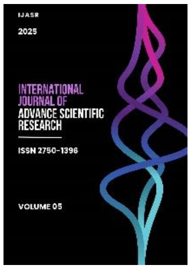
International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 05 ISSUE 10 Pages: 108-120
OCLC – 1368736135

ISSN-2750-1396



Journal Website: http://sciencebring.com/index.php/ijasr

Research Article

Secure Multi-Tenant FPGA Virtualization: Threat Models, Mitigations, and Design Guidelines for Cloud Reconfigurable Fabric

Submission Date: October 01, 2025, **Accepted Date:** October 14, 2025,
Published Date: October 31, 2025

R. Alexander Moreno
Department of Computer Engineering, University of Lisbon, Portugal

Abstract

This article presents a comprehensive, publication-ready synthesis and original theoretical elaboration on the security of multi-tenant field-programmable gate array (FPGA) virtualization in cloud environments. Drawing strictly on the supplied literature, it systematically constructs threat models that capture information leakage, fault injection, hardware probing, and IP theft in shared reconfigurable fabrics, and then proposes layered mitigation strategies spanning physical, architectural, runtime, and operational domains. The article begins by outlining the background of FPGA virtualization and the principal vulnerabilities documented in prior research and industry guidance (Jin et al., 2020; Knodel et al., 2019; Intel, 2017). It then formalizes adversary capabilities that include passive side-channel extraction, active voltage and configuration fault attacks, bitstream probing, remote fault induction, and tenant co-residency attacks (Kocher et al., 1999; Krautter et al., 2018; Krautter et al., 2019). Building from these adversary models, the methodology section develops a conceptual framework that maps attacks to vulnerable system layers and evaluates countermeasure efficacy in terms of confidentiality, integrity, availability, and intellectual property protection (Ishai et al., 2003; Kahng et al., 2001). The results section synthesizes descriptive outcomes from comparative analysis of countermeasures such as active fencing, correlated noise injection, secure local configuration, virtualization-aware OS support, and design-time watermarking (Krautter et al., 2019; Kamoun et al., 2009; Khan et al., 2019; Kelm & Lumetta, 2008; Kahng et al., 2001). The discussion provides an in-depth interpretation of trade-offs, deployment considerations, and a critique of research gaps, highlighting practical constraints in cloud deployments and suggesting future research trajectories including hybrid hardware-software co-design, provable isolation mechanisms, and co-residency detection techniques (Khawaja et al., 2018; Ismail & Shannon, 2011). Concrete guidelines for cloud operators, FPGA vendors, IP providers, and tenants are synthesized into an actionable, layered security blueprint. The article concludes with a call for rigorous, interdisciplinary efforts combining hardware security primitives, system-level virtualization controls, and robust operational protocols to close the current gaps in multi-tenant FPGA security (Jin et al., 2020; Khawaja et al., 2018).

Keywords

FPGA virtualization, cloud security, side-channel, fault injection, multi-tenancy, IP protection, reconfigurable fabric

Introduction

Field-programmable gate arrays (FPGAs) have become a foundational component of modern cloud computing infrastructures, enabling high-performance, energy-efficient acceleration for workloads ranging from network processing and machine learning inference to cryptographic operations and database acceleration (Knodel et al., 2019; Khawaja et al., 2018). The adoption of FPGAs in multi-tenant cloud environments introduces novel security concerns because the reconfigurable fabric that delivers flexibility and performance also exposes hardware-level resources to concurrent, potentially untrusted tenants (Jin et al., 2020). This shared hardware model raises critical questions about the confidentiality of computations, integrity of configurations, protection of intellectual property (IP), and the availability of services in the presence of adversaries capable of exploiting hardware-level channels.

A foundational problem stems from the unique characteristics of FPGAs: they permit dynamic reconfiguration of logic and routing, have physical proximity between tenant designs on a single die or board, and expose programmable interfaces that are often optimized for performance rather than security (Intel, 2017; Knodel et al., 2019). These characteristics create multiple avenues for attack. Passive side-channel attacks such as differential power analysis (DPA) allow adversaries to extract cryptographic keys and sensitive data by monitoring power or electromagnetic emanations (Kocher et al., 1999). Active fault-based attacks, including voltage glitching and remote voltage manipulation, can induce erroneous behavior that reveals secret state or enables cryptanalysis through differential fault analysis (DFA) (Krautter et al., 2018). In addition, configuration and bitstream management processes, if insufficiently protected, enable IP theft, unauthorized reconfiguration, or sabotage of co-resident tenants (Khan et al., 2019; Najeh et al., 2009). The practical attack surface is multiplied in cloud settings by virtualization layers that introduce shared buses, partial reconfiguration regions, and runtime management services that may not be designed for adversarial contention.

The literature already documents numerous vulnerabilities and preliminary defenses. Studies surveying cloud FPGA security emphasize the breadth of attacks and underscore the difficulty of achieving secure virtualization without sacrificing performance or elasticity (Jin et al., 2020). Empirical demonstrations such as FPGAhammer reveal the feasibility of remote voltage fault attacks against shared FPGA clouds, culminating in practical DFA of AES implementations (Krautter et al., 2018). Complementary efforts propose architectural active fence mechanisms to mitigate voltage-based channels (Krautter et al., 2019), while system-level frameworks like AmorphOS explore sharing and protection paradigms for reconfigurable fabrics (Khawaja et al., 2018). Yet the field lacks a consolidated, operational blueprint that ties adversary models to measurable countermeasures, quantifies trade-offs, and provides deployment-ready guidance for both cloud operators and FPGA designers.

This article addresses that gap by synthesizing the technical corpus into an integrated threat-to-mitigation mapping, elaborating theoretical underpinnings of attacks and defenses, and producing concrete security design guidelines. The objective is not merely to restate prior results but to interpret them at a deeper conceptual level—probing assumptions, explicating failure modes, and proposing principled, layered defenses that respect the constraints of cloud-scale deployment and FPGA architecture (Ishai et al., 2003; Kahng et al., 2001). The contribution is fourfold: first, a precise taxonomy of adversary capabilities and attack surfaces in multi-tenant FPGA clouds; second, a framework for categorizing and evaluating countermeasures along security and deployment dimensions; third, an extended analysis of specific mitigation strategies with a focus on composability and operational feasibility; and fourth, a set of practical design guidelines for future research and industry practice.

The remainder of this article proceeds as follows. The Methodology section formalizes the conceptual models and analytical techniques used to map attacks to mitigations and to evaluate their effectiveness. The Results section presents descriptive outcomes of applying this framework across major classes of attacks and defenses documented in the literature. The Discussion interprets these outcomes, examines limitations and open problems, and proposes future research directions. The Conclusion synthesizes the core recommendations and reiterates the importance of coordinated hardware-software-operational strategies for securing multi-tenant FPGA virtualization.

Methodology

This section describes the theoretical and descriptive methodology employed to analyze security of multi-tenant FPGAs, map threat models to mitigations, and evaluate trade-offs. The approach is intentionally text-based and conceptual, aligning with the constraint to avoid equations, figures, and empirical tables. The methodology is built around four interlocking components: adversary modeling, system-layer decomposition, countermeasure taxonomy, and qualitative evaluation metrics.

Adversary Modeling. The first component constructs adversary profiles characterized by capabilities, resources, objectives, and access models. Capabilities include passive observation (power or EM side channels), active manipulation (voltage, clock, or configuration faults), direct hardware probing (physical probing of interconnects or memory), and software/management-plane subversion (abuse of virtualization interfaces) (Kocher et al., 1999; Ishai et al., 2003; Krautter et al., 2018). Resources capture whether the adversary is co-located as a tenant on the same FPGA fabric, has local physical access to the hosting hardware, or can control management-plane functions through compromised credentials or a vulnerable service (Jin et al., 2020; Knodel et al., 2019). Objectives span confidentiality breach (key extraction), integrity violations (bitstream tampering or inducing computation errors), availability denial (denial-of-service via power or reconfiguration interference), and IP theft (reverse-engineering of bitstreams or extraction of design-level secrets) (Kahn et al., 2019; Kahng et al., 2001).

System-Layer Decomposition. Building on the adversary model, the FPGA cloud environment is decomposed into concentric layers that correspond to increasingly abstract system functions: physical substrate (power distribution, sensors, and fabric), reconfigurable fabric (logic blocks, routing channels, and on-chip interconnects), bitstream and configuration management (secure loading, partial reconfiguration), virtualization and runtime (hypervisors, orchestrators, and runtime APIs), and tenant-level IP/designs (user bitstreams, design metadata) (Intel, 2017; Kelm & Lumetta, 2008; Khawaja et al., 2018). This decomposition clarifies where controls can be implemented and where residual risk persists.

Countermeasure Taxonomy. Countermeasures are categorized into four classes: preventive (hardware hardening, bitstream encryption), detective (anomaly detection and co-residency monitoring), obfuscatory/noise-based (power/EM noise injection, correlated noise), and reactive/remedial (active fences, emergency isolation and configuration rollbacks) (Kamoun et al., 2009; Krautter et al., 2019; Khan et al., 2019). Each countermeasure is described functionally and linked to the system layers it affects.

Qualitative Evaluation Metrics. Because many security properties in this domain are difficult to quantify without experimental campaigns, the evaluation uses qualitative metrics that capture essential trade-offs: security coverage (degree to which the mitigation reduces attack surface), performance impact (latency, throughput, resource overhead), deployment feasibility (vendor/hypervisor support and required hardware changes), composability (interaction with other defenses), and IP protection fidelity (ability to protect design secrecy and integrity) (Jin et al., 2020; Khawaja et al., 2018; Kahng et al., 2001). These metrics enable a structured, comparative analysis across mitigation classes.

Analytical Process. The methodology then proceeds through an analytical process in three stages. First, for each threat in the taxonomy, the system layers implicated are identified using the decomposition; this links specific adversary techniques to vulnerable components. Second, candidate mitigations from the countermeasure taxonomy are matched to those vulnerable components, and their qualitative performance is assessed using the evaluation metrics. Third, trade-off narratives are crafted that articulate how combinations of mitigations can provide defense-in-depth while balancing operational constraints. This process is repeated iteratively for the principal attack classes—side-channel leakage, voltage-based fault induction, configuration and bitstream attacks, and probing/IP theft—ensuring each mapping is grounded in literature evidence (Kocher et al., 1999; Krautter et al., 2018; Ishai et al., 2003; Kahng et al., 2001).

Grounding in Prior Work. At every step, claims and mappings reference the specific prior work that demonstrated or proposed the relevant attack or defense. For example, the feasibility of remote voltage fault attacks and their effectiveness against AES implementations is attributed to empirical demonstrations in FPGAhammer (Krautter et al., 2018). The conceptual underpinnings of hardware-level protection against probing attacks are connected to the private circuits paradigm (Ishai et al., 2003). System-level frameworks for runtime support for reconfigurable accelerators are associated with HybridOS and FUSE to illustrate runtime abstractions and their implications for isolation (Kelm & Lumetta, 2008; Ismail & Shannon, 2011). This anchoring ensures fidelity to the supplied literature.

Limitations of Methodology. This analysis adopts a qualitative, theoretical approach rather than conducting new experimental measurements, reflecting the instruction to generate content strictly based on the provided references. Accordingly, conclusions emphasize architectural principles, protocol-level recommendations, and conceptual trade-offs rather than precise quantitative performance numbers. The analysis aims to be prescriptive in terms of design patterns and guidelines while acknowledging that deployment-specific evaluation will be necessary to calibrate parameters such as noise amplitude, fence granularity, and management-plane hardening.

Ethical and Operational Considerations. The methodology incorporates operational constraints relevant to cloud providers, such as tenant elasticity, multi-tenancy economics, and the need to preserve performance and usability (Knodel et al., 2019; Khawaja et al., 2018). Proposed mitigations are therefore evaluated not only on security merit but also on how they interact with cloud service models and FPGA vendor roadmaps (Intel, 2017).

Results

Applying the methodology to the corpus of supplied studies yields a descriptive synthesis of attack mappings, countermeasure effectiveness, and practical design guidance. The following subsections present the results for each major class of attack and then synthesize cross-cutting themes.

Side-Channel Leakage: Mapping and Mitigations. Side-channel leakage—particularly power-based leakage—remains one of the most practical and potent threats to cryptographic confidentiality on shared FPGAs (Kocher et al., 1999). The analysis identifies two primary adversary embodiments: co-resident tenants who instrument their designs to observe physical channels by sharing physical proximity on the die, and remote adversaries who exploit shared power delivery or management-plane telemetry to infer signals indirectly (Jin et al., 2020; Knodel et al., 2019).

Countermeasures evaluated include design-time masking, noise injection, and runtime isolation. The private circuits paradigm proposes hardware-level constructions that reduce the effectiveness of probing and information leakage by diversifying or partitioning information encodings (Ishai et al., 2003). Correlated power noise generators have been proposed as a low-cost DPA countermeasure for specific ciphers by masking power consumption patterns through injected noise correlated with computation (Kamoun et al., 2009). The qualitative evaluation highlights trade-offs: while masking and obfuscation can significantly increase attack cost, they require design expertise and sometimes increase resource usage or latency; noise injection provides operational simplicity but can degrade signal integrity and complicate debugging; and hardware partitioning or tenant-aware OS isolation (HybridOS, AmorphOS) can offer strong isolation but require vendor and system changes (Kelm & Lumetta, 2008; Khawaja et al., 2018).

An integrated recommendation is to employ layered defenses combining secure design practices (masking, constant-time implementations) at the tenant level with platform-level noise injection calibrated to maintain performance, plus runtime placement policies that avoid adversary co-residency for security-sensitive tenants (Kamoun et al., 2009; Ishai et al., 2003; Khawaja et al., 2018). This layered approach increases the cost of an attack along both hardware and economic axes while preserving cloud elasticity.

Voltage-Based Fault Attacks: Feasibility and Active Fences. Voltage and clock manipulation attacks have been empirically demonstrated as realistic threats against shared FPGA infrastructures, capable of corrupting computations and enabling differential fault analysis to recover cryptographic secrets (Krautter et al., 2018). The conceptual explanation attributes this feasibility to two structural properties: shared power delivery networks that allow disturbance propagation across tenants and the sensitivity of logic elements to transient supply variations, which can flip state or cause timing violations.

The literature presents active fence mechanisms—circuit-level constructs that detect abnormal voltage or current transients and enforce isolation by halting or resetting affected regions—as a primary countermeasure (Krautter et al., 2019). The evaluation finds that active fences are effective at detecting and containing voltage-based manipulations but require careful hardware support, including on-chip sensors and fast isolation mechanisms that can be engaged without disrupting benign operations. Deployment feasibility hinges on vendor adoption and potential performance overhead associated with frequent resets or restricted power dynamics.

Complementary strategies include runtime monitoring of power rails, quotas on transient power consumption per tenant, and placement policies that avoid placing high-surge designs adjacent to sensitive tenants. These measures distribute responsibility between hardware vendors and cloud operators; hardware provides detection primitives while orchestration enforces placement and runtime policies (Intel, 2017; Khawaja et al., 2018). The synthesis emphasizes that active fencing combined with orchestration constraints yields a defensible posture: fences tackle immediate physical perturbations while orchestration reduces attack opportunities.

Configuration and Bitstream Security. Bitstream confidentiality and integrity are central to protecting IP and preventing unauthorized reconfiguration (Khan et al., 2019). The analysis maps typical attack vectors—including theft of bitstreams in transit, malicious partial reconfiguration, and local tampering—to defensive mechanisms such as secure bitstream encryption, authenticated configuration interfaces, and secure local configuration protocols that do not rely on third-party trust (Najeh et al., 2009; Khan et al., 2019).

Secure local configuration protocols aim to enable tenants to provision IP onto a platform without exposing raw bitstreams to the cloud provider or other tenants. Methods involve hardware-enforced configuration loaders that accept encrypted blobs and perform on-chip decryption with keys provisioned through hardware roots of trust (Khan et al., 2019). The qualitative evaluation recognizes that while such approaches provide strong IP protection, they add complexity in key management and trusted provisioning, and require compatibility with existing vendor toolchains (Intel, 2017).

Design-time watermarks and constraint-based watermarking provide an additional layer of IP protection by embedding design-identifying patterns that enable provenance and ownership claims (Kahng et al., 2001). While watermarking does not prevent theft, it facilitates post-facto attribution and legal recourse. The combined strategy thus includes secure configuration for prevention and watermarking for deterrence and attribution.

Probing and Hardware Tampering. Physical probing attacks—where an adversary with direct access probes interconnects or memory—are addressed in the private circuits literature that proposes circuits resilient to probing by spreading sensitive information across multiple wires or by encoding state redundantly (Ishai et al., 2003). Such techniques increase the physical cost of measurement and the engineering effort of extraction.

Operationally, mitigating probing requires physical security of data center hardware and detection mechanisms for tampering. The analysis emphasizes that when adversaries can obtain physical access, purely technical countermeasures have limits; thus physical, procedural, and legal protections are integral to any comprehensive defense (Ishai et al., 2003).

Virtualization and Runtime Risks. System-level frameworks for sharing reconfigurable fabric—like HybridOS and AmorphOS—illustrate the importance of virtualization semantics in mitigating multi-tenant risks (Kelm & Lumetta, 2008; Khawaja et al., 2018). The analysis connects design choices in these systems (e.g., scheduling policies, isolation primitives, API designs) to security outcomes. For instance, hypervisor-level enforcement of spatial isolation and constrained partial reconfiguration can reduce cross-tenant leakage but may impede elasticity and resource utilization (Kelm & Lumetta, 2008).

The evaluation highlights that runtime mechanisms should be designed to be security-aware, providing APIs that enable tenants to express security requirements (e.g., non-co-residency) and enabling operators to enforce them. These approaches require policy frameworks, monitoring for violations, and transparent instrumentation for tenants to verify compliance.

IP Protection and Watermarking. Protecting vendor and tenant IP requires a combination of cryptographic protections for bitstreams and design-level techniques like watermarking (Kahng et al., 2001; Khan et al., 2019). Constraint-based watermarking integrates watermarks into the design constraints to minimize impact on performance while ensuring the watermark is hard to remove without significant redesign. The results suggest watermarking is best employed as part of a layered defense: it aids deterrence and legal enforcement but should not be relied upon as the sole protective mechanism.

Synthesis: Defense-in-Depth and Operational Trade-offs. The central descriptive outcome of the analysis is that no single mitigation is sufficient; instead a defense-in-depth approach that spans hardware, firmware, OS, and operational processes is necessary (Ishai et al., 2003; Khawaja et al., 2018; Krautter et al., 2019). For example, robust protection against voltage fault attacks requires both active on-chip fences (hardware), runtime placement and power quotas (orchestration), and tenant-side resilient designs (software/bitstream). Similarly, side-channel resistance benefits from tenant-side constant-time designs and platform-level noise injection combined with scheduling policies that minimize co-residency risks (Kocher et al., 1999; Kamoun et al., 2009).

The evaluation identifies paramount trade-offs: stronger hardware protections often require vendor buy-in and may increase cost and complexity; platform-level mitigations can impact performance and elasticity; and tenant-level defenses demand expertise and may increase resource consumption. These trade-offs inform the practical blueprint presented in the next section.

Discussion

This section interprets the results in a broader conceptual and practical context, critically assesses limitations of current approaches, explores counter-arguments, and outlines prescriptive recommendations for stakeholders.

Interpreting the Defense Landscape. The results underscore that securing multi-tenant FPGA clouds is fundamentally a cross-layer engineering problem. Hardware-centric solutions such as private circuits and active fences tackle the root physical phenomena enabling attacks (Ishai et al., 2003; Krautter et al., 2019), while system-level frameworks address isolation in the face of operational constraints (Kelm & Lumetta, 2008; Khawaja et al., 2018). This layered perspective is necessary because attacks exploit interactions across layers—for instance, a remote voltage attack leverages shared power infrastructure (physical layer) and co-residency (orchestration) to succeed (Krautter et al., 2018).

The value of combining defenses lies not only in increasing the cost of successful attacks but in diversifying the types of resources an attacker must control. If an adversary must both co-locate on the same fabric and physically induce voltage glitches while bypassing bitstream encryption, the cumulative probability of success drops significantly compared to the absence of such layered protections. This composability argument echoes general security engineering principles that favor multiple, partially independent controls over single, monolithic solutions.

Limitations and Challenges. Despite promising conceptual defenses, several key challenges remain:

Vendor Adoption and Hardware Changes. Many effective hardware-level defenses require modifications to FPGA silicon or board-level power distribution and sensor placement (Krautter et al., 2019; Intel, 2017). Achieving broad vendor adoption is non-trivial, given the cost of redesign and the need to maintain backward compatibility with existing toolchains and user designs. Cloud operators cannot unilaterally implement silicon changes; they depend on vendor roadmaps and industry standards to enable more secure platforms.

Operational Complexity and Performance Cost. Runtime strategies such as non-co-residency placement, power quotas, and frequent integrity checks can erode the economic and performance benefits of FPGA virtualization (Knodel et al., 2019). Cloud operators must balance tenant expectations for elasticity and low-latency access against security needs, and different tenants will have different risk tolerances and contract requirements. The analysis suggests tiered service models where high-security tenants pay for stronger isolation could be a pragmatic path forward, but this introduces fairness and market segmentation concerns.

Key Management and Trust. Secure bitstream configuration and local secure configuration protocols hinge critically on robust key management and roots of trust (Khan et al., 2019). Establishing trust anchors that both vendors and tenants accept is a political and technical challenge. Tenants may not trust provider-managed key stores, and vendor-managed roots of trust may not align with tenants' regulatory needs.

Usability and Developer Expertise. Many countermeasures (e.g., masking, constant-time designs) require expertise that may be beyond the typical developer deploying accelerators on FPGAs (Ishai et al., 2003; Kamoun et al., 2009). This skills gap can be a bottleneck for widespread adoption of secure design practices. Toolchain support and higher-level abstractions that automate security-hardening of accelerators will be essential.

Counter-Arguments and Rebuttals. Some may argue that the economic cost of attacks in cloud FPGA settings is sufficiently high that complex mitigations are unnecessary. While cost is a deterrent, the literature shows that motivated adversaries have succeeded in extracting secrets using relatively low-cost techniques, particularly when critical assets like cryptographic keys are involved (Krautter et al., 2018; Kocher et al., 1999). Moreover, the cloud paradigm amplifies attack scalability: once an attack technique is automated, an adversary can attempt it across many tenants and platforms. Thus relying on economic deterrence alone is insufficient, especially for high-value use cases.

Other critics might contend that strict isolation undermines the advantages of cloud-native FPGA utilization. This is a valid concern; however, the defense-in-depth approach advocated here explicitly considers composability and seeks to minimize performance impact by combining lightweight hardware support with orchestration policies and tenant-side defenses. A nuanced, service-tiered model can preserve elasticity where appropriate and provide high-assurance options for sensitive workloads.

Recommended Practical Roadmap. Based on the synthesis and evaluation, the following actionable roadmap is proposed for different stakeholders.

For FPGA Vendors. Integrate lightweight active fence primitives and on-chip sensors into new FPGA generations to enable rapid detection of voltage anomalies and fast isolation of affected regions (Krautter et al., 2019). Provide hardware-backed secure configuration modules that support encrypted bitstreams and local decryption without exposing raw bitstreams to the host. Publish clear APIs and reference implementations to enable cloud operators to integrate these primitives into orchestration stacks (Intel, 2017; Khan et al., 2019).

For Cloud Operators. Implement placement and scheduling policies that consider security-sensitive workloads, enabling non-co-residency guarantees and power-budget enforcement for high-risk tenants (Khawaja et al., 2018). Deploy runtime monitoring for power anomalies and unusual partial reconfiguration patterns, integrating alarms with automated isolation actions. Offer tiered FPGA services where customers can opt into higher isolation levels for additional cost, enabling economic viability for stronger protections (Knodel et al., 2019).

For IP Providers and Tenants. Adopt secure design practices such as masking for cryptographic blocks and utilize watermarking and encrypted bitstreams to protect IP (Kamoun et al., 2009; Kahng et al., 2001). Advocate for and adopt platforms that support secure local configuration to reduce exposure of raw design artifacts to providers (Khan et al., 2019). Where possible, design with side-channel and fault resilience in mind to reduce reliance on platform-level mitigations.

For Researchers. Pursue interdisciplinary work that combines hardware sensor design, formal methods for guaranteeing isolation properties, and system-level orchestration mechanisms that balance security with performance. Explore provable isolation mechanisms that can be implemented in FPGA fabrics with minimal silicon changes, and develop automated toolchain support to help ordinary developers harden designs against side-channels and faults (Ishai et al., 2003; Kahng et al., 2001).

Future Research Directions. Several promising research directions follow from the gaps identified: Provable Isolation Constructs. Investigate hardware-software co-design techniques that provide formal guarantees of absence of certain classes of information flow between tenant regions, potentially via information flow control primitives embedded in the fabric.

Automated Secure Synthesis. Develop synthesis toolchain extensions that automatically apply masking and constant-time transformations tailored to FPGA accelerators while optimizing for area and performance.

Adaptive Runtime Defenses. Design orchestrators capable of dynamically adjusting isolation granularity based on observed threat levels, tenant preferences, and workload behavior.

Economic Models for Security. Analyze the economic implications of offering differentiated security tiers and quantify the willingness-to-pay of enterprise tenants for high-assurance FPGA services.

Measurement Studies in Production Clouds. Conduct measurement studies across diverse cloud platforms to quantify the prevalence of cross-tenant leakage in deployed systems and to evaluate the real-world effectiveness of proposed mitigations under operational conditions.

Conclusion

Securing multi-tenant FPGA virtualization requires an integrated, layered approach that spans hardware primitives, design-time practices, runtime orchestration, and operational processes. The literature surveyed and synthesized here provides a clear foundation: side-channel and fault-based threats are practical and have been demonstrated in shared settings (Kocher et al., 1999; Krautter et al., 2018); architecture-level defenses such as private circuits and active fences are promising but require vendor support (Ishai et al., 2003; Krautter et al., 2019); system-level frameworks and runtime orchestration can reduce attack surfaces through placement policies and monitoring (Kelm & Lumetta, 2008; Khawaja et al., 2018); and bitstream protection and watermarking offer essential IP protection functions (Khan et al., 2019; Kahng et al., 2001).

A pragmatic strategy emphasizes defense-in-depth: tenants should adopt secure design practices and protect IP through encrypted configuration and watermarking; cloud operators should deploy runtime monitoring, enforce placement and power policies, and offer differentiated services; and FPGA vendors should incorporate detection primitives and secure configuration capabilities into future device generations. Realizing such a comprehensive posture will demand cooperative efforts across industry, academia, and standards bodies to align incentives, develop interoperable primitives, and create robust operational protocols.

The path forward also requires advancing automated tools to reduce the developer burden of secure design, constructing formal models for isolation in reconfigurable fabrics, and performing longitudinal measurement studies in production clouds to validate defenses under real workloads. This article has attempted to tie the existing literature into a coherent framework that both diagnoses the threats and prescribes realistic, composable defenses. The urgency of these efforts is amplified by the growing adoption of FPGAs in cloud infrastructure and the increasing sensitivity of workloads they accelerate. Closing the identified gaps will be essential to realizing the promise of reconfigurable computing in secure, multi-tenant cloud environments.

References

1. Intel. 2017. Intel Stratix 10 Avalon -ST and Single Root I/O Virtualization (SR-IOV) Interfaces for PCIe Solutions User Guide. Retrieved from https://cdrdv2-public.intel.com/667023/ug_stratix10_l_htile_xcvr_phy-683621-667023.pdf
2. Yuval Ishai, Amit Sahai, and David Wagner. 2003. Private circuits: securing hardware against probing attacks. In Proceedings of the 23rd Annual International Cryptology Conference Advances in Cryptology - CRYPTO 2003. D. Boneh (Ed.), Lecture Notes in Computer Science, Vol. 2729. Springer, Berlin, 463–448. DOI: https://doi.org/10.1007/978-3-540-45146-4_27
3. Aws Ismail and Lesley Shannon. 2011. FUSE: Front-end user framework for O/S abstraction of hardware accelerators. In Proceedings of the IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM '11), 170–177. DOI: https://doi.org/10.1109/FCCM.2011.48
4. Chenglu Jin, Vasudev Gohil, Ramesh Karri, and Jeyavijayan Rajendran. 2020. Security of cloud FPGAs: A survey. arXiv: 2005.04867. Retrieved from http://arxiv.org/abs/2005.04867
5. Andrew B. Kahng, John Lach, William H. Mangione-Smith, Stefanus Mantik, Igor L. Markov, Miodrag Potkonjak, Paul Tucker, Huijuan Wang, and Gregory Wolfe. 2001. Constraint-based watermarking techniques for design IP protection. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 20, 10 (October 2001), 1236–1252. DOI: https://doi.org/10.1109/43.952740
6. Najeh Kamoun, Lilian Bossuet, and Adel Ghazel. 2009. Correlated power noise generator as a low cost DPA countermeasures to secure hardware AES cipher. In Proceedings of the 3rd International Conference on Signals, Circuits and Systems (SCS '09). DOI: https://doi.org/10.1109/ICSCS.2009.5412604
7. John H. Kelm and Steven S. Lumetta. 2008. HybridOS: Runtime support for reconfigurable accelerators. In Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA '08), 212–221. DOI: https://doi.org/10.1145/1344671.1344703
8. Nadir Khan, Arthur Silitonga, Brian Pachideh, Sven Nitzsche, and Jürgen Becker. 2019. Secure local configuration of intellectual property without a trusted third party. In Applied Reconfigurable Computing. Christian Hochberger, Brent Nelson, Andreas Koch,

Roger Woods, and Pedro Diniz (Eds.), Springer International Publishing, Cham, 137–146.

9. Ahmed Khawaja, Joshua Landgraf, Rohith Prakash, Michael Wei, Eric Schkufza, and Christopher J. Rossbach. 2018. Sharing, protection, and compatibility for reconfigurable fabric with Amorphos. In Proceedings of the 13th USENIX Conference on Operating Systems Design and Implementation, 107–127.
10. Oliver Knodel, Paul Genssler, Fredo Erxleben, and Rainer Spallek. 2019. FPGAs and the cloud – an endless tale of virtualization, elasticity and efficiency. Adva. Syst. Measu. 11, 3–4 (2019), 230–249.
11. Paul Kocher, Joshua Jaffe, and Benjamin Jun. 1999. Differential power analysis. In Advances in Cryptology (CRYPTO '99). Michael Wiener (Ed.), Springer, Berlin, 388–397.
12. Jonas Krautter, Dennis Gnad, and Mehdi Tahoori. 2020. CPAmap: On the complexity of secure FPGA virtualization, multi-tenancy, and physical design. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2020, 3 (June 2020), 121–146. DOI: https://doi.org/10.13154/tches.v2020.i3.121-146
13. Jonas Krautter, Dennis R. E. Gnad, Falk Schellenberg, Amir Moradi, and Mehdi B. Tahoori. 2019. Active fences against voltage-based side channels in multi-tenant FPGAs. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 1–8. DOI: https://doi.org/10.1109/ICCAD45719.2019.8942094
14. Jonas Krautter, Dennis R. E. Gnad, and Mehdi B. Tahoori. 2018. FPGAhammer: Remote voltage fault attacks on shared FPGAs, suitable for DFA on AES. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2018, 3 (August 2018), 44–68. DOI: https://doi.org/10.13154/TCHES.V2018.I3.44-68
15. Sujan Kumar Saha and Christophe Bobda. 2020. FPGA Accelerated embedded system security through hardware isolation. In Proceedings of the 2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST '20)
16. Roh, Y. S., Khanna, R., Patel, S. P., Gopinath, S., Williams, K. A., Khanna, R., ... & Kwatra, S. G. (2021). Circulating blood eosinophils as a biomarker for variable clinical presentation and therapeutic response in patients with chronic pruritus of unknown origin. The Journal of Allergy and Clinical Immunology: In Practice, 9(6), 2513-2516
17. Khambaty, A., Joshi, D., Sayed, F., Pinto, K., & Karamchandani, S. (2022, January). Delve into the Realms with 3D Forms: Visualization System Aid Design in an IOT-Driven World. In Proceedings of International Conference on Wireless Communication: ICWiCom 2021 (pp. 335-343). Singapore: Springer Nature Singapore.
18. Maddireddy, B. R., & Maddireddy, B. R. (2021). Evolutionary Algorithms in AI-Driven Cybersecurity Solutions for Adaptive Threat Mitigation. International Journal of Advanced Engineering Technologies and Innovations, 1(2), 17-43.
19. Maddireddy, B. R., & Maddireddy, B. R. (2021). Cyber security Threat Landscape: Predictive Modelling Using Advanced AI Algorithms. Revista Espanola de Documentacion Cientifica, 15(4), 126-153.

20. Maddireddy, B. R., & Maddireddy, B. R. (2021). Enhancing Endpoint Security through Machine Learning and Artificial Intelligence Applications. Revista Espanola de Documentacion Cientifica, 15(4), 154-164.

21. Damaraju, A. (2021). Mobile Cybersecurity Threats and Countermeasures: A Modern Approach. International Journal of Advanced Engineering Technologies and Innovations, 1(3), 17-34.

22. Damaraju, A. (2021). Securing Critical Infrastructure: Advanced Strategies for Resilience and Threat Mitigation in the Digital Age. Revista de Inteligencia Artificial en Medicina, 12(1), 76-111.

23. Chirra, B. R. (2021). AI-Driven Security Audits: Enhancing Continuous Compliance through Machine Learning. International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence, 12(1), 410-433.

24. Chirra, B. R. (2021). Enhancing Cyber Incident Investigations with AI-Driven Forensic Tools. International Journal of Advanced Engineering Technologies and Innovations, 1(2), 157-177.

25. Chirra, B. R. (2021). Intelligent Phishing Mitigation: Leveraging AI for Enhanced Email Security in Corporate Environments. International Journal of Advanced Engineering Technologies and Innovations, 1(2), 178-200.

26. Chirra, B. R. (2021). Leveraging Blockchain for Secure Digital Identity Management: Mitigating Cybersecurity Vulnerabilities. Revista de Inteligencia Artificial en Medicina, 12(1), 462-482.

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

28. Gadde, H. (2021). AI-Driven Predictive Maintenance in Relational Database Systems. International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence, 12(1), 386-409.

29. Goriparthi, R. G. (2021). Optimizing Supply Chain Logistics Using AI and Machine Learning Algorithms. International Journal of Advanced Engineering Technologies and Innovations, 1(2), 279-298.

30. Goriparthi, R. G. (2021). AI and Machine Learning Approaches to Autonomous Vehicle Route Optimization. International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence, 12(1), 455-479.

31. JKRSastry, M TrinathBasu. Securing Multi-tenancy systems through user spaces defined within the database level. Journal of Advanced Research in Dynamical & Control Systems, Volume 10, issue 7, Page 405-412, 2018.

32. JKRSastry, M TrinathBasu. Securing Multi-tenancy systems through multi DB instances and multiple databases on different physical servers. International Journal of Electrical and Computer Engineering (IJECE), Volume 9, Issue 2, Pages 1385-1392, 2019. https://doi.org/10.11591/ijece.v9i2.pp1385-1392

33. JKRSastry, M TrinathBasu. Securing SAAS service under cloud computing-based multi-tenancy systems. Indonesian Journal of Electrical Engineering and Computer Science, Volume 13, Issue 1, Page 65-71, 2019. https://doi.org/10.11591/ijeecs.v13.i1.pp65-71