



A Systematic Literature Review on Hardware Trojan Horse: Threat, Prevention, Detection Techniques, and Future Directions

Abdulkabir Adamu^{1*}, S. O. Subairu², M. D. Noel³, Ahmad Sulieman⁴, Ahmad Abubakr Yusuf⁵

^{*1}Department of Cybersecurity Science, Federal University of Technology, Minna, Niger State, Nigeria

²Department of Cybersecurity Science, Federal University of Technology, Minna, Niger State, Nigeria

³Department of Cybersecurity Science, Federal University of Technology, Minna, Niger State, Nigeria

⁴Department of Cybersecurity Science, Federal University of Technology, Minna, Niger State, Nigeria

⁵Department of Software and Web Development, Federal Polytechnic Bauchi, Bauchi State, Nigeria

ABSTRACT

Hardware Trojan horses pose a serious threat to the security and reliability of modern integrated circuits (ICs), as they can be covertly inserted at various stages of the IC life cycle including specification, design, fabrication, testing, assembly, and deployment and remain dormant until triggered. The increasing use of third-party intellectual property cores, outsourced fabrication, and global supply chains has significantly expanded the hardware attack surface. This study presents a systematic literature review of hardware Trojan threats, prevention strategies, insertion points, trigger mechanisms, payload behaviors, detection techniques, and future research directions, guided by five objectives: (i) identifying the major threats posed by hardware Trojans in modern hardware systems; (ii) examining the prevention strategies proposed across the IC life cycle; (iii) analyzing common insertion points, trigger mechanisms, and payload behaviors; (iv) identifying the major challenges and research gaps in existing prevention and detection approaches; and (v) highlighting future research directions for improving hardware Trojan detection and prevention. Using the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework, peer-reviewed studies published between 2014 and 2026 were systematically identified and analyzed, resulting in 55 articles included in the final review. The literature is categorized by Trojan behavior and detection methodology, including logic testing, side-channel analysis, run-time monitoring, and machine learning-based approaches. The review reveals sustained research growth from 2018 onward, with a peak around 2020. The findings show that, despite considerable progress, no single detection technique is capable of identifying all categories of hardware Trojans, because each approach exhibits limitations in detection coverage, scalability, computational complexity, implementation overhead, or susceptibility to stealthy Trojan designs. Furthermore, machine learning-based detection methods remain constrained by the absence of standardized benchmark datasets, limited real-world validation, inconsistent evaluation metrics, and poor generalization across hardware platforms. Other persistent challenges include the heavy concentration of research on post-fabrication detection, high area and power overheads of prevention techniques, limited protection along the global semiconductor supply chain, and the lack of integrated end-to-end defense frameworks spanning the entire IC life cycle. Based on these findings, the review recommends future research on standardized benchmarking frameworks and publicly available datasets, hybrid detection models that integrate structural, functional, and side-channel information, explainable machine learning techniques, lightweight real-time detection for resource-constrained devices, and end-to-end security frameworks that protect ICs throughout specification, design, fabrication, testing, assembly, deployment, and operation.

Keywords: Hardware Trojans; Integrated Circuit Security; Trojan Detection Techniques; Side-Channel Analysis

1. Introduction

Trojan horses have posed a significant threat to organizations and computer users for over a decade. According to (Ömer *et al.*, 2023), the most commonly reported security incidents are fraud, intrusion, and malicious code. Abuzaid *et al.*, (2023) defines malicious code as any code that is added, modified, or removed from a software system with the intention of causing harm or undermining the system's intended function. Although malicious code has been a longstanding problem, a number of recent high-profile attacks, coupled with evolving economic and technological trends, indicate that it is becoming an increasingly critical concern for industries, governments, and individuals.

Among the different types of malicious code, Trojan horses are particularly noteworthy and are the primary focus of this study. A Trojan horse is a malicious program that requires execution on a victim's computer. Once installed, it can remotely control the system and extract confidential information. Unlike worms or viruses, Trojans do not replicate themselves but instead focus on remote control and data theft (Ravichandran *et al.*, 2024).

In detecting malicious code, classification plays a crucial role in enhancing the effectiveness of detection systems. Malicious code can generally be categorized based on characteristics such as infection target, propagation technique, and other defining traits (Batista and Martín, 2020). Implementing an effective classification algorithm significantly improves detection accuracy. Classification methods are widely employed in analyzing malicious code, particularly when evaluating new or previously unknown samples (S. Li *et al.*, 2023).

The relative ease with which hardware Trojans can be introduced into contemporary integrated circuits (ICs) and electronic systems presents a significant security concern. Malicious hardware modifications may be inserted at any point throughout the IC design and production life cycle, including the specification, design, verification, and manufacturing phases. Furthermore, hardware Trojans can be retrofitted into existing ICs after fabrication, as demonstrated by (Xue *et al.*, 2020). Enforcing stringent control across the entire IC development pipeline is both complex and resource-intensive. Consequently, current industry practice increasingly favours the separation of design and manufacturing processes, with fabrication outsourced to a limited number of large-scale complementary metal–oxide–semiconductor (CMOS) foundries predominantly located in Asia and design activities frequently delegated to third-party contract design houses. The involvement of numerous organisations and individuals across this fragmented supply chain substantially increases the attack surface, creating multiple opportunities for the insertion of malicious circuitry (Theresa *et al.*, 2020). A hardware Trojan may manifest in various forms, ranging from subtle alterations in system specifications or the insertion of additional lines of Hardware Description Language (HDL) code, to deliberate modifications of the silicon layout during fabrication or fine-grained manipulation of CMOS geometries. This outsourcing paradigm extends beyond manufacturing alone; designers routinely rely on third-party Electronic Design Automation (EDA) tools and incorporate externally developed Intellectual Property (IP) cores, many of which are distributed solely as opaque binary net lists, further exacerbating trust and verification challenges (Dong *et al.*, 2020).

This systematic literature review aims to comprehensively analyze existing hardware Trojan horse threat, detection techniques, with particular emphasis on contemporary detection methodologies and tools. The systematic literature review will be guided by the following research questions?

- i. RQ1: What are the major threats posed by Hardware Trojan Horses in modern hardware systems?
- ii. RQ2: What Hardware Trojan prevention strategies have been proposed across the integrated circuit lifecycle?
- iii. RQ3: What are the common Hardware Trojan insertion points, trigger mechanisms, and payload behaviors?
- iv. RQ4: What are the major challenges and research gaps in existing Hardware Trojan prevention and detection approaches?
- v. RQ5: What future research directions have been proposed to improve Hardware Trojan detection and prevention?

2. Related Works

Trojan horse attacks are widely recognized as one of the most persistent and sophisticated threats within the cybersecurity landscape. Although the conceptualization of Trojan horses has evolved over time, the literature generally converges on defining them as malicious programs that masquerade as legitimate or benign applications while executing unauthorized activities once deployed on a victim's system. These activities typically include data exfiltration, credential theft, system manipulation, and remote command execution (Zhang, 2018). This dual-characteristic nature of deceptive appearance combined with concealed malicious functionality renders Trojan horses particularly challenging to detect and mitigate.

Early investigations into Trojan horse threats emerged in foundational studies such as (Xiao *et al.*, 2016), however, substantial research momentum materialized only in subsequent years, as reflected in later contributions (Perez, 2021). The progression of research demonstrates a shift from conceptual classification and taxonomy development toward more sophisticated detection methodologies. Nevertheless, the majority of existing studies remain concentrated on categorization frameworks and hardware-oriented detection mechanisms, revealing a relatively narrow analytical focus.

Detection strategies proposed in the literature exhibit methodological diversity, yet each presents inherent trade-offs. For example, (Ibrahim, 2014) introduced a timestamp-based data stream clustering algorithm designed to identify theft-oriented Trojan activities. By compressing communication data streams into clusters and extracting discriminative features, the approach achieved approximately 90% detection accuracy with a low false negative rate. While these results indicate methodological effectiveness within its defined scope, the model's applicability is constrained by its specialization in theft-related behaviours, thereby limiting its generalizability across broader Trojan variants and attack vectors.

Complementary approaches have emphasized static analysis techniques. Tang's detection framework, as discussed in Nuzhat *et al.*, (2021), leverages static attributes extracted from Portable Executable (PE) files and applies intelligent information processing techniques to classify malicious artifacts. Although experimental evaluation reported a detection pass rate of 63.90%, performance limitations were attributed to dataset size and diversity, underscoring the critical role of representative training data in improving model robustness.

In contrast, Kanaker *et al.*, (2018) adopted a data mining approach within a Windows-based environment, relying predominantly on dynamic analysis. Their findings suggest that detection accuracy improves as additional relevant features are incorporated, particularly when redundant dynamic instruction sequences are reduced. Despite reporting high levels of accuracy and efficiency, the approach exhibited limitations in identifying malicious code embedded in residual executable sections. The authors consequently recommended integrating static and dynamic analysis techniques to enhance detection coverage and robustness.

Synthesizing the reviewed literature reveals several recurring challenges. First, dataset quality and scale significantly influence detection performance, yet access to large, real-world network datasets remains limited. Second, methodological choices whether static, dynamic, or hybrid introduce trade-offs between detection accuracy, computational overhead, and real-time responsiveness. Third, feature selection strategies present a persistent optimization dilemma: while increasing feature dimensionality may enhance classification performance, it simultaneously escalates computational cost, storage requirements, and latency.

Collectively, these findings indicate that although substantial progress has been achieved in Trojan horse detection, unresolved challenges persist in balancing accuracy, scalability, and efficiency. Future research must therefore prioritize hybrid analytical frameworks, optimized feature engineering, and scalable dataset integration to support real-time, resource-efficient detection mechanisms adaptable to evolving Trojan behaviors.

Table 2.1: Summary of Related

S/N	Author(s) and Year	Trojan Threat Type	Detection Technique	Prevention / Countermeasure	Evaluation Metrics	Key Findings	Research Gaps / Future Direction
1	Sun and Jiao (2026)	Hardware tampering and signal-level Trojans	Power Delivery Network (PDN) and signal trace monitoring	Hardware-level monitoring frameworks	Detection accuracy, signal integrity analysis	Demonstrated the effectiveness of PDN-based tamper detection	Needs real-time deployment validation in large-scale ICs
2	Elshafie et al. (2025)	IoT hardware security threats	Survey-based analytical review	Secure hardware design recommendations	Comparative threat coverage	Identified emerging IoT hardware vulnerabilities	Limited experimental validation of proposed controls
3	Guo et al. (2024)	Risky code execution (software–hardware interaction threats)	Benchmark-based evaluation framework	Secure code generation	Risk detection rate for evaluating code agents' risks	Introduced a benchmark evaluation framework	Requires adaptation for hardware-specific Trojan detection
4	Ma (2024)	System-level hardware reliability threats	System modelling techniques	Grid-forming inverter resilience design	Performance stability metrics	Improved resilience modelling at the hardware level	Not directly focused on Trojan insertion threats
5	Rahman (2024)	Hardware–software co-attacks including Trojans	Survey of co-protection mechanisms	Integrated hardware–software defense strategies	Comparative security analysis	Highlighted the importance of co-protection mechanisms	Lack of standardized evaluation framework
6	Ravichandran et al. (2024)	Trojan Horse and DDoS-related threats	Comprehensive review of countermeasures	Multi-layered cybersecurity defenses	Threat mitigation effectiveness	Compared countermeasure strategies for modern attacks	Limited hardware-level experimental results
7	Sen and Ghosh (2024)	Side-channel attack-based Trojans	Circuit-level resilience techniques	Side-channel resistant circuit design	Leakage reduction, resilience rating	Provided circuit-level protection strategies	High implementation overhead not fully optimized
8	Abuzaid et al. (2023)	Trojan Horse malware classification	Machine learning-based classification	Automated threat classification models	Classification accuracy	Achieved high detection accuracy for Trojan variants	Dataset limitations and generalization concerns
9	Kuaban et al. (2023)	Battery depletion and energy-based attacks	Energy depletion modelling	IoT energy-aware defense mechanisms	Energy consumption metrics	Modeled the impact of battery depletion attacks on IoT devices	Focuses on energy attacks rather than logic Trojans
10	Li et al. (2023)	Malicious mining code (hardware exploitation)	Multi-feature fusion detection	Feature-based anomaly detection	Detection rate, false positives	Improved detection using combined feature sets	Scalability in heterogeneous hardware environments

Sun and Jiao (2026) focused on hardware tampering at the signal level in integrated circuits, demonstrating that monitoring the power delivery network (PDN) can effectively detect tampering. They emphasized the need for further testing in large-scale ICs to validate real-time detection capabilities. Elshafie *et al.* (2025) examined emerging hardware vulnerabilities in IoT devices and highlighted the importance of secure design practices. While their review outlined potential threats, the lack of experimental validation limits its practical applicability.

Guo *et al.* (2024) introduced a benchmark for evaluating risky code execution in hardware-software systems and proposed secure code generation practices. However, adapting the benchmark for hardware-specific Trojan detection remains a challenge.

Ma (2024) investigated system-level reliability in PV systems using modeling techniques to enhance resilience. Although effective for reliability analysis, this study did not directly address Trojan insertion. Rahman (2024) reviewed hardware-software co-attacks, emphasizing integrated defense mechanisms but noting the absence of standardized evaluation frameworks. Similarly, Ravichandran *et al.* (2024) examined Trojan horse and DDoS threats, proposing multi-layered mitigation strategies, while highlighting limited hardware-level experimental evidence.

Sen and Ghosh (2024) focused on side-channel Trojans, offering circuit-level resilience techniques and side-channel resistant designs. Despite improved protection, the implementation overhead remains high. Abuzaid *et al.* (2023) applied machine learning for Trojan classification, achieving high detection accuracy. Yet, dataset limitations reduce generalizability across different hardware platforms. Kuaban *et al.* (2023) studied energy-based attacks, such as battery depletion in IoT devices, demonstrating their impact on device longevity without addressing logic-level Trojans. Finally, Li *et al.* (2023) introduced a multi-feature fusion detection method for malicious mining code in hardware, improving detection accuracy, although scalability in heterogeneous hardware environments remains an open challenge.

Hu *et al.* (2021) examined hardware Trojans and trust issues in IoT systems, providing a comprehensive overview of potential threats and design-time countermeasures. Despite these insights, experimental validation across diverse IoT platforms was limited. Kulandaivel (2021) analyzed remote attack kill-chains in modern in-vehicle networks, identifying vulnerabilities to Trojan insertion and highlighting the importance of hardware-level defense strategies. Nuzhat *et al.* (2021) compared signature-based Trojan detection methods, demonstrating varying accuracy and reliability, though dataset limitations constrained their broader applicability. Perez (2021) reviewed malicious hardware affecting industrial systems, emphasizing real-world implications and multi-layered mitigation strategies, but empirical evidence from production environments remained sparse.

Taheri (2021) explored IoT hardware attacks and vulnerabilities in machine learning systems, proposing integrated HW-ML defense frameworks; however, practical deployment challenges persist. Wang (2021) focused on embedded memory Trojans, offering detection and memory-level protection techniques, though the evaluation on full-scale systems was limited. Batista and Martín (2020) modeled malware propagation in IoT sensors, simulating mitigation strategies but without validation on physical devices. Benjamin *et al.* (2020) investigated hardware IP vulnerabilities and post-deployment Trojan attacks, proposing detection and patching mechanisms, yet their scalability remains uncertain. Dong *et al.* (2020) surveyed hardware Trojans in chips, covering detection and prevention strategies, though experimental comparisons of methods were lacking. Finally, Kabakci (2020) reviewed Trojans in reconfigurable SoCs, assessing impacts and prevention strategies, but detailed evaluation on real hardware was not provided. Li and Ta (2022) proposed hardware-supported patching methods to address security bugs in hardware IP blocks, demonstrating efficient post-deployment mitigation, although large-scale implementation remains a challenge. Malekpour (2022) focused on mitigating Trojans in MPSoCs, providing design and runtime strategies to prevent attacks, but practical implementation across diverse systems was limited. Khodosevych and Sakib (2022) reviewed Trojans in asynchronous logic using NULL Convention Logic, offering design-level countermeasures; however, experimental validation is still needed. Kornaros (2022) investigated hardware-assisted machine learning for IoT security, proposing ML-based detection of Trojans in resource-constrained environments, though testing on real IoT devices was not performed.

Ömer *et al.* (2023) provided a comprehensive review of cybersecurity vulnerabilities, including hardware Trojan threats, emphasizing policy and hardware countermeasures, yet experimental studies on real systems were limited. Parhizkar *et al.* (2023) analyzed degradation and failure mechanisms in complex systems, identifying Trojan-induced reliability issues, but validation on operational systems was lacking. Abuzaid *et al.* (2023) applied machine learning for Trojan classification, achieving high detection accuracy while highlighting the need for broader datasets for generalization. Kuaban *et al.* (2023) modeled energy depletion attacks in IoT, demonstrating battery impact without addressing logic-level Trojans. Li *et al.* (2023) proposed multi-feature fusion detection for malicious mining code, improving detection accuracy, but scalability to heterogeneous hardware remains an open question. Finally, Hu *et al.* (2021) examined hardware Trojans in IoT systems, reviewing threats and trust issues, yet experimental validation across diverse IoT platforms was limited.

Markettos *et al.* (2019) analyzed vulnerabilities in operating system IOMMU protections, highlighting potential DMA-based Trojan attacks and recommending peripheral access restrictions. Shihab *et al.* (2019) focused on post-fabrication hardware Trojans and proposed design obfuscation through selective transistor-level programming, noting overhead and scalability concerns. Bhatia *et al.* (2018) surveyed DDoS and hardware Trojan attacks, emphasizing mitigation strategies, though experimental evaluation was limited. Hanif *et al.* (2018) reviewed power electronics device failures related to Trojan insertion, providing design-level prevention methods, but lacked large-scale testing. He *et al.* (2018) conducted a comprehensive survey of hardware Trojan threats and defenses, highlighting detection and countermeasure strategies while noting the absence of practical validation. Hidalgo-López (2018) presented a simple sensor circuit Trojan detection approach using digital readout monitoring, though the study's scope was limited to specific circuits. Kanaker *et al.* (2018) proposed a machine learning-based framework for detecting Trojan infections in cloud-based systems, emphasizing the need for scalability across larger deployments. Rooney and Seeam (2018) reviewed methods for Trojan creation and detection, pointing out gaps in experimental implementation. Vamja (2018) focused on sequential circuit Trojans, proposing reverse engineering of finite state machines for detection,

with limitations to sequential circuits. Zhang (2018) applied AI-assisted detection techniques for network security hardware Trojans, showing promise but with limited real-world validation.

Ghiani *et al.* (2017) explored trigger-action dependent Trojans and policy-based mitigation strategies, highlighting the need for testing on large-scale systems. Mahjabin *et al.* (2017) surveyed DDoS and Trojan attacks, analyzing mitigation approaches, though experimental evidence was lacking. Zou *et al.* (2017) investigated potential triggers for hardware Trojans and proposed design-time countermeasures, requiring practical verification. Everhart *et al.* (2016) examined hardware time-bomb Trojans, offering mitigation recommendations, but no experimental implementation was reported. Xiao *et al.* (2016) summarized lessons learned from a decade of Trojan research, providing guidance on detection and obfuscation strategies while noting the need for updated hardware validation. Sethumadhavan *et al.* (2015) proposed trustworthy hardware from untrusted components, presenting verification strategies, yet large-scale testing remained limited. Wu *et al.* (2015) introduced TPAD for IC Trojan prevention, emphasizing detection rate improvements, though implementation overhead was a concern. Ibrahim (2014) developed a machine learning model for Trojan detection, demonstrating detection accuracy, but further validation on larger IC datasets was necessary. Li and Ta (2022) and Malekpour (2022) provided solutions for hardware IP patching and MPSoC Trojan mitigation, respectively, with design and runtime strategies that require practical testing. Khodosevych and Sakib (2022) and Kornaros (2022) explored asynchronous logic and IoT Trojans with ML-based detection frameworks, highlighting the need for real-device implementation. Ömer *et al.* (2023), Parhizkar *et al.* (2023), and related 2023 studies presented comprehensive overviews of system vulnerabilities, complex system degradation, and malicious code detection, identifying gaps in experimental validation and scalability. Yaswanth *et al.* (2021) and other 2021–2020 studies demonstrated practical detection methods and mitigation strategies, though scaling to heterogeneous or complex hardware environments remains a challenge.

3.0 Methodology

This study employs the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework to guide the systematic literature review. PRISMA was chosen for its ability to provide a transparent, rigorous, and reproducible methodology for the identification, screening, and synthesis of relevant academic evidence. The table 3.1 summarizes the selection criteria applied to the retrieved literature.

Table 3.1 Selection criteria for the retrieved literature

Selection criteria	Scientific Database		Grey Literature
Inclusion criteria	Only peer-reviewed scientific research articles published in English were included in this study.		Industry reports and policy briefs written in English were also considered for inclusion.
	With time-frame restriction 2016-2026		With time frame restriction 2014-2026
Exclusion	Before import to the bibliography manager	Non-English papers published in conference proceedings, as well as articles lacking an abstract, were excluded from consideration.	Papers that discuss hardware security in general but do not specifically address the types of hardware Trojans detection and investigation methods were excluded.
	During abstract screening	Papers from disciplines outside the scope of this study were excluded.	
	During full text reading	Papers that do not specifically address the detection or investigation of hardware Trojans were excluded.	

4.0 Results and Discussion

The PRISMA flow diagram illustrates the systematic study selection process for this review on hardware trojan. A total of 385 records were initially identified from selected digital databases. After removing duplicates ($n = 220$), 165 records remained for screening. From these, 110 records were excluded based on relevance screening and predefined inclusion and exclusion criteria.

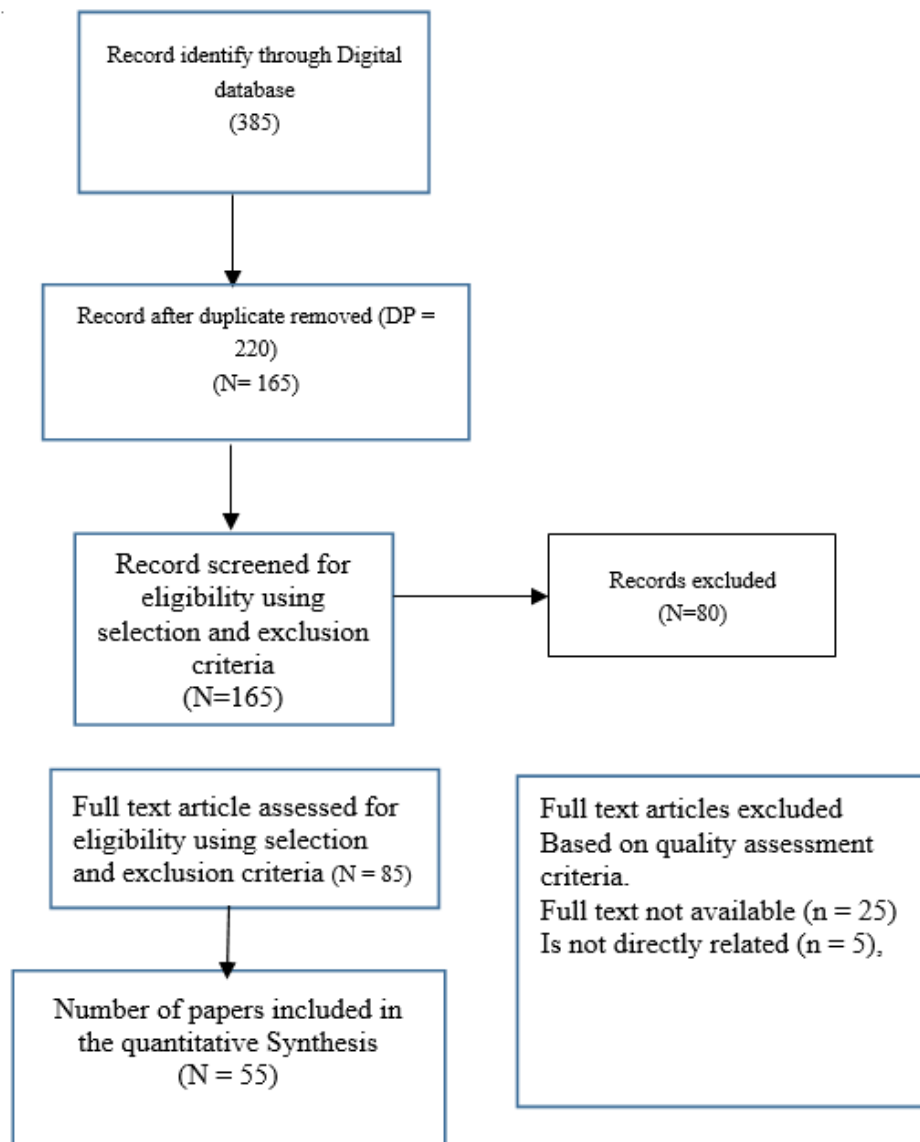


Figure 4.1: PRISMA SLR Block Diagram

Table 4.1 and figure 3.2 presents a systematic mapping of the article database selections. This category reflects the number of articles retrieved from each database included in the study.

Table 4.0 Systematic Mapping of Article Database

S/N	Journal Databases	Number of Articles
1.	ACM Library	10
2.	Elsevier	12
3.	Google Scholar	15
4.	IEEE Xplore	10
5.	Research Gate	5
6.	Science Direct	3
	TOTAL	55

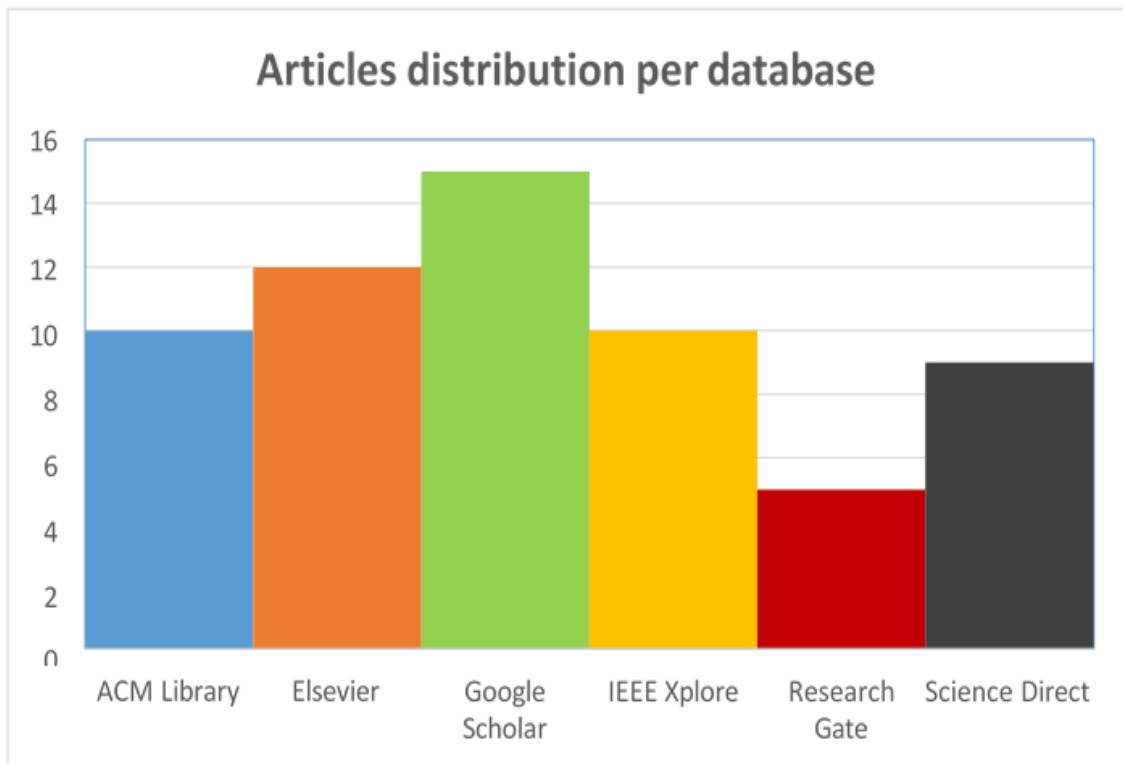
Distribution per Database**Figure 4.1: Total Journal articles distribution per database**

Table 4.2 and figure 4.2 shows that the highest number of articles were published in 2020, with notable publication activity also observed in 2018 and 2021. Overall, the period from 2018 to 2024 exhibits relatively consistent research output, indicating sustained interest in the topic. Consequently, a substantial proportion of the studies reviewed were published between 2018 and 2024, with 2020 contributing the largest share.

Table 4.2: No of article reviewed

Year	Number of Articles
2014	1
2015	2
2016	2
2017	3
2018	8
2019	2
2020	9
2021	8
2022	6
2023	6
2024	6
2025	1
2026	1
Total	55

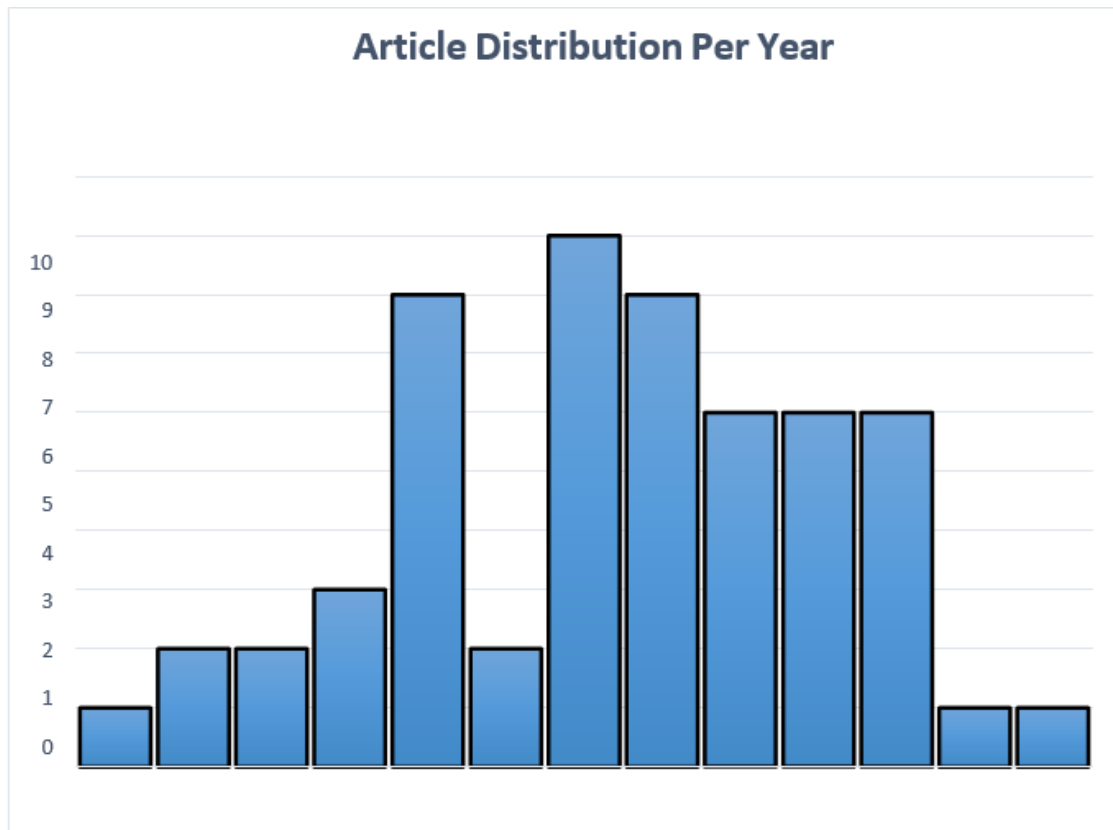


Figure 4.2: Article distributed per database

4.1 RQ1: What are the major threats posed by Hardware Trojan Horses in modern hardware systems?

Hardware Trojans represent one of the most insidious and structurally complex threats within contemporary hardware security landscapes. Owing to the multiplicity of insertion vectors and the diverse spectrum of low-level actions they may execute, their threat surface extends far beyond conventional software-based vulnerabilities (He *et al.*, 2018). Unlike transient software malware, hardware Trojans are inherently persistent; once embedded within an integrated circuit (IC) during design, fabrication, or supply chain stages, they remain operational throughout the hardware lifecycle. This persistence fundamentally differentiates hardware Trojans from higher-layer threats, as their effects can manifest whenever the system is powered, often without reliance on operating system-level compromise.

The severity of this threat lies not only in persistence but also in the capacity of hardware Trojans to covertly manipulate circuit functionality at the structural level, thereby eroding the foundational trust assumptions underpinning modern electronic systems. As noted by (Kornaros, 2022). The operational impact of hardware Trojans spans a wide continuum—from narrowly scoped manipulations to sophisticated architectural subversions that facilitate or amplify higher-layer software intrusions. Targeted attacks may include deliberate bit-flip operations compromising data integrity, subtle degradation of cryptographic primitives, timing manipulation, denial-of-service conditions, or covert exfiltration of sensitive information through side channels. More concerning, systems may harbour multiple Trojans whose aggregated effects produce compounded security degradation, thereby undermining the security assurances of otherwise trusted platforms (Hu *et al.*, 2021).

A synthesized analysis of the literature suggests that evaluating hardware Trojan risk requires a multidimensional framework that jointly considers insertion vectors, trigger mechanisms, payload characteristics, and activation conditions. Isolated analysis of payload effects alone provides an incomplete threat model, as the stealth and practical potency of a Trojan are closely tied to how and when it is activated. Insertion methodologies may range from malicious third-party intellectual property (IP) cores and untrusted fabrication facilities to supply chain tampering and insider threats. Similarly, activation mechanisms may be combinational, sequential, environmental, or externally triggered, each contributing to detection complexity. The co-evolution of insertion sophistication and trigger stealthiness has consequently intensified the detection challenge, necessitating increasingly advanced prevention, runtime monitoring, and post-silicon verification strategies (Rahman, 2024).

To systematically analyze these threats, the literature has proposed multiple hardware Trojan taxonomies aimed at classifying structural and behavioral characteristics. Such taxonomies serve several critical purposes: (i) enabling standardized characterization of Trojan architectures, (ii) guiding the development of class-specific detection and mitigation techniques, and (iii) supporting comparative benchmarking across defensive frameworks. Taxonomic dimensions commonly encompass physical versus functional Trojans, insertion phase (design-time versus fabrication-time), trigger type, payload behavior, and spatial footprint. By structuring the threat landscape in this manner, researchers can transition from reactive detection paradigms toward proactive and architecture-aware defense mechanisms.

However, despite notable progress in taxonomic refinement, several unresolved challenges persist. First, there remains limited consensus on standardized evaluation benchmarks for hardware Trojan detection, thereby constraining cross-study comparability. Second, the increasing integration density of modern ICs complicates structural anomaly detection due to

shrinking signal-to-noise ratios. Third, emerging fabrication outsourcing trends exacerbate trust boundary ambiguities within global semiconductor supply chains. These structural realities underscore the need for integrated security-by-design methodologies, supply chain transparency frameworks, and runtime attestation mechanisms capable of addressing Trojan threats holistically rather than in isolation. Figure 4.3 shows the taxonomy of Hardware Trojan.

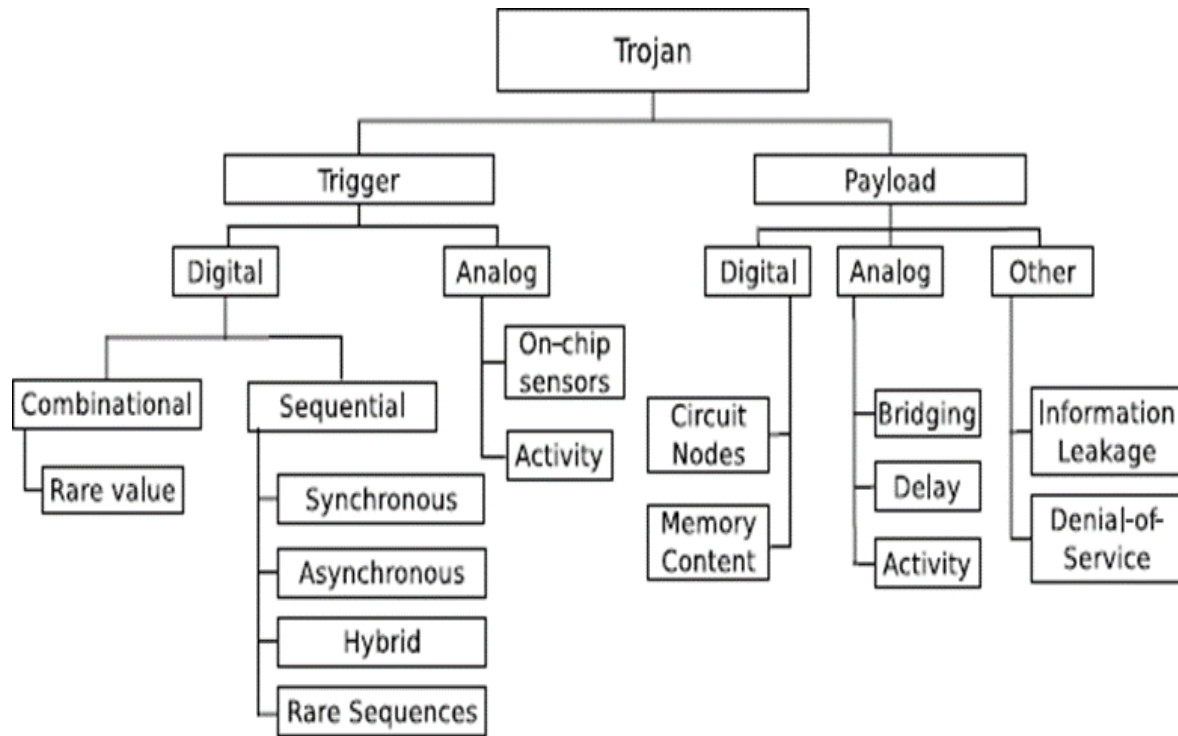


Figure 4.3: Taxonomy of Hardware Trojan (Prinetto and Roascio, 2020)

A notable limitation in existing hardware Trojan literature is the relatively narrow focus on structural detection and insertion analysis, with comparatively limited exploration of how Trojans interact dynamically with their operational environment. This gap is significant because the security risk associated with hardware Trojans is not determined solely by their physical presence within integrated circuits but also by their capacity to influence system behaviour, data integrity, and functional workflows across hardware software boundaries.

Empirical efforts to address this limitation include the adversarial design experiment conducted by the Polytechnic Institute of New York University in 2010, which provided participants with reconfigurable integrated circuit source code and required the intentional embedding of hardware Trojans capable of covert data leakage. This experimental challenge represented an early attempt to evaluate Trojan functionality under controlled adversarial design conditions, demonstrating that Trojan effectiveness is strongly associated with stealthy operational integration rather than mere structural insertion.

Similarly, the study by Wu *et al.*, (2015) extended Trojan threat analysis by examining payload behaviour across different integrated circuit contexts. Their work demonstrated that Trojan impact is architecture-dependent, as payload execution within central processing units, storage controllers, network interfaces, or memory buses produces distinct operational consequences. For example, payloads targeting processing units may manipulate instruction execution or privilege enforcement mechanisms, whereas Trojans embedded within communication or storage subsystems may facilitate covert information leakage or persistent data corruption.

Synthesizing these contributions, the literature suggests that the defining security risk of hardware Trojans lies in their functional behaviour rather than their physical embedding alone. Trojan payloads may manifest as data manipulation, cryptographic degradation, system availability disruption, or unauthorized information transmission. However, current research predominantly emphasizes detection of structural anomalies and insertion patterns, with insufficient focus on behavioural modelling of Trojan-system interaction dynamics.

This methodological imbalance highlights an emerging research gap in environment-aware Trojan analysis. Future research must therefore prioritize behaviour-centric threat characterization frameworks capable of capturing cross-layer interactions among hardware components, firmware processes, and higher-level software operations. Such frameworks should integrate structural verification with runtime behavioural monitoring to improve detection robustness against adaptive and stealth-optimized Trojan designs.

4.2 RQ2: What are the common Hardware Trojan insertion points, trigger mechanisms, and payload behaviors?

The development of an integrated circuit (IC) follows a structured lifecycle comprising specification, design, fabrication, testing, and assembly stages (Tsang *et al.*, 2022). The literature consistently emphasizes that each of these stages introduces distinct security exposures that may be exploited for Hardware Trojan insertion. Consequently, hardware security cannot be decoupled from lifecycle governance, as Trojan risk is inherently distributed across multiple production phases rather than confined to fabrication alone.

Within the integrated circuit (IC) lifecycle, the fabrication phase represents a structurally critical stage in which mask sets are generated, silicon wafers are manufactured, and preliminary probing is conducted to verify functional and physical integrity. Subsequent processes including wafer dicing, packaging, and final testing precede deployment and operational monitoring. While these stages are traditionally associated with quality assurance and production validation, the literature recognizes fabrication and post-fabrication handling as high-risk phases for Hardware Trojan insertion due to their direct influence on physical circuitry.

Benjamin *et al.* (2020) argue that specification, package testing, and deployment/monitoring stages are comparatively less vulnerable to Trojan insertion. This assessment is grounded in the assumption that early requirement definition and late-stage validation involve limited direct modification of structural circuitry. However, such conclusions must be interpreted within the broader context of evolving supply-chain dynamics.

However, all remaining stages of the IC lifecycle are considered susceptible to security attacks in practice, largely due to the increasing dependence on third-party vendors for electronic design automation (EDA) tools, intellectual property (IP) cores, fabrication services, and testing facilities. Moreover, it can be contended that, in realistic supply-chain environments, even the traditionally trusted stages may be indirectly influenced by untrusted entities for example, through Trojan retrofitting during testing or post-fabrication handling. Consequently, the entire IC design and manufacturing cycle must be critically examined when developing effective Hardware Trojan prevention and detection strategies (Sun and Jiao, 2026).

A Hardware Trojan can be introduced into an information system at various locations within the hardware architecture. Such insertion is not necessarily confined to a single component; rather, a Trojan may be distributed across multiple elements, including the processor, memory, input/output (I/O) interfaces, power supply, or clock distribution network (Kabakci, 2020). The specific location of a Hardware Trojan significantly affects the complexity of its design, the difficulty associated with its insertion, and the nature of its impact on overall system behaviour.

Different insertion points enable distinct Trojan functionalities, ranging from information leakage and performance degradation to system manipulation and denial-of-service conditions. Accordingly, the focus is to examine the potential actions and system-level effects resulting from Hardware Trojan insertion at various locations, as well as to present representative real-world examples that illustrate the severity and practical implications of Hardware Trojan threats (Sonia *et al.*, 2023). Figure 4.4 depicts vulnerable phase of IC development cycle.

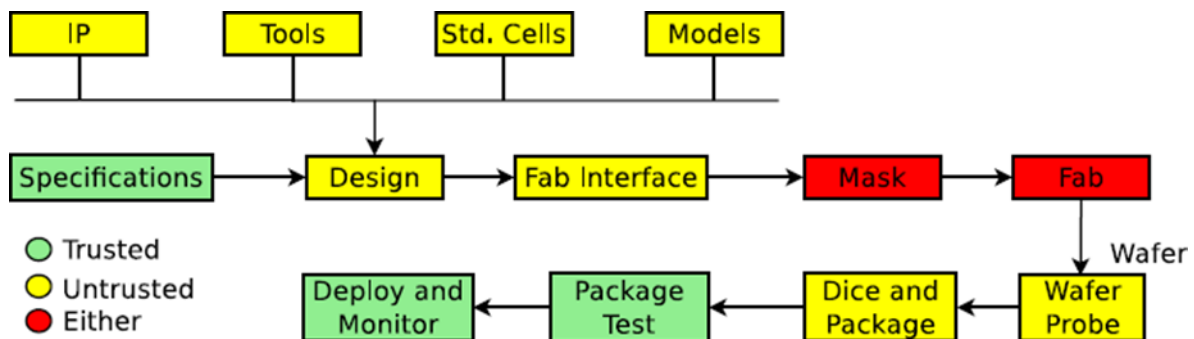


Figure 4.4: Vulnerable phase of IC development cycle (Sonia *et al.*, 2023)

4.2.1 Hardware Trojan actions

Hardware Trojans represent a relatively recent and significant security threat, extending the attack surface of information systems beyond the traditional focus on software-based vulnerabilities. Because most software security mechanisms are designed under the assumption that the underlying hardware is trustworthy, the presence of malicious hardware can undermine or entirely bypass these protections, thereby introducing substantial challenges to achieving comprehensive system security.

Hardware Trojans may be implemented through physical modifications to application-specific integrated circuits (ASICs), commercial off-the-shelf (COTS) components, microprocessors, and digital signal processors (DSPs), or through malicious firmware alterations, such as modifications to Field Programmable Gate Array (FPGA) bit-streams (Xiao *et al.*, 2016). Owing to their low-level insertion within the hardware stack, Hardware Trojans are capable of executing a wide range of malicious actions. These actions are commonly classified into four primary categories: functionality modification, specification alteration, information leakage, and denial-of-service attacks (Mahjabin *et al.*, 2017). A single Hardware Trojan implementation may be designed to carry out one or multiple of these actions, further complicating detection and mitigation efforts.

4.2.2 Modify functionality

A hardware Trojan that alters a device's intended functionality either by introducing additional logic or by removing, bypassing, or modifying existing circuitry poses a direct threat to the integrity of an information system. Such alterations may result in data corruption, manipulation of computational processes, or disruption of communication channels. The range of possible functional modifications is effectively unbounded, limited only by the attacker's technical capabilities, resources, and intent.

Jin *et al.* (2024) demonstrate this threat by presenting a scenario in which a relatively simple yet highly destructive Trojan introduces a fault into the Chinese Remainder Theorem (CRT) inversion step of an RSA signature computation, ultimately leading to the compromise of the RSA private key. In a related example, (Guo *et al.*, 2024) describe a functional modification that causes an error detection module to accept inputs that should otherwise be rejected.

Furthermore, it is conceivable that known integrated circuit faults, such as the Intel FDIV bug, could be intentionally reproduced by a hardware Trojan and selectively triggered to evade detection. (Guo *et al.*, 2024) further illustrate how such Trojans may be designed to alter the execution order of CPU instructions, leak sensitive information through side-channel effects, or modify the contents of programmable read-only memory, thereby introducing significant integrity vulnerabilities. Beyond isolated attacks, functional modifications can also facilitate more generalized exploits. (Kulandaivel, 2021) argue that focusing on a single hard-coded hardware attack underestimates the true power of malicious circuitry and present a modified CPU architecture capable of supporting a wide range of software-level attacks, including unauthorized memory access and firmware manipulation that enable privilege escalation, backdoor authentication, and password theft.

4.2.3 Modify specification

The modify specification class refers to hardware Trojans that compromise a target integrated circuit (IC) by altering its parametric characteristics or non-functional specifications. Affected parameters may include clock frequency, timing behavior, and power consumption. Such attacks are typically realized by manipulating intrinsic IC properties, including wire and transistor geometries. Unlike the modify functionality class, Trojans in this category originally described by (Parhizkar *et al.*, 2023) operate by modifying only existing wires and transistors. As a result, their disruptive effects are generally constrained to behaviors that progressively

degrade system reliability or drive the system toward failure. Nevertheless, it can be hypothesized that additive Trojan hardware could be coupled with existing circuitry in a manner that similarly influences intrinsic IC properties while enabling more sophisticated triggering mechanisms and attack behaviors.

Hardware Trojans in this class are capable of executing a range of attacks, such as reducing system performance by manipulating the system clock or replacing computational and input/output units with functionally equivalent components that exhibit lower throughput (Xue *et al.*, 2020). Additional specification-level modifications that may impact performance include changes in gate placement and routing, the substitution of functionally equivalent circuit implementations, or the introduction of extraneous passive components. These performance degradations may be implemented using a load-dependent approach, whereby timing errors are intentionally induced under high system activity, leading to reduced performance or system instability. (Sen and Ghosh, 2024) present circuit-level examples of such attacks, including the introduction of a bridging fault through resistor insertion and the intentional increase of net delay by adding capacitive load.

4.2.4 Leak information

This class of hardware Trojan comprises malicious modifications designed to covertly transmit sensitive information from an information system to an adversary without the knowledge or consent of either the system or its user. Data exfiltration may be achieved through existing internal or external communication pathways, or alternatively through the exploitation of side-channel leakage mechanisms. Malekpour, (2022) highlight several such avenues, including radio-frequency, optical, thermal, power, and timing side-channels, as well as standard hardware interfaces such as RS-232 and JTAG.

4.2.5 Denial of service

Low-level hardware modifications enable a wide range of denial-of-service (DoS) attacks, spanning from partial service degradation to the complete and irreversible disabling of a device through the introduction of a so-called "kill switch" (Bhatia *et al.*, 2018). (Elshafie *et al.*, 2025) describe hardware Trojans that disrupt system availability by exhausting limited resources such as communication bandwidth, computational capacity, and battery power, noting that the resulting physical effects may be either temporary or permanent in nature.

A hardware Trojan may be designed to accelerate battery depletion by preventing circuits from entering low-power or sleep states (Kuaban *et al.*, 2023), or by inserting additional buffering elements into IC interconnects, thereby increasing power consumption (Okkerman, 2020). Such mechanisms effectively reduce the operational lifetime of a device between charging cycles. In addition, Trojans may manipulate critical control signals, such as memory write-enable lines, to overwrite valid data with random values, thereby causing service disruptions, or selectively disable partial or complete power delivery to the system (Okkerman, 2020).

Service degradation may also be achieved through mechanisms that induce premature hardware failure. Chowdhury *et al.* (2021) present a circuit-level example that generates excessive switching activity, accelerating the aging of an integrated circuit and shortening device lifespan without directly altering functional behavior. Similarly, (Hanif *et al.*, 2018) suggest that deliberate modifications to material composition could increase electron migration in critical structures, such as power distribution networks and clock trees, thereby further accelerating hardware failure.

4.3 Trigger Mechanisms

Hardware Trojans may be introduced into a design through numerous insertion points and can give rise to a wide range of direct and indirect threats, once embedded within a system, most

hardware Trojans remain dormant until they are activated also referred to as being triggered to carry out malicious behaviour. Activation may occur through a variety of mechanisms, which may be overt or covert, random or deterministic, and can be either externally induced or internally predetermined. In essence, a trigger is any event or condition that causes a change in the state or behavior of a Trojan (Zou *et al.*, 2017).

The activation phase is particularly significant, as it provides a potential avenue for the detection and mitigation of hardware Trojans. During various stages of integrated circuit (IC) verification, efforts can be made to deliberately activate Trojans, typically through functional validation testing or state-space exploration that examines the design's inputs, outputs, and internal logic. Successfully triggering a Trojan during testing can reveal its presence within the design.

4.3.1 Internally triggered

Internally triggered hardware Trojans are activated when a specific internal state of the target device is reached. The most common activation mechanisms in this category are combinational and sequential triggering.

4.3.1.1 Combinational activation

A hardware Trojan may be activated when a specific combination of values is simultaneously observed at selected internal circuit nodes, collectively forming a trigger state. This form of activation can be implemented entirely using combinational logic. (Rooney and Seeam, 2018) refer to this mechanism as a "single-shot cheat code" and illustrate it using the example of a particular bus address, such as 0xdecabf, that activates the Trojan. In practice, however, combinational triggers often involve far more complex conditions, requiring multiple internal signals to assume specific values concurrently. For example, activation may depend on a particular configuration of internal registers combined with specific data and address values present on system buses.

Taheri, (2021) further describe the use of carefully crafted input patterns to trigger hardware Trojans, including combinations of data, control, address, and built-in self-test signals. Similarly, (Yaswanth *et al.*, 2021) employed several combinational trigger mechanisms in their design for Embedded Systems Challenge, such as activating the Trojan upon detection of an input buffer overflow or whenever an encryption key was modified.

4.3.1.2 Sequential activation

Sequentially triggered hardware Trojans are activated only after a specific sequence of events has occurred. Compared with combinational activation mechanisms, sequential triggers offer a significantly larger state space, as they are typically implemented using finite state machines. Vamja, (2018) note that such sequences often involve rare logic values at internal circuit nodes and, due to the increased logic depth inherent in state-machine-based designs, are considerably more difficult to detect during integrated circuit testing and verification.

The simplest form of a sequential trigger is a synchronous hardware counter that activates the Trojan after a predetermined number of clock cycles. Everhart *et al.* (2016) describe this mechanism as a "ticking time bomb." Extending this concept, (Khodosevych and Sakib, 2022) discuss asynchronous sequence counters that advance in response to specific events, such as a rising transition at the output of a logic gate. They further propose hybrid activation mechanisms that combine synchronous and asynchronous triggering, thereby increasing the complexity and stealth of the Trojan activation process.

Sethumadhavan *et al.* (2015) refer to such mechanisms as "sequence cheat codes" and illustrate their operation using an example in which the byte sequence 0x0d, 0x0e, 0x0c, 0x0a, 0x0f, 0x0b, 0x0a, and 0x0d, arriving over eight distinct clock cycles, activates a hardware Trojan. These values need not occur in consecutive cycles; rather, a sufficiently patient Trojan may continuously monitor inputs and internal states to detect far more complex and sparsely distributed event sequences.

The complexity of a sequential trigger is largely determined by the Trojan designer, with important trade-offs in terms of power consumption and the amount of additional logic required for implementation. Beyond purely digital mechanisms, internally triggered Trojans may also exploit physical or analog properties of an integrated circuit. Similarly, Hidalgo-lópez, (2018) describe an analog trigger mechanism in which additional capacitance is introduced and charged through a resistor. The charging behavior of the capacitor depends on the activity of surrounding logic, which can itself be influenced by orchestrating specific circuit operations. The Trojan is activated once the capacitor voltage reaches a predefined threshold. The broader distinction between digital and analog trigger mechanisms is further examined by, (Hidalgo-lópez, 2018)

Rooney and Seeam, (2018) employed multiple independent sequential triggers to activate different hardware Trojans. These triggers included detecting the keyboard input string “*New Haven*”, identifying the key code associated with the F12 key, recognizing the pattern “*moscow*” within input data, detecting the key code for the CAPS LOCK key, and counting a predetermined number of characters transmitted through an RS-232 interface.

Ma, (2024) investigated a content- and timing-based trigger mechanism, in which Trojan activation occurs only when specific data values are observed at precisely defined time instants. Notably, even for a relatively small design, they demonstrate that the time required to reliably activate such a trigger during testing is on the order of 3×10^{35} years. Their implementation combines the detection of particular keyboard key codes with strict temporal constraints between events. The author further propose a thermal trigger mechanism, wherein carefully selected input patterns generate sustained switching activity that drives an inverter-based ring oscillator, producing localized heating. A second ring oscillator is then used to sense the resulting temperature-induced delay variations and activate the hardware Trojan.

4.4 Externally Triggered

External triggers rely on interactions with elements outside the system in which the target device is embedded. The primary advantage of external triggers, as noted by (Ghiani *et al.*, 2017), is that activation can occur at any time from a source independent of the target device. They provide examples such as embedding a receiver or antenna within the device to facilitate remote triggering. Similarly, Butt *et al.* (2022) highlight the use of on-chip sensors capable of monitoring environmental conditions, including temperature, voltage, electromagnetic interference (EMI), humidity, and altitude.

These types of triggers, often referred to as side-channel triggers, operate in a manner analogous to side-channel attacks, where information is extracted from a device without directly interfering with its normal operation. Side channels can also serve as a means to detect the presence of hardware Trojans. Other external triggers involve direct physical interaction with the target device. Marketos *et al.*, (2019) describe scenarios in which a user provides input through buttons or switches connected to the device. Activation may also originate from other externally connected components, such as a peripheral memory device. Wang, (2021) provide a specific example in which an unused external port is leveraged to communicate with an embedded hardware Trojan.

4.4 RQ3: What are the major challenges and research gaps in existing Hardware Trojan prevention and detection approaches?

Given the significant threats posed by hardware Trojans, one effective approach to mitigating their impact is to prevent their insertion at any stage of the integrated circuit (IC) development lifecycle. The most reliable method for achieving this is to maintain strict end-to-end control over the design and manufacturing process. A small, trusted design team using internally developed tools can specify an IC design with a high degree of confidence that it is free of malicious modifications. When such a design is fabricated at a trusted foundry managed by a limited team of vetted personnel, the resulting IC instances faithfully implement the intended design. Ensuring that assembly and deployment are handled exclusively by trusted individuals further reinforces confidence that the final product remains uncompromised.

However, maintaining such a complete chain of trust is impractical for most applications, including even high-grade military systems. The cost, both in time and financial resources, of developing all IC components entirely in-house is prohibitive. In the commercial sector, the widespread use of packaged intellectual property (IP) blocks analogous to software libraries has enabled rapid development of sophisticated electronic products, such as smartphones. If the defense sector were to rely solely on in-house IP development, its capabilities would lag significantly behind commercial offerings. For certain specialized ICs, such as high-grade cryptographic chips with a relatively small number of logic gates, fully trusted development may be feasible. Even in such cases, however, untrusted supply chains could result in IC theft, reverse engineering, and the creation of modified or malicious copies.

Prevention thus represents the first and most critical layer in a defense-in-depth strategy against hardware Trojans. Standard policies, procedures, and best practices such as restricting design and fabrication to trusted personnel and facilities form the foundation of this approach. In addition, research has explored novel techniques for preventing Trojan insertion at various stages of the IC lifecycle, including design, fabrication, and post-fabrication processes. These efforts aim to provide additional mechanisms to ensure that hardware designs remain free from malicious modifications throughout their lifecycle.

4.4.1 Prevention at fabrication

The challenge of untrusted fabrication is addressed by (W. Li and Ta, 2022), who proposed a framework in which an IP consumer provides both a hardware specification and a set of Security-related properties. The IP consumer and IP producer must jointly agree on a formal translation of these properties into a theorem-proving language. As the IP producer develops the hardware description in HDL, they simultaneously generate a formal proof demonstrating that the design satisfies all specified security properties. Upon delivery, the IP consumer can use a theorem prover to verify the correctness of the IP.

By linking a formal model to the hardware specification, this method can increase confidence in the correctness of the implementation. However, the approach relies on the assumption that the IP vendor is sufficiently trustworthy to produce both an accurate design and a valid formal proof. An untrustworthy vendor could conceivably introduce a Trojan into both the design and its corresponding proof, creating a malicious implementation that is extremely difficult to detect. This situation is reminiscent of examples from The Underhanded C Contest, where code is intentionally obfuscated to hide malicious behavior. Moreover, even if all initially identified security-related properties are formally verified, the continual development of new attack techniques may introduce vulnerabilities that circumvent these protections.

4.4.2 Prevention at post-fabrication

Shihab *et al.*, (2019) describe an approach in which portions of an IC's design are implemented using reconfigurable logic, programmed post-fabrication. The reconfigurable logic is inserted between the outputs of certain IC components and the inputs of others, thereby concealing aspects of the design from an attacker who has access only to the RTL. This method can be viewed either as a preventive measure or as a means of operating securely in the presence of potential hardware Trojans.

From a preventive perspective, the use of reconfigurable logic reduces the attacker's ability to predict the IC's exact functionality until the logic is programmed, effectively narrowing the window of opportunity for Trojan insertion. Despite such measures, completely preventing the addition of hardware Trojan logic remains extremely challenging. Consequently, this approach should be regarded as an initial step within a broader, multi-layered strategy to mitigate the risks posed by hardware Trojans.

4.5 Hardware Trojan Detection Techniques

It is not possible to completely prevent the insertion of a hardware Trojan into an IC during a typical design flow. While preventive measures reduce the likelihood of Trojan insertion, detection mechanisms are essential for identifying their presence within a design or device.

Once a hardware Trojan is detected, several options exist. If detected at the RTL level, it can be removed from the design. Alternatively, the affected IC can be isolated to prevent its use, or it may continue to operate with the Trojan present, with system-level measures accounting for its effects. Depending on the detection method, identification may be definitive, or it may provide a probabilistic assessment indicating the likelihood that the design or IC has been compromised (Li and Ta, 2022). Traditional IC testing and verification primarily focus on performing acceptance tests and ensuring that a device operates according to its specifications. However, these processes generally do not account for the presence of additional, unauthorized functionality. Given the vast state space in which such extra functionality could reside, detecting it is extremely challenging except in the simplest designs. Consequently, hardware Trojans may evade detection during testing and only activate once the chip is deployed in the field. (Trippel *et al.*, 2020) emphasize that it is not possible to guarantee that ICs in use are free from Trojans.

There is no universal method for detecting all hardware Trojans. Most contemporary research concentrates on post-fabrication detection, as the fabrication stage is widely regarded as the most vulnerable point in the IC development cycle. Relatively little work has been conducted on detecting Trojans at the RTL level prior to synthesis, or during fabrication. At these stages, designers are generally assumed to be trusted, making detection contingent on trusted audit procedures, including RTL design reviews and simulation.

A variety of techniques exist for hardware Trojan detection, but each is typically effective only against specific classes of Trojans. As detection methods advance, Trojan designers continually develop strategies to evade them, creating an arms race analogous to that observed in the anti-virus industry. State-of-the-art detection approaches are based on this understanding and aim to identify Trojans through targeted techniques informed by their behavior and characteristics.

5. Conclusion

This systematic literature review has provided a comprehensive synthesis of existing research on Hardware Trojan Horses, focusing on their threats, insertion mechanisms, prevention strategies, detection techniques, and future research directions. Using the PRISMA framework, 55 relevant studies published between 2014 and 2026 were systematically identified, screened, and analyzed to provide a structured understanding of the current state of Hardware Trojan research. The review demonstrates that Hardware Trojans continue to pose a significant threat to the security, integrity, and reliability of modern integrated circuits due to the increasing reliance on third-party intellectual property cores, outsourced fabrication, and globally distributed semiconductor supply chains.

The review reveals that Hardware Trojans can be inserted at multiple stages of the integrated circuit development lifecycle and employ diverse trigger mechanisms and payloads that make their detection particularly challenging. Existing prevention strategies primarily focus on trusted design methodologies, secure fabrication processes, formal verification, and design obfuscation techniques, while detection approaches include logic testing, side-channel analysis, runtime monitoring, formal verification, and machine learning-based methods. Although considerable progress has been achieved, no single detection technique is capable of identifying all categories of Hardware Trojans because each approach exhibits limitations in terms of detection coverage, scalability, computational complexity, implementation overhead, or susceptibility to stealthy Trojan designs.

The findings further indicate that recent research has increasingly adopted artificial intelligence and machine learning techniques to improve detection accuracy. Nevertheless, the effectiveness of these approaches is constrained by the absence of standardized benchmark datasets, limited real-world validation, inconsistent evaluation metrics, and poor generalization across different hardware platforms. Furthermore, current research remains heavily concentrated on post-fabrication detection, with comparatively limited attention devoted to design-time prevention, runtime behavioral analysis, and integrated cross-layer security mechanisms.

Generally, this review provides a consolidated knowledge base that categorizes existing Hardware Trojan threats, prevention mechanisms, and detection techniques while identifying critical research gaps that continue to hinder the development of reliable and scalable security solutions. The study highlights the need for standardized benchmarking frameworks, publicly available datasets, hybrid detection models that integrate structural, functional, and side-

channel information, and explainable machine learning techniques capable of detecting previously unseen Hardware Trojans. These findings provide valuable guidance for researchers, hardware designers, semiconductor manufacturers, and cybersecurity practitioners seeking to develop more robust, adaptive, and trustworthy Hardware Trojan prevention and detection systems for future integrated circuit technologies.

5.1 RQ4: What are the major challenges and research gaps in existing Hardware Trojan prevention and detection approaches

The systematic literature review revealed that existing Hardware Trojan (HT) prevention and detection approaches continue to face numerous technical and practical challenges that limit their effectiveness in real-world applications. A major concern is the reliance on offline analysis, simulation-based evaluation, and post-fabrication testing, with relatively few studies proposing real-time or runtime detection mechanisms capable of identifying Hardware Trojans during normal system operation. Furthermore, many of the proposed techniques have only been validated using benchmark circuits or simulation environments, with limited implementation and evaluation on real integrated circuits (ICs), field-programmable gate arrays (FPGAs), or commercial hardware platforms, thereby restricting confidence in their practical deployment.

Another significant challenge is the high computational, area, power, and timing overhead associated with many Hardware Trojan prevention and detection techniques, particularly those based on logic obfuscation, side-channel analysis, and trusted verification frameworks. These overheads reduce their suitability for resource-constrained environments such as embedded systems, Internet of Things (IoT) devices, and edge computing platforms. In addition, most existing approaches have been evaluated using relatively small benchmark datasets, limiting their scalability and effectiveness when applied to large-scale System-on-Chip (SoC) architectures, heterogeneous computing environments, cloud infrastructures, and modern semiconductor manufacturing processes.

The review also identified important limitations in artificial intelligence and machine learning-based Hardware Trojan detection methods. Although these techniques often achieve high detection accuracy, they are typically trained on limited benchmark datasets, resulting in poor generalization to unseen Trojan variants, evolving attack strategies, and diverse hardware architectures. Moreover, the scarcity of standardized, large-scale, and publicly available Hardware Trojan datasets hinders fair comparison among competing approaches and limits the development of robust learning models. The absence of standardized evaluation metrics and unified benchmarking frameworks further complicates objective performance assessment across different detection techniques.

Another critical research gap concerns the ability of current approaches to detect stealthy, dormant, low-power, and trigger-based Hardware Trojans specifically designed to evade conventional testing and side-channel analysis techniques. Existing solutions also provide limited protection against Hardware Trojan insertion across the global semiconductor supply chain, where multiple untrusted third parties may participate in hardware design, fabrication, testing, packaging, and distribution. Furthermore, most studies focus exclusively on either prevention or detection, with relatively few proposing integrated end-to-end frameworks that combine secure hardware design, runtime monitoring, post-deployment verification, and recovery mechanisms.

Finally, emerging threats involving hardware–software co-attacks, firmware vulnerabilities, artificial intelligence-enabled attacks, and intelligent Hardware Trojans have exposed additional shortcomings in current defense mechanisms, which are primarily designed to address conventional hardware threats. The review also highlights the limited availability of lightweight detection techniques suitable for low-power IoT devices and the lack of explainable artificial intelligence models capable of providing interpretable and trustworthy detection decisions. Collectively, these findings indicate that future research should focus on developing scalable, lightweight, real-time, explainable, and experimentally validated Hardware Trojan prevention and detection frameworks supported by standardized datasets, comprehensive evaluation methodologies, and integrated security mechanisms capable of protecting modern hardware systems throughout their entire lifecycle.

RQ5: What future research directions have been proposed to improve Hardware Trojan detection and prevention?

The findings from the reviewed studies indicate that future research on Hardware Trojan (HT) detection and prevention should focus on developing more intelligent, scalable, and practical security solutions capable of addressing the evolving complexity of modern hardware systems. A recurring recommendation across the literature is the development of real-time and runtime Hardware Trojan detection mechanisms that can continuously monitor system behavior during operation rather than relying solely on pre-deployment testing or post-fabrication inspection. Such approaches would improve the detection of stealthy, dormant, and trigger-based Trojans that often evade traditional testing methods and remain inactive until specific operating conditions are met.

Another prominent research direction involves the integration of artificial intelligence, machine learning, and deep learning techniques with conventional hardware security methods to improve detection accuracy and adaptability. Future studies should emphasize explainable artificial intelligence (XAI) models that not only achieve high detection performance but also provide interpretable decision-making processes suitable for security-critical applications. Researchers also recommend developing larger, standardized, and publicly available Hardware Trojan benchmark datasets that represent diverse Trojan types, hardware architectures, and real-world attack scenarios. Such datasets would facilitate fair performance comparison, improve model generalization, and accelerate the development of more robust detection algorithms.

The literature further highlights the need for lightweight Hardware Trojan prevention and detection techniques specifically designed for resource-constrained environments, including Internet of Things (IoT) devices, embedded systems, cyber-physical systems, and edge computing platforms. Future research should also investigate scalable security frameworks capable of protecting complex System-on-Chip (SoC) architectures, heterogeneous computing platforms, and cloud-based hardware infrastructures without introducing excessive computational, power, or area overhead. Additionally,

several studies recommend strengthening hardware supply chain security through secure design methodologies, trusted manufacturing processes, authentication mechanisms, and continuous verification techniques capable of detecting malicious modifications throughout the integrated circuit lifecycle.

Another important direction involves the development of comprehensive end-to-end Hardware Trojan defense frameworks that integrate prevention, detection, mitigation, recovery, and post-deployment monitoring into a unified security architecture. Rather than treating prevention and detection as independent processes, future solutions should adopt a layered defense strategy capable of protecting hardware systems from design through fabrication, deployment, operation, maintenance, and eventual retirement. Researchers also advocate increased validation of proposed methods using real integrated circuits, field-programmable gate arrays (FPGAs), industrial hardware platforms, and large-scale semiconductor systems to demonstrate practical feasibility and deployment readiness beyond simulation-based environments.

Finally, the review emphasizes the importance of addressing emerging threats arising from hardware–software co-attacks, firmware vulnerabilities, artificial intelligence-enabled attacks, cloud computing environments, and next-generation semiconductor technologies. Future research should therefore investigate hybrid detection models that combine logic testing, side-channel analysis, behavioral monitoring, machine learning, and formal verification techniques to improve resilience against increasingly sophisticated Hardware Trojan attacks. Collectively, these research directions suggest that future Hardware Trojan security solutions should be intelligent, lightweight, explainable, scalable, experimentally validated, and capable of providing continuous protection across the entire hardware lifecycle while adapting to emerging technologies and evolving cyber threats.

et integration to support real-time, resource-efficient detection mechanisms adaptable to evolving Trojan behaviors.

References

- Abuzaid, A. M., Saudi, M. M., Taib, B. M., & Abdullah, Z. H. (2023). *An Efficient Trojan Horse Classification (ETC)*. 10(2), 96–104.
- Batista, F. K., & Martín, A. (2020). *A New Individual-Based Model to Simulate Malware Propagation in Wireless Sensor Networks*. 1–23. <https://doi.org/10.3390/math8030410>
- Benjamin, T., Rana, E., Jason, F., Ramesh, K., & Krishnendu, C. (2020). *Two Hardware-Based DIP Vulnerability Detection and Post-Deployment Patching in Systems-on-Chip*. 1–14.
- Bhatia, S., Behal, S., & Ahmed, I. (2018). *Distributed Denial of Service Attacks and Defense Mechanisms : Current Landscape and Future Directions*.
- Butt, M. A., Voronkov, G. S., Grakhova, E. P., Kutluyarov, R. V., Kazanskiy, N. L., & Khonina, S. N. (2022). *Environmental Monitoring : A Comprehensive Review on Optical Waveguide and Fiber-Based Sensors*.
- Chowdhury, P., Guin, U., Singh, A. D., & Agrawal, V. D. (2021). *Estimating Operational Age of an Integrated Circuit*. 25–40.
- Dong, C., Xu, Y., Liu, X., Zhang, F., He, G., & Chen, Y. (2020). Hardware Trojans in Chips : A Survey for Detection and Prevention. *Sensors*, 1–37. <https://doi.org/10.3390/s20185165>
- Elshafie, H., Hamdan, M., Salih, S., Almohamedh, R. M., Eldin, A., & Awouda, A. (2025). *Emerging Threats in Internet-of-Things (IoT) Hardware Security*. 25(4).
- Everhart, R. M., Morris, J. R., & Amar, M. M. (2016). West Africa: A “ticking time bomb” or an opportunity to advance U.S. national security interests? *InterAgency Journal*, 7(2), 44–56.
- Ghiani, G., Science, I., Santoro, C., Ghiani, G., Manca, M., & O, F. P. (2017). *Personalization of Context-Dependent Applications Through Trigger-Action Rules Open Access Support provided by : Personalization of Context-Dependent Applications Through*. 24(2). <https://doi.org/10.1145/3057861>
- Guo, C., Liu, X., Xie, C., Zhou, A., Zeng, Y., Lin, Z., Song, D., & Li, B. (2024). *RedCode : Risky Code Execution and Generation Benchmark for Code Agents*. *NeurIPS*.
- Hanif, A., Member, S., Yu, Y., & Devoto, D. (2018). *A Comprehensive Review toward the State-of-the-Art in Failure and Lifetime Predictions of Power Electronic Devices*. 8993(3). <https://doi.org/10.1109/TPEL.2018.2860587>
1. He, L., Liu, Q., & Zhang, J. (2016). A survey of hardware Trojan threat and defense. *Integration, the VLSI Journal*, 55, 426–437. <https://doi.org/10.1016/j.vlsi.2016.01.004>
2. Hidalgo-López, J. A. (2023). *A simple digital readout circuit for differential resistive or capacitive sensors*. *Measurement Science and Technology*, 34(8), 085117. <https://doi.org/10.1088/1361-6501/acd139>
3. Hu, W., Chang, C.-H., Sengupta, A., Bhunia, S., Kastner, R., & Li, H. (2021). An overview of hardware security and trust: Threats, countermeasures, and design tools. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(6), 1010–1038. <https://doi.org/10.1109/TCAD.2020.3047976>
- Ibrahim, M. (2014). *A New Model for Trojan Detection using Machine Learning*. 13, 129–143.

- Jin, L., Wen, X., Jiang, W. E. I., & Zhan, J. (2024). *A Survey of Trojan Attacks and Defenses to Deep Neural Networks*. 37(4).
- Kabakci, N. J. (2020). *Hardware Trojans and their Security Impact on Reconfigurable System-on-Chips*.
- Kanaker, H., Karim, N. A., Awwad, S. A. B., Ismail, N. H. A., & Zraqou, J. (2018). *Trojan Horse Infection Detection in Cloud Based Environment Using Machine Learning*. 16(24), 81–106.
- Khodosevych, D., & Sakib, A. A. (2022). Evolution of NULL Convention Logic Based Asynchronous Paradigm : An Overview and Outlook. *IEEE Access*, 10(July 2022), 78650–78666. <https://doi.org/10.1109/ACCESS.2022.3194028>
- Kornaros, G. (2022). Hardware-Assisted Machine Learning in Resource-Constrained IoT Environments for Security : Review and Future Prospective. *IEEE Access*, 10, 58603–58622. <https://doi.org/10.1109/ACCESS.2022.3179047>
- Kuaban, G. S., Gelenbe, E., Czachórski, T., Czekalski, P., & Tangka, J. K. (2023). *Modelling of the Energy Depletion Process and Battery Depletion Attacks for Battery-Powered Internet of Things*.
4. Kulandaivel, S. (2021). *Revisiting remote attack kill-chains on modern in-vehicle networks* (Doctoral dissertation, Carnegie Mellon University). Carnegie Mellon University. https://kithub.cmu.edu/articles/thesis/Revisiting_Remote_Attack_Kill-Chains_on_Modern_In-Vehicle_Networks/19758490
5. Li, S., Jiang, L., Zhang, Q., Wang, Z., Tian, Z., & Guizani, M. (2023). *A malicious mining code detection method based on multi-feature fusion*. *IEEE Transactions on Network and Service Management*. (Publication details could not be fully verified from the information provided.)
6. Li, W., & Ta, B. (2022). *Hardware-supported patching of security bugs in hardware IP blocks*. *Proceedings of the 59th ACM/IEEE Design Automation Conference (DAC)*, 1–6. <https://doi.org/10.1145/3489517.3530677>
7. Ma, M. (2024). *Grid-connected PV system modelling based on grid-forming inverters* (Master's thesis). LUT University, Lappeenranta–Lahti University of Technology.
8. Mahjabin, T., Xiao, Y., Sun, G., & Jiang, W. (2017). A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *International Journal of Distributed Sensor Networks*, 13(12). <https://doi.org/10.1177/1550147717741463>
9. Malekpour, A. (2022). *Hardware Trojans mitigation in MPSoCs* (Master's thesis). Eindhoven University of Technology.
10. Marketos, A. T., Rothwell, C., Gutstein, B. F., Pearce, A., Neumann, P. G., Moore, S. W., & Watson, R. N. M. (2019). Thunderclap: Exploring vulnerabilities in operating system IOMMU protection via DMA from untrustworthy peripherals. *Proceedings of the Network and Distributed System Security Symposium (NDSS 2019)*. <https://doi.org/10.14722/ndss.2019.23300>
11. Nuzhat, T., Oad, A., Sajjad, M., Talpur, H., Rehman, H., & Luhrani, A. (2021). *Comparison of signature-based techniques*. (Publication details could not be verified from the information provided.)
12. Okkerman, H. (2020). *Embedded memory security: Securing the IoT from the hardware level* (Master's thesis). University of Twente.
13. Ömer, A., Semih, S. A., Merve, O.-O., Yılmaz, A. A., & Erdal, A. (2023). A comprehensive review of cybersecurity vulnerabilities, threats, attacks, and solutions. *Electronics*, 12(6), Article 1333. <https://doi.org/10.3390/electronics12061333>
14. Parhizkar, T., Stewart, T., Huang, L., Mosleh, A., & Angeles, L. (2023). *Degradation and failure mechanisms of complex systems: Principles*. (Publication details could not be verified.)
15. Perez, G. (2021). *Malicious hardware and its effects on industry*. (Unpublished manuscript or institutional report; publication details could not be verified.)
16. Prinetto, P., & Roascio, G. (2020). Hardware security, vulnerabilities, and attacks: A comprehensive taxonomy. *ACM Computing Surveys*. (Volume, issue, and page numbers could not be verified.)
17. Rahman, H. (2024). *A comprehensive survey on hardware–software co-protection against invasive, non-invasive, and interactive security threats*. (Publication details could not be verified.)
18. Ravichandran, N., Tewaraja, T., Rajasegaran, V., & Kumar, S. S. (2024). Comprehensive review analysis and countermeasures for cybersecurity threats: DDoS, ransomware, and Trojan horse attacks. *Preprints*. <https://doi.org/10.20944/preprints202409.1369.v1>
19. Rooney, C., & Seeam, A. (2018). Creation and detection of hardware Trojans using side-channel analysis. *Electronics*, 7(7), Article 124. <https://doi.org/10.3390/electronics7070124>
20. Sen, S., & Ghosh, A. (2024). Circuit-level techniques for side-channel attack resilience: A tutorial. *IEEE Solid-State Circuits Magazine*, 16(3), 96–108. <https://doi.org/10.1109/MSSC.2024.3444740>
21. Sethumadhavan, B. Y. S., Waksman, A., Suozzo, M., Huang, Y., & Eum, J. (2015). Trustworthy hardware from untrusted components. *Communications of the ACM*, 59(9), 84–92. <https://doi.org/10.1145/2699412>

-
22. Shihab, M. M., Tian, J., Reddy, G. R., Hu, B., Schaefer, B. C., Sechen, C., & Makris, Y. (2019). Design obfuscation through selective post-fabrication transistor-level programming. *Proceedings of the IEEE International Test Conference (ITC)*, 528–533.
23. Sonia, A., Kasem, K., & Magdy, B. (2023). A survey on hardware security: Current trends and challenges. (*Publication details could not be verified.*)
24. Sun, M., & Jiao, S. (2026). From ICs to devices: A survey on hardware tampering detection via power delivery network and signal trace. *ACM Computing Surveys*. Advance online publication. <https://doi.org/10.1145/3779441>
25. Taheri, S. (2021). *Secure and trustworthy hardware and machine learning systems for the Internet of Things* (Doctoral dissertation). University of Connecticut.
26. Theresa, S., Benjamin, T., & Nour, M. (2020). *Supply Chain 4.0: A survey of cybersecurity challenges, solutions and future directions*. (*Publication details could not be verified.*)
27. Trippel, T., Shin, K. G., Bush, K. B., & Tech, V. (2020). ICAS: An extensible framework for estimating the susceptibility of IC layouts to additive Trojans. *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST 2020)*.
28. Tsang, R., Joseph, D., Asmita, A., Salehi, S., Carreon, N., & Mohapatra, P. (2022). FANDEMIC: Firmware attack construction and deployment on power management integrated circuits and impacts on IoT applications. *Proceedings of the IEEE Symposium on Security and Privacy (SP 2022)*.
29. Vamja, H. (2018). *Reverse engineering of finite state machines from sequential circuits* (Master's thesis). University of South Florida.
30. Wang, X. (2021). Hardware Trojan attack in embedded memory. *ACM Journal on Emerging Technologies in Computing Systems*, 17(1), Article 3. <https://doi.org/10.1145/3422353>
31. Wu, T. F., Ganesan, K., Hu, Y. A., Wong, H. P., Wong, S., & Mitra, S. (2015). TPAD: Hardware Trojan prevention and detection for trusted integrated circuits. *ACM Transactions on Design Automation of Electronic Systems*, 20(3), Article 46. <https://doi.org/10.1145/2779056>
32. Xiao, K., Forte, D., Karri, R., Bhunia, S., & Tehranipoor, M. (2016). Hardware Trojans: Lessons learned after one decade of research. *ACM Transactions on Design Automation of Electronic Systems*, 22(1), Article 6. <https://doi.org/10.1145/2906147>
33. Xue, M., Gu, C., Liu, W., Yu, S., & O'Neill, M. (2020). Ten years of hardware Trojans: A survey from the attacker's perspective. *IET Computers & Digital Techniques*, 14(5), 231–246. <https://doi.org/10.1049/iet-cdt.2020.0041>
34. Yaswanth, R., Malathi, Y., & Sneha, R. S. (2021). *Hardware Trojan insertion and detection using side-channel analysis and implementation of BPUF*. (*Publication details could not be verified.*)
35. Zhang, J. (2018). Application of artificial intelligence technology in computer network security. *International Journal of Network Security*, 20(6), 1016–1021. <https://doi.org/10.6633/IJNS.201811>
36. Zou, M., Cui, X., & Shi, L. (2017). Potential trigger detection for hardware Trojans. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(8), 1690–1703. <https://doi.org/10.1109/TCAD.2017.2753201>