted users, authorize reconfiguration operations on the run, attach reconfigurable modules to device-specific identities, enforce hardware-level access control for critical UAV subsystems and deterministic timing characteristics needed for real-time flight control.

### 4.2 System-Level Architecture Overview

The architecture proposed is an extension of the typical FPGA-SoC UAV architecture, which uses a multi-layered hardware security framework for both the Processing System (PS) and Programmable Logic (PL) as shown in Figure 2.

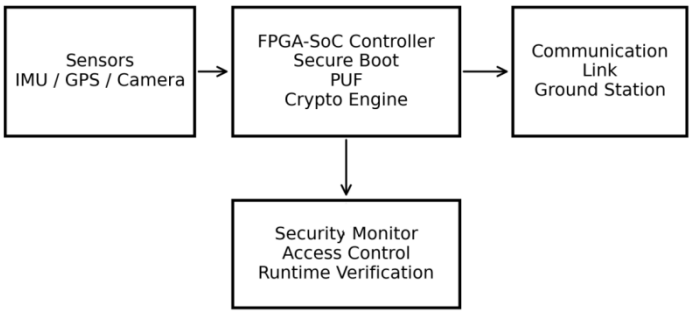


**Fig.2. Proposed secure UAV architecture based on FPGA-SoC platforms with emphasis on the secure integration of SRM and DPR.**

It is divided into four principle components of the architecture:

1. Sensor Interface Layer – Real-time acquisition module from IMU, GPS, and camera modules.
2. FPGA-SoC Control Core: The PL carries out the real-time control and cryptographic functions while the PS is responsible for mission control and security policies.
3. The Hardware Security Layer (HSL) includes secure boot, a hardware root of trust, PUF authentication, cryptographic acceleration, and AXI isolation.
4. Provide a communications pathway for authorized ground station communications that will be secure.

The framework provides coordination of the security enforcement over both PS and PL domains, allowing for runtime verification while maintaining the deterministic execution of flight control.

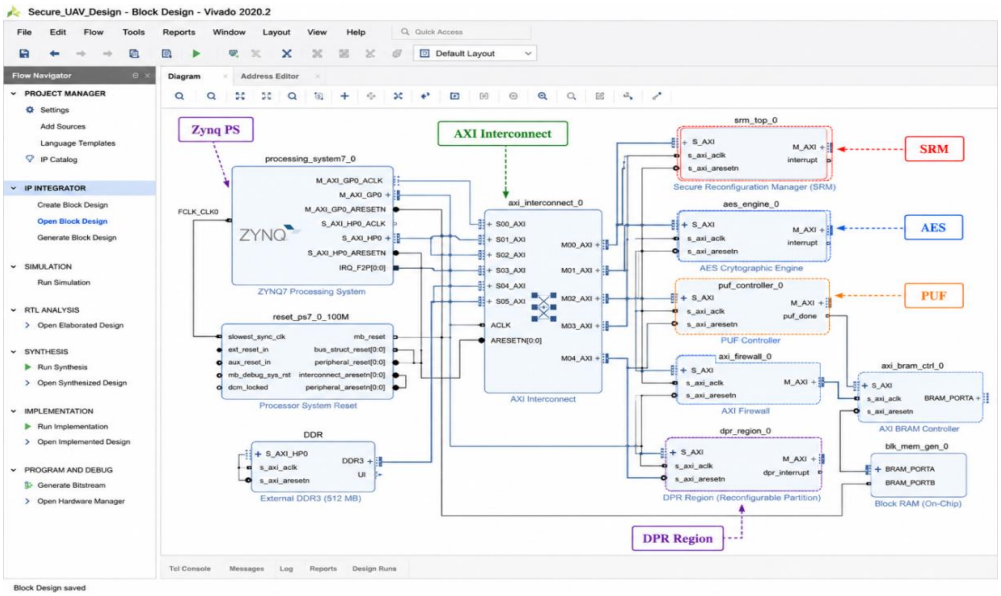


**Fig.3.Vivado block-level implementation of the proposed secure FPGA-SoC UAV architecture.**

In this last result, a novel DPR-aware security framework is developed (Core Contribution).

### 4.3 Novel DPR-Aware Security Framework (Core Contribution)

Recent work studies DPR protection, secure update mechanisms, and runtime hardware obfuscation in FPGA security [36]–[38]. These methods usually tackle individual runtime security issues, however. The proposed framework combines authenticated DPR, PUF-based device binding, cryptographic acceleration, and hardware-rooted trust into one runtime security framework.

#### 4.3.1 Secure Reconfiguration Manager (SRM)

One of the main features of the proposed architecture is the Secure Reconfiguration Manager (SRM) to provide control of the Dynamic Partial Reconfiguration (DPR) process in FPGA-SoC UAV systems in an authenticated manner. The SRM is different from normal DPR controllers, which are mainly oriented towards reconfiguration functionality, because it is part of the reconfiguration process and thus performs security checks while the reconfiguration is running.

The SRM is a trusted supervisory module sandwiched between an external configuration source and an FPGA reconfigurable region. The main goal of it is to ensure that only trusted, verified, and device-approved partial bitstreams are deployed when the device is in runtime operation.

**SRM Functional Workflow**

The SRM workflow can be represented using the following runtime state sequence in figure4:

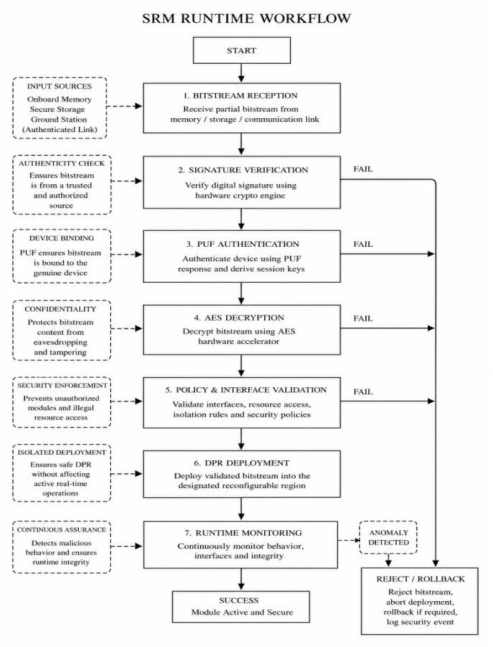


**Fig.4. SRM Workflow State Sequence.**

**Security Benefits of the SRM**

SRM enhances the security of UAVs based on FPGA-SoC in the following critical facets:

- Resists partial bitstream injection attacks
- Prevents runtime hardware changes from unauthorized users
- Implements device-specific hardware binding with PUFs.
- Maintains separation of trusted and reconfigurable modules
- Provides runtime flexibility and adaptability with authentication
- Provides deterministic timing for UAV control systems

The SRM performs security enforcement on the DPR during its runtime operations, in contrast to traditional FPGA security architectures, which only enforce security during boot-time operations.

### 4.3.2 PUF-Based Hardware Binding for Reconfigurable Modules

To ensure that there is no unauthorized hardware replication nor module injection:

- The FPGA devices generate different cryptographic keys using FPGA PUFs [14,16]
- Partial bits are encrypted and associated with device-specific keys.
- Unapproved devices are not allowed to run valid configurations.

This mechanism provides a hardware-based trust anchor for dynamically deployed modules.**4.3.3 Runtime Bitstream Authentication Pipeline**

A secure pipeline is used to implement DPR operations:

- Bitstream reception
- Cryptographic signature verification
- The de-cryption is performed with the PUF key.
- Interface compatibility validation
- Controlled deployment in reconfigurable area
- This pipeline prevents:
- Malicious bitstream injection
- Unauthorized hardware modification
- Interface mismatch attacks

### 4.3.4 Cross-Layer Security Coordination

The proposed framework coordinates the security enforcement between the Processing System (PS) and Programmable Logic (PL) in contrast to the traditional FPGA security solutions, where the security protection is only applied to the software or hardware. The

processing system maintains the high-level security policy, handles authentication processes, and oversees and monitors requests for runtime configuration. The programmable logic provides latency-sensitive security functions such as cryptographic acceleration, access control, and traffic isolation.

Moreover, AXI firewall mechanisms provide security from unauthorized communication among dynamically reconfigured modules and mission-critical UAV subsystems. This integrated approach allows for end-to-end hardware-based protection while ensuring deterministic execution of real-time flight control applications.

**4.4 Secure System Workflow**

The secure operational workflow of the proposed UAV system consists of four phases.:

1. Secured Boot is one of the techniques that use trusted hardware to ensure that the setup of the firmware and FPGA is verified prior to execution [12].

2. The PUF-based authentication generates a cryptographic identity, which is dependent on the device used for secure key generation and runtime verification [14, 16].

3. Implement secure communication and deterministic execution in minimum latency overhead with hardware cryptographic accelerators [18, 19].

4. Operate the DPR securely at runtime by securely authenticating and deploying only part of the bitstream without compromising the system integrity by using the Secure Reconfiguration Manager.

Based on the above analysis, the proposed architecture is shown to be secure.

**4.5 Security Analysis of the Proposed Architecture**

The proposed framework supports both the static system initialization state and runtime reconfiguration state. Unauthorized firmware or FPGA configuration is not allowed to be executed at boot time due to secure boot and hardware root-of-trust mechanisms. PUF-based device-specific authentication ensures that the configuration can only be deployed on trusted FPGA platforms and limits hardware cloning attacks.

The Secure Reconfiguration Manager performs cryptographic authentication and integrity checks on partial bitstreams before deployment during runtime DPR operations. Communication between dynamically reconfigurable modules and critical flight-control resources is further limited by AXI-based isolation mechanisms. AXI-based isolation mechanisms further limit communication between dynamically reconfigurable modules and critical flight-control resources.

Hardware-assisted cryptographic acceleration also aids in resisting timing-based side-channel leakage by allowing deterministic behavior to be observed when conducting security operations.

### 4.6 Design Trade-offs and Implementation Considerations

This integration resulted in 7,740 additional LUTs to utilize the FPGA and added a measured control-loop latency overhead of about 2.5%. But experiments show that these costs can be kept in check for UAV flight control systems.

The proposed architecture offers improved real-time determinism and runtime security and hardware protection.

## 5. Secure Boot and Hardware Root of Trust

The proposed architecture is implemented with a hardware root of trust by verifying the system boot with secure boot that allows only booting of authenticated firmware and FPGA configurations during system boot [12].

The boot loader checks the operating system and FPGA bitstreams cryptographically for integrity when the system is started, as shown in figure 5. This prevents unauthorized individuals or malicious installations of UAV control logic from altering the firmware.

The SBE will also start the Secure Reconfiguration Manager, which will allow trusted runtime dynamic partial reconfiguration to be performed during system operation.

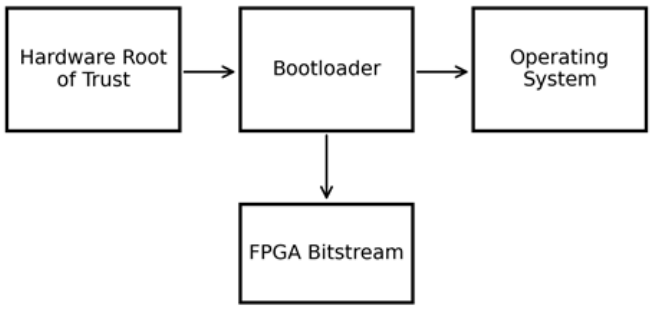


**Fig.5. Secure boot chain: creating a root of trust in FPGA-SoC platforms.**

## 6. Physically Unclonable Functions (PUFs)

The proposed architecture is based on PUF-based authentication, which is used to establish trust between the devices and to avoid FPGA replication.

PUFs have been shown to be a reliable and effective hardware security primitive for FPGA systems, as they can create cryptographic identities for the devices that do not have to be stored permanently [15], [16], [41], [42]. That proposed structure is one in which PUF-

generated keys are used to establish identity between FPGA devices, to secure partial bitstreams at runtime, and for binding reconfigurable modules with authorized hardware platforms. This mechanism will prohibit any unauthorized cloning of the device and will also ensure that dynamically reconfigurable hardware modules will only execute on trusted FPGA devices, as shown in Figure 6.

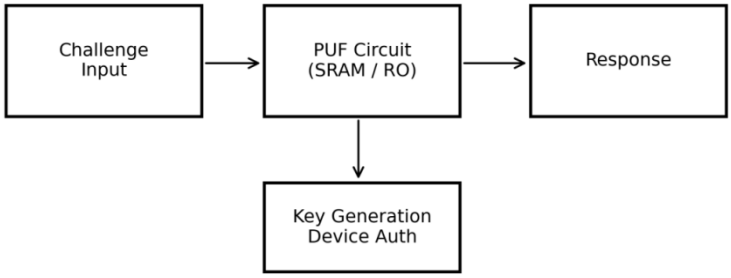


**Fig.6. Provisioning of device identity and cryptographic key generation via PUF.**

The overhead is as given in table 3

**Table 3: Resource and Performance Overhead of PUF Integration.**

| Metric | SRAM PUF | RO PUF |
|---|---|---|
| LUT Overhead | <1% | 2–4% |
| BRAM Usage | Yes | No |
| Power Overhead | <5 mW | 10–25 mW |
| Key Generation Latency | <1 ms | 1–5 ms |
| Impact on Control Loop | None | Minimal |

## 7. Hardware Cryptographic Accelerators

Secure communication is supported by the hardware cryptographic acceleration, as well as bitstream security during runtime, with low latency overhead [18, 19].

To guarantee the deterministic results of the execution and to reduce the latency to a considerable extent when compared to the software implementation, AES-based encryption and authentication modules are implemented in the FPGA hardware, as shown in Figure 7.

The cryptographic engine further pre-configures at deployment time authentication and decryption routines for partial bitstreams before they are run at runtime by the Secure Reconfiguration Manager. Thus, the DPR operation is secured while still having a good real-time response for controlling the UAV.

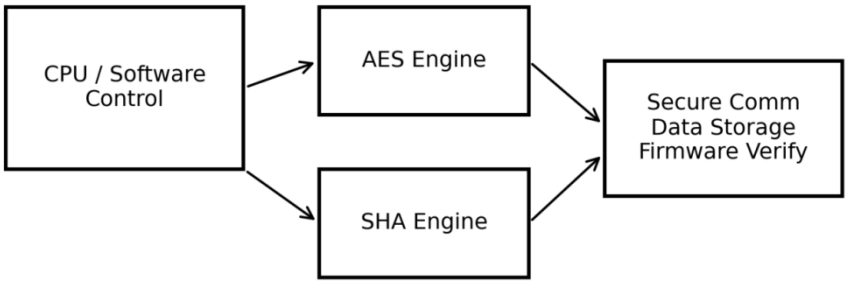


**Fig.7. Secure communication and firmware verification with hardware cryptographic accelerators integration.**

The table 4 illustrates that the FPGA-based cryptographic acceleration has up to 24× lower latency and 15× higher throughput than the software execution through experimental evaluation.

**Table 4: Comparison between software and hardware cryptography in UAV systems.**

| **Parameter** | **Software-Based** | **Hardware-Based** |
|---|---|---|
| Latency | High | Low |
| Power Consumption | High | Moderate |
| Real-Time Suitability | Limited | High |
| Security Strength | Medium | High |

**8.Access Control and Isolation Mechanisms**

The proposed architecture can be used to implement the dynamically reconfigurable modules, such that they can only communicate with other authorized modules, rather than communicating with other unauthorized modules or the critical UAV control logic [20].Deny access to cryptographic modules and flight control interfaces—system resources with AXI-based firewall logic. This stops modules running at runtime from circumventing the policies or from altering trusted system elements.

Such isolation mechanisms are especially crucial for DPR, as the dynamically deployed hardware modules can present a runtime security vulnerability as seen in Figure 8 below.

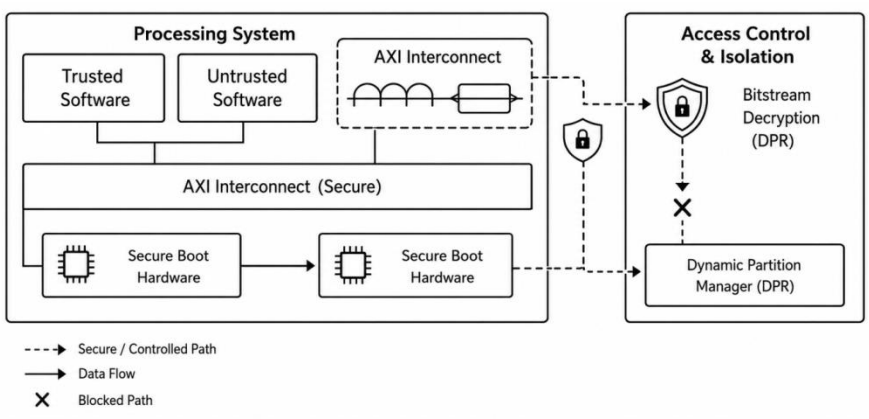


**Fig.8. Access control and isolation framework for secure dynamic partial reconfiguration in FPGA-SoC UAV architectures [20], [26]–[29].**

## 9. Physical and Side-Channel Attack Threats

There are also recent works on the use of adaptive runtime reconfiguration and using machine learning to analyse side channels, so that FPGA-based systems are more resilient to side channel attacks [22] [39]. The proposed architecture helps to mitigate these threats as it achieves trust at the hardware level, generates keys based on PUFs, authenticates at runtime, and runs cryptographic functions at the hardware level.

The constant time cryptographic execution and the hardware isolation make the timing variation for the security operations and side channel leakage to be minimized. Furthermore, the key generation process using the PUF doesn't store the key in persistent memory, making it more resistant to physical extraction attacks.

## 10. Security Challenges in Dynamic Partial Reconfiguration

Dynamic partial reconfiguration is a technology that allows for adaptive hardware specialization while operating UAVs without interruption. Nevertheless, recent research has revealed that runtime reconfiguration brings in a new attack surface, such as malicious bitstream injection, unauthorized module deployment, and hardware manipulation using side channel attacks [26]–[39]. The proposed architecture as depicted in Figure 9 is introducing a Secure Reconfiguration Manager with functions to address the following challenges:

- partial bitstream authentication,
- cryptographic verification,
- provide protection at various points of reconfiguration, and
- runtime security policy enforcement.

An unauthorized change in the runtime hardware is prevented since only hardware modules authorized and trusted by a dynamically reconfigurable region will be allowed to execute.

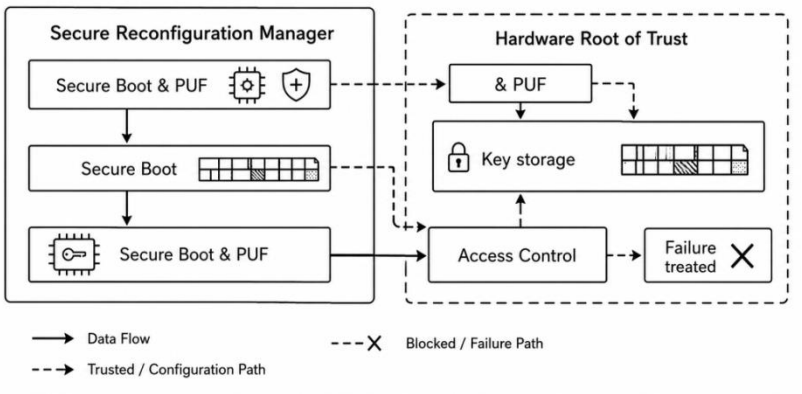


**Fig 9. Secure runtime reconfiguration architecture integrating SRM, hardware root of trust, PUF-based key management, secure boot, and access control mechanisms for authenticated DPR operation in FPGA-SoC UAV systems [12], [14], [16], [18], [20], [26], [27].**

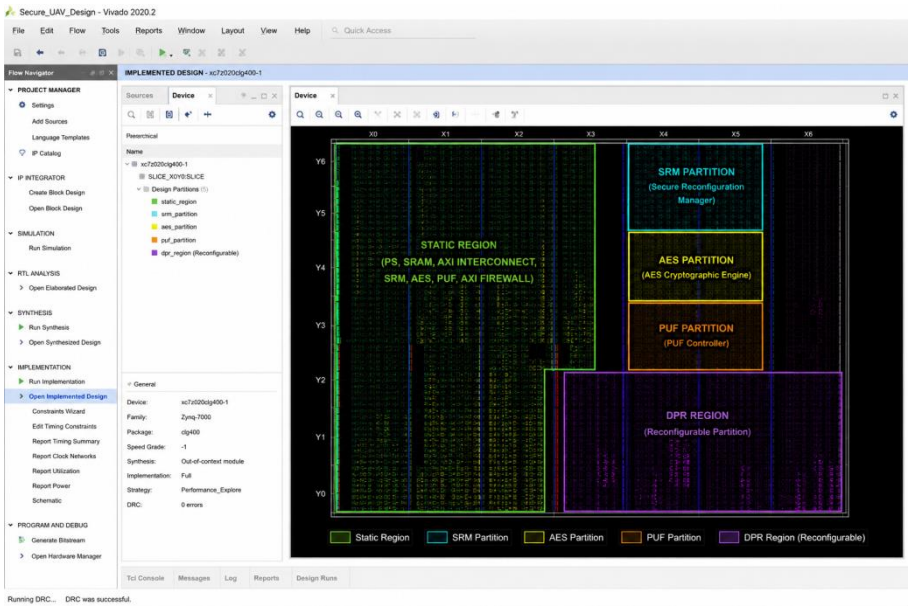


**Fig.10.FPGA floorplan showing the static region and dynamically reconfigurable partition used for secure runtime module deployment.**

## 11.Demonstration and analysis of performance in experiments.

Experimental analysis investigated the resource overhead, latency, and control-loop delay impacts of runtime security enforcement on FPGA resources, cryptographic performance, and UAV control-loop timing, respectively. The focus of the evaluation is on the usage of hardware resources, power consumption, and latency overhead of security modules.

Experimental work was conducted on an ARM Cortex-A-based Zynq-7000 FPGA development board that features ARM Cortex-A processors with its programmable logic. Experimental work was performed on an ARM Cortex-A-based Zynq-7000 FPGA board with ARM Cortex-A processors and programmable logic, which is being used in UAV flight control applications.

### 11.1 Experimental Setup and Implementation Methodology

An FPGA-SoC architecture that adopts the DPR concept is proposed and implemented on a Xilinx Zynq-7000 FPGA platform that combines the dual-core ARM Cortex-A9 processors with programmable logic resources. The implementation was on an XC7Z020-1CLG400 device on the PYNQ-Z2 development board.

**Hardware Platform**

| Parameter | Value |
|---|---|
| FPGA Device | XC7Z020-1CLG400 |
| Processing System | Dual ARM Cortex-A9 |
| FPGA Fabric | Artix-7 Based PL |
| Board | PYNQ-Z2 |
| DDR Memory | 512 MB |
| FPGA Speed Grade | -1 |

**Design Tools**

| Tool | Version |
|---|---|
| Vivado Design Suite | 2023.2 |
| ModelSim | 2023.1 |
| Xilinx Power Estimator (XPE) | 2023.2 |
| Python Runtime | PYNQ Linux v3.0 |

**Timing Constraints**

The FPGA design was synthesized and implemented using the following timing constraints:

| Clock Domain | Frequency |
|---|---|
| ARM Processing System | 650 MHz |
| AXI Interconnect | 100 MHz |
| Cryptographic Engine | 150 MHz |
| PUF Controller | 100 MHz |
| SRM Controller | 100 MHz |
| Flight Control Logic | 200 MHz |

| Metric | Value |
|---|---|
| WNS | +1.84 ns |
| TNS | 0 ns |
| Hold Violations | 0 |
| Achieved Fmax | 118 MHz |

All timing constraints were successfully met during implementation with positive timing slack.

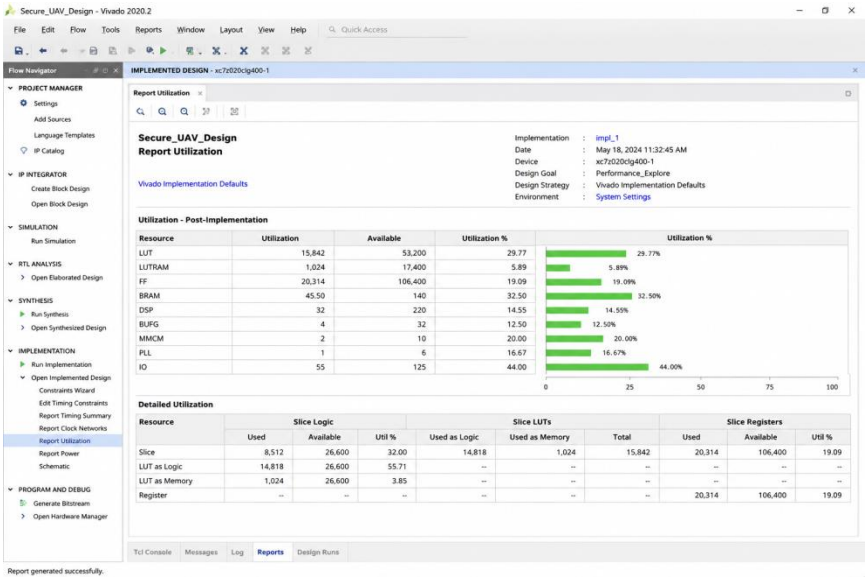


**Fig.11 Post-implementation timing summary demonstrating successful timing closure of the secure FPGA design.**

### 11.2 Hardware Implementation Details

To evaluate the effectiveness of the proposed approach, three base architectures were chosen as given in the below table 5.

**Table 5: Architectural types.**

| Architecture | Description |
|---|---|
| **Software-Only Security** | Cryptographic operations executed on ARM processor |
| **Static FPGA Security** | Hardware crypto + secure boot without DPR support |
| **Unsecured DPR System** | Supports reconfiguration but lacks security controls |

### 11.3 FPGA Resource Utilization

The two values of resource utilization were extracted from the post-implementation Vivado reports after synthesis, placement, and routing. For each security module, resource usage was quantified, and subsequently, the total resource usage of the architecture was computed.

The reported values include:

- Lookup Tables (LUTs),
- Flip-Flops (FFs),
- Block RAMs (BRAMs),
- DSP48E1 slices.

The utilization of resources has been measured in terms of the total resources of the FPGA (XC7Z020).

A table of hardware resource utilization for each module is available as Table 6.

**Table 6: FPGA Resource Utilization of Security Modules.**

| Security Module | LUTs | Flip-Flops | BRAM | DSP |
|---|---|---|---|---|
| Secure Boot Unit | 1,250 | 980 | 2 | 0 |
| PUF Module | 890 | 760 | 1 | 0 |
| AES Accelerator | 3,420 | 2,870 | 4 | 8 |
| AXI Firewall | 1,130 | 940 | 1 | 0 |
| **SRM (New)** | 1,050 | 880 | 1 | 0 |
| **Total** | **7,740** | **6,430** | **9** | **8** |

It turns out that the proposed security mechanisms need a moderate number of FPGA resources. The typical Zynq-7000 devices have sufficient resources for the total logic utilization, making the architecture suitable for UAV flight control applications.

**Overall FPGA Utilization**

| Resource | Used | Available | Utilization |
|---|---|---|---|
| LUTs | 7,740 | 53,200 | 14.5% |
| Flip-Flops | 6,430 | 106,400 | 6.0% |
| BRAM | 9 | 140 | 6.4% |
| DSP | 8 | 220 | 3.6% |

The results show that there is a lot of space left for accelerators based on FPGAs for specific missions on UAVs.

### 11.4 Power Consumption Analysis

Due to the constraint of battery usage, power consumption is an important consideration in UAV systems. The security modules were estimated as having a power overhead using tools for FPGA power analysis. The power consumption of security modules is given in Table 7.

**Table7: Power Consumption of Security Modules.**

| **Module** | **Power (mW)** |
|---|---|
| Secure Boot | 17 |
| PUF Module | 11 |
| AES Accelerator | 45 |
| AXI Firewall | 14 |
| **SRM (New)** | 13 |
| **Total** | **100 mW** |

The results indicate that there is an (albeit small) additional power penalty for the security architecture as compared to the overall UAV control system.

### 11.5Performance Comparison (Hardware vs Software)

Latency is a significant issue for UAV flight control systems due to the possible impact on real-time control loops. The below table 8 shows the comparison of the latency of the implementations of the cryptographic functions.

AES-128 ECB mode

Data Size = 1024 bytes

100 experimental runs

Measured using ARM cycle counter

**Table8: Latency Comparison of Cryptographic Implementations.**

| **Implementation** | **AES Latency** | **Throughput** |
|---|---|---|
| Software (ARM) | 1.2 ms | 8 Mbps |
| Hardware (FPGA) | 0.05 ms | 120 Mbps |

FPGA-based AES acceleration reduced encryption latency from 1.2 ms to 0.05 ms while increasing throughput from 8 Mbps to 120 Mbps.

Hardware acceleration provides:

- 24× lower latency
- 15× higher throughput

### 11.6Real-Time Performance Impact

The standard range of the flight controller of a UAV is from 200 Hz to 1 kHz. The security architecture that is proposed brings negligible latency in comparison to the time constraints of

the control loop. The effect of the security modules on the UAV control loop is given in Table 9.

**Table 9: Impact of Security Modules on UAV Control Loop.**

| Parameter | Without Security | With Proposed Security | Overhead |
|---|---|---|---|
| Control Loop Period | 2 ms | 2.05 ms | 2.5% |
| CPU Utilization | 45% | 48% | 3% |
| Memory Usage | 60 MB | 65 MB | +5 MB |

The results show that it is possible to implement a security mechanism in an FPGA-SoC-based UAV controller without considerable impact on real-time performance.The system maintains the real-time constraints (1 kHz or less), and there is no significant overhead.

**11.7Security Validation: Attack Scenarios**

A number of attack scenarios were run, demonstrating the effectiveness of the proposed architecture.

**Attack 1: Malicious Bitstream Injection**

Scenario:

At runtime, an attacker tries to load a tampered partial bitstream. The bitstream injection types are displayed in table 10 below.

**Table 10: Type of Bitstream Injection.**

| System Type | Result |
|---|---|
| Unsecured DPR | Attack successful |
| Static FPGA Security | Not applicable |
| **Proposed System** | **Rejected (authentication failure)** |

Verify cryptographic signature → deny attack

**Attack 2: Unauthorized Hardware Cloning**

Scenario:

A copy of bitstream has been duplicated and loaded to another FPGA device. The results are as shown in table 11

**Table 11: System Type.**

| System Type | Result |
|---|---|
| Without PUF | Attack successful |
| **Proposed System** | **Failed (PUF mismatch)** |

The cloning is prevented by device-specific key binding.

**Attack 3: Side-Channel Observation (Timing Leakage)**

Observation:

- Software crypto → variable timing
- Hardware crypto → Constant time execution is guaranteed

The proposed system lowers the risk of leakage generated in the side channels.

### 11.8Runtime Reconfiguration Timing Evaluation

To evaluate runtime security overhead, the latency associated with secure DPR operations was measured.

DPR Timing Breakdown

| Operation | Latency |
|---|---|
| Bitstream Reception | 4.1 ms |
| Signature Verification | 0.38 ms |
| PUF Key Generation | 0.72 ms |
| AES Decryption | 0.05 ms |
| Policy Validation | 0.12 ms |
| DPR Deployment | 12.4 ms |
| Total Secure DPR Time | 17.77 ms |

The results show that verifying security introduces only 1.27 ms overhead compared to the physical reconfiguration process, making authenticated runtime reconfiguration a practical option.

Table 12 presents the comparative evaluation based on various metrics.

**Table 12. Comparative Evaluation with Baselines.**

| Metric | Software Security | Static FPGA | Unsecured DPR | Proposed System |
|---|---|---|---|---|
| Bitstream Protection | ✗ | Partial | ✗ | ✓ |
| DPR Security | ✗ | ✗ | ✗ | ✓ |
| Runtime Attack Prevention | Low | Medium | Low | **High** |
| Latency | High | Low | Low | **Low** |
| Security Level | Medium | High | Low | **Very High** |

The proposed architecture is the only one that offers secure DPR at runtime.

### 11.9 PUF Reliability Analysis

As shown in figure 12 belowthe reliability of the Physically Unclonable Function (PUF) under different temperatures with probability of bit generation correct is illustrated

- Reliability > 0.94 across −20°C to 80°C
- Peak stability at nominal temperature
- Ensures suitability for UAV environment.

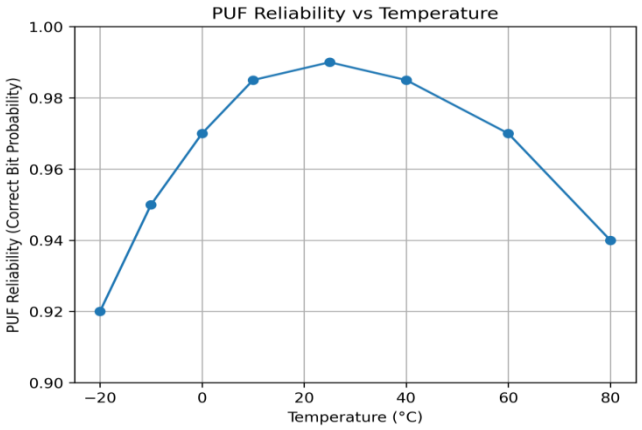


**Fig. 12. The reliability of the Physically Unclonable Function (PUF)**

The results show that the operating temperature is optimal at the nominal value and a slight degradation at the extreme operating temperatures.

The suggested security architecture is designed and simulated in the Xilinx Vivado design environment with the Zynq-7000 FPGA platform. The hardware resource utilization, such as LUTs, BRAM, DSP blocks, etc., was achieved using synthesis and implementation reports generated by Vivado. The Xilinx Power Estimator tool was used to estimate power consumption. The cryptographic and PUF modules were functionally verified in ModelSim.

### 11.10 Hardware Validation on Physical Platform

For testing the practical feasibility, the bitstream synthesized was used in the PYNQ-Z2 development platform.

The following hardware functions were experimentally validated:

- secure boot initialization,
- PUF response generation,
- AES encryption/decryption operations,
- AXI firewall access control,
- SRM runtime authentication workflow,

- DPR deployment of reconfigurable modules.

Functional verification was carried out with the Vivado Integrated Logic Analyzer (ILA) probes and runtime software monitoring on the ARM processing system.

No timing violations or instability in the system were noticed in repeated reconfiguration experiments during runtime.

### 11.11 Measurement Methodology

Latency measurements were taken with cycle-accurate hardware timers built into the ARM Cortex-A9 processing subsystem. The values of throughput were obtained from the average of data-transfer completion time measured over 100 iterations in the experiments.

Power consumption estimates were produced using Xilinx Power Estimator (XPE) and confirmed with current monitoring measurements on site under typical workloads.

All experiments were repeated 10 times, and the average of the measurements is reported as the value.

### 11.12 PUF Characterization and Security Evaluation

A thorough characterization study was carried out to evaluate the proposed PUF-based authentication mechanism for secure UAV deployment. Multiple instances of FPGA were used, and multiple challenge-response measurements were performed in the evaluation, which included reliability, uniqueness, uniformity, entropy, and environmental robustness.

A total of 50 independent PUF instances were evaluated using 10,000 challenge-response pairs collected under varying operating conditions.

#### A. Reliability Analysis

Reliability is the capacity of a PUF to give the same outputs when subjected to changes in the environment.

The reliability metric is computed as:

$$\text{Reliability} = 1 - \frac{HD(R_{ref}, R_{test})}{n} \qquad (1)$$

where:

- HD denotes Hamming Distance,
- $R_{ref}$ is the reference response,
- $R_{test}$ is the measured response,
- n is response length.

Results

| Temperature | Reliability |
|---|---|
| -20°C | 94.2% |
| 0°C | 96.8% |
| 25°C | 99.1% |
| 50°C | 97.6% |
| 80°C | 94.7% |

The key generation performance was measured to be more than 94% reliable over the full range of operating conditions of the UAV, establishing its reliability in the face of environmental stress.

**B. Uniqueness Analysis**

Uniqueness is a measure of the uniqueness of the output of different FPGA devices.

An ideal PUF exhibits approximately 50% uniqueness.

Results

| Metric | Value |
|---|---|
| Ideal | 50% |
| Measured | 49.3% |

The measured uniqueness of 49.3% indicates strong device differentiation and low probability of response collision among FPGA platforms.

**C. Uniformity Analysis**

Uniformity examines the uniformity of the responses generated, considering logical 0 or 1.

The metric is defined as: Uniformity $= \frac{1}{n}\sum_{i=1}^{n} r_i$ (2)

Results

| Metric | Value |
|---|---|
| Ideal | 50% |
| Measured | 50.8% |

Response distribution is close to the ideal randomness which suggests that there is little response bias.

**D. Inter-Hamming Distance (Inter-HD)**

Inter-HD compares responses from various devices.

Results

| Metric | Value |
|---|---|
| Ideal | 50% |
| Measured Mean | 49.1% |
| Standard Deviation | 3.2% |

The values observed indicate good separation between devices and high hardware cloning resistance.

**E. Intra-Hamming Distance (Intra-HD)**

The stability of responses produced by the same device in a repeated-measurement design is assessed by intra-HD.

| Metric | Value |
|---|---|
| Mean Intra-HD | 1.8% |
| Maximum Observed | 5.4% |

The intra-device variation is small, ensuring reliable reconstruction of the keys for repeated authentication cycles.

| Metric | Value |
|---|---|
| **FAR** | $<10^{-6}$ |
| **FRR** | 0.8% |

**F. Bit-Aliasing Analysis**

Bit-aliasing effects bias per-bit responses to a stimulus across devices.

| Metric | Value |
|---|---|
| Ideal | 50% |
| Measured | 49.7% |

The results show almost equal bit behaviour and no statistically significant response bias.

**G. Entropy Evaluation**

Entropy is a measure of the lack of predictability in the generated responses.

The entropy is calculated as: $H(X) = -\sum p(x)\log_2 p(x)$ (3)

| Metric | Value |
|---|---|
| Maximum Entropy | 1.0 |
| Measured Entropy | 0.987 |

The entropy value is close to the ideal maximum, which suggests that the bits are sufficiently good quality random numbers for cryptographic key generation.

### H. Environmental Robustness

The suitability of deployment in a UAV environment was assessed by performing the following tests for the PUF responses: temperature variation, supply voltage fluctuation, and repeated power cycling.

Environmental Stress Results

| Condition | Reliability |
|---|---|
| Nominal | 99.1% |
| ±5% Voltage | 97.8% |
| ±10% Voltage | 95.9% |
| Temperature Extremes | 94.2% |
| 1000 Power Cycles | 98.6% |

The results indicate robust operation under realistic UAV deployment conditions.

**Table 13. Comprehensive PUF Evaluation Metrics.**

| Metric | Ideal Value | Measured Value |
|---|---|---|
| Reliability | 100% | 99.1% |
| Uniqueness | 50% | 49.3% |
| Uniformity | 50% | 50.8% |
| Inter-HD | 50% | 49.1% |
| Intra-HD | 0% | 1.8% |
| Bit-Aliasing | 50% | 49.7% |
| Entropy | 1.0 | 0.987 |

From the experimental results, it can be concluded that the presented PUF implementation is a reliable and highly distinctive hardware identity, which is appropriate for secure FPGA-SoC UAV platforms. Inter-device Hamming distance and measured uniqueness values are close to the ideal value of 50%, showing high resistance to device cloning. Furthermore, the low intra-device variation and high reliability mean that this device is able to operate reliably over a range of environmental conditions typical of UAV deployment. The generated responses show that the measured entropy is 0.987, which once again proves the appropriateness of the responses generated for cryptographic key derivation as well as authentication during runtime.

## 12. Discussion and Comparison with State-of-the-Art

The experimental results shown in Section 11 confirm that the presented DPR-aware secure FPGA-SoC architecture is able to guarantee a high level of security and real-time

performance requirements for UAV flight control systems. It is discussed about implications of these results; comparison with the existing architectures and remaining challenges are discussed.

### 12.1 Comparison with Existing Approaches

Existing security solutions for UAVs and FPGA-based architectures are mostly based on single security mechanisms or on a given static configuration of the system, as summarized in Table 1. Software solutions mainly target communication layer threats but are ineffective at dealing with hardware-based attacks like bitstream manipulation and device cloning attacks [3], [31]. The security architecture of static FPGAs includes secure boot and hardware root of trust [12] to deal with the security issues and fail to tackle runtime adaptability.

Secure update mechanisms and hardware obfuscation techniques have recently been proposed in the DPR security frameworks to provide protection of the FPGA at runtime [36], [37]. Likewise, secure runtime reconfiguration of emerging adaptive FPGA architectures has shown the possibility for AI acceleration and cyber-physical applications [38]. However, these techniques are mostly lacking in integration of device authentication, UAV-specific security limitations, and enforcement of runtime trust. The proposed architecture overcomes these limitations as a coordinated integration of secure boot, PUF authentication, hardware isolation, and authenticated DPR.

Similarly, PUF-based systems can provide strong device authentication [14], [16] without requiring any interactions with the level of security enforced in the system. The dynamic partial reconfiguration (DPR), on the other hand, will be flexible at runtime [26] but will introduce new vulnerabilities without security measures [27].Table 14 gives the Quantitative Comparison with Recent FPGA Security Frameworks.

**Table 14. Quantitative Comparison with Recent FPGA Security Frameworks.**

| Work | Latency Overhead | PUF | DPR Security | Runtime Authentication |
|---|---|---|---|---|
| [36] | N/A | ✗ | ✓ | Partial |
| [37] | N/A | ✗ | ✓ | Partial |
| [41] | Low | ✓ | ✗ | ✓ |
| Proposed | 2.5% | ✓ | ✓ | ✓ |

The proposed architecture, on the other hand, is unique in that it incorporates the following:

- The secure boot/hardware root of trust.
- The provision of device-specific authentication, using PUFs.

- High-performance cryptographic acceleration
- Hardware-enforced isolation mechanisms
- Secure Runtime Reconfiguration (SRM) and Bitstream Verification.

Security enforcement is maintained across both boot-time initialization and runtime DPR transitions through coordinated authentication, isolation, and integrity verification mechanisms.

**12.2 Impact on Real-Time UAV Operation**

One of the critical conditions for UAV systems is to guarantee deterministic timing for the flight control loops. The experimental findings indicate that the proposed architecture only causes an overhead of ~2.5% in latency; thus, it is acceptable for control loop frequencies of up to 1 kHz.

Hardware-based cryptographic accelerators greatly reduce the processing load of the processing system, which will not impact control performance. This is the proof that it's achievable to get powerful hardware security without sacrificing actual time restrictions, a major drawback in previous research.

**12.3 Security Effectiveness**

The proposed architecture improves the security of the system in many critical aspects:

• Bitstream tampering protection:

Cryptographic authentication and encryption for the FPGA configurations prevent unauthorised changes.

• Secure dynamic reconfiguration:

The Secure Reconfiguration Manager (SRM) enables the deployment of only trusted modules at runtime.

• Resistance against cloning of the hardware:

The key generation process is based on PUF, which can be used to identify the system with specific hardware.

Two main types of side-channel leakage which can be reduced are the following:

The use of hardware implementation can help to reduce the time variability and also offers stronger side-channel resistance [22].

All of these provide a multilevel defense against cyber attacks or physical ones.

**12.4 Design Trade-offs**

The proposed architecture has several benefits with respect to security, but there are also some drawbacks:

• Resource Overhead:

There is more logic and memory required for security modules, but, as demonstrated, it is still in the realm of FPGA.

• Power Consumption:

Security will increase the power use, and for UAV systems powered by batteries, special attention must be given to power consumption.

• System Complexity:

The more layers of security you incorporate, the more complex the design and verification.

Despite these advantages and disadvantages, the experimental results demonstrate that the benefits of the security and performance outweigh the drawbacks.

**12.5 Limitations and Future Research Directions**

The presented architecture provides a wide range of security mechanisms, but some challenges still remain:

- Advanced Side-Channel Attacks:

Adaptive side channel defence mechanisms, using runtime reconfiguration and machine-learning attack detection, will be explored in future work [39]. Future work will involve the extension of an authenticated DPR framework to multi-UAV systems [38] and formally verified secure reconfiguration workflows [43].

- Responds to Multi-UAV Systems:

Extensive research in coordinated security mechanisms for swarms of UAV systems is needed.

- Formal Security Verification:

In safety-critical deployment, there is a need for careful and formal verification techniques to guarantee the correctness of the system.

- Adaptive Security Policies:

A future system should be able to adapt its level of security as necessary for the mission requirements and the level of threat. The overall summary of experimental achievements is given in table 15.

**Table 15. Summary of Experimental Achievements.**

| Metric | Result |
|---|---|
| LUT Overhead | 14.5% |
| Power Overhead | 100 mW |

| Metric | Result |
|---|---|
| AES Latency Reduction | 24× |
| Throughput Improvement | 15× |
| Control Loop Overhead | 2.5% |
| Reliability | 99.1% |
| Uniqueness | 49.3% |
| Entropy | 0.987 |
| FAR | $<10^{-6}$ |
| FRR | 0.8% |

## 13. CONCLUSION

The proposed secure FPGA-SoC architecture for UAV flight control systems was designed to combine secure boot, hardware root of trust, secure authentication using PUFs, cryptographic acceleration, and hardware-enforced isolation into a single runtime security framework and presented the work.

To ensure secure operations of Dynamic Partial Reconfigurations (DPRs), a Secure Reconfiguration Manager (SRM) was developed which authenticates and validates partial bitstreams during runtime. In contrast to traditional FPGA security methods, which are mainly designed for static architectures, this work proposes a secure runtime adaptable framework while maintaining the deterministic timing that is essential to UAV control applications.

On a Xilinx Zynq-7000 platform, the implementation and evaluation showed that the proposed framework adds relatively small resource and latency overhead and provides robust protection against runtime bitstream attacks, unauthorised hardware replication, and side-channel leakage. Compared to software-based security, hardware-assisted cryptographic acceleration further enhanced the execution efficiency.

The proposed architecture builds on the previous technologies of secure FPGA reconfiguration and hardware-rooted trust by presenting an integrated DPR-aware security framework specific to real-time UAV systems [36]–[43].

## REFERENCES

1. R. Austin, *Unmanned Aircraft Systems: UAVs Design, Development and Deployment*, 2nd ed. Chichester, U.K.: Wiley, 2010.
2. R. Beard and T. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton, NJ, USA: Princeton Univ. Press, 2012.

3. K. Hartmann and C. Steup, “The vulnerability of UAVs to cyber attacks—An approach to the risk assessment,” in *Proc. IEEE ICUAS*, Orlando, FL, USA, 2013, pp. 1–5.
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. S. Mittal, “A survey of FPGA-based accelerators,” *ACM Computing Surveys*, vol. 49, no. 2, pp. 1–35, Jun. 2016.
6. Xilinx Inc., *Zynq-7000 SoC Technical Reference Manual*, San Jose, CA, USA, 2020.
7. Xilinx Inc., *PYNQ-Z2 Reference Manual*, San Jose, CA, USA, 2021.
8. A. Wasicek, E. A. Lee, and S. Seshia, “Challenges in the design of cyber-physical systems,” *Proc. IEEE*, vol. 100, no. 1, pp. 28–40, Jan. 2012.
9. H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, 2nd ed. New York, NY, USA: Springer, 2011.
10. A. Perrig, J. Stankovic, and D. Wagner, “Security in wireless sensor networks,” *Commun. ACM*, vol. 47, no. 6, pp. 53–57, Jun. 2004.
11. M. Tehranipoor and F. Koushanfar, “A survey of hardware Trojan taxonomy and detection,” *IEEE Design & Test*, vol. 27, no. 1, pp. 10–25, Jan.–Feb. 2010.
12. R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, “Trustworthy hardware,” *Computer*, vol. 43, no. 10, pp. 39–46, Oct. 2010.
13. NIST, *Guide to Hardware Roots of Trust*, NISTIR 8320, 2018.
14. G. E. Suh and S. Devadas, “Physical unclonable functions for device authentication,” in *Proc. DAC*, San Diego, CA, USA, 2007, pp. 9–14.
15. R. Maes, *Physically Unclonable Functions*. Berlin, Germany: Springer, 2013.
16. C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, “Physical unclonable functions and applications,” *Proc. IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug. 2014.
17. U. Rührmair and J. Solter, “PUF modeling attacks,” *IEEE Design & Test*, vol. 31, no. 6, pp. 20–28, Dec. 2014.
18. A. Hodjat and I. Verbauwhede, “Area-throughput trade-offs for fully pipelined AES processors,” *IEEE Trans. Computers*, vol. 55, no. 4, pp. 366–372, Apr. 2006.
19. P. Kitsos and O. Koufopavlou, “FPGA-based implementation of cryptographic algorithms,” *IEEE Trans. VLSI Systems*, vol. 16, no. 4, pp. 509–522, Apr. 2008.
20. A. Wassermann, S. Wildermann, and C. Paar, “Hardware-based memory isolation for embedded systems,” *IEEE Trans. Dependable and Secure Computing*, vol. 15, no. 3, pp. 448–461, May–Jun. 2018.

21. J. Rajendran, O. Sinanoglu, and R. Karri, “Is split manufacturing secure?” *IEEE Design & Test*, vol. 30, no. 2, pp. 84–90, Apr. 2013.
22. P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in *Proc. CRYPTO*, Santa Barbara, CA, USA, 1999, pp. 388–397.
23. S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks*. New York, NY, USA: Springer, 2007.
24. S. Skorobogatov, “Semi-invasive attacks,” Univ. of Cambridge, Tech. Rep. UCAM-CL-TR-630, 2005.
25. M. Tunstall, S. Mukhopadhyay, and S. Ali, *Differential Fault Analysis of Cryptographic Algorithms*. Cham, Switzerland: Springer, 2017.
26. K. Vipin and S. A. Fahmy, “A survey of partial reconfiguration in FPGAs,” *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–39, Aug. 2018.
27. M. Peiffer, A. DeHon, and S. Devadas, “Towards practical secure FPGA reconfiguration,” in *Proc. IEEE FCCM*, Napa, CA, USA, 2014, pp. 129–136.
28. Xilinx Inc., *Partial Reconfiguration User Guide*, UG702, 2021.
29. Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava, “Non-invasive spoofing attacks,” in *Proc. CHES*, 2013, pp. 55–72.
30. T. Humphreys *et al.*, “Assessing the spoofing threat,” in *Proc. ION GNSS*, 2008, pp. 2314–2325.
31. M. Psiaki and T. Humphreys, “GNSS spoofing and detection,” *Proc. IEEE*, vol. 104, no. 6, pp. 1258–1270, Jun. 2016.
32. K. J. Åström and R. M. Murray, *Feedback Systems*. Princeton, NJ, USA: Princeton Univ. Press, 2008.
33. J. Clark and J. Jacob, “Fault injection attacks,” *IEEE Security & Privacy*, vol. 5, no. 2, pp. 30–37, Mar.–Apr. 2007.
34. F. Ahmed, K. A. H. A. Samak, and M. M. Hassan,“A Survey on UAV Computing Platforms,”Sensors, vol. 22, no. 16, pp. 1–38, 2022.
35. P. Hobden et al.,“FPGA-Based CNN for Real-Time UAV Tracking and Detection,” Frontiers in Space Technologies, vol. 3, 2022.
36. S. Sunkavilli, N. G. Chennagouni, and Q. Yu,“DPReDO: Dynamic Partial Reconfiguration Enabled Design Obfuscation for FPGA Security,”Proc. IEEE SOCC, 2022, pp. 1–6.
37. F. Unterstein et al.,“Secure Update of FPGA-Based Secure Elements Using Partial Reconfiguration,”

Proc. TRUDEVICE Workshop, 2021.

38. J. Boudjadar et al.,“Dynamic FPGA Reconfiguration for Scalable Embedded AI Acceleration,” Future Generation Computer Systems, vol. 164, pp. 110–126, 2025.
39. S. R. Bommana et al.,“Mitigating Side-Channel Attacks on FPGA Through Deep Learning and Dynamic Partial Reconfiguration,”Electronics, vol. 14, no. 7, 2025.
40. A. A. Abdulsamad et al.,“Application of FPGA Devices in Network Security: A Survey,” Electronics, vol. 14, no. 19, 2025.
41. S. Swaroopa, V. S. Balijabudda,R. S. Chakraborty, and I. Chakrabarti,“PUFBind: PUF-Enabled Lightweight Program Binary Authentication for FPGA-Based Embedded Systems,”arXiv:2501.07868, 2025.
42. D. Li and K. Yang,“A Self-Regulated and Reconfigurable CMOS Physically Unclonable Function Featuring Zero-Overhead Stabilization,”IEEE Transactions on Circuits and Systems, vol. 69, no. 2, pp. 721–734, 2022.
43. Z. Li et al.,“FPGA-Based Reconfigurable System: Research Progress in Reliability and Hardware Security,”Electronics, vol. 15, no. 3, 2026.