INTERNATIONAL JOURNAL OF NEXT-GENERATION ENGINEERING AND TECHNOLOGY (IJNGET)

eISSN: 3087-4319

Volume. 02, Issue. 09, pp. 12-21, September2025

Securing Multi-Tenant FPGA Accelerators for Cloud Cryptography: Architectures, Threat Models, and Practical Countermeasures

Dr. Adrian K. Morales
Department of Computer Engineering, Global Institute of Technology

Article received: 23/09/2025, Article Accepted: 11/10/2025, Article Published: 06/11/2025

ABSTRACT

This article presents a comprehensive, original research treatment of securing multi-tenant Field Programmable Gate Array (FPGA) accelerators used in cloud environments for cryptographic workloads. Motivated by the growing deployment of reconfigurable logic in cloud settings and the particular suitability of FPGAs for computationally intensive cryptography—such as homomorphic encryption accelerators—this work synthesizes architectural techniques, threat models, defensive design patterns, and operational controls into an integrated framework for secure FPGA multi-tenancy. We begin by situating the study within recent advances in in-fabric memory architectures and network-attached reconfigurable computing that enable efficient sharing and remote access (Chung et al., 2011; Conger et al., 2005). We then examine the distinct security challenges introduced by multi-tenant FPGA use, drawing on studies that analyze side channels, configuration integrity, and isolation failures (Dessouky et al., 2021; Diktopoulos et al., 2022). Building on prior work in hardware countermeasures and secure update protocols (Danger et al., 2009; Drimer & Kuhn, 2009), we propose a layered methodological approach combining architectural isolation (hardware and hypervisor level), cryptographic isolation (key and state management), active runtime monitoring, and formal configuration verification. The methodology emphasizes threat-driven design decisions, guided by practical constraints typical in cloud deployments, such as I/O virtualization standards and remote management interfaces (Intel, 2023). Results are presented descriptively from a rigorous thought experiment and design evaluation that models attacks (fault injection, side-channel leakage, covert channels, and malicious partial reconfiguration) against realistic FPGA cloud stacks, and shows how each proposed countermeasure mitigates specific attack vectors, often in complementary and overlapping ways (Dessouky et al., 2021; Diktopoulos et al., 2022; Drimer et al., 2008). The discussion provides deep analysis of trade-offs—including performance, area, power, manageability, and trust boundaries—alongside limitations of current techniques and a roadmap for future research such as hardware-assisted fuzzing for configuration interfaces and secure in-fabric memory provisioning (Ding et al., 2021; Chung et al., 2011). We conclude with practical recommendations for cloud providers, FPGA vendors, and researchers to prioritize a zero-trust posture, strengthen configuration authenticity, and invest in active fence and dual-rail logic strategies to raise the cost of attacks to impractical levels (Hariharan, 2025; Danger et al., 2009; Diktopoulos et al., 2022). This comprehensive treatment serves as a resource for secure deployment of cryptographic accelerators in multi-tenant cloud environments and highlights urgent research directions to close persistent gaps.

Keywords: FPGA security, multi-tenancy, homomorphic encryption, configuration integrity, side-channel mitigation, in-fabric memory, zero trust

INTRODUCTION

Field-Programmable Gate Arrays (FPGAs) have emerged as a foundational technology enabling cloud providers and enterprise systems to accelerate computationally intensive tasks. Unlike fixed-function ASICs, FPGAs offer reconfigurability that supports a broad spectrum of applications, notably cryptographic workloads that benefit from parallelism and bit-level optimization. The adoption of FPGA accelerators in multi-tenant cloud environments is driven by the need for flexible, high-throughput hardware support for services such as homomorphic encryption, machine learning inference, and network processing (Chung et al., 2011; Cilardo & Argenziano, 2016). However, this flexibility introduces novel security challenges: the same reconfigurability that enables rapid deployment of new accelerators also opens avenues for cross-tenant interference, side-channel leakage, and configuration tampering if proper isolation and management controls are not rigorously enforced (Dessouky et al., 2021).

This research article analyzes the intersection of FPGA architectural capabilities and the threat landscape of multi-tenant cloud deployments, with a particular focus on cryptographic acceleration. It addresses several fundamental questions that practitioners and researchers face: How should cloud providers design FPGA platforms to support concurrently running, untrusted tenants without compromising confidentiality, integrity, and availability? What are the practical countermeasures that raise attack costs while keeping acceleration benefits? How do hardware features—such as in-fabric memory and I/O virtualization—affect both attack surfaces and defensive options? To answer these, we synthesize existing peer-reviewed findings with novel, rigorous theoretical analysis to produce a publication-ready blueprint for secure FPGA multi-tenancy.

The need for this synthesis is acute. Reconfigurable computing is being pushed into shared infrastructure, and studies show that attackers can exploit side channels, configuration update mechanisms, and interconnects to exfiltrate data or sabotage computation (Dessouky et al., 2021; Diktopoulos et al., 2022). For cryptographic workloads, the stakes are particularly high: confidentiality breaches yield the direct compromise of secrets, and weaker integrity models undermine trust in computation that might be used to process sensitive data or enforce confidentiality properties (Cilardo & Argenziano, 2016; Costache & Smart, 2015). Moreover, cloud customers frequently assume isolation provided by virtualization and tenancy management is sufficient; however, FPGA architectures do not always cleanly map to virtual machine abstraction models, and device-level resources can be shared across tenants in subtle ways (Conger et al., 2005; Intel, 2023).

This article contributes three principal elements. First, a thorough characterization of attack surfaces and adversary capabilities in multi-tenant FPGA clouds, built on prior taxonomies and extended to cover emerging configuration and in-fabric resource sharing practices (Dessouky et al., 2021; Chung et al., 2011). Second, a layered methodology combining architectural partitioning, cryptographic controls, runtime checks, and operational practices intended to be implementable within existing cloud stacks and forthcoming FPGA feature sets (Intel, 2023; Drimer & Kuhn, 2009). Third, a reasoned evaluation showing how the approach mitigates canonical attacks and identifying residual risks that require vendor or standards action, such as secure partial reconfiguration chains and active fence refinements (Diktopoulos et al., 2022; Danger et al., 2009).

The remainder of this article elaborates each of these elements in depth. The literature grounding and background elaboration highlight pivotal prior work on in-fabric memory, network-attached reconfigurable computing, hardware countermeasures, and secure update protocols (Chung et al., 2011; Conger et al., 2005; Danger et al., 2009; Drimer & Kuhn, 2009). The methodology section specifies threat models, design principles, and concrete mechanisms. Results present a descriptive evaluation that links defenses to attack mitigations. The discussion interrogates tradeoffs and limitations and outlines future research directions, including the need for harmonized standards for FPGA multi-tenancy and hardware-assisted fuzzing of management interfaces (Ding et al., 2021; Intel, 2023). The conclusion synthesizes actionable recommendations for cloud providers and hardware vendors to adopt a zero-trust approach to FPGA tenancy (Hariharan, 2025).

METHODOLOGY

This section presents the methods used to derive secure design principles and evaluate their effectiveness against established attack vectors. Given the nature of hardware and cloud research, our methodology uses a structured, reasoning-driven approach that synthesizes empirical findings from the literature with design analysis, threat modeling, and thought experiments that emulate realistic cloud operational conditions. This textual methodology is intentionally detailed because its purpose is to provide a replicable blueprint that practitioners can apply and researchers can operationalize into experiments.

Threat Modeling and Adversary Capabilities

A starting point for any security design is a precise threat model. Building on comprehensive surveys of FPGA multi-tenant risks, we consider adversaries with varying capabilities. The adversary taxonomy includes:

1. **Co-resident tenant adversary:** a tenant who shares an FPGA device and can deploy custom logic via partial or full reconfiguration, with the objective of extracting

secrets or disrupting execution (Dessouky et al., 2021; Diktopoulos et al., 2022).

2. **Remote update adversary:** an attacker who compromises update channels or supply chains to deliver malicious or tampered configurations, exploiting weaknesses in remote update protocols (Drimer & Kuhn, 2009).

3. **Side-channel adversary:** an attacker employing electromagnetic, power analysis, timing, or other side channels to recover private data processed by co-located accelerators, exploiting leakage pathways across shared resources (Danger et al., 2009; Diktopoulos et al., 2022).

4. **Covert channel adversary:** an attacker that establishes a low bandwidth communication channel across shared resources (memory, routing, I/O blocks) to exfiltrate data from an isolated tenant (Dessouky et al., 2021).

5. **Fault injection adversary:** one who induces transient faults or manipulates environmental parameters to alter state and bypass cryptographic protections (Danger et al., 2009).

6. **Management plane adversary:** an attacker with access to hypervisor, orchestration, or remote management infrastructure (e.g., cloud control plane) who attempts to reconfigure FPGAs or alter resource assignment (Intel, 2023).

For each class, we enumerate concrete capabilities and constraints such as the ability to perform partial reconfiguration, time resolution for side-channel measurements, and access to cloud vendor APIs. This granularity enables targeted mapping of defenses to threat vectors.

Design Principles

From the literature and operational considerations, we distill the following design principles:

● Principle of strong hardware partitioning: Provide isolated physical partitioning within the FPGA fabric or use in-fabric memory allocation strategies that reduce leakage across tenant boundaries (Chung et al., 2011).

● Principle of authenticated configuration and update: Ensure every bitstream, partial reconfiguration frame, and firmware artifact is cryptographically authenticated with provenance and rollback protections (Drimer & Kuhn, 2009).

● Principle of minimized shared micro-resources: Where possible, avoid shared clocks, power rails, and I/O primitives across tenants; when sharing is necessary, employ active fences or sanitization to reduce leakage (Diktopoulos et al., 2022; Danger et al., 2009).

● Principle of runtime attestation and monitoring: Continuously validate runtime behavior and configuration integrity through hardware-anchored measurement facilities and telemetry to detect anomalies (Dessouky et al., 2021).

● Principle of defense in depth: Layer defenses so that multiple mitigations address the same class of attacks, raising the resource cost of successful exploitation (Danger et al., 2009; Ding et al., 2021).

Architectural Mechanisms

Given these principles, we define practical architectural mechanisms that can be implemented in cloud FPGA stacks:

1. Secure Boot and Authenticated Bitstreams: At device power-up and during partial reconfiguration, require cryptographic verification of bitstreams using signatures tied to a device root of trust. This prevents injection of unauthorized configurations (Drimer & Kuhn, 2009).

2. In-Fabric Memory Partitioning: Extend the notion of CoRAM and in-fabric memory abstractions to include tenant-aware allocation and zeroing semantics, such that per-tenant buffers and caches are isolated and scrubbed on tenant transitions (Chung et al., 2011).

3. Active Fence and Routing Constraints: Use active fence circuits and routing policies that sever or gate unintended signal couplings between tenant designs. Active fences can be software-configured to dynamically adjust isolation levels (Diktopoulos et al., 2022).

4. Dual-Rail and Precharge Logic for Sensitive Blocks: For tenant logic executing cryptography, employ dual-rail with precharge logic or other balancing techniques to reduce differential power and electromagnetic leakages (Danger et al., 2009).

5. Hypervisor-Aware Resource Allocation: Use I/O virtualization standards and timesharing policies that map virtualized endpoints to physical resources

deterministically, minimizing multiplexing-induced leakage (Intel, 2023).

6. Runtime Hardware Monitors and Anomaly Detection: Leverage on-board sensors and designed hardware monitors to collect telemetry—power variance, timing jitter, error rates—and feed these into anomaly detection pipelines that flag suspicious co-residency behavior (Ding et al., 2021; Dessouky et al., 2021).

7. Secure Partial Reconfiguration Chains: Create chained verification for partial reconfiguration where a trusted monitor validates compatibility and resource claims of incoming partial bitstreams before allowing reconfiguration (Drimer & Kuhn, 2009).

Operational Controls

Mechanisms at the hardware level must be complemented by operational controls that cloud providers can enact:

- Tenant Scheduling Policies: Schedule high-risk cryptographic workloads on physically isolated devices or provide dedicated FPGA instances for sensitive tenants.

- Bitstream Provenance Policies: Maintain a registry and certificate infrastructure for approved accelerators, enabling revocation and audit trails.

- Continuous Fuzzing of Management Interfaces: Adopt hardware-assisted fuzzing techniques to test configuration APIs, management interfaces, and firmware to uncover logic bugs and misconfigurations (Ding et al., 2021).

- Oblivious Data Handling Practices: Encourage application designs that reduce sensitive state residency on shared resources and adopt cryptographic schemes that minimize side-channel exposure, for example selecting schemes with favorable leakage characteristics for encrypted computation (Costache & Smart, 2015; Cilardo & Argenziano, 2016).

Evaluation Methodology

To assess the efficacy of the proposed layered defenses, we perform a structured design evaluation combining mapping attacks to defenses and quantifying mitigation in descriptive terms grounded in empirical findings from the literature. We frame the evaluation around canonical attack scenarios:

1. Side-channel recovery of cryptographic keys from co-resident tenant on shared FPGA.

2. Covert channel exfiltration via shared in-fabric memory or routing.

3. Malicious partial reconfiguration to inject a spy core that reads sensitive internal bus traffic.

4 Compromise of update channels to install unauthorized bitstreams.

For each scenario, we trace the attack steps, identify required capabilities, and map countermeasures that raise the attack cost or block it outright. Where available, we reference empirical mitigation effectiveness from the literature (e.g., active fence assessments and dual-rail logic analysis) to support claims about relative mitigation strength (Diktopoulos et al., 2022; Danger et al., 2009). When empirical data are not directly available for a measure, we use conservative engineering estimates and reasoning based on hardware properties and design constraints.

Ethical and Practical Considerations

We intentionally rely on descriptive and theoretical evaluation rather than performing potentially risky practical attacks. Our approach adheres to responsible disclosure and safety practices: we discuss attack techniques only to the extent necessary for defenders to understand and mitigate them, and we recommend vendor and standards actions to further reduce risk exposure. This methodology ensures the research is actionable for defenders without providing a free blueprint for attackers.

RESULTS

The results of our analysis synthesize how the layered methodology mitigates the defined attack scenarios. Results are presented as descriptive evaluations; each subsection describes an attack scenario, the steps an attacker would take, the defenses engaged, and the residual risk after mitigation. Claims are supported by citations to the literature or logical derivation from hardware characteristics.

Side-Channel Key Recovery from Co-resident Tenant

Attack description: An adversary deploys a tenant bitstream on a shared FPGA instance co-residing with a victim cryptographic accelerator. The adversary seeks to exploit power, timing, or electromagnetic leakage to recover cryptographic keys processed by the victim logic. The attacker may use localized sensors, carefully scheduled operations, and high-resolution timers to correlate observations with victim activity (Diktopoulos et al., 2022; Danger et al., 2009).

Defense mapping: Dual-rail and precharge logic for cryptographic modules make distinguishing signal transitions from balanced currents much harder, reducing leakage amplitude. Active fencing and routing constraints reduce the ability of a co-resident tenant to place sensors or to route signals close to victim nets, decreasing signal coupling. In-fabric memory partitioning ensures that shared buffers do not transiently store victim state accessible to the adversary (Danger et al., 2009; Diktopoulos et al., 2022; Chung et al., 2011).

Effectiveness and residual risk: Dual-rail logic and precharge provide substantial reductions in side-channel signal-to-noise but typically come with area and performance penalties; they are most cost-effective when applied selectively to high-sensitivity modules rather than across all fabric (Danger et al., 2009). Active fences, when correctly sized and placed, can effectively segregate signal domains; empirical assessments indicate substantial mitigation of on-chip SCA leakage in multi-tenant contexts, while still requiring careful design to avoid creating new timing-based leakages (Diktopoulos et al., 2022). In sum, layered application of balancing logic, routing isolation, and runtime monitoring significantly increases the attacker's required measurement resolution and adversary sophistication, often rendering the attack economically infeasible for opportunistic cloud attackers. However, a persistent residual risk remains against highly motivated attackers with direct physical access or those who can subvert the supply chain (Drimer & Kuhn, 2009).

Covert Channel via Shared In-Fabric Memory or Routing

Attack description: A co-resident tenant creates a covert channel by modulating shared resources—such as in-fabric memory blocks, shared routing resources, or clock domains—to transmit bits to another colluding tenant or to an external endpoint. Covert channels may exploit timing, contention, or resource reclaim behavior (Dessouky et al., 2021).

Defense mapping: In-fabric memory partitioning with strict allocation semantics and mandatory scrubbing upon context switches disrupts traditional covert channels by removing state persistence and contention patterns. Hypervisor-aware resource allocation and deterministic scheduling reduce the opportunity for contention-based modulation. Active monitoring of resource usage patterns and performance counters enables anomaly detection when tenants disproportionately stimulate shared resources (Chung et al., 2011; Intel, 2023; Ding et al., 2021).

Effectiveness and residual risk: Partitioning and scrubbing remove many naive covert channel vectors by eliminating shared state. Deterministic scheduling can remove temporal multiplexing opportunities but may sacrifice utilization; this trade-off must be tuned based on tenant performance requirements. Active monitoring can detect anomalous contention but may generate false positives that require operational tuning. The residual covert channel risk primarily involves microarchitectural subtleties (such as subtle routing congestion patterns) that are difficult to fully eliminate without restrictive partitioning, underscoring the need for layered defenses and conservative scheduling for security-sensitive tenants (Dessouky et al., 2021).

Malicious Partial Reconfiguration to Install Spy Cores

Attack description: Using partial reconfiguration abilities, an attacker installs a stealthy core that taps into internal buses or memory to leak sensitive data. The attacker may attempt to obfuscate the core's footprint to evade monitoring (Drimer & Kuhn, 2009).

Defense mapping: Authenticated bitstreams and chained verification for partial reconfiguration prevent unauthorized bitstreams from being accepted by the device. A trusted monitor or configuration manager validates resource claims and enforces placement constraints before allowing a partial reconfiguration operation to proceed. Runtime attestation of the post-reconfiguration fabric state against a signed baseline can detect unauthorized logic (Drimer & Kuhn, 2009).

Effectiveness and residual risk: Authenticating bitstreams is a near-complete prevention for injected malicious cores when the device root of trust and key management are uncompromised. The practicality of this defense depends on robust key provisioning and secure update infrastructure; if management plane credentials or the update signing chain is compromised, attackers can bypass this defense (Drimer & Kuhn, 2009; Intel, 2023). Thus, operational controls such as strict key lifecycle management and supply chain integrity are critical complements.

Compromise of Update Channels and Supply Chain

Attack description: Compromise of the update signing process, repositories, or deployment pipelines enables attackers to distribute tampered bitstreams or firmware at scale (Drimer & Kuhn, 2009).

Defense mapping: Hierarchical provenance verification, multi-party signing for critical artifacts, and audit trails increase the difficulty of widespread compromise. Hardware-anchored attestation requires that only artifacts signed with appropriate governance keys are accepted. Continuous monitoring of repository interactions and integrity checks during deployment reduce the window of exposure (Drimer & Kuhn, 2009; Intel, 2023).

Effectiveness and residual risk: These operational measures greatly raise the cost and coordination required for global compromise. However, sophisticated supply-chain attacks that compromise vendor signing infrastructure or embed vulnerabilities in design tools are challenging to mitigate at the cloud provider level and require industry collaboration and policy measures. The residual risk is best addressed via a combination of vendor-led controls, diversified signing authorities, and continuous repository hygiene.

Holistic Risk Posture

Mapping all scenarios to the layered defenses demonstrates that the approach provides overlapping protections: authenticated bitstreams block some attack vectors decisively; architectural partitioning and fencing limit leakage and exfiltration; runtime monitoring and anomaly detection provide detection and response capabilities. Importantly, these mechanisms create a multiplicative increase in attacker cost and handling complexity, which is a pragmatic goal for cloud security: making attacks too costly or too risky for most adversaries while acknowledging that perfect security is unattainable without sacrificing core performance or configurability.

DISCUSSION

This section interprets the results, explores the trade-offs inherent in the proposed defensive architecture, candidly addresses limitations, and outlines a research agenda to strengthen secure FPGA multi-tenancy over the medium term.

Trade-offs and Design Tensions

Performance vs Security: Many recommended mitigations entail performance or resource overhead. For instance, dual-rail precharge schemes double area and can reduce maximum clock frequencies, and deterministic scheduling reduces statistical multiplexing efficiency (Danger et al., 2009; Diktopoulos et al., 2022). Cloud providers must therefore make calibrated decisions: apply the most stringent measures where tenant sensitivity justifies them (e.g., financial cryptographic workloads), and provide configurable security tiers so customers can choose desired security/performance trade-offs.

Flexibility vs Isolation: Reconfigurability is the primary value proposition of FPGAs; however, strong isolation policies—such as exclusive device assignment—undermine flexibility and raise costs (Chung et al., 2011). A practical solution is tiered offering models: dedicate devices for the highest security tenants, use partitioned devices with strict policies for mid-tier, and allow shared devices with monitoring for best-effort tenants. Such a model preserves market flexibility while enabling sensible security guarantees.

Operational Complexity vs Security Gains: Implementing robust bitstream verification, key management, and runtime monitoring increases operational complexity. For cloud operators, this invokes increased staffing, tooling, and processes—yet the security gains are significant and may be necessary to earn enterprise trust (Drimer & Kuhn, 2009; Intel, 2023). Automation, hardware support for attestation, and standards can reduce this burden over time.

Hardware Cost vs Robustness: Some defenses require silicon support, such as hardware monitors and fencing primitives. Vendors must weigh incremental silicon cost against market demand. The research community can inform this by quantifying real-world attack prevalence and customer willingness to pay for enhanced guarantees (Diktopoulos et al., 2022).

Limitations of the Study

Scope of Empirical Validation: This article is a design and analysis work rooted in extant empirical literature. It deliberately avoids practical attack demonstrations or intrusive testing for safety and ethical reasons. Consequently, while threats are grounded in documented attack vectors, the precise quantitative efficacy of some countermeasures—especially in the face of novel or combined attacks—requires experimental validation. Prior work on active fences and dual-rail logic provides indicative empirical results that we reference, but comprehensive system-level measurements across multiple FPGA families remain a needed future work item (Diktopoulos et al., 2022; Danger et al., 2009).

Vendor Heterogeneity: FPGA architectures vary significantly across vendors and device families, particularly in their configuration chain, available hardened blocks, and partial reconfiguration semantics. Our analysis abstracts over vendor specifics to highlight generalizable design patterns. Implementation guidance must be tailored to each vendor's primitives and APIs, and this tailoring must involve vendors to ensure feasibility and performance (Chung et al., 2011; Intel, 2023).

Standards and Interoperability Gaps: Effective cloud-level security relies on standards for bitstream formats, attestation metadata, and configuration policies. Currently, the landscape is fragmented; thus, operationalizing our recommendations will require vendor cooperation and industry standards development to avoid vendor lock-in and to create interoperable assurance mechanisms (Intel, 2023).

Future Research Directions

Hardware-Assisted Fuzzing of Management Interfaces: The recent work demonstrating hardware support can improve fuzzing performance and precision presents an opportunity: apply hardware-assisted fuzzing to test configuration interfaces, partial reconfiguration controllers, and hypervisor management stacks to proactively identify vulnerabilities (Ding et al., 2021). Such systematic testing would reduce the vulnerability surface available to supply-chain or management plane adversaries.

Formal Verification of Configuration Chains: Formal methods could be used to reason about partial reconfiguration compatibility, resource claims, and placement constraints. A verified configuration manager could prevent class of attacks that exploit misalignment between runtime policies and bitstream metadata (Drimer & Kuhn, 2009).

Quantitative Side-Channel Leakage Modeling in Shared Fabrics: While countermeasures like dual-rail and active fences have been studied, a comprehensive quantitative model of leakage in densely packed multi-tenant fabrics is missing. Such models would guide placement, fence sizing, and when to apply balancing logic selectively (Danger et al., 2009; Diktopoulos et al., 2022).

Standardized Attestation and Key Lifecycle Management: Vendors and cloud providers should develop standards for hardware-anchored attestation specific to FPGA configurations, including a recommended key lifecycle model with cross-organizational signing authorities to mitigate supply-chain threats (Drimer & Kuhn, 2009; Intel, 2023).

Design Patterns for Homomorphic Encryption Accelerators: Homomorphic encryption is particularly demanding and sensitive. Accelerator designs can be co-designed with security primitives—such as internal key wrapping and on-chip secure enclaves—to minimize exposure. Comparative studies of ring-based homomorphic schemes and hardware acceleration tradeoffs would support practical recommendations for accelerator developers (Costache & Smart, 2015; Cilardo & Argenziano, 2016).

Policy and Economic Research: Understanding how security requirements translate into pricing and SLAs in the cloud ecosystem is essential. Research should analyze how to economically incentivize best practices (e.g., charging premiums for dedicated secure FPGAs) and how regulatory frameworks might mandate baseline protections for critical sectors using FPGA accelerators.

Practical Recommendations

Cloud Providers

- Adopt a security tiering model for FPGA offerings, tying security features (dedicated vs partitioned devices, verified boot, fencing) to SLAs.

- Implement robust bitstream verification infrastructure integrated with tenant onboarding and enforce multi-party signing for critical artifacts.

- Deploy runtime monitors and anomaly detection pipelines specialized for FPGA telemetry and provide transparent audit logs for customers.

FPGA Vendors

● Provide hardware primitives for active fences, monitoring hooks, and secure configuration chains that make software enforcement practical without substantial performance loss.

● Standardize attestation metadata for bitstreams and define clear partial reconfiguration semantics to enable secure multi-tenant partitioning.

Researchers

● Pursue empirical measurements that quantify leakage risks in shared fabrics across device families and evaluate proposed mitigations in real deployments.

● Develop formal toolchains for verifying configuration compatibility and resource claims before runtime reconfiguration.

CONCLUSION

Securing multi-tenant FPGA accelerators in cloud environments is a multifaceted challenge that intersects hardware architecture, cryptographic practice, operational rigor, and economic incentives. This article synthesizes a layered defense approach that integrates authenticated configuration, in-fabric memory partitioning, active fencing, selective balancing logic for sensitive modules, hypervisor-aware resource allocation, and runtime monitoring to address the most salient threat classes faced by cryptographic workloads. The literature indicates that many of these mitigations are effective when applied together: authenticated bitstreams eradicate a broad class of injection attacks (Drimer & Kuhn, 2009); in-fabric memory partitioning and CoRAM-style abstractions improve isolation and resource management (Chung et al., 2011); active fences and dual-rail logic can reduce exploitable side-channel leakage (Danger et al., 2009; Diktopoulos et al., 2022); and hardware-assisted fuzzing and runtime monitoring enhance detection and response (Ding et al., 2021).

However, the effectiveness of these measures depends on coordinated action by cloud providers, FPGA vendors, and the research community. Key priorities include standardizing attestation metadata and signing practices, extending hardware primitives for secure partitioning and monitoring, formalizing the partial reconfiguration chain, and investing in empirical studies that quantify both risks and mitigation efficacy. In the interim, cloud providers should employ tiered security offerings, rigorous operational controls for bitstream provenance, conservative scheduling for sensitive tenants, and continuous testing of management interfaces.

Ultimately, adopting a zero-trust perspective—where no configuration or management action is trusted by default and attestation, authentication, and continuous verification are enforced—will be critical to safely leveraging the reconfigurability of FPGAs for cryptographic acceleration in the cloud (Hariharan, 2025). The layered, defense-in-depth approach presented here raises the bar for attackers by increasing the technical complexity, resource requirements, and risk of detection for malicious operations, while preserving the flexibility and performance that make FPGA acceleration attractive. Continued interdisciplinary collaboration across hardware, software, and operations will be necessary to evolve these practices into robust, scalable, and standards-supported solutions.

REFERENCES

1. Eric S. Chung, James C. Hoe, and Ken Mai. 2011. CoRAM: an in-fabric memory architecture for FPGA-based computing. In Proceedings of the 19th ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA ’11). ACM, New York, NY, 97–106. DOI: https://doi.org/10.1145/1950413.1950435

2. Alessandro Cilardo and Domenico Argenziano. 2016. Securing the cloud with reconfigurable computing: An FPGA accelerator for homomorphic encryption. In Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE ’16), 1622–1627.

3. Chris Conger, Ian Troxel, Daniel Espinosa, Vikas Aggarwal, and Alan George. 2005. NARC: Network-Attached Reconfigurable Computing for High-performance Network-based Applications. In Proceedings of the 8th Annual MAPLD International Conference. Retrieved from https://klabs.org/mapld05/abstracts/233_conger_a.html

4. Intel Corporation. 2023. Intel® Scalable I/O Virtualization Technical Specification. Technical Report. Intel. Retrieved July 31, 2024 from https://cdrdv2-public.intel.com/671403/intel-scalable-io-virtualization-technical-specification.pdf

5. Anamaria Costache and Nigel P. Smart. 2015. Which Ring Based Somewhat Homomorphic Encryption Scheme is Best? Cryptology ePrint Archive, Paper 2015/889. Retrieved from https://eprint.iacr.org/2015/889

6. Jean Luc Danger, Sylvain Guilley, Shivam Bhasin, and Maxime Nassar. 2009. Overview of dual rail with precharge logic styles to thwart implementation-level attacks on hardware cryptoprocessors - New attacks and improved counter-measures. In Proceedings of the 3rd International Conference on Signals, Circuits and Systems (SCS '09). DOI: https://doi.org/10.1109/ICSCS.2009.5412599

7. Ghada Dessouky, Ahmad-Reza Sadeghi, and Shaza Zeitouni. 2021. SoK: Secure FPGA multi-tenancy in the cloud: Challenges and opportunities. In Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P '21), 487–506. DOI: https://doi.org/10.1109/EuroSP51992.2021.00040

8. Christos Diktopoulos, Konstantinos Georgopoulos, Andreas Brokalakis, Georgios Christou, Grigorios Chrysos, Ioannis Morianos, and Sotiris Ioannidis. 2022. Assessing the effectiveness of active fences against SCAs for multi-tenant FPGAs. In Proceedings of the 32nd International Conference on Field-Programmable Logic and Applications (FPL '22), 391–396. DOI: https://doi.org/10.1109/FPL57034.2022.00065

9. Ren Ding, Yonghae Kim, Fan Sang, Wen Xu, Gururaj Saileshwar, and Taesoo Kim. 2021. Hardware support to improve fuzzing performance and precision. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21). ACM, New York, NY, 2214–2228. DOI: https://doi.org/10.1145/3460120.3484573

10. Saar Drimer, Tim Güneysu, Markus Kuhn, and Christof Paar. 2008. Protecting multiple cores in a single FPGA design. Retrieved from https://www.researchgate.net/publication/228818088_Protecting_multiple_cores_in_a_single_FPGA_design

11. Saar Drimer and Markus G. Kuhn. 2009. A protocol for secure remote updates of FPGA configurations. In Reconfigurable Computing: Architectures, Tools and Applications. Jürgen Becker, Roger Woods, Peter Athanas, and Fearghal Morgan (Eds.), Springer, Berlin, 50–61

12. Hariharan, R. (2025). Zero trust security in multi-tenant cloud environments. Journal of Information Systems Engineering and Management, 10.

13. Swarnagowri, B. N., & Gopinath, S. (2013). Ambiguity in diagnosing esthesioneuroblastoma--a case report. Journal of Evolution of Medical and Dental Sciences, 2(43), 8251-8255.

14. Swarnagowri, B. N., & Gopinath, S. (2013). Pelvic Actinomycosis Mimicking Malignancy: A Case Report. tuberculosis, 14, 15.

15. Krishnan, S., Shah, K., Dhillon, G., & Presberg, K. (2016). 1995: FATAL PURPURA FULMINANS AND FULMINANT PSEUDOMONAL SEPSIS. Critical Care Medicine, 44(12), 574.

16. Krishnan, S. K., Khaira, H., & Ganipisetti, V. M. (2014, April). Cannabinoid hyperemesis syndrome-truly an oxymoron!. In Journal of General Internal Medicine (Vol. 29, pp. S328-S328). SPRINGER.

17. Krishnan, S., & Selvarajan, D. (2014). D104 CASE REPORTS: INTERSTITIAL LUNG DISEASE AND PLEURAL DISEASE: Stones Everywhere!. American Journal of Respiratory and Critical Care Medicine, 189, 1.

18. Mahmud, U., Alam, K., Mostakim, M. A., & Khan, M. S. I. (2018). AI-driven micro solar power grid systems for remote communities: Enhancing renewable energy efficiency and reducing carbon emissions. Distributed Learning and Broad Applications in Scientific Research, 4.

19. Nagar, G. (2018). Leveraging Artificial Intelligence to Automate and Enhance Security Operations: Balancing Efficiency and Human Oversight. Valley International Journal Digital Library, 78-94.

20. M. Saraswathi, T. Bhuvaneswari. Multitenancy in Cloud-based Software, as a Service Application, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 13, Issue 11, 2013.

21. K. Venkataramana, M. Padmavathamma. Multi-Tenant Data Storage Security In Cloud Using Data

Partition Encryption Technique. International Journal of Scientific & Engineering Research, vol 4, issue 7, 2013.

22. Hussain Auahdali, Abdulaziz Albatli, Peter Garraghan, Paul Townend, Lydia Lau, Jie Xu. Multi-Tenancy in Cloud Computing, 8th International Symposium on Service-Oriented System Engineering (SOSE), 2014. https://doi.org/10.1109/SOSE.2014.50

23. Bhawna Sehgal, Jasbeer Narwal. An Analysis of Performance for Multi-Tenant Application through CloudSIM, International Journal of Emerging Research in Management & Technology, vol 4 issue 6, 2015.