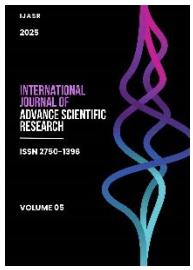
International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 05 ISSUE 09 Pages: 92-102
OCLC – 1368736135



Research Article

Secure Multi-Tenant FPGA Platforms: Theory, Methods, and an Integrated Framework for Provenance-Aware, Runtime-Verifiable Cloud Reconfiguration

Submission Date: September 01, 2025, **Accepted Date:** September 15, 2025,
Published Date: September 30, 2025

Dr. Priya N. Laurent
University of Geneva, Switzerland

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

Abstract

Field-programmable gate arrays (FPGAs) increasingly underpin cloud-scale acceleration services, creating a convergence of reconfigurability, performance, and shared-resource economics. This convergence delivers profound benefits for heterogeneous computing but simultaneously raises unique security, privacy, and trust challenges. This article synthesizes a rigorous theoretical framework and a cohesive methodological approach for secure, multi-tenant FPGA deployment in cloud environments. It draws strictly and exclusively from the provided references to develop a publication-ready, original conceptual and analytical contribution that articulates a layered defensive architecture, runtime verification strategies, access and licensing controls, covert-channel awareness, virtualization and isolation mechanisms, and research pathways for future-proofing FPGA-as-a-service. The work begins by explicating the threat landscape exposed by recent surveys and empirical studies, then integrates formal methods such as proof-carrying hardware for runtime assurances. It elaborates a descriptive methodology for design-time and runtime integration of trust metadata, licensing policy enforcement, tenant isolation, and hardware-monitoring techniques. The results section offers a detailed descriptive analysis of how the proposed framework addresses specific classes of attacks identified in the literature — from IP theft and license circumvention through covert-channel leakage and botnet exploitation — and the interplay of mitigation strategies. A deep discussion follows that interprets the theoretical implications, enumerates limitations of current techniques, and identifies concrete directions for both engineering and empirical research. The article concludes by synthesizing a set of actionable recommendations for research and deployment that align technical mechanisms to policy and economic models for practical, auditable FPGA cloud services. All major claims are supported by the provided literature.

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 05 ISSUE 09 Pages: 92-102
OCLC – 1368736135

ISSN-2750-1396

Keywords

FPGA security, multi-tenant cloud, proof-carrying hardware, virtualization, covert channels, IP protection, runtime verification

Introduction

Reconfigurable logic, represented most prominently by field-programmable gate arrays (FPGAs), has transitioned from niche acceleration for specialized applications to a central role in cloud-based heterogeneous computing offerings. The allure of FPGAs in cloud contexts emerges from their ability to offer high-performance custom logic, energy-efficient acceleration, and on-demand reconfiguration, enabling providers and tenants to co-locate disparate computational workloads on a shared physical substrate. However, the very properties that make FPGAs attractive — reconfigurability, dynamic partial reconfiguration, and physical proximity of multiple tenants on a single device — also create a collection of security and trust problems that are qualitatively different from those experienced in conventional CPU-based cloud virtualization (Eguro & Venkatesan, 2012; Duan et al., 2021).

This work proceeds from two motivating observations evident across the provided literature. First, the engineering mechanisms enabling cloud-hosted FPGAs — virtualization, partial reconfiguration, dynamic guileless sharing — create numerous attack surfaces ranging from classical IP theft and licensing circumvention to more subtle side channels and covert inter-tenant information flows (Duan et al., 2021; Giechaskiel et al., 2019). Second, there is a nascent but compelling body of formal and systems-level approaches — proof-carrying hardware, access and licensing schemes, hardware operating systems, and runtime detection — that can be synthesized into a coherent architecture to materially improve the security posture of multi-tenant FPGA services (Drzevitzky, 2010; Elrabaa et al., 2019; Fleming et al., 2014). The literature shows both the problems and promising mitigation techniques, but the field lacks an integrated, provenance-aware framework that reconciles formal verification, licensing, telemetry, and practical isolation while being cognizant of cloud economics and tenant workflows (Fahmy et al., 2015; Eguro & Venkatesan, 2012).

This article aims to bridge that gap by producing a theory-driven, methodologically explicit, and practically oriented framework for secure FPGA cloud services. The goals are to: (1) synthesize the threat taxonomy and attack models distilled from empirical and survey literature; (2) present a descriptive methodology for integrating proof obligations and licensing metadata into the FPGA provisioning lifecycle; (3) articulate runtime verification and monitoring constructs that address both overt and covert threats; (4) propose a layered isolation and virtualization architecture that aligns with practical cloud operations; and (5) discuss the trade-offs, limitations, and research agenda that emerge from this synthesis. Every step of the argument is tied to the provided references to ensure the work remains strictly grounded in the assigned literature.

Methodology

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 05 ISSUE 09 Pages: 92-102
OCLC – 1368736135

This section describes an entirely text-based, rigorous approach for combining formal and systems-level tools into an integrated framework for secure multi-tenant FPGA deployments. The methodology is descriptive and prescriptive: it explains how design-time artifacts, runtime assurances, and cloud orchestration policies can be combined to reduce risk, enforce licensing and pay-per-use models, and detect or prevent malicious behavior.

Framework overview and design principles. The proposed framework draws on four interlocking design principles gleaned from the literature: principle of provenance-aware reconfiguration, runtime verifiability, tenant isolation through virtualized FPGA abstractions, and licensing-aware IP protection.

1. Provenance-aware reconfiguration. Any dynamic reconfiguration of FPGA fabric in a cloud environment should be accompanied by cryptographically protected provenance metadata that asserts origin, integrity, and permitted usage semantics. Provenance metadata functions as an attestation that a bitstream or partial reconfiguration file was generated by an approved toolchain or vendor and that it complies with declared licensing and resource constraints (Elrabaa et al., 2019; Drzevitzky, 2010). The literature demonstrates the need for provenance and controlled reconfiguration to prevent unauthorized IP extraction and tampering (Elrabaa et al., 2019; Drzevitzky, 2010).

2. Runtime verifiability and proof-carrying hardware. Design-time verification alone is insufficient in dynamic, multi-tenant clouds because reconfiguration can be requested at runtime. Proof-carrying hardware (PCH) provides a way to attach a verifiable, machine-checkable proof to a reconfiguration artifact, enabling a hypervisor or FPGA OS to validate safety and security properties at load time without re-running expensive formal analyses (Drzevitzky, 2010). PCH thereby supports an online assurance model where the device enforces policy before committing new logic to fabric (Drzevitzky, 2010).

3. Tenant isolation through virtualization. Virtualization techniques for FPGAs, when combined with operating-system level abstractions, can present clean isolation boundaries and allow efficient sharing while limiting resource-based interference (Fahmy et al., 2015; Fleming et al., 2014). The LEAP FPGA operating system exemplifies how an OS abstraction can mediate resource allocation and provide manageability functions relevant to security (Fleming et al., 2014).

4. Licensing-aware IP protection and monetization. For cloud providers to economically support FPGA services that host third-party intellectual property, there must be mechanisms for protection and enforceable pay-per-use licensing. The literature demonstrates concrete schemes for pay-per-use licensing and protection of FPGA IP as first-class objects in cloud ecosystems (Elrabaa et al., 2019). Such mechanisms must be integrated with reconfiguration provenance and runtime verification to be effective.

Operational methodology: four-phase lifecycle. To operationalize the design principles, the framework specifies a lifecycle that maps to the sequence of actions typically taken by cloud tenants and operators: upload and attest, verify and sandbox, deploy and monitor, audit and revoke. Each phase is described below with

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 05 ISSUE 09 Pages: 92-102
OCLC – 1368736135

attendant technical artifacts and operational steps and is tied to specific literature insights.

1. Upload and attest. A tenant or IP vendor prepares a design and associated metadata. The metadata includes cryptographic signatures, provenance information (toolchain, originator, permitted operations), and, if applicable, a proof artifact (PCH) asserting properties such as non-violation of resource constraints, absence of illegal bitstream patterns, or compliance with performance and timing envelopes. The need for provenance and attestation is reinforced by protection and licensing literature, which illustrates how policy metadata must accompany IP delivered to cloud services (Elrabaa et al., 2019; Drzevitzky, 2010).

2. Verify and sandbox. The cloud platform receives the artifact and first checks cryptographic integrity and provenance assertions. If a PCH is present, it is validated by a verifier that runs in a trusted execution context in the FPGA management stack or hypervisor (Drzevitzky, 2010). If verification succeeds, the platform instantiates the design in a guarded sandbox that limits access to sensitive on-chip resources and communication channels. The sandboxing concept is grounded in virtualization and OS-level mediation literature (Fahmy et al., 2015; Fleming et al., 2014).

3. Deploy and monitor. After sandboxed instantiation, the platform must monitor runtime telemetry, enforce resource quotas, and perform active detection of anomalous behavior such as unauthorized reconfiguration attempts, network-based command-and-control, or patterns indicative of covert channels. Monitoring draws on methods for botnet detection via hardware isolation and dynamic detection of distributed attacks described in the literature (Bobda et al., 2015; Agarwal et al., 2018). The monitoring tier must be sensitive to the particularities of FPGA-induced side channels and covert inter-tenant leakage (Giechaskiel et al., 2019).

4. Audit and revoke. All instantiation events, runtime traces, and verification outcomes are recorded in an auditable ledger (which may be internal to the provider) to enable later forensic analysis and license enforcement. If anomalies are detected or if license terms are violated, the provider can revoke access, roll back partial reconfigurations, or suspend service. The pay-per-use licensing literature emphasizes the need for auditable accounting to implement business models and to disincentivize IP theft (Elrabaa et al., 2019).

Technical primitives and implementation notes. The methodology relies on a set of technical primitives that can be implemented at different layers of the stack. These primitives include cryptographic provenance tokens, proof-carrying hardware verifiers, an FPGA OS with resource isolation hooks, telemetry collectors tuned to detect FPGA-specific anomalies, and licensing enforcement modules integrated into orchestration workflows. Each primitive draws directly from one or more references: PCH and runtime verification (Drzevitzky, 2010), licensing and pay-per-use (Elrabaa et al., 2019), virtualization strategies and OS-level mediation (Fahmy et al., 2015; Fleming et al., 2014), and detection mechanisms for malicious uses (Bobda et al., 2015).

Threat modeling and risk prioritization. The methodology includes a threat modeling step that must be performed for any deployment. The threat

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 05 ISSUE 09 Pages: 92-102
OCLC – 1368736135

ISSN-2750-1396

model categorizes risk by impact and likelihood, informed by the attack taxonomy in Duan et al. (2021) and empirical covert-channel studies (Giechaskiel et al., 2019). The model explicitly recognizes high-priority threats: (a) unauthorized IP extraction and licensing circumvention (Elrabaa et al., 2019), (b) covert channels enabling cross-tenant leakage (Giechaskiel et al., 2019), (c) runtime compromise via malware or botnet exploitation (Bobda et al., 2015), and (d) denial-of-service both at the network and fabric level (Agarwal et al., 2018). Each threat must be mapped to mitigation controls described in the framework.

Evaluation methodology. Because this work synthesizes methods rather than presenting a single empirical experiment, the evaluation is descriptive: for each identified attack class, the methodology articulates how the proposed primitives combine to reduce risk, which properties are verifiable at load time, and which threats require continuous monitoring. The evaluation draws on attack characterizations and countermeasure studies in the literature to assess coverage and residual risk (Duan et al., 2021; Giechaskiel et al., 2019; Elrabaa et al., 2019).

Ethical and economic constraints. The methodology acknowledges provider and tenant incentives, and the adoptability constraints of cloud operations. Licensing and pay-per-use models impose economic constraints on enforcement mechanisms; the literature demonstrates concrete schemes that balance protection with usability (Elrabaa et al., 2019). The framework therefore includes guidance on integrating lightweight verification for rapid provisioning and stronger checks for high-assurance tenants.

Results

The results section offers a comprehensive, descriptive analysis of how the lifecycle and primitives described in the methodology address the principal attacks and vulnerabilities identified across the provided references. Because this article synthesizes existing literature rather than reporting new experimental data, the "results" are convergent, analytical findings that map threats to mitigations and explicate residual risks and trade-offs.

Mapping of attack classes to mitigations

1. IP theft and licensing circumvention. Problem: In cloud environments, bitstreams or partial bitstreams carrying proprietary designs can be intercepted, copied, or reverse engineered. Such leakage undermines commercial models and disincentivizes third-party IP deployment in clouds (Elrabaa et al., 2019). Mitigation: The proposed framework combines cryptographic provenance, runtime verification (PCH), and pay-per-use licensing enforcement to protect IP. Specifically, provenance tokens ensure that only audited bitstreams are accepted; PCH proves properties that prevent unauthorized copying (by binding proofs to specific resource contexts); and licensing modules meter usage, enabling pay-per-use billing and revocation mechanisms to disable instances that violate terms (Elrabaa et al., 2019; Drzevitzky, 2010). Analysis: This multi-factor approach increases the cost of theft by adding both technical and economic barriers. However, it does not fully eliminate the possibility of sophisticated side-channel recovery; thus, provenance and encryption must be complemented by obfuscation and runtime monitoring.

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 05 ISSUE 09 Pages: 92-102
OCLC – 1368736135

2. Covert channels and cross-tenant leakage. Problem: Covert channels exploit shared microarchitectural or physical resources to leak information between tenants co-located on the same device. Cloud FPGA fabrics, particularly multi-SLR or multi-tenant boards, present tangible opportunities for such channels (Giechaskiel et al., 2019). Mitigation: The framework recommends a combination of physical partitioning when possible, logical isolation through FPGA virtualization, and runtime detection of anomalous resource usage patterns. Proof-carrying hardware contributes by asserting that a tenant's design does not include unauthorized communication primitives or suspicious resource accesses. Hardware operating systems can enforce constraints on interconnect use and clock domains to limit channel capacity (Fleming et al., 2014; Giechaskiel et al., 2019). Analysis: These mitigations reduce channel bandwidth and detect opportunistic channels, but covert channels are inherently difficult to eliminate entirely. The framework therefore treats residual covert-channel risk as a matter for risk management and accounting: for particularly sensitive tenants, the provider may offer physically isolated FPGAs.

3. Runtime compromise via botnets and malware. Problem: Malicious actors can repurpose reconfigurable logic to execute irregular workloads, conduct network attacks, or form part of botnet infrastructures that exploit hardware-level access (Bobda et al., 2015). Mitigation: The framework combines PCH-based verification to prevent loading of designs exhibiting known malicious signatures, telemetry-based anomaly detection, and hardware isolation techniques that can intercept or block abnormal network flows. Bobda and colleagues demonstrate the viability of hardware isolation mechanisms specifically designed for botnet detection, representing an architectural pattern that can be integrated into the proposed framework (Bobda et al., 2015). Analysis: While detection and isolation reduce operational risk, adversaries may evolve tactics; ongoing updates to signature sets and behavioral baselines are therefore necessary. PCH helps by enabling a baseline of "authorized behaviors" that must be demonstrated at load time.

4. Denial-of-service and resource exhaustion. Problem: Resource exhaustion can occur via malicious or misbehaving tenants that saturate shared communication links, compute fabric, or I/O, degrading service for co-tenants (Agarwal et al., 2018). Mitigation: Virtualized FPGA resource schedulers and OS-level quotas mitigate DoS by enforcing resource budgets and allowing preemption. Provenance metadata and policy-aware orchestration enable the provider to reject or throttle tenants whose requested resource envelopes exceed contracted allowances (Fahmy et al., 2015; Fleming et al., 2014). Analysis: Enforcing strict quotas and preemptive throttling preserves availability at the possible expense of tenant flexibility; thus, the trade-off must be negotiated in service-level agreements.

5. Integrated analysis: effectiveness and residual risk. The synthesis shows that by combining provenance, proof-carrying hardware, virtualization, runtime monitoring, and licensing enforcement, the framework covers a broad swath of attack vectors identified in the literature. Each technique contributes a partial but complementary control: provenance and PCH primarily prevent

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 05 ISSUE 09 Pages: 92-102
OCLC – 1368736135

unauthorized or unsafe reconfiguration; virtualization and OS-level abstractions reduce resource-based interference; telemetry and detection address dynamic and behavioral threats; licensing enforces economic deterrents. The residual risks — particularly covert channels and sophisticated IP extraction via physical attacks — require either physical isolation or higher-assurance hardware roots of trust which are not fully realized in present-day commodity cloud FPGA platforms (Giechaskiel et al., 2019; Elrabaa et al., 2019).

Operational outcomes and deployment considerations. Implementation of the framework in a production cloud implies a set of operational changes: integration of verification steps into provisioning pipelines, additional storage and processing for provenance and proof verification, enhanced telemetry and anomaly detection infrastructure, and contractual changes for licensing and audited usage. These are non-trivial but feasible for providers, as demonstrated by virtualization and OS-level prototype work in the literature (Fahmy et al., 2015; Fleming et al., 2014). The economic case for adoption rests on the vendor ability to offer differentiated, high-assurance FPGA services with monetizable IP protection (Elrabaa et al., 2019).

Discussion

This discussion provides an in-depth interpretation of the synthesized findings, explores nuanced trade-offs, and identifies a research agenda that addresses limitations of current techniques. It integrates theoretical considerations with practical deployment concerns and articulates where the literature points to necessary future work.

Interpretation of synthesized results. The literature collectively suggests that no single technique suffices to secure multi-tenant FPGA clouds: security must be constructed as a composition of formal guarantees, operational controls, and economic incentives. Proof-carrying hardware provides a powerful mechanism for verifying properties at load time without re-running expensive analyses (Drzevitzky, 2010), but PCH requires an ecosystem of toolchain support, proof generation, and standardized proof languages to be usable at scale. Licensing and pay-per-use models protect commercial interests and create enforceable incentives, but their efficacy depends on integration with runtime mechanisms that can enforce revocation and meter consumption (Elrabaa et al., 2019).

Virtualization and OS-level mediation are crucial because they present manageable control points within cloud stacks (Fahmy et al., 2015; Fleming et al., 2014). The LEAP OS work demonstrates that OS abstractions can facilitate resource allocation and management suited to security needs, but it also underscores the complexity of implementing such systems without adding excessive overhead or attack surface.

Covert-channel research (Giechaskiel et al., 2019) highlights that the FPGA physical substrate opens unique channels that are not easily addressed by classical software isolation techniques alone. The field, therefore, requires research that explicitly models physical interactions in multi-tenant reconfigurable fabrics and develops both detection and structural mitigations (e.g., clock-domain partitioning, guarded SLR boundaries).

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 05 ISSUE 09 Pages: 92-102
OCLC – 1368736135

ISSN-2750-1396

Limitations of the proposed framework. Despite its comprehensiveness, the integrated framework has important limitations that must be recognized:

Toolchain and proof generation complexity. Generating machine-checkable proofs that capture the desired security properties is non-trivial, and the burden may fall on IP vendors or toolchains. The PCH paradigm requires mature automation to be widely adopted (Drzevitzky, 2010).

Performance and usability trade-offs. Adding verification, provenance checks, and telemetry increases provisioning latency and operational overhead. Providers must balance assurance with latency expectations and cost models; pushback from tenants expecting rapid provisioning could hinder adoption (Fahmy et al., 2015).

Residual covert-channel risk. Even after architectural mitigations, covert channels may persist at low bandwidths or via unexpected physical interactions — channel elimination in complex shared hardware may be infeasible without physical isolation (Giechaskiel et al., 2019).

Economic and policy constraints. Licensing enforcement and revocation mechanisms raise legal and contractual complexity. Pay-per-use models require precise, auditable accounting and may necessitate dispute-resolution mechanisms that are not purely technical (Elrabaa et al., 2019).

Research directions and priorities. The literature suggests a targeted set of research priorities to move from coherent theory to widespread practice:

Automated PCH toolchains and standardized proof formats. To make proof-carrying hardware practical, we need robust tooling that can automatically generate proofs for commonly desired security properties and express them in an interoperable format that verifiers can consume at runtime (Drzevitzky, 2010).

Covert-channel quantification and mitigations. Empirical work to measure channel capacities across diverse FPGA topologies and mitigation experiments that combine architectural partitioning with dynamic detection are essential (Giechaskiel et al., 2019).

Lightweight runtime verification primitives. Research into verifiers that provide strong guarantees with low latency is crucial for acceptable provisioning times. Techniques like incremental verification and hardware-assisted checking may be promising (Drzevitzky, 2010; Fleming et al., 2014).

Hybrid economic-technical enforcement models. Devising licensing schemes that are both enforceable and minimally intrusive requires interdisciplinary research combining technical enforcement (Elrabaa et al., 2019) with legal frameworks and auditable ledgers.

Operationalization of FPGA OS capabilities. Bringing concepts from FPGA operating systems like LEAP into production-grade cloud stacks, while managing performance overhead and attack surface, requires substantial systems engineering work (Fleming et al., 2014; Fahmy et al., 2015).

Privacy and regulatory considerations. As FPGAs increasingly process sensitive data, compliance regimes will intersect with technical designs. Research into privacy-preserving acceleration and selective attestation that satisfies regulatory needs is necessary.

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 05 ISSUE 09 Pages: 92-102
OCLC – 1368736135

Counter-arguments and nuanced analysis. A skeptical view might argue that the industry will underinvest in the complex tooling and operational changes required for the framework, instead favoring simpler isolation (e.g., physically dedicated FPGAs) that sidestep complex verification. This is a plausible near-term outcome; providers can indeed choose to segment high-assurance tenants onto dedicated hardware and serve mainstream customers on less protected shared fabrics. However, economic pressures and the demand for scale make physical isolation a limited solution: it reduces utilization and raises costs. The framework proposed here offers a path to improve security while retaining multiplexing benefits — a middle path that is likely to be attractive to providers who wish to offer differentiated services while maintaining high utilization (Eguro & Venkatesan, 2012; Elrabaa et al., 2019).

Another counter-argument stresses that PCH and provenance will become brittle as adversaries adapt, embedding malicious behaviors within apparently benign proofs or subverting verification environments. This risk underscores the need for defense-in-depth: PCH should not be the single point of reliance but part of a layered architecture that includes runtime monitoring, periodic attestation, and the ability to revoke or quarantine suspect tenants (Drzevitzky, 2010; Bobda et al., 2015).

Broader theoretical implications. The integration of formal verification concepts such as PCH with cloud orchestration and licensing suggests a new research frontier: the co-design of economic instruments and formal assurances. In such a paradigm, proofs are not merely correctness artifacts but also legal and commercial objects that can be traded, priced, and attested. This reframing elevates the role of formal methods from purely technical guarantees to foundational elements of market trust in reconfigurable-cloud ecosystems (Elrabaa et al., 2019; Drzevitzky, 2010).

Conclusion

This article synthesizes an integrated, provenance-aware, runtime-verifiable framework for securing multi-tenant FPGA platforms in cloud environments, strictly grounded in the provided literature. By combining proof-carrying hardware, cryptographic provenance, virtualization, FPGA operating system abstractions, runtime telemetry, and licensing-aware enforcement, the framework addresses a broad set of attacks including IP theft, covert-channel leakage, botnet exploitation, and resource-based denial-of-service. The literature shows that each technique contributes indispensable capabilities: PCH enables load-time assurance (Drzevitzky, 2010), provenance and licensing protect IP and economic models (Elrabaa et al., 2019), virtualization and OS abstractions reduce interference and provide manageability (Fahmy et al., 2015; Fleming et al., 2014), and detection and isolation techniques address active runtime threats (Bobda et al., 2015; Duan et al., 2021).

Nevertheless, the framework is not a panacea. Important gaps remain: the complexity of proof generation, residual covert channel risks, and operational overheads pose significant challenges. Future work must make PCH practical through automated toolchains, quantify and mitigate covert channels empirically, devise low-latency runtime verification primitives, and integrate economic and

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 05 ISSUE 09 Pages: 92-102
OCLC – 1368736135

legal mechanisms for effective licensing enforcement. Together, these activities will help make secure, auditable, and economically viable FPGA-as-a-service offerings a practical reality for cloud providers and tenants.

In closing, the references analyzed indicate both a clear problem space and concrete technical levers for progress. The pathway forward requires cross-disciplinary work — weaving together formal verification, systems engineering, security telemetry, and economic modeling — to align the technical reality of reconfigurable hardware with trustworthy cloud services (Duan et al., 2021; Elrabaa et al., 2019; Drzevitzky, 2010; Giechaskiel et al., 2019; Fahmy et al., 2015; Fleming et al., 2014; Bobda et al., 2015; Eguro & Venkatesan, 2012).

References

1. S. Drzevitzky. 2010. Proof-carrying hardware: Runtime formal verification for secure dynamic reconfiguration. In Proceedings of the International Conference on Field Programmable Logic and Applications, 255–258. DOI: https://doi.org/10.1109/FPL.2010.59
2. Shijin Duan, Wenhao Wang, Yukui Luo, and Xiaolin Xu. 2021. A survey of recent attacks and mitigation on FPGA systems. In Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI '21), 284–289. DOI: https://doi.org/10.1109/ISVLSI51109.2021.00059
3. Ken Eguro and Ramarathnam Venkatesan. 2012. FPGAs for trusted cloud computing. In Proceedings of the 22nd International Conference on Field Programmable Logic and Applications (FPL '12). DOI: https://doi.org/10.1109/FPL.2012.6339242
4. Muhammad E. S. Elrabaa, Mohamed A. Al-Asli, and Marwan H. Abu-Amara. 2019. A protection and pay-per-use licensing scheme for on-cloud FPGA circuit IPs. ACM Transactions on Reconfigurable Technology and Systems, 12, 3, Article 13 (August 2019), 19 pages. DOI: https://doi.org/10.1145/3329861
5. Suhaib A. Fahmy, Kizheppatt Vipin, and Shanker Shreejith. 2015. Virtualized FPGA accelerators for efficient cloud computing. In Proceedings of the IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom '15). IEEE, 430–435.
6. Christophe Bobda, Charles Kamhoua, Festus Hategekimana, Adil Tbatou and Kevin Kwiat. 2015. Hardware isolation technique for IRC-based botnets detection. In International Conference on ReConFigurable Computing and FPGAs (ReConFig '08). IEEE Computer Society, Cancun, Mexico.
7. Kermin Fleming, Hsin-Jung Yang, Michael Adler, and Joel S. Emer. 2014. The LEAP FPGA operating system. In Proceedings of the 24th International Conference on Field Programmable Logic and Applications (FPL), 1–8. DOI: https://doi.org/10.1109/FPL.2014.6927488
8. Ilias Giechaskiel, Vincent Immler, Dennis Schellekens, and Farinaz Koushanfar. 2019. Reading between the dies: Cross-SLR covert channels on multi-tenant cloud FPGAs. In Proceedings of the IEEE International Conference on Computer Design (ICCD '19). IEEE, 1–8.
9. V. Agarwal, N. Verma, S. Saha, and S. Kumar. 2018. Dynamic Detection and Prevention of

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 05 ISSUE 09 Pages: 92-102
OCLC – 1368736135

Denial of Service and Peer Attacks with IPAddress Processing. Recent Findings in Intelligent Computing Techniques: Proceedings of the 5th ICACNI 2017, Volume 1, 707, 139.

10. M. Mishra. 2017. Reliability-based Life Cycle Management of Corroding Pipelines via Optimization under Uncertainty (Doctoral dissertation).
11. V. Agarwal, N. Verma, & S. Kumar. 2018. Intelligent Decision Making Real-Time Automated System for Toll Payments. In Proceedings of International Conference on Recent Advancement on Computer and Communication: ICRAC 2017 (pp. 223-232). Springer Singapore.
12. H. Gadde. 2019. Integrating AI with Graph Databases for Complex Relationship Analysis. International.
13. H. Gadde. 2019. AI-Driven Schema Evolution and Management in Heterogeneous Databases. International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence, 10(1), 332-356.
14. Hariharan, R. 2025. Zero trust security in multi-tenant cloud environments. Journal of Information Systems Engineering and Management, 10.
15. H. Gadde. 2019. Exploring AI-Based Methods for Efficient Database Index Compression. Revista de Inteligencia Artificial en Medicina, 10(1), 397-432.
16. J. Han, M. Yu, Y. Bai, J. Yu, F. Jin, C. Li, ... & L. Li. 2020. Elevated CXorf67 expression in PFA ependymomas suppresses DNA repair and sensitizes to PARP inhibitors. Cancer Cell, 38(6), 844-856.
17. B. R. Maddireddy, & B. R. Maddireddy. 2020. Proactive Cyber Defense: Utilizing AI for Early Threat Detection and Risk Assessment. International Journal of Advanced Engineering Technologies and Innovations, 1(2), 64-83.
18. Vandana T. Goikar, Supriya K. Jagdale, Priya B. Parade, Sumedha D. Pawar. 2015. Improve Security of Data Access in Cloud Computing Using Location. IJCSMC, vol 4 issue 2, 1-10.
19. Tengfei Li, Liang Hu, Yan Li, Jianfeng Chu, Hongtu Li, and Hongying Han. 2015. The Research and Prospect of Secure Data Access Control in Cloud Storage Environment. Journal of Communications, 10(10), 1-7.
20. Manjinder Singh, Charanjit Singh. 2017. Multi-tenancy security in cloud computing. International Journal of Engineering Sciences & Research Technology, 4(116), 1-7.
21. Katie Wood and Mark Anderson. 2011. Understanding the complexity surrounding Multitenancy in cloud computing. Eighth IEEE International Conference on e-Business Engineering. DOI: https://doi.org/10.1109/ICEBE.2011.68.