IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 25 August 2023, accepted 9 October 2023, date of publication 2 November 2023, date of current version 7 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3329572

SURVEY

Wireless Network-on-Chip Security Review: Attack Taxonomy, Implications, and Countermeasures

LASHMI KONDOTH, RAJAN SHANKARAN, (Senior Member, IEEE), QUAN Z. SHENG, (Member, IEEE), AND RICHARD HAN
School of Computing, Macquarie University, Sydney, NSW 2109, Australia

Corresponding author: Lashmi Kondoth (lashmi.kondoth@students.mq.edu.au)

This work was supported by the International Research Training Program (iRTP) Scholarship Scheme.

ABSTRACT Network-on-chip (NoC) is a critical on-chip communication framework that underpins high-performance multicore computing and network system architectures. Its adoption has become widespread due to the ongoing emergence of cutting-edge NoC technologies. However, the introduction of wireless interfaces has expanded potential vulnerabilities, impacting confidentiality, integrity, and system availability. While there is a growing body of research in NoC security, a comprehensive study of security threats in wireless NoC is lacking. Our article provides an extensive review of recent advancements and research in wireless NoC security. This encompasses examinations, summaries, comparative assessments, and the constraints of existing studies. We conclude by delineating future research directions in this crucial field.

INDEX TERMS Countermeasures, network-on-chip, on-chip security, wireless communication, denial-of-service (DoS), eavesdropping, spoofing.

I. INTRODUCTION

System-on-Chip (SoC) contains various hardware elements embedded into one integrated circuit [1]. An SoC may integrate 200×300 diverse components such as microprocessors, memory units, and communication interfaces like Inter-Intergrated circuits (I2C), Serial Peripheral Interface (SPI), and Universal Synchronous/Asynchronous Receiver/Transmitter (USART). These different components in an SoC were historically connected with a shared bus [2], as shown in Figure 1. Bus architectures may face performance limitations, particularly as the number of connected devices and data transfer demands increase. As a consequence, a groundbreaking technology, termed ''Network-on-Chip,'' surfaced, enhancing traditional System-on-Chip (SoC) architectures by facilitating communication among numerous cores [3]. This packet-switching technique used in NoC provides optimal performance in a multicore environment by segmenting packets into multiple flits, thereby improving link utilization compared to circuit switching, which requires a dedicated communication channel.

As depicted in Figure 2, a Network-on-Chip (NoC) architecture comprises routers, each linked to a processing core through a network interface. Additionally, each router is interconnected with four adjacent routers. A data packet from a source travels through one or more routers to reach the destination, thus limiting the interconnection wires between communicating modules. These integrated circuits gained popularity due to their high performance [4]. This has led to the development of several NoC architectures, such as wireless NoC, hybrid wireless NoC, and photonics NoC.

Wireless NoC, the most predominant NoC technology, facilitates low delay, high bandwidth, and long-distance on-chip communication. In addition to the conventional components of an NoC, such as links, routers, and network interfaces, wireless NoCs also incorporate a wireless transmitter to facilitate single-hop, long-range wireless links. The placement of these wireless interfaces is crucial to minimize network size and overhead [5]. A number of topologies that

The associate editor coordinating the review of this manuscript and approving it for publication was Muhamamd Aleem.

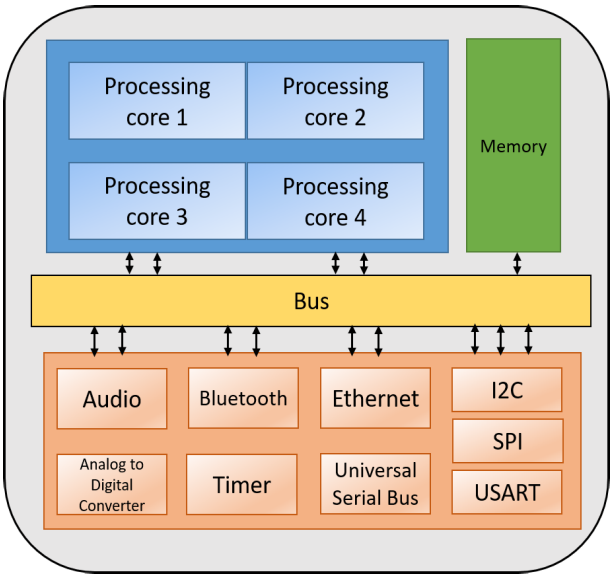

FIGURE 1. A sample system-on-chip (SoC) architecture.

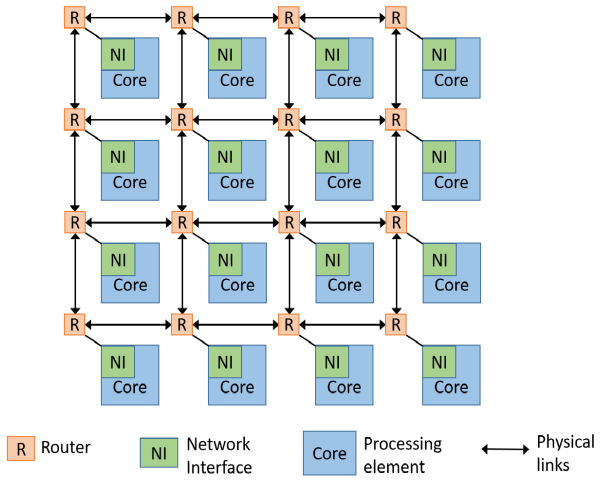

FIGURE 2. A typical wired newtork-on-chip (NoC) architecture.

make use of wireless connectivity have been proposed, such as fully wireless topology, hybrid wired-wireless topology with a wireless interface assigned to a single node or shared by multiple nodes, and small world-based wireless NoC topology [6]. Since replacing every physical interconnection with a wireless link consumes high power and offers poor scalability due to channel limitation, a hybrid wired-wireless NoC architecture is proposed, where the topology is divided into multiple clusters, as shown in Figure 3. This architecture is commonly adopted to attain an optimal tradeoff between performance and power consumption. Each cluster contains a cluster head and multiple nodes. Physical wires are used to connect all nodes within a cluster, and inter-cluster communication is carried out via the central nodes or cluster heads through wireless transmission links, as explained below.

Data packets within the cluster follow the commonly used deterministic routing, specifically the XY routing algorithm [7]. For inter-subnet communication, packets are directed through single-hop wireless links situated at the central node of the cluster. Although the Tera Hertz (THz) antenna is proven to have high bandwidth and performance, it is prone to high error rates. Therefore, antennas operating in the millimeter wave (mm-wave) spectrum are the most preferred technology [8], [9].

Over the past few years, there has been significant research activity directed toward enhancing the performance and energy efficiency of wireless Network-on-Chip (NoC) systems. However, the adoption of wireless communication in NoCs brings about the potential for cyber-attacks due to the use of wireless channels. These malicious attacks have the potential to interfere with the operation of the entire NoC subsystem. Each component of a NoC is vulnerable to a distinct type of attack. Managing security attacks, as well as the identification and isolation of attackers poses significant challenges within the NoC environment due to its limited resources, as outlined in [10]. This situation calls for an immediate and thorough response that includes the development of strategies and the implementation of solutions.

A. RELATED SURVEY AND PAPER SELECTION

Extensive surveys are presented in [2] and [11], providing a taxonomy of attacks based on security goals and defenses for NoCs. While [11] categorizes the attacks under confidentiality, integrity, and availability, [2] categorizes attacks on the basis of six security services: confidentiality, integrity, authenticity, availability, anonymity, and freshness. These surveys conduct a thorough analysis of proposed countermeasures and emphasize the limitations associated with these countermeasures. Proposed countermeasures are evaluated in terms of system overheads, which are critical for assessing their practicality and impact on NoC performance. Another such work is [12], where the security attacks on wired NoC and their countermeasures are discussed. However, this paper considers only a limited number of security attacks, such as eavesdropping, spoofing, and denial of service attacks.

This survey conducts a detailed examination of NoC security on a component-by-component basis. It examines each component in isolation, identifies potential attacks on each, and provides a synopsis of various approaches employed to counteract such attacks. A component-wise evaluation of NoC allows for a detailed analysis of each module's security properties, focusing on specific vulnerabilities and attack surfaces. This enables a more targeted and effective security assessment. Different components may have different security requirements and challenges. Component-wise evaluation enables the implementation of customized countermeasures for each module, addressing their specific security concerns. If a security breach or attack occurs, using

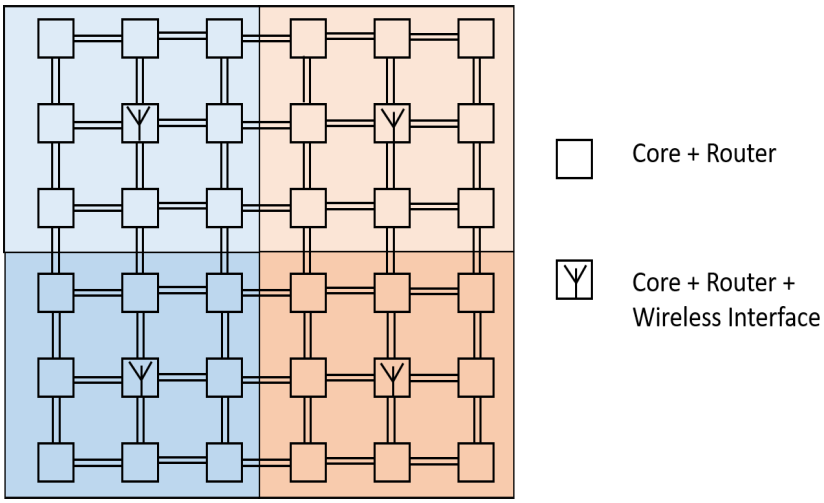

FIGURE 3. A wireless NoC architecture.

this type of evaluation can aid in isolating and containing the impact. By understanding which component is compromised, it becomes possible to limit the spread of the attack to other parts of the NoC. NoC architectures are often designed with modularity in mind. This type of evaluation aligns well with this approach, allowing individual components to be replaced or updated without affecting the entire system's security. This approach is particularly crucial in wireless NoCs, where the presence of multiple wireless interfaces and the unique properties of wireless communication introduce new challenges and attack vectors. To identify relevant research studies, we conducted a comprehensive literature search using IEEE Xplore, ACM Digital Library, and Scopus to identify sources that are current, relevant as well as authoritative. We also manually screened the reference lists to look for articles that were overlooked in the initial search process. Furthermore, we excluded various studies that were purely theoretical which did not involve any experimental evaluation. A thorough review of data collected from several quality papers reflecting the latest research and novelty from different perspectives further enhances this survey.

B. CONTRIBUTION

This paper aims to provide a thorough survey and analysis of the current state of wireless NoC security. The key contributions of this paper are outlined as follows:

- This paper introduces a component-wise review of security attacks on wireless NoC, marking the first work of its kind, with a thorough examination of various types of attacks and their impact on the underlying network infrastructure.
- This paper examines the feasibility and constraints of proposed countermeasures within the NoC framework. The aim is to support designers in crafting resilient and scalable solutions capable of addressing the varied requirements of future applications. Moreover, the analysis offers valuable insights into security and performance, empowering readers to make well-informed decisions regarding the optimal mitigation strategy tailored to their specific needs and requirements.
- This paper highlights open research issues and outlines possible research directions in this field: Despite progress, several open challenges still remain in the area of wireless NoC security. This paper identifies possible research directions to address these open challenges in wireless NoC, aiming to contribute to advancements and innovations in the field.

The rest of this article is organized as follows. Section II provides the preliminaries for the study of wireless NoC security, and NoC security attack taxonomy is examined in Section III. A survey on countermeasures implemented to defend against those attacks in NoC and wireless NoC is discussed in Sections IV and V, respectively. Finally, Section VI provides some concluding remarks with suggestions for future research.

II. PRELIMINARIES

The factors contributing to security attacks on NoC are manifold, and consequently, attack detection, analysis, and mitigation pose challenges that prevent a wide-scale deployment of wireless NoC to support diverse applications [10]. Even though restricting physical access to the system minimizes physical attacks, malicious sources within the NoC system could initiate attacks that are not physical in nature. These attacks pose a high risk and require complex methods of detection and defense. It is common for designers to use third-party intellectual property (IP) when designing NoC to save time and money. Due to this, vulnerabilities that lead to security breaches are mainly introduced during the integrated circuit (IC) design process. The following are the scenarios where malicious sources are planted during the manufacturing phase:

- **Design Stage:** The process of developing application software plays a crucial role in the design phase of NoC. The primary aim during this phase is to create models that depict the function and behavior of ICs through the utilization of logic blocks, thus achieving IC functionality. Within this stage, any potential design flaws can act as vulnerabilities, providing opportunities for the insertion of hardware trojans or authorized design modifications, thereby introducing security risks [13]. These design adjustments have the potential to significantly contribute to security breaches, and their detection can pose considerable challenges [13]. An obvious example would be a case that involves the manipulation of ''don't-care bits'' in register-transfer level (RTL) code, which can be exploited to gain access to restricted states and compromise the confidentiality of sensitive data.
- **Synthesis and Layout Stage:** Once the RTL blocks are implemented, the synthesis stage involves converting the high-level software into digital logic gates and hardware. This is followed by the generation of a physical representation and placement of components and routing logic. Consequently, modeling the electrical characteristics and timing requirements from the given layout can lead to various side-channel attacks. Furthermore, since adversaries have unrestricted access to the transistor gates and interconnections (netlist) through synthesis tools, they can also bring about a change in the functionality of these components by changing the logic states of the established connections [14].
- **Fabrication Stage:** After all the design checks have been carried out, the layout file is shared with the manufacturer, who then proceeds to fabricate the IC by assembling all the components. Various threats can be introduced when manufacturing in an untrusted facility, including Trojans, reverse engineering, and IP privacy violations [15]. Furthermore, altering the chemical concentrations during the fabrication can lead to changes in electrical or electromagnetic properties, which may, in turn, impact the performance and lifetime of an integrated circuit.

While internal attacks can come from various sources, as mentioned earlier, external attackers also present credible threats. Although attackers within a subsystem can be identified and prevented from launching further attacks, the same does not hold true for malicious activity resulting from external attacks, which can be detected but not prevented.

III. NOC SECURITY ATTACK TAXONOMY

Figure 4 shows a broad classification of NoC security attacks based on three parameters [16]: *attack vector, attack target, and attack impact*. An adversary can penetrate the system through software or a hardware event by exploiting the application or physical architecture.

Different attack vectors can be classified as follows:

- **Side-Channel attacks:** This is one of the most commonly occurring attacks, which is realized by exploring NoC implementation through observation of hardware properties such as operation time, power requirement, or electromagnetic leaks. A timing attack, for instance, exploits information leakage and data leakage within AES-encrypted (Advanced Encryption Standard) traffic. These attacks are discussed in [17] and [18], respectively. Similarly, the work in [19] has explored how circuit and packet switching techniques within NoC can be exploited to launch timing and cache-based attacks. Specifically, this research addresses a particular attack where malicious routers intercept flits passing through them to gather information about the nature of network traffic and related timing data. To counter this threat, circuit switching is recommended as a safer alternative to packet switching. This is because this technique makes it harder for the attacker to gain information as it establishes a dedicated communication path between the sender and receiver for the duration of the session, preventing eavesdropping and interception of data packets that are common in packet-switched networks.
- **Supply Chain Attacks:** In the course of the fabrication process, IPs, including specifications, source code, or other essential data required for producing a specific semiconductor or any software, are vulnerable to compromise. This can occur through the introduction of malicious files, theft of IP, or reverse engineering, ultimately jeopardizing the integrity of the supply chain. Although ICs use various obfuscation methods to lock out external access, emerging de-obfuscation technologies still pose a challenge. According to [20], such attacks are prevented by exponentially increasing the netlist cycles.
- **Network Attacks:** With the introduction of network-based communication within the SoC platform, various novel attacks have been introduced that target not just the wireless interfaces but also the NoC subsystem as a whole. Such attacks can track data, gain access to sensitive information, or degrade network performance by exhausting resources [21]. While computer network security has been extensively studied over recent years, the resource-constrained environment of NoC poses various challenges in terms of area, power, and performance requirements, previously not found in traditional networks.

The support for diverse topologies and routing techniques exposes NoC to different sets of vulnerabilities, as depicted in Figure 5:

- **Denial-of-Service (DoS):** A DoS attack has the capability to introduce redundant packets into a network, leading to the inefficient utilization of network bandwidth, thereby degrading the performance of an NoC [21]. The DoS attack can further cause persistent jamming

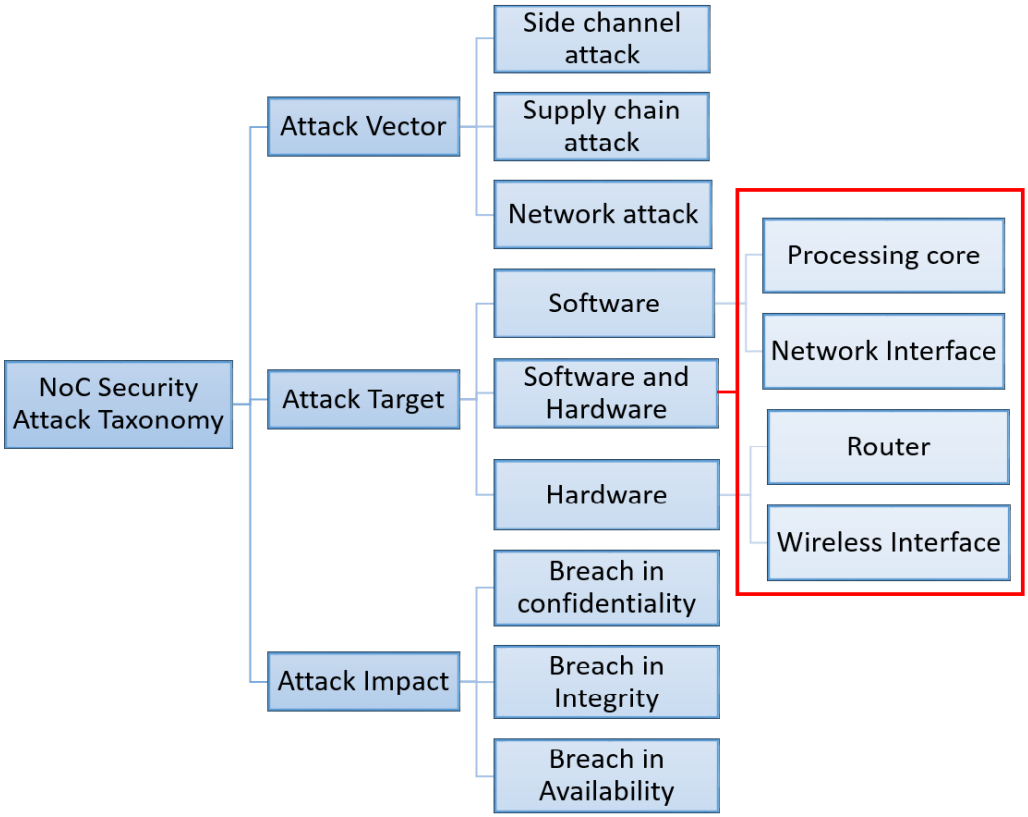

FIGURE 4. NoC attack taxonomy.

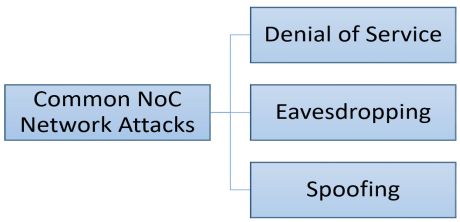

FIGURE 5. NoC network attack taxonomy.

on wireless interfaces, leading to collisions and thus depriving legitimate use of the interface.

- **Eavesdropping:** A malicious eavesdropper can intercept the network traffic and listen to all data communicated over the network to extract critical information. In particular, the broadcast nature of wireless interfaces makes them easy targets for eavesdroppers.
- **Spoofing:** An impersonation attack is an attempt by a malicious party seeking unauthorized access to a system to alter its configuration, causing inaccurate behavior or system failure.

Furthermore, these attacks can be hardware or software-based, or a combination thereof [11]. An example illustrating the deployment of hardware trojans involves unauthorized alterations to the NoC circuit during the design cycle. Usually, it is triggered during the operation of the trojan-injected module in order to alter the circuit behavior. Software-based malware injections are a common type of attack on SoCs, in which the device's firmware is modified to cause network disruptions to gain illegal access to resources or to leak sensitive data. Consequently, both the security and functionality of the entire system are compromised.

Irrespective of the attack type or its origin, an NoC security attack aims to violate the CIA triad model of information security [11]: (1) confidentiality, (2) integrity, and (3) availability, as shown in Figure 6. Within the realm of NoC, the role of the confidentiality service is to safeguard sensitive information from unauthorized extraction. The integrity service is responsible for guaranteeing that the content remains unaltered, and the availability service ensures that system resources can be consistently accessed. Compromising any of these vital services results in a decline in NoC performance and functionality.

Moreover, each component and its implementation style make it a target for particular attacks. As a result, future research must emphasize more on a modular approach to designing NoC security [12]. The following two sections delve into the significant advances made in recent years in countermeasures employed to safeguard wireless on-chip interconnects, some of which are also generally applicable to wired NoCs.

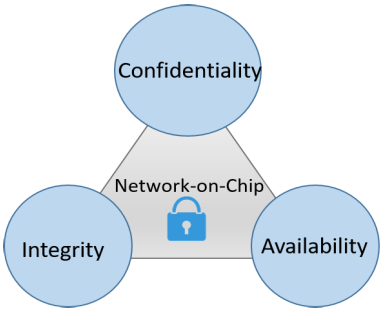

FIGURE 6. CIA triad model of security.

IV. NOC SECURITY ATTACKS AND COUNTERMEASURES

On-chip interconnect security countermeasures usually comprise three stages: 1) detection of an attack, 2) tracking the source of the attack, and 3) preventing the attack. These countermeasures can be implemented through various techniques, including the use of encryption for confidentiality, and security. Other techniques, such as secure zones, offer a protective layer to ensure the security of data as it travels through the network. A wired NoC typically consists of processing elements, network interfaces, and routers. An attack can target one or more of these components or a combination of them. Due to this, the unit that acts as the adversary and the unit in which the mitigation technology is integrated become the two key factors of security countermeasures in NoC.

A. PROCESSING CORE

NoC integrates various processing cores for multiple applications, all connected by the underlying communication fabric. These processing cores are mainly procured from third-party IPs to lower the cost and design cycle times of NoCs. Hence, there is a high risk of inadvertently or intentionally injecting hardware trojans (HTs) or software malware by the manufacturer, consequently triggering malicious behavior, which most likely seeks to access or tamper with the flits as they move through the fabric.

The authors of [22] propose a packet validation technique that combines algebraic manipulation detection (AMD) and cyclic redundancy check (CRC) codes for error detection to achieve enhanced data integrity and security. However, as stated in [23], this proposed approach leaves the header vulnerable, potentially risking the exposure of sensitive information. Therefore, tunnel-based encapsulation is used within an assumed trusted network interface to encrypt the header flit, exposing only the destination address [23]. In this implementation, the AES algorithm encrypts the data, and the Siphash function authenticates its access. While AES offers robust security, its application in a resource-constrained fabric increases latency and processing overheads, rendering its practical use unfeasible. Consequently, this implementation restricts itself to securing the integrity of data flits in the architecture through encryption, in exchange for incurring a significant overhead due to the utilization of AES.

Due to the latency and overhead, many solutions opt for a lightweight countermeasure. For example, arbitrary failure like the Byzantine fault is addressed in [24], where these faults are caused by HTs or DoS attacks, targeting the network's availability. In response to these potential security threats, a novel approach is proposed which serves the dual purpose of guaranteeing the dependable transmission of data flits or packets and identifying any malicious nodes within the network. However, this incurs latency of 10-40%, leading to performance overhead. In [25], a network traffic monitoring technique is developed to address DoS attacks launched by malicious sources, flooding the network with junk packets. First, statistical analysis of communication patterns computes congestion path information and packet latency information within each router. Eventually, a broadcast alert mechanism localizes the source of the attack, leading to 6% area and 4% power overhead for each attack scenario. In [26], a spiking neural network (SNN) is used to detect traffic anomalies resulting from DoS attacks. The use of SNN is an effective method for identifying patterns in data in real-time. In this instance, the SNN classifies the attack by monitoring the temporal duration of Request-to-Send (RTS) signals. However, this method's accuracy depends on the duration of the attack in the temporal sample, as accuracy increases from 60% to 87% when the attack duration progresses from 30% to 50%.

The secure zone concept, as shown in Figure 8, is another widely employed mechanism in networking to provide authenticated access. A ''secure zone'' refers to a designated area or region that is isolated within the on-chip network to ensure protection against unauthorized access or tampering. Accordingly, an architecture preventing non-secure packets from entering the protected area is presented in [27]. Moreover, the Diffi-Hellman protocol is utilized to generate group keys by the secure zone's border IPs, and these keys are subsequently used to secure communication within the zone.

A zone-based architecture is proposed in [28], which uses a single router in a cluster with four cores, memory, and a bus. This architecture dynamically generates the secure zone at run-time by isolating the communication resources. As a result, malicious parties are prevented from accessing shared resources. Consequently, sensitive data is kept confidential, and DoS attacks are mitigated.

In [29], secure zones are established through the utilization of multiple managerial cores, which are strategically distributed across various regions within the NoC. The creation of secure zones for sensitive applications permits packets from trusted and secure sources to enter the zone. Packets that are not destined for sensitive applications are rerouted to circumvent the zone. However, the placement of manager cores at fixed locations imposes architectural limitations, which could lead to hardware overhead because they may not effectively adapt to changing traffic patterns or scalability

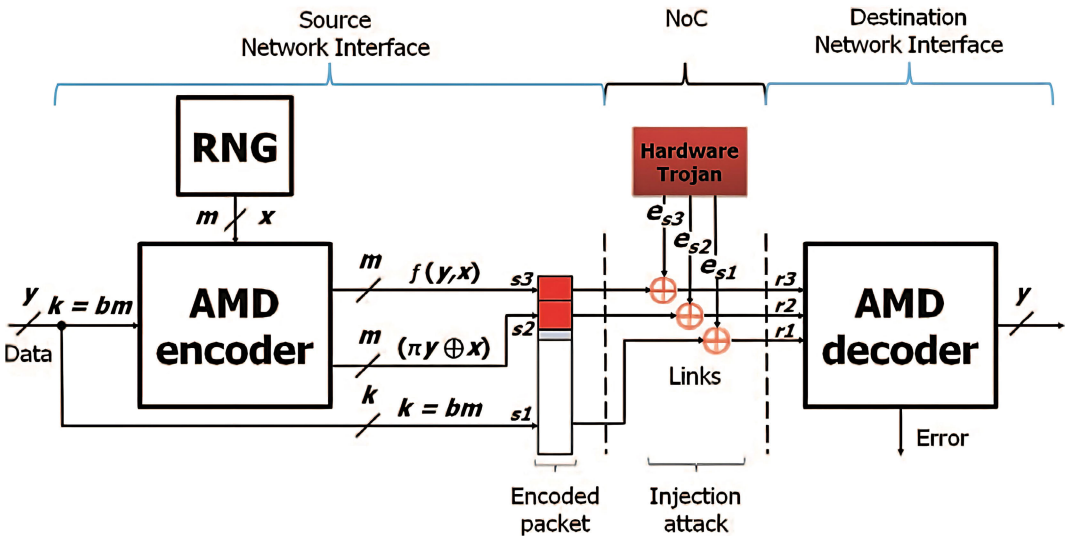

FIGURE 7. Packet validation using AMD encoder [22].

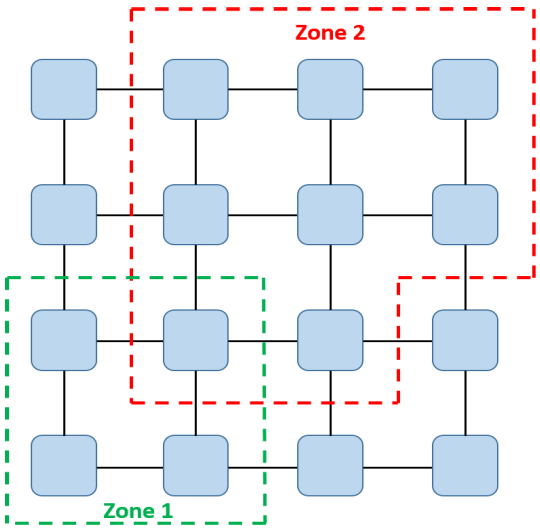

FIGURE 8. A sample secure zone isolation with two secure zones (Zone 1 and Zone 2).

requirements in the NoC. An approach similar to [28] is proposed in [30], wherein sharing of resources is prevented within the secure zone. As a result, only one application can run in the region, and this application is given exclusive access to the resources. Once a secure zone is formed, wrappers are used to discard all the packets coming into the zone, irrespective of from where they originate. Due to the wrappers isolating the IP cores, packets within the secure zone are not encrypted, resulting in faster execution. In [31], a dynamic, secure zone CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) NoC architecture is proposed, showing the feasibility of CAESAR cores in a multicore environment. Manager cores encrypt packets in this architecture, and network firewalls within the network interface are used for packet authentication. Nevertheless, the algorithms employed for authentication and encryption introduce significant latency and overhead to the NoC. A Secure zone implementation offers a proactive solution for maintaining NoC security. It involves various trade-offs concerning both area and performance.

As shown in Table 1, despite implementing various attack mitigation strategies, latency and performance overhead remains a concern. The processing element (PE) demands more resources compared to other NoC components. However, its constant access and generation of raw data render it more susceptible, necessitating the implementation of advanced data protection measures without incurring excessive overhead.

B. NETWORK INTERFACE (NI)

Network interfaces connect various heterogeneous processing elements and routers. Its primary function is to packetize and depacketize the data, as well as to generate flits for transmission within the network, as shown in Figure 9. According to [32], buffer-based NI is predominantly implemented using First In First Out (FIFO) scheduling due to low latency and reduced complexity. The Advanced eXtensible Interface (AXI) protocol (AXI) compliant NI architecture is extensively employed in transaction-based designs due to its compact footprint and minimal power consumption. Furthermore, NI could establish a set of security rules to function as a hardware firewall, as discussed in [33]. These rules form a security map that delineates different regions and their respective access permissions. Based on this map, memory access is determined for a specific source. However, NI is susceptible to various other attacks impacting flow control, routing, and packet manipulation due to its full access to the raw data.

TABLE 1. Processing element security countermeasures.

Security Attack	Attack Impact	Countermeasures	Limitations
DoS Attack	Network availability	Lightweight handshake authentication [24]	Increased latency by a factor of 10-40%.
		Traffic monitoring system [25]; monitors communication pattern and congestion	Introduces a 10% increase in area and power usage.
		Spiking neural network [26]; monitors the RTS signals over time	Low accuracy for short duration attacks with intervals less than 30%.
Data Tampering	Data integrity	AMD and CRC error detection [22]	Susceptible to traffic flow analysis as the header is left unprotected, thereby exposing sensitive payload information.
Eavesdropping	Unauthorized access to data and sensitive data extraction	Tunnel based encapsulation using AES encryption and Siphash authentication [23]	Results in an increased area overhead by 4.5%.
		Static secure zone implementation [27], [29]	Hardware overhead exceed by 10% due to architectural limitations.
		Dynamic secure zone implementation [28], [30], [31]	Hardware overhead exceed by 10% due to secure zone implementation.

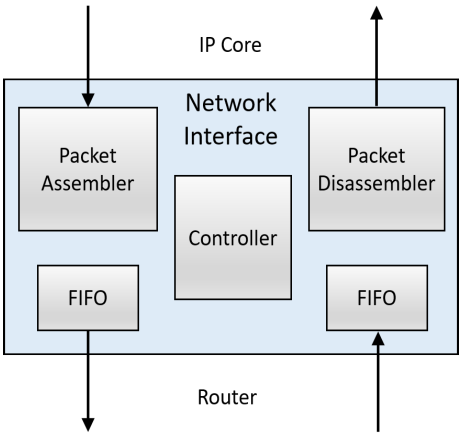

FIGURE 9. A typical network interface microarchitecture.

As suggested in [34], static allocation of links is done to ensure compliance with the principles of Confidentiality, Integrity, and Availability (CIA) in NoC security. In a statically allocated network, the routing paths for data transmission are predetermined during the design phase. This predictability means that different communication flows are less likely to overlap and interfere with each other, as they follow dedicated paths. This scheme allows uniform network traffic without user interference, thus ensuring the availability of the network. The NI employs a fixed bandwidth allocation scheme based on Time Division Multiplexing (TDM), along with temporal and data obfuscations, to enhance security significantly. Static allocation has the advantage of low overhead but lacks the efficiency of a dynamic allocation scheme, which enables rerouting, power scaling, and runtime configuration changes.

The work presented in [35] addresses duplication attacks where an HT injected into NI manipulates the header pointer. A header pointer is a crucial component used to manage the order and transmission of packets in a circular flit queue. When a flit is introduced by the packetizer, the tail pointer undergoes an increment. Subsequently, when a flit is conveyed to a router, the head pointer is raised to facilitate the transmission of the following flit. As an integral component of the proposed HT detection mechanism, the PE generates a cryptographic key by incorporating the outgoing data count, buffer ID, and destination ID into the header flit. Once the header flit is received, the NI calculates its own key using the same parameters and performs a cross-validation with the received key. In scenarios where packets have either invalid or duplicate destination addresses, the computed keys will not correspond, leading to the automatic discarding of such packets. This mechanism serves as a security measure to identify and eliminate potentially malicious or erroneous data packets within the network. Furthermore, the detector module utilizes a digital and analog implementation to monitor the ratio of incoming to outgoing packets over time. This approach assists in identifying the source of the attack and also serves as a preventive measure against reverse engineering. However, it could take ∼2.5 days to detect such an attack.

The work discussed in [36] examines an NI in which the attack focus is on the Finite State Machine (FSM) control unit. This control unit is targeted by exploiting its FIFO-based queuing mechanism. Critical bits are added to the finite state machine and dummy states are inserted to mitigate tampering attacks. Any attacker targeting the FSM control unit without the knowledge of the critical bits will make the unit jump to these illegal states and cannot revert to a legal state. Therefore, constant monitoring of state transitions

TABLE 2. Network interface security countermeasures.

Security Attack	Attack Impact	Countermeasures	Limitations
DoS Attack	Network jamming	Static allocation of links; time division multiplexing [34]	Not efficient when the NoC exhibits dynamic characteristics, such as rerouting, power scaling, and run-time configuration changes.
Tampering Attack	Manipulates header pointer	Encoded key cross-validation and monitoring the packet transmission ratio [35]	Attack detection is not significant since it may take in the worst case up to 2.5 days to identify and recognize the attack.
	FSM tampering	Addition of dummy states using key bits [36]	Attacker can monitor the state transitions to extract the keys.
Spoofing and Eavesdropping	Sensitive data extraction	Secure zone implementation; use of GCM authenticated encryption [37]	Results in an area overhead of greater than 20%.

leads to the detection of HT, considering that the key remains unextracted. Furthermore, the concept of partitioning the NoC into secure and non-secure zones is explored [37]. An authenticated encryption mechanism is utilized within the network interface of secure cores to prevent malicious parties located outside the zone from engaging in inter-zone communication. The authenticated encryption mechanism uses Galois/Counter mode (GCM) [37] to verify the packet's source and prevents inter-zone communication, thus securing access to core IPs and maintaining confidentiality.

Nonetheless, NI is completely exposed to raw data, rendering it vulnerable to various forms of attacks that can affect flow control, routing, and packet forwarding. Furthermore, NI is responsible for the formation of flits and the order in which they will be transmitted. As it acts as a gateway for the PE to communicate with the rest of the system, a failure or compromise of the NI would have a detrimental impact on the PE's functionality. Incorporating the firewall functionality into the network interface involves augmenting the existing infrastructure with advanced security features and access control mechanisms, as shown in Table 2, fortifying the network against potential threats and unauthorized access.

C. ROUTERS

Routers in NoC play a critical role in managing the data traffic and communication between different processing elements within the system. These routers are responsible for packet routing, ensuring efficient data transfer, and maintaining the overall performance and connectivity of the on-chip network. The traditional NoC router architecture comprises a crossbar switch, input buffers, allocators, and arbitration logic. Due to the efficacy of its channel utilization model, virtual-channel (VC) flow control is the most widely used flow-control technique in NoC routers [8] as shown in Figure 10. By using VC flow control, each channel's flit buffers are divided into several lanes, and each lane can allocate buffers independently. This provides a flexible method for channel assignment, enhancing both channel utilization and throughput within the system.

Routers within a NoC system employ a pipelined architecture [38], and contemporary research primarily concentrates on optimizing various stages to minimize latency and hardware overhead, thus bolstering overall performance [39]. However, routers' functionality to store, route, and forward data flits enables them raw access to all packets passing in the network. The work presented in [40] examines a black hole router attack that initiates a DoS attack wherein packets are intentionally discarded at a rate ranging from 5% to 34%, contingent upon factors such as the infected router's location and the number of infected routers. Furthermore, routers hold the routing table and routing logic. Through the insertion of a hardware trojan into the router, an attacker gains the capability to manipulate routing table entries, leading to routing anomalies and network disruptions. The outlined reasons provide a compelling rationale for investigating router security in NoC environments.

Attacks that specifically target the packet header by altering the destination address are discussed in [41]. Therefore, a trojan-aware 3-phase routing process is proposed in [42]. As part of the first phase, a detector module monitors input ports for violations of XY routing rules. Upon detecting a malicious router, other routers in the neighborhood are made aware of it through update messages, thereby shielding them from this compromised router. This way, the malicious router can be bypassed. The proposed bypass algorithm reroutes packets in network communication, addressing the risk of network deadlock due to deviations from XY routing. It introduces the concept of intermediate destinations. When a packet is rerouted, an intermediate destination is assigned, ensuring that it adheres to XY routing guidelines. By temporarily ejecting and re-injecting packets at these intermediate points, XY routing integrity is maintained, effectively eliminating the risk of network deadlocks. The

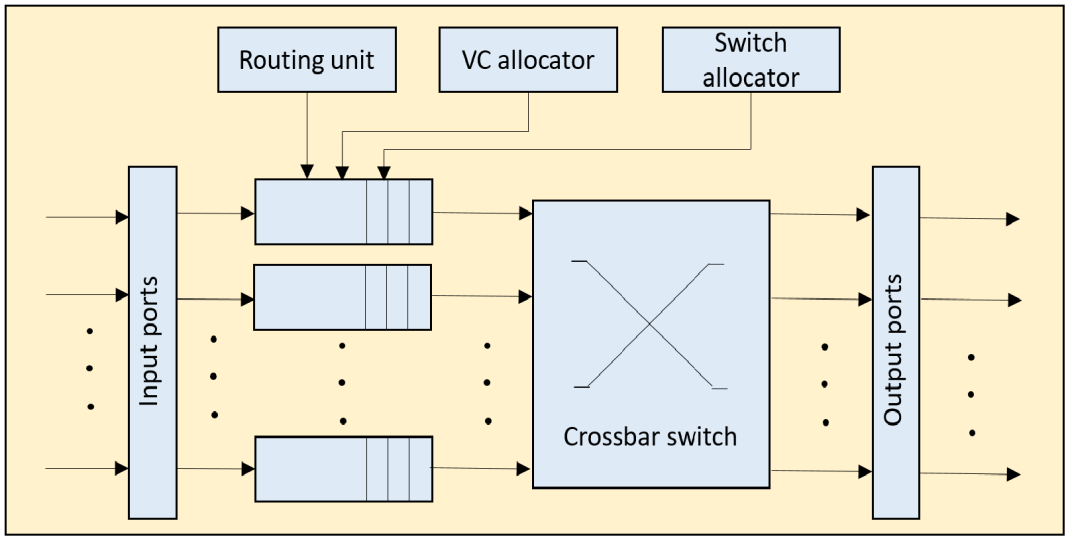

FIGURE 10. A router microarchitecture based on VC flow control.

work in [17] considers timing attacks initiated by multiple malicious routers to exploit information leakage. NoC monitors the bandwidth and generates alert messages if an attack is detected. Upon receiving the alert message, an alternate route is selected i.e., instead of XY, YX, is used. The same countermeasure is used for AES traffic leakage in [18]. However, this is done under the assumption that the malicious router is not aware of the alternate route. Furthermore, the countermeasures discussed in [17] and [42] are limited to XY routing only. Similarly, in [19], an obfuscation block is proposed to change the response time and add arbitrary delays to mask the intended behavior. However, area and power overheads incurred are 16% and 18%, respectively, which is relatively high compared to baseline NoC.

Encryption is the most widely used method to ensure integrity in NoCs. An encryption algorithm scrambles plaintext into incoherent text so that only authorized parties with the key can decipher the original text. Some of the most commonly used encryption algorithms in network security are AES, Triple Data encryption standard (3-DES), RSA, and Elliptic curve cryptography. Despite these conventional encryption methods, parts of the packet remain unencrypted, like the source, destination address, or prefix. A malicious node may sniff this information from the headers to launch various attacks. The work in [43] proposes a novel routing mechanism in which secure packets traverse through a pre-computed path. The destination address is encrypted so that it remains anonymous to the router. To ensure correct packet forwarding, path information such as XY direction and the number of permissible turns is encoded within the secure packet. A lightweight on-the-fly encryption mechanism is proposed in [44] in which the key, generated by the destination node, is split into two halves, enabling a key exchange with the intended source. While the right half of the key is sent using XY routing, the left half is sent via YX routing. Consequently, the source node sends acknowledgment two times for each half of the key through XY routing. For the proposed encryption architecture, DES, a symmetric key algorithm is proposed. The use of this algorithm ensures confidentiality and integrity of sensitive data at the expense of increased latency stemming from multiple key exchange rounds, which comes along with a 19% expansion in area and a 16.4% growth in power overhead.

Alternatively, a machine learning-based approach is proposed in [45] to detect eavesdropping (ED) attacks. The proposed scheme uses probes that are attached to the router to gather NoC traffic-related data. Subsequently, all data that is gathered is transmitted to a dedicated processing core, referred to as the decision unit, for further analysis. This analysis incorporates the implementation of an ED-sensing algorithm to discern the occurrence of ED attacks. Furthermore, the scheme, through the use of multiple instances of ML models that are deployed in all routers, facilitates a collective decision-making process to determine the occurrence of an ED attack. Although this approach achieves high accuracy within a very short time frame, the adoption of multiple ML models to identify a singular attack instance may introduce high latency, adversely impacting overall system performance.

In [46], a run time protector is proposed for each row in the 2×4 mesh topology to monitor routers and the acknowledgment packets they generate in response to the packets sent from the secure to the non-secure zone. Another approach described in this work uses a restart time protector, which pre-computes the time taken to reach the destination route and uses it to compute a counter threshold. A security auditor is alerted if no acknowledgment is received within the threshold. This security auditor gathers the routing

TABLE 3. Router security countermeasures.

Security Attack	Attack Impact	Countermeasures	Limitations
DoS Attack	Packet misrouting and Deadlock [41]	Trojan aware 3-phase routing process [42]	Tailored to XY routing limiting their adaptability.
		Secure zoning with run time monitoring and restart time protector [46]	Pre-computation of counter threshold setting contributes to an increased startup time.
Timing Attack	Traffic and Information leakage	Mask behaviour; obfuscation blocks and arbitrary delay [19]	Results in increased area and power overhead by 16% and 18% respectively.
		Gossip NoC; monitors link bandwidth and YX alternate routing [17], [18]	Tailored to XY routing limiting their adaptability.
Eavesdropping	Sensitive data extraction	Pre-computeation of path routing; encryption of destination address [43]	Tailored to XY routing limiting their adaptability.
		Collective decision making; using ML models [45]	Use of multiple instances of ML models lead to increased processing times and greater communication delays.
		DES algorithm; multiple rounds of key exchange [44]	Results in an increase in area and power overhead by 19% and 16.4% respectively.
		ECC and Flit scrambling [47]	Results in an increase in area and power overhead by 39% and 13% respectively.

table data from the malicious router to conduct a more thorough examination. Although these methods show a promising result, the increase in area overhead compared to a conventional router is significantly high, along with a corresponding increase in startup time. A mechanism to prevent packet tampering against malicious routers is proposed in [47]. This approach employs an integrity check with appropriate permutations on the flit to achieve the desired objective. Error control coding (ECC) encodes a packet's required fields, and permutation scrambles the flits, where each router operates a physically unclonable function (PUF) to select unique permutation patterns. In this case, additional modules are required to decode the incoming packets, resulting in a 39% increase in area and a 13% rise in power overhead.

Malicious routers can affect the processing element they are attached to and the packets that are routed through them. Consequently, routers themselves become sources from which malicious content is generated and spread throughout the network. Notably, XY routing is the most widely used routing algorithm, so most of the countermeasures are restricted to XY routing only, as shown in Table 3.

V. WIRELESS NOC SECURITY ATTACKS AND COUNTERMEASURES

Conventional NoCs are typically vulnerable at the processing core, network interface, and router. Furthermore, with wireless NoCs, wireless interfaces open up an additional attack point. Although all the countermeasures mentioned in the previous section also apply to wireless NoCs, security for wireless interfaces, in particular, requires careful consideration. While ECC improves communication reliability [6], it does not provide any guarantee to prevent attacks on wireless interfaces. Moreover, the implementation topology and features, such as broadcast capability, token-based channel access, etc., can be vulnerable to various attacks as well.

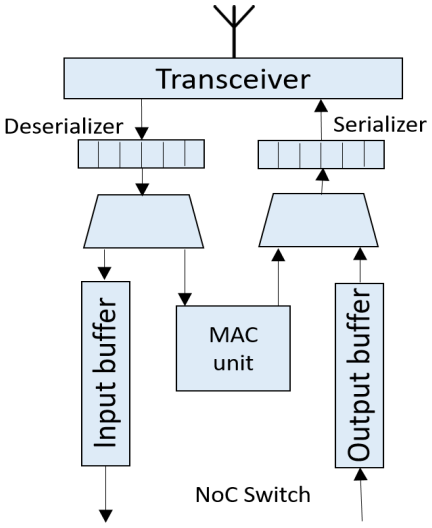

FIGURE 11. A wireless interface architecture.

A. WIRELESS INTERFACE (WI)

In wireless NoCs, wired links are used for short-distance communication, while single-hop wireless links are used for long-distance communication [9]. The use of wireless

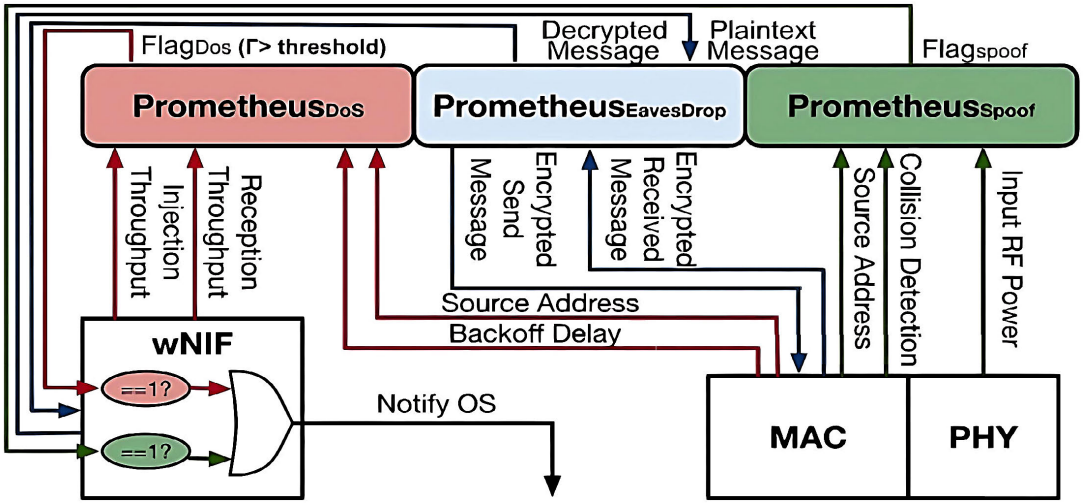

FIGURE 12. Prometheus microarchitecture [54].

interfaces bolsters energy efficiency improvements and provides an attractive high bandwidth alternative for long-distance communication [5], thereby eliminating most of the performance bottlenecks of wired I/Os. Figure 11 shows the wireless interface architecture. On-off keying modulation and medium access control (MAC) are the most popular techniques to ensure simple circuit and collision-free wireless communication. Token-based MAC is frequently used to allocate WI slots, making it an easy target for attacks that aim to disrupt communication through unauthorized access to data or tokens [48], [49]. The use of error control coding in wireless interfaces can improve communication reliability [50], but this does not prevent attacks from taking place. Although recently, the use of many new adaptive algorithms which are traffic aware has been considered to improve efficient utilization of underlying wireless bandwidth [51], [52], NoCs are still falling short of providing robust and reliable communication due to significant data loss and other pressing security related issues.

According to [53], the small-world topology is the most suitable network for wireless NoCs as it offers resistance to DoS attacks, resulting in high network availability. A "small-world topology or network" refers to a type of network topology that is characterized by short average path lengths between nodes, which means that most nodes can be reached from any other node in a relatively small number of steps. Simultaneously, small-world networks display a level of clustering, where nodes tend to have connections only with their nearby neighbors. This work explores network disruptions caused by hardware trojans, caused by the injection of junk packets into an IP core. As a result, to mitigate the impact of DoS attacks, the network setup is followed by simulated annealing (SA) heuristics. Subsequently, a metric quantifying the impact of the DoS attack is computed and then optimized to create a revised network configuration. Hence, a small-world network, characterized by an optimal average metric, is iteratively formed until the metric average nears zero. Despite achieving a high throughput, this approach does not take into consideration diverse traffic patterns and communication priorities of different application types.

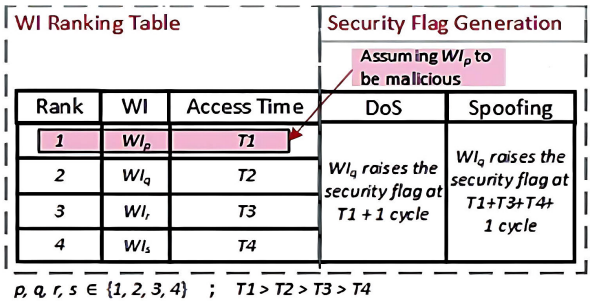

FIGURE 13. Ranking table and security flag generation [48], [49].

In [54], a secure wireless NoC architecture is proposed in which the underlying Operating System (OS) is equipped with robust security features to provide superior defense against spoofing, eavesdropping, and DoS attacks. As shown in Figure 12, a hardware module named Prometheus is designed to work in conjunction with a wireless interface. It is strategically located between the wireless network interface (wNIF) and the physical layer (PHY) of the network to monitor collisions. It derives an unfairness index by utilizing throughput and back-off delay as the two metrics to identify a selfish node. However, the threshold against which the ratio is compared must be optimized for each architecture. The architecture is additionally protected against spoofing and eavesdropping by RF analysis which is used to verify the packet ID and Py-based encryption, respectively.

Broadcast-capable wireless interfaces are considered in [55], which use Express Coherence Notification (ECONO) protocol to manage cache coherence. The proposed architecture performs an authentication check by comparing the

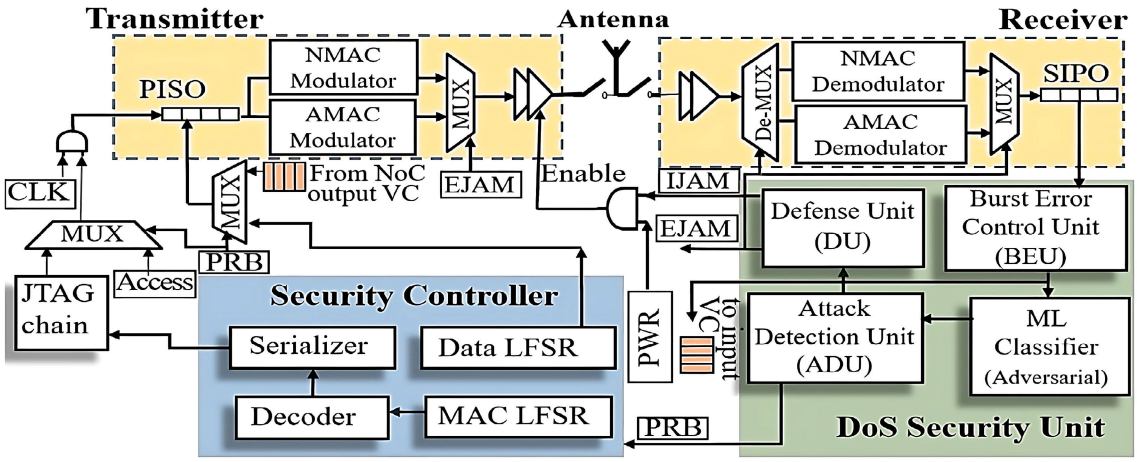

FIGURE 14. Proposed security framework [59], [60].

sender's and receiver's keys. A counter is used to track the number of transmissions, ensuring freshness. Additionally, to maintain integrity, the lightweight hashing algorithm SPONGENT is used. The name "Spongent" is derived from the contraction of "SPONGE" and "SUBSTITUENT," the two primary functions of the algorithm. Although these countermeasures prevent data modification by impersonation, eavesdropping, and replay attacks, they do not consider flooding caused by the repeated malicious injection of packets into the network, which incurs a performance overhead of 30%.

Apart from the traditional encryption algorithms, various other mechanisms have evolved to protect NoCs against various types of attacks. For example, in [56], the threat of spoofing is detected using the received signal power of wireless interconnections. However, this technique enforces placement limitations for nodes with wireless interfaces/antennas due to the constraints imposed by the equidistance algorithm. These placement limitations introduce considerable challenges in terms of performance, power, and area.

Wireless interface-based availability and integrity attacks are tackled in [48] and [49]. Shared single channels and allocation of slots using a token-based scheme are also considered here. By deploying a distributed channel ranking controller and majority voter within NI, a security mechanism to safeguard against DoS and spoofing attacks is conceived. Figure 13 illustrates the ranking table formed by the controller with the required channel access time of each WI. When access time is violated, the WI monitoring the transmission generates attack flags for notifying other NIs. The WIs examine the ranking table to confirm the correctness of the flag and, in response, generate a support flag. This mechanism operates as a form of a voting process. Through the collected responses, the controller identifies the malicious wireless interface and subsequently issues instructions to disable it. Similarly, spoofing is detected by monitoring the channel against a threshold idle time. In [57], various novel attacks, namely, Malicious threshold configuration attacks, Disruptive token passing attacks, Data stealing by broadcast attacks, and Hybrid attacks in wireless interfaces, are described. These attacks involve tampering with the router, NI, and wireless hub configuration (HC) registers to create various security vulnerabilities. The authors propose a defense mechanism against malicious configuration attacks, which involves implementing distance checks in routers to determine the shortest transmission path, along with using token-wait counters to check for the number of clock cycles without a token. In addition, packet transmission/reception counters in wireless hubs are used to maintain a tally of transmitted and received packets. Finally, threshold values are used to cross-check these counters and detect DoS and token-passing attacks. The hubs are deactivated when an attack is detected through the previously described mechanisms, and local cluster hub connection routers are informed of their unavailability. Using the wired NoC, packets in the hub connection routers waiting for wireless transmission are redirected to their destination node using a detour-based routing mechanism, as shown in Figure 15. However, these mitigation technologies are limited to a particular wireless architecture, such as a token-passing mechanism commonly used to regulate channel access.

Machine learning is another attractive solution which is capable of detecting attacks with a relatively high accuracy. Even though machine learning (ML) is widely employed in the framework of NoC designs for traffic congestion awareness [58], it has seldom been used for securing wireless NoCs. DoS attack based on persistent jamming has been discussed in [59] and [60] for wireless NoC interconnection. This work assumes that in the event of continuous jamming, the medium may no longer be available for legitimate use, thereby causing disruptions resulting in severe network performance degradation. These research studies aim to detect burst errors and then employ an ML classifier to sense

TABLE 4. Wireless interface security countermeasures.

Security Attack	Attack Impact	Countermeasures	Limitations
DoS Attack	Network jamming	Small-world topology with SA heuristics [53]	Diverse traffic patterns and communication priorities of different application types are not taken into consideration.
	Network collisions	Prometheus [54]; monitors transmission throughput	The technique needs to be optimized to suit different NoC architecture types.
	Access time manipulation	Distributed channel ranking controller and security flag generator [48], [49]	Tailored to suit token-based channel access mechanism limiting their adaptability.
		Monitors packet transmission, reception, and token access counts [57]	Tailored to suit token-based channel access mechanism limiting their adaptability.
	Persistent jamming	Burst error detection and ML-based attack classifier [59], [60]	Proposed countermeasure fails when packets are broadcast.
		PN encoded CDMA [61], [62]	Proposed countermeasure fails when packets are broadcast.
Spoofing	Impersonation	Prometheus [54]; RF-power analyzer	Impose placement constraints for WIs, thereby limiting the adaptability of this approach.
		RF-power profiler [56]	Impose placement constraints for WIs, thereby limiting the adaptability of this approach.
		Secret key authentication [55]	Results in an increased overhead by a factor of 30% contributing to lower overall performance.
		Distributed channel ranking controller security flag generator [48], [49]	Tailored to suit token-based channel access mechanism, thereby limiting the adaptability of approach.
Eavesdropping	Sensitive data leakage	Prometheus [54]; Py-based Encryption	The utilization of Py in high injection rate scenarios results in high latency overhead.
		SPONGENT [55]	Results in an increased overhead by a factor of 30% contributing to lower overall performance.
		XOR-based data scrambler and address checker [59]	Assumes a single attack source, thereby overlooking scenarios involving coordinated attacks by multiple sources.
		Data encoded with Pseudo-random Noise sequence [61], [62]	Assumes a single attack source, thereby overlooking scenarios involving coordinated attacks by multiple sources.

the attack as shown in Figure 14 and switch off wireless communication in wireless NoCs, using wired links instead of wireless interface. In contrast to earlier research efforts, the approaches here also consider external jamming attacks by transferring packet transfer control to the underlying wired NoC, thus withstanding attacks on wireless network communication. Additionally, external eavesdroppers are prevented from listening in by a data scrambler, and internal eavesdroppers are prevented by an address checker, followed by turning off the malicious WI. Nevertheless, broadcast packets would cause the network to get overloaded and eventually fail, as mentioned in [11].

A similar approach is taken in [61] and [62] for securing Network-in-Package (NiP). The architecture consists of wired mesh-based NoCs communicating with each other using wireless links. However, instead of switching off all WIs in the case of external jamming attacks, Code Division Multiple Access (CDMA) is utilized along with pseudo-random noise to encode the data. This filters out external jamming signals at the receiving end, preventing eavesdropping and thereby ensuring confidentiality. However, the proposed countermeasure assumes only one attacker, while multiple attackers could be involved. Table 4 summarizes all the countermeasure techniques employed in securing wireless interfaces against various types of attacks.

While there are multiple defense mechanisms in place to counter the attacks mentioned above, such as encryption, channel assessment, and burst error detection, there is a significant scarcity of all-encompassing solutions designed for the detection, mitigation, and pinpointing of wireless interface attacks.

VI. FUTURE DIRECTIONS AND CONCLUSION

The rapid development of silicon technologies is leading to the emergence of Network-on-Chip (NoC) as a promising on-chip communication infrastructure. Furthermore, the use of single-integrated antennas for intra-chip communication has led to the design of wireless NoC architecture.

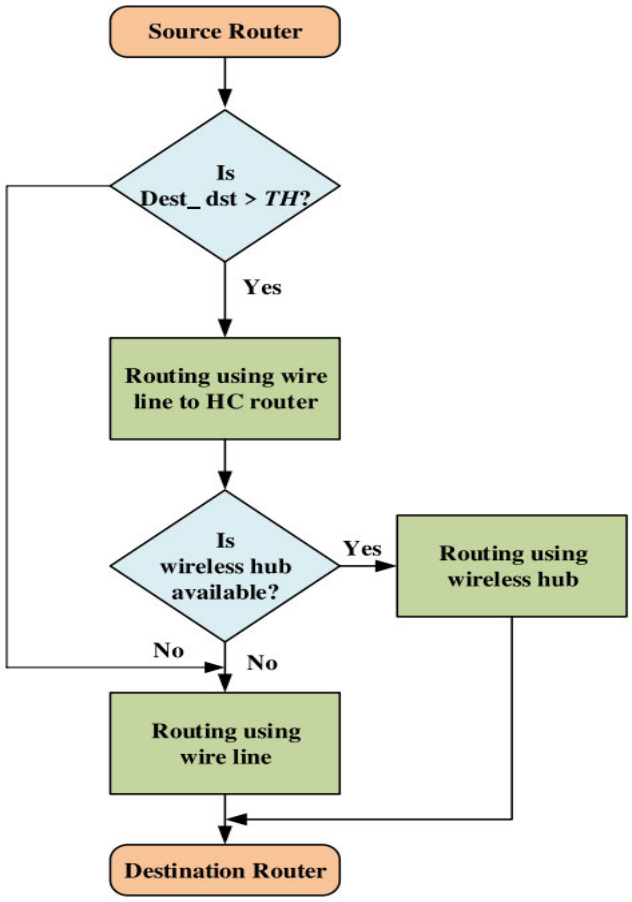

FIGURE 15. Detour-based routing mechanism [57].

Nonetheless, even with the growing attention on enhancing the security of wireless NoCs in recent times, devising effective countermeasures remains problematic due to the network's diverse nature, resource limitations, and dynamic attributes. This section summarizes various countermeasures discussed in this paper and provides future research directions.

A. SECURE ZONE

Although secure zoning offers a promising solution to NoC security by ensuring confidentiality, some challenges are associated with it. An example is the hardware overhead a static security zone adds to an architecture [27], [29], [46]. Although the hardware overhead in an NoC system may be minimal, the need for added security in interzone communication using shared resources results in a limitation of communication routes and an augmentation of processing overhead. A dynamic, secure zone overcomes this limitation [28], [30], [31]. However, this approach tends to introduce delays as secure zone allocation is done at run time. In this context, it is imperative to find a balance between performance and security using encryption and trust. Consequently, there is significant scope for further research in the secure zone field of NoC design.

B. ENCRYPTION AND DECRYPTION

The research results of [23] and [37] reveal that the use of encryption techniques has an impact on the system bandwidth since the available bandwidth decreases with a corresponding increase in encryption key size. However, the use of such encryption techniques is necessary to achieve the desired levels of integrity and confidentiality. Many schemes, such as those outlined in [44] and [47], divide encryption keys and data scrambling to minimize overhead. Even though these are lightweight encryption methods, they could also add several rounds of key exchange and handshakes. Consequently, the suitability of these techniques for use in NoC must be carefully studied and examined.

C. MACHINE LEARNING

In recent years, Machine Learning algorithms and techniques for detecting various security anomalies have gained some traction. Continuing with this trend, ML-based approaches are found to be especially useful for detecting abnormal traffic patterns in NoC architectures, and many approaches employ machine learning-based techniques to detect DoS attacks [26], [58], [59] and eavesdropping [45]. However, the accuracy of such techniques is achieved at the cost of memory and processing time since such techniques normally tend to aggressively consume limited resources at a phenomenal rate. One possible future research direction would be to develop resource-aware ML algorithms that are specifically designed to address the underlying NoC limitations and requirements.

Various countermeasures are used as defense mechanisms to safeguard NoC from malicious attacks, and yet only a handful of these can be used to accurately identify the attacker, especially when the attack originates from multiple sources concurrently.

In summary, this article provides a detailed examination of various different security vulnerabilities and attacks on NoC architecture. In particular, the article examines novel types of attacks that specifically emanate in a wireless NoC environment and explores the current state-of-the-art countermeasures that are employed to detect, prevent, and mitigate these attacks. Some countermeasures that are common to both wired and wireless NoC links are also included in this study. However, despite these countermeasures, several challenges still exist. This survey brings to light the prevailing security-related challenges in this area, which need to be urgently addressed for the wider adoption of wireless technology in NoC in the near future.

REFERENCES

[1] S. D. Chawade, M. A. Gaikwad, and R. M. Patrikar, "Review of XY routing algorithm for network-on-chip architecture," *Int. J. Comput. Appl.*, vol. 43, pp. 20–23, Oct. 2012.
[2] H. Weerasena and P. Mishra, "Security of electrical, optical and wireless on-chip interconnects: A survey," 2023, *arXiv:2301.09738*.
[3] W.-C. Tsai, Y.-C. Lan, Y.-H. Hu, and S.-J. Chen, "Networks on chips: Structure and design methodologies," *J. Electr. Comput. Eng.*, vol. 2012, pp. 1–15, 2012.

[4] I. A. Alimi, R. K. Patel, O. Aboderin, A. M. Abdalla, R. A. Gbadamosi, N. J. Muga, A. N. Pinto, and A. L. Teixeira, "Network-on-chip topologies: Potentials, technical challenges, recent advances and research direction," in *Network-on-Chip: Architecture, Optimization, and Design Explorations*. London, U.K.: IntechOpen, 2021.

[5] P. P. Pande, A. Ganguly, K. Chang, and C. Teuscher, "Hybrid wireless network on chip: A new paradigm in multi-core design," in *Proc. 2nd Int. Workshop Netw. Chip Architectures*, Dec. 2009, pp. 71–76.

[6] S. Wang and T. Jin, "Wireless network-on-chip: A survey," *J. Eng.*, vol. 2014, no. 3, pp. 98–104, 2014.

[7] S. D. Chawade, M. A. Gaikwad, and R. M. Patrikar, "Review of XY routing algorithm for network-on-chip architecture," *Int. J. Comput. Appl.*, vol. 43, pp. 20–23, Oct. 2012.

[8] S. Deb, A. Ganguly, P. P. Pande, B. Belzer, and D. Heo, "Wireless NoC as interconnection backbone for multicore chips: Promises and challenges," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 2, no. 2, pp. 228–239, Jun. 2012.

[9] P. Wettin, P. P. Pande, D. Heo, B. Belzer, S. Deb, and A. Ganguly, "Design space exploration for reliable mm-wave wireless NoC architectures," in *Proc. IEEE 24th Int. Conf. Appl.-Specific Syst., Architectures Processors*, Jun. 2013, pp. 79–82.

[10] A. Ganguly, N. Mansoor, M. S. Shamim, M. M. Ahmed, R. S. Narde, A. Vashist, and J. Venkataraman, "Intra-chip wireless interconnect: The road ahead," in *Proc. 10th Int. Workshop Netw. Chip Architectures*, Oct. 2017, p. 6.

[11] A. Sarihi, A. Patooghy, A. Khalid, M. Hasanzadeh, M. Said, and A. A. Badawy, "A survey on the security of wired, wireless, and 3D network-on-chips," *IEEE Access*, vol. 9, pp. 107625–107656, 2021.

[12] S. Charles and P. Mishra, "A survey of network-on-chip security attacks and countermeasures," *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–36, May 2021.

[13] Z. Pan and P. Mishra, "A survey on hardware vulnerability analysis using machine learning," *IEEE Access*, vol. 10, pp. 49508–49527, 2022.

[14] M. Fyrbiak, S. Wallat, P. Swierczynski, M. Hoffmann, S. Hoppach, M. Wilhelm, T. Weidlich, R. Tessier, and C. Paar, "HAL—The missing piece of the puzzle for hardware reverse engineering, trojan detection and insertion," *IEEE Trans. Depend. Sec. Comput.*, vol. 16, no. 3, pp. 498–510, May 2019.

[15] P. Mishra and S. Charles, "Trustworthy system-on-chip design using secure on-chip communication architectures," in *Network-on-Chip Security and Privacy*. Cham, Switzerland: Springer, 2021.

[16] J. Wei and J. Wei, "Survey of network and computer attack taxonomy," in *Proc. IEEE Symp. Robot. Appl. (ISRA)*, Jun. 2012, pp. 294–297.

[17] C. Reinbrecht, A. Susin, L. Bossuet, and J. Sepúlveda, "Gossip NoC—Avoiding timing side-channel attacks through traffic management," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jul. 2016, pp. 601–606.

[18] C. Reinbrecht, A. Susin, L. Bossuet, G. Sigl, and J. Sepúlveda, "Side channel attack on NoC-based MPSoCs are practical: NoC prime+probe attack," in *Proc. 29th Symp. Integr. Circuits Syst. Design (SBCCI)*, Aug. 2016, pp. 1–6.

[19] C. Reinbrecht, A. Aljuffri, S. Hamdioui, M. Taouil, B. Forlin, and J. Sepulveda, "Guard-NoC: A protection against side-channel attacks for MPSoCs," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jul. 2020, pp. 536–541.

[20] S. Roshanisefat, H. M. Kamali, H. Homayoun, and A. Sasan, "SAT-hard cyclic logic obfuscation for protecting the IP in the manufacturing supply chain," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 4, pp. 954–967, Apr. 2020.

[21] L. Daoud, "Secure network-on-chip architectures for MPSoC: Overview and challenges," in *Proc. IEEE 61st Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2018, pp. 542–543.

[22] T. Boraten and A. K. Kodi, "Packet security with path sensitization for NoCs," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2016, pp. 1136–1139.

[23] J. Sepúlveda, A. Zankl, D. Flórez, and G. Sigl, "Towards protected MPSoC communication for information protection against a malicious NoC," *Proc. Comput. Sci.*, vol. 108, pp. 1103–1112, Jan. 2017.

[24] S. Ellinidou, G. Sharma, O. Markowitch, G. Gogniat, and J.-M. Dricot, "A novel network-on-chip security algorithm for tolerating Byzantine faults," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFT)*, Oct. 2020, pp. 1–6.

[25] S. Charles, Y. Lyu, and P. Mishra, "Real-time detection and localization of distributed DoS attacks in NoC-based SoCs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 12, pp. 4510–4523, Dec. 2020.

[26] K. Madden, J. Harkin, L. McDaid, and C. Nugent, "Adding security to networks-on-chip using neural networks," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 1299–1306.

[27] J. Sepúlveda, D. Flórez, and G. Gogniat, "Efficient and flexible NoC-based group communication for secure MPSoCs," in *Proc. Int. Conf. Reconfigurable Comput. FPGAs (ReConFig)*, Dec. 2015, pp. 1–6.

[28] M. M. Real, P. Wehner, V. Migliore, V. Lapotre, D. Göhringert, and G. Gogniat, "Dynamic spatially isolated secure zones for NoC-based many-core accelerators," in *Proc. 11th Int. Symp. Reconfigurable Commun.-Centric Syst.-on-Chip (ReCoSoC)*, Jun. 2016, pp. 1–6.

[29] L. L. Caimi, V. Fochi, E. Wachter, D. Munhoz, and F. G. Moraes, "Activation of secure zones in many-core systems with dynamic rerouting," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.

[30] L. L. Caimi, V. Fochi, E. Wachter, and F. G. Moraes, "Runtime creation of continuous secure zones in many-core systems for secure applications," in *Proc. IEEE 9th Latin Amer. Symp. Circuits Syst. (LASCAS)*, Feb. 2018, pp. 1–4.

[31] S. P. Azad, M. Tempelmeier, G. Jervan, and J. Sepúlveda, "CAESAR-MPSoC: Dynamic and efficient MPSoC security zones," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jul. 2019, pp. 477–482.

[32] B. Aghaei, M. Reshadi, M. Masdari, S. H. Sajadi, M. Hosseinzadeh, and A. Darwesh, "Network adapter architectures in network on chip: Comprehensive literature review," *Cluster Comput.*, vol. 23, no. 1, pp. 321–346, Mar. 2020.

[33] S. Saponara, T. Bacchillone, E. Petri, L. Fanucci, R. Locatelli, and M. Coppola, "Design of an NoC interface macrocell with hardware support of advanced networking functionalities," *IEEE Trans. Comput.*, vol. 63, no. 3, pp. 609–621, Mar. 2014.

[34] A. Shalaby, Y. Tavva, T. E. Carlson, and L.-S. Peh, "Sentry-NoC: A statically-scheduled NoC for secure SoCs," in *Proc. 15th IEEE/ACM Int. Symp. Netw.-on-Chip (NOCS)*, Oct. 2021, pp. 67–74.

[35] V. Y. Raparti and S. Pasricha, "Lightweight mitigation of hardware trojan attacks in NoC-based manycore computing," in *Proc. 56th ACM/IEEE Design Autom. Conf. (DAC)*, Jun. 2019, pp. 1–6.

[36] J. Frey and Q. Yu, "Exploiting state obfuscation to detect hardware trojans in NoC network interfaces," in *Proc. IEEE 58th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2015, pp. 1–4.

[37] K. Sajeesh and H. K. Kapoor, "An authenticated encryption based security framework for NoC architectures," in *Proc. Int. Symp. Electron. Syst. Design*, Dec. 2011, pp. 134–139.

[38] G. Dimitrakopoulos, A. Psarras, and I. Seitanidis, *Microarchitectures of Network-on-Chip Routers*. New York, NY, USA: Springer, 2015.

[39] B. Chemli and A. Zitouni, "Design and evaluation of optimized router pipeline stages for network on chip," in *Proc. Int. Image Process., Appl. Syst. (IPAS)*, Nov. 2016, pp. 1–5.

[40] L. Daoud and N. Rafla, "Analysis of black hole router attack in network-on-chip," in *Proc. IEEE 62nd Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2019, pp. 69–72.

[41] V. J. Kulkarni, R. Manju, R. Gupta, J. Jose, and S. Nandi, "Packet header attack by hardware trojan in NoC based TCMP and its impact analysis," in *Proc. 15th IEEE/ACM Int. Symp. Netw.-on-Chip (NOCS)*, Oct. 2021, pp. 21–28.

[42] R. Manju, A. Das, J. Jose, and P. Mishra, "SECTAR: Secure NoC using trojan aware routing," in *Proc. 14th IEEE/ACM Int. Symp. Netw.-on-Chip (NOCS)*, Sep. 2020, pp. 1–8.

[43] A. Sarihi, A. Patooghy, M. Hasanzadeh, M. Abdelrehim, and A. A. Badawy, "Securing network-on-chips via novel anonymous routing," in *Proc. 15th IEEE/ACM Int. Symp. Netw.-on-Chip (NOCS)*, Oct. 2021, pp. 29–34.

[44] A. Sarihi, A. Patooghy, M. Hasanzadeh, M. Abdelrehim, and A. A. Badawy, "Securing on-chip communications: An on-the-fly encryption architecture for SoCs," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2021, pp. 741–746.

[45] C. Sudusinghe, S. Charles, S. Ahangama, and P. Mishra, "Eavesdropping attack detection using machine learning in network-on-chip architectures," *IEEE Design Test.*, vol. 39, no. 6, pp. 28–38, Dec. 2022.

[46] A. K. Biswas, S. K. Nandy, and R. Narayan, "Network-on-chip router attacks and their prevention in MP-SoCs with multiple trusted execution environments," in *Proc. IEEE Int. Conf. Electron., Comput. Commun. Technol. (CONECCT)*, Jul. 2015, pp. 1–6.

[47] J. Frey and Q. Yu, "A hardened network-on-chip design using runtime hardware trojan mitigation methods," *Integration*, vol. 56, pp. 15–31, Jan. 2017.
[48] S. S. Rout, A. Singh, S. B. Patil, M. Sinha, and S. Deb, "Security threats in channel access mechanism of wireless NoC and efficient countermeasures," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5.
[49] S. Pasricha, J. Jose, and S. Deb, "Electronic, wireless, and photonic network-on-chip security: Challenges and countermeasures," *IEEE Design Test.*, vol. 39, no. 6, pp. 90–98, Dec. 2022.
[50] D. DiTomaso, A. Kodi, D. Matolak, S. Kaya, S. Laha, and W. Rayess, "A-WiNoC: Adaptive wireless network-on-chip architecture for chip multiprocessors," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 12, pp. 3289–3302, Dec. 2015.
[51] M. Devanathan, V. Ranganathan, and P. Sivakumar, "Congestion-aware wireless network-on-chip for high-speed communication," *Automatika*, vol. 61, no. 1, pp. 92–98, Jan. 2020.
[52] Q. Gao, W. Song, Z. Lu, L. Li, and Y. Fu, "Dynamic and traffic-aware medium access control mechanisms for wireless NoC architectures," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.
[53] A. Ganguly, M. Y. Ahmed, and A. Vidapalapati, "A denial-of-service resilient wireless NoC architecture," in *Proc. Great Lakes Symp. VLSI*, May 2012, pp. 259–262.
[54] B. Lebiednik, S. Abadal, H. Kwon, and T. Krishna, "Architecting a secure wireless network-on-chip," in *Proc. IEEE/ACM Int. Symp. Netw.-on-Chip (NOCS)*, Oct. 2018, pp. 1–8.
[55] F. Pereñíguez-García and J. L. Abellán, "Secure communications in wireless network-on-chips," in *Proc. 2nd Int. Workshop Adv. Interconnect Solutions Technol. Emerg. Comput. Syst.*, Jan. 2017, pp. 27–32.
[56] B. Lebiednik, S. Abadal, H. Kwon, and T. Krishna, "Spoofing prevention via RF power profiling in wireless network-on-chip," in *Proc. 3rd Int. Workshop Adv. Interconnect Solutions Technol. Emerg. Comput. Syst. (AISTECS)*, 2018, pp. 1–4.
[57] A. K. Biswas, N. Chatterjee, H. K. Mondal, G. Gogniat, and J.-P. Diguet, "Attacks toward wireless network-on-chip and countermeasures," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 2, pp. 692–706, Apr. 2021.
[58] E. Kakoulli, V. Soteriou, and T. Theocharides, "Intelligent hotspot prediction for network-on-chip-based multicore systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 3, pp. 418–431, Mar. 2012.
[59] A. Vashist, A. Keats, S. M. P. Dinakarrao, and A. Ganguly, "Securing a wireless network-on-chip against jamming-based denial-of-service and eavesdropping attacks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 12, pp. 2781–2791, Dec. 2019.
[60] A. Vashist, A. Keats, S. M. P. Dinakarrao, and A. Ganguly, "Unified testing and security framework for wireless network-on-chip enabled multi-core chips," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 5s, pp. 1–20, Oct. 2019.
[61] M. Ahmed, A. Vashist, S. M. P. Dinakarrao, and A. Ganguly, "Architecting a secure wireless interconnect for multichip communication: An ML approach," in *Proc. Asian Hardw. Oriented Secur. Trust Symp. (AsianHOST)*, Dec. 2020, pp. 1–6.
[62] M. M. Ahmed, A. Ganguly, A. Vashist, and S. M. P. Dinakarrao, "AWARe-Wi: A jamming-aware reconfigurable wireless interconnection using adversarial learning for multichip systems," *Sustain. Comput., Informat. Syst.*, vol. 29, Mar. 2021, Art. no. 100470.

LASHMI KONDOTH received the bachelor's degree in electronics and communication engineering from the University of Calicut, in 2017, and the master's degree in embedded systems from Amrita University, India, in 2019. From 2019 to 2022, she was an Application Engineer with Microchip Technology, India. She is currently a Research Scholar with the School of Computing, Macquarie University, Australia. Her research interests include network-on-chip, wireless communication, and the Internet of Things. She was a recipient of the International Research Training Program Scholarship.

RAJAN SHANKARAN (Senior Member, IEEE) received the M.B.A. (MIS) degree in information systems from the Maastricht School of Management, in 1994, and the M.Sc. (Hons.) and Ph.D. degrees in network communications and security from the University of Western Sydney, in 1999 and 2005, respectively. He is currently the Course Director of the School of Computing, Macquarie University, Sydney, Australia. His primary focus areas encompass D2D communications, medical implant security, network security, mobile computing, and on-chip networks. He has actively participated as a program committee member in numerous computer networking and security conferences. He served as a Technical Program Committee Member for WMNC 2017, Mobility 2017, and WCNC 2018. Additionally, he contributed to the organization of IEEE/IFIP NOMS 2017 and CNSM 2017.

QUAN Z. SHENG (Member, IEEE) is currently a Professor and the Head of the School of Computing, Macquarie University, Australia. Before moving to Macquarie University, he spent ten years with the School of Computer Science, The University of Adelaide, serving in a number of senior leadership roles, including the Acting Head and the Deputy Head of the School of Computer Science. He is the Associate Director of the Smart Green Cities Research Center, Macquarie University. His research interests include the Internet of Things (IoT), services computing, big data analytics, machine learning, and web technologies. He is a member of the Australian Computer Society (ACS) Technical Advisory Board on IoT. He was a recipient of the AMiner Most Influential Scholar on IoT, in 2018; the Australian Research Council (ARC) Future Fellowship, in 2014; the Chris Wallace Award for Outstanding Research Contribution, in 2012; and the Microsoft Research Fellowship, in 2003. He is ranked by Microsoft Academic as one of the Most Impactful Author in Services Computing (ranked top 5 all time) and in Web of Things (ranked top 20 all time). He is the Vice Chair of the Executive Committee of the IEEE Technical Community on Services Computing (IEEE TCSVC).

RICHARD HAN received the B.Sc. degree (Hons.) in electrical engineering from Stanford University, in 1989, and the Ph.D. degree in electrical engineering from the University of California at Berkeley, in 1997. He was a Research Staff Member with the IBM Thomas J. Watson Research Center, from 1997 to 2001. He was an Assistant, an Associate, and a Full Professor with the Department of Computer Science, University of Colorado Boulder, from 2001 to 2020. He was a Professor of networks and distributed systems with the School of Computing, Macquarie University, in 2021. His research interests include mobile and ubiquitous computing, wireless sensor networks, drone systems, cloud computing, cybersecurity, operating systems, and cybersafety. His research has received four best paper awards or equivalents and more than 13000 citations according to Google Scholar. He has also received the NSF CAREER Award, the IBM Faculty Awards, and the Frank Moyes Award in entrepreneurship.