EMITTER International Journal of Engineering Technology
Vol. 10, No. 2, December 2022, pp. 370~388
DOI : 10.24003/emitter.v10i2.716

Hardware Trojan Detection and Mitigation in NoC using Key authentication and Obfuscation Techniques

Thejaswini P[1], Vivekananda G[1,2], Anu H[1], Priya R[1], Krishna Prasad B S[1], Nischay M[1]

[1]Electronics and Communication Engineering Department, JSS Academy of Technical Education, Bangalore
[2]Computer Science and Engineering Department, Indian Institute of Technology Guwahati
Correspondence Author: thejaswinip@jssateb.ac.in

Received August 20, 2022; Revised September 22, 2022; Accepted October 25, 2022

Abstract

Today's Multiprocessor System-on-Chip (MPSoC) contains many cores and integrated circuits. Due to the current requirements of communication, we make use of Network-on-Chip (NoC) to obtain high throughput and low latency. NoC is a communication architecture used in the processor cores to transfer data from source to destination through several nodes. Since NoC deals with on-chip interconnection for data transmission, it will be a good prey for data leakage and other security attacks. One such way of attacking is done by a third-party vendor introducing Hardware Trojans (HTs) into routers of NoC architecture. This can cause packets to traverse in wrong paths, leak/extract information and cause Denial-of-Service (DoS) degrading the system performance. In this paper, a novel HT detection and mitigation approach using obfuscation and key-based authentication technique is proposed. The proposed technique prevents any illegal transitions between routers thereby protecting data from malicious activities, such as packet misrouting and information leakage. The proposed technique is evaluated on a 4x4 NoC architecture under synthetic traffic pattern and benchmarks, the hardware model is synthesized in Cadence Tool with 90nm technology. The introduced Hardware Trojan affects 8% of packets passing through infected router. Experimental results demonstrate that the proposed technique prevents those 10-15% of packets infected from the HT effect. Our proposed work has negligible power and area overhead of 8.6% and 2% respectively.

Keywords: Network-on-Chip (NoC), Hardware Security, Hardware Trojan, Obfuscation, Misrouting.

1. INTRODUCTION

System-on-chip (SoC) is the heart of computing systems, dominating the electronic/mobile computing market becoming an integral part of our

daily life [1][2]. Sophisticated SoCs, a MultiProcessor System-on-Chip (MPSoC) encompass thousands of integrated processing units into a single IC including a central processing unit (CPU), graphics and memory interfaces with a promising high throughput, low latency and preferably low energy utilization [3]. As the dimensions of semiconductor chips are shrinking and more IP cores are added to them, many SoC chips are too complex to utilize traditional data bus or crossbar interconnect approach for communication [4]. The physical resources carrying data on the chips and quality of services begin to crumble due to large number of wires, reduced chip reliability, increased energy consumption and electro-magnetic interference. The notion of employing shared bus architecture to solve these drawbacks lacks scalability. [5]. NoC has gained widespread acceptance as a suitable interconnection mechanism for MPSoCs. as it provides structured architecture, lowering the complexity and cost of chip design. It brings notable improvements over traditional data bus and crossbar communication architectures. NoC scheme allows reuse of components, architectures, design methods and tools by organizing communication between operating modules located on the same chip [6]. This scheme simplifies switching, routing functions to achieve higher operating frequencies by establishing a relationship between the processor blocks. It also provides flexibility, easy accommodation of changes which improves the scalability of communication architecture and power efficiency of complex SoCs. NoC provides a clear demarcation between computation and communication in MPSoC [7]. It significantly reduces wire routing congestion and avoids timing closure issues in a SoC thereby increasing efficiency and high-performance interconnection.

With substantial pressure of time-to-market, reduction in design costs and in the wake of globalized semiconductor enterprise model, modern MPSoCs combine several distinct pre-verified hard or soft IP blocks from a wide pool of untrustworthy Third Party Intellectual Property (3PIP) providers. This has given rise to Hardware Trojans (HTs), a malicious modification by an adversary. These modifications are unknown and undesired to hardware designers which can have system-wrecking impact which poses threat to the security and reliability of a whole chip [8]. Chip size, unpredictability and integration density will make future NoCs progressively helpless. So it is vital to address the presence of HTs in 3PIP NoC itself. For example, in 2011 United States got to know that the microchips they had brought for military application from transponder to missiles from chip was found to be counterfeit. Report says that microchips could have been shut down remotely at any time.

Hardware Trojans have three key characteristics: rarity of activation, evasion of detection and malicious intention [9]. The primary goal of an HTs has always been the same, which is to carry out an unanticipated task to middle ground the authentication, confidentiality and integrity of the hardware. A unique mechanism activates these HTs, referred to as the

payload. These HTs exist in many forms, small or large concerning the rest of the circuit which is often undetectable in manufacturing testing and verification processes [10]. They can cause threats to the system. HT detection and mitigation techniques for general ICs were investigated in the past decade. However, effective remedies that primarily strengthen the NoC architecture are lacking [11]. One significant problem is to enable safe and reliable communication in the SoC even in the presence of untrusted IPs, as a hacked 3PIP may cause havoc in the SoC by launching a slew of assaults due to HTs embedded in it [1]. Because of its unique and fundamental function in SoC communication, the NoC has direct access to all SoC resources and information. The HTs placed into NoCs will result in unauthorized memory access, data corruption, information leakage and Denial-of-Service (DoS) attacks, like deadlock, livelock and incorrect routing [11]. Compared to the prevailing techniques, our technique takes into account all of the NoC characteristics, consisting of parallel communication links, scalability and high modularity, wherein we reinforce the NoC functionality to resist potential HT attacks. Hence, we employ an Obfuscation technique to prevent the effects of HTs in NoC. As the effect of Hardware Trojans and our proposed methodology is independent of size of architecture. So, for ease of implementation and interpretation of results we have considered a standard 4x4 mesh NoC as shown in Figure 1. The HT model considered in our work misroutes the packets in the network and leaks information to a malicious receiver node.

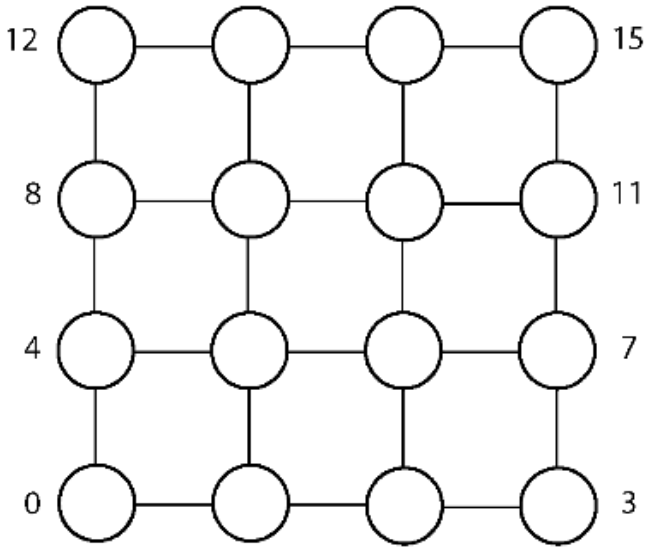

Figure 1. 4 x 4 mesh NoC

To secure NoC from the adverse effect of HTs, we propose an approach for detecting and mitigating HT during runtime that involves two main phase. In the first phase, a key-based encryption technique to detect and prevent any illegal transitions between the routers. In the second phase, we make use

of bit-obfuscation technique wherein the true functionality of the circuit is hidden is proposed to protect the data from malicious activities [12] such as packet loss and information leakage. We make the following significant contributions in this paper:

- The implementation of packet misrouting and information leakage HTs in NoC leads to performance degradation.
- Our key based authentication and bit-obfuscation detect misrouting and information leakage HTs.
- We experimentally demonstrate that our technique effectively mitigates the effect of HTs and route packets to the proper destination address.

The remainder of this work is organized as follows. We summarize the existing HT detection and mitigation methods for NoC in Section 2. Section 3 discusses the motivation of the proposed method. The HT insertion scenarios, HT detection by key based authentication and mitigation by obfuscation technique are described in Section 4. Section 5 presents the detailed experimental data gained from the simulations and synthesis performed. Conclusion is discussed in section 6.

2. RELATED WORKS

Significant research is carried out to increase robustness against HT attacks in NoC by presuming that HT leaks or peeps data, cause Denial of Service (DoS). Dynamic packet tagging and tag scrambling [13], use two techniques with customization in NoC at runtime to prevent the effects of HTs. The tagging of packet which is dynamic in nature will generate tags by making use of the minimized or lookup table designs and the word-based rotations for data scrambling for lower overhead. This work offered a good detection rate with a trade-off with critical over of storing lookup tables for computing the tag information for each packet.

In EETD: An Energy Efficient Design for Runtime Hardware Trojan Detection [14], a security structure for NoC based architectures is proposed wherein he work combined encryption, authentication, and network coding methods. The proposed scheme primarily focuses on restricting the unauthorised access to important information contained inside a secure core. The performance overhead introduced is relatively minimal and doesn't impact on the overall performance of the system but the mitigation model is done using the same tampered network which in in turn exposed to HT attacks.

A mechanism to defend against the HTs [15] is proposed along with four different HTs, where a complicated bit pattern in the incoming data causes the Trojan to fool packets from the HTs . A mitigation method is presented to reduce the adverse effects of Trojan by a bit-shuffling mechanism within the router and a directly extracted key from the input data. On a 4x4 NoC, the proposed HT and mitigation system are demonstrated. The simulation results show that the proposed strategy is

effective in preventing Trojan attacks with a negligible area overhead in comparison with local processor.

A new lightweight Target-Activated Sequential Payload (TASP) Hardware Trojan model is introduced [16] in a system that inspects the packets and injects faults in order to launch a DoS attack. A numerous switch-to-switch link obfuscation techniques including shuffling, inverting and scrambling within each router is proposed to avoid triggering HTs which retains the usage of links instead of rerouting packets. To prevent HT from getting activated and hence gracefully reduce network performance rather than disconnecting links and rerouting around them, threat detection and mitigation via proposed L-Ob s2s obfuscation is necessary.

For mitigation of HTs which snoops the data from the packets, a light weight mitigation technique has been proposed [17]. The packet here is being trapped in the Network Interface of the NoC itself and the information is being collected by an attacker or the packet is trapped until the next packet overwrites it. The proposed technique notifies this kind of packets and updates in a packet incoming and outgoing ratio for each router to dodge HT affected routers. This work provides significantly stronger protection with reduced overheads than recently proposed approaches.

In Exploiting state obfuscation to detect Hardware Trojans in NoC network interfaces [18], key bits are added to the Finite State Machine (FSM) of Network Interface (NI) control unit and creates the dummy states to enhance the toughness for the accomplishment of Trojan activities. An illegal state and restricted state transition induced by an incorrect key are examined and HTs occurrences are detected. The suggested solution is intended for runtime HT detection, and it is designed to complement existing offline HT detection methods.

For protection of hardware IP using data obfuscation and securing at System-Level, Sentry-NoC [19] exploited the NoC constraints in the traffic flow. The conventional protocol isolated the traffic and doesn't interfere in communication. The paper proposed temporary data obfuscation which is not dynamic and infused with the rime domain multiplexing in NoCs, this led to randomize the source and destination patterns of the packets in turn scrambling the switching activity over the links. This work has good resilience towards the side channel attacks but remain vulnerable to most of the on chip HT attacks.

In Routing Aware and Runtime Detection [20], a new routing algorithm is proposed to combat against the HTs which caused DoS attacks. The proposed work detects the HT from the router's incoming port and the destination of the packets in the router. Once the HT router is identified the information is propagated to the neighbouring routers and the routing algorithm is obtained to effectively misroute the packets away from the HT. This work will know the exact location of the HT in the network, in some advanced HTs the information passing to neighbours can also be tampered and the degradation of the performance by misrouting is also a minus.

In Fortified NoC [21], a robust approach to prevent HT is proposed. The prevention is focused n NI level with 3-layer protection for securing the information and resources. A new routing algorithm has also been proposed which isolated the HT from packets being transmitted through it. The data in the packet is being scrambled to mislead the HT by bypassing these will prevent data leakage and snooping. This work is not effective in prevention against illegal data request and data flushing also the overhead is more which takes more time for processing.

3. ORIGINALITY

The current trends in technology scaling has empowered the integration of hundreds of processing units into a single IC chip. NoC has been an effective solution for the fast parallel communication between processing elements. As all information that must be transmitted must pass through routers, the attackers can infect routers and release the sensitive information they require. Attackers can easily insert HTs in routers and make packets to misroute to their desired router and leak the information there. This has been the most common procedure for attackers to illegally access the data. These attacks are all possible when the system is vulnerable. Improper protection of data and improper restriction of unauthorized activities/access are major concerns.

One possible way of preventing illegal access to information is by using key-based authentication which checks the key in each router and proceeds further if the key is correct [22]. Many such key-based authentications are in use and they provide security to a good extent. The key lengths are 3 to 5 bits wide due to resource or data width limits in the packets used to store keys. So, these keys can be cracked by any random combination within a few attempts. To have higher security dynamic keys are being used, wherein keys change with each hop in the routers.

Another way of misleading attackers is to obfuscate the packets. This process hides the true identity of the packet by changing any information which can be reversed whenever required by the trusted region [23]. Presently, there are many types of obfuscation techniques some of them are: shuffling, scrambling, rotating, inverting, etc. Out of all these techniques inverting is the more secure option. When a certain combination of data or information is shuffled or rotated it gives back the same data or information. We employ the inverting obfuscation approach to prevent and produce a distinct and different output regarding the input packet to be obscured.

To have a more secure transmission of packets, a combination of key-based authentication and obfuscation is used. In this combination we first create the dynamic key generation in every router and the packet will be obfuscated. The packet will get back to its original form (i.e., de-obfuscation) only when the key is verified (when it reaches the correct router/destination). This way we can prevent the HTs in the router which causes misrouting and information leakage.

4. SYSTEM DESIGN

Network-on-Chip (NoC) consists of Network Interfaces (NIs), routers, links and several IP cores. We assume that NIs and links are not a good place

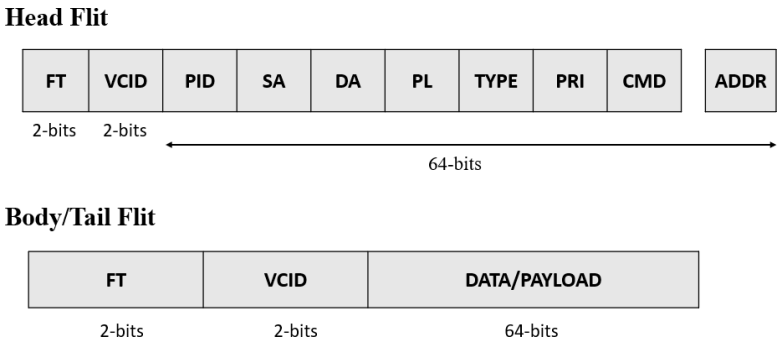

Figure 2. Packet structure

to insert Trojans. Therefore, we focus on the insertion of Trojans in routers alone as it can impact major number of packets passing through it. Trojans in router can be placed within buffer or routing unit or port selection unit. Directly targeting the corruption of data in router is very difficult, as the probability of getting caught during verification process is very high. In our proposed work, we design and insert the trojan model in the router and propose the key based authentication and obfuscation technique for the detection and mitigation of the trojan. Detailed explanation of packet structure, implementation of Trojan and proposed technique is explained in the further subsections.

We considered a standard 2D 4x4 mesh NoC design as shown in Figure 1 to discuss our proposed work. The model has 16 cores each of which are connect to individual routers through Network Interface (NI), The packets in NoC are routed to its destination by hopping through routers using XY routing.

4.1 Packet Structure

NoC employs communication between cores in the form of packets. A router is a basic building block that routes the packets passing through it to an appropriate destination. The packets here are further divided into smaller units called flits. The structure of flits are depicted in Figure 2. The Head flit contains all the indispensable information required for routing such as- Packet ID which is a unique packet identification number to identify packet in the network, Source address- where the packet is originated, Destination address- where the packet has to reach and Packet length- defines the number of flits in a packet. Since NoC carries all categories of packets exchanged between cores, the TYPE field specifies the packet type and priority is assigned to every packets which will be used in routing when there is a port conflict with other packets. The CMD(command) field is an extra bit

space where the additional information about the packet can be stored. The physical address to which the data has to be transmitted across the cores is stored in the ADDR field. So, each packets will be of 64-bit in length, the flit type and VCID fields are external to this 64-bit length of flits and common for all flit types. The body flit and tail flit will contain the flit type and VCID field along with the data which has to be transmitted to destination. The request packets will contain only head flit whereas, the response packets will contain one head flit followed by body flits and terminated by one tail flit.

4.2 Hardware Trojan Model

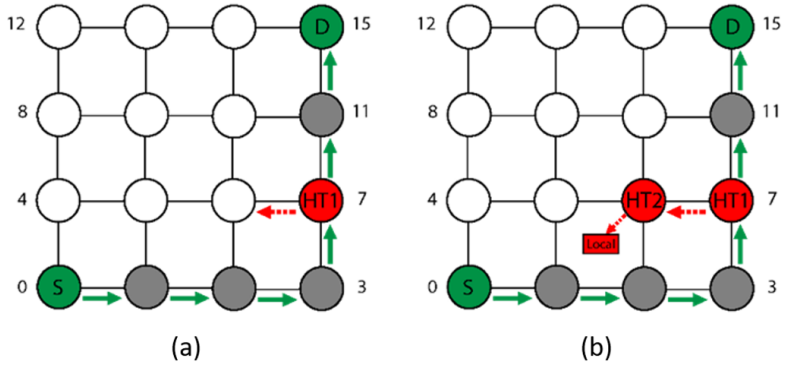

Figure 3. (a) 4 x 4 mesh NoC with HT at router 7, (b) 4 x 4 mesh NoC with HT at router 6 & 7.

Hardware Trojans are the malevolent alteration to the envisioned functionality of any hardware circuit. These unwanted and ambiguous alterations can have catastrophic effects on the underlying circuit. The key characteristics of any Trojans are- malicious intention, evasion of detection, and rarity of activation. This overall aims to bring down the integrity and trust of the circuit.

In our work, we consider Trojan that is inactive and only gets triggered when the packets pass though it meeting a specific condition. The trojan here targets the source and destination address field of the specific packets. Its main purpose is to misroute and leak information or cause Denial-of-Service (DoS). When a packet comes out of the routing unit, the packet has to go through virtual channel allocation and switch/port allocation. During this port allocation if the trojan gets triggered, the packet's port direction is changed from intended direction to the direction which trojan benefits to tamper data or degrade the performance of system.

The trojan is secretly introduced into routers which benefits the intention of the attacker. Initially, the Trojan model is in inactive state, i.e, the necessary triggering condition is not yet met. Once the trojan is activated, it affects conventional routing of packet and pushes the packet into different port/direction, thereby misrouting the packets and hindering it from reaching the desired node. A malicious node might misroute to some random

direction to cause delay in packet transmission over the network or route the packet to its core leading to information leakage. This causes performance degradation of the system. We explain the behaviour of the threat model case wise as depicted in Figure 3. Consider that router 6 & 7 are affected by trojan. Whenever a packet reaching a specific destination (say 15) or a packet originating from a specific source (say 0) passes through the HT affected router, trojan is triggered and is active.

Case 1: Misrouting of the packets

When the packet bearing source address 0 and destination address 15 reaches router 7 through conventional routing technique, the trojan in router 7 gets triggered. According to the conventional routing protocol the intended next router is 11 (green path), but the HT affected router misroutes the packet to router 6 which is not desired as show in Figure 3(a). This makes the packet to stay longer in the network increasing the average packet latency.

Case 2: Information Leakage or packet drop

The packets which are misrouted from router 7 to router 6 due to the presence of HT as specified in Case 1, may or may not be redirected to the destination. Instead, router 6 can act as a malicious receiver and forwards the packets to its local node by local switch or port selection resulting in information leakage as show in Figure 3(b). Other way of delivering the attack is to drop packet which has met the activation condition. The overall outcome of this malicious activity is to leak information or drop the packet causing network to re-transmit packet. This will reduce network capacity and increases the average packet latency as the number of packets in network increase.

As the trojan should not be easily detected and not affect all or most of the packets passing through it, the considered trojan will only affect 10-15% of the packets passing through the HT affected router. Overall it only affects 0.5-1% of the total packets generated in the network. The number of packets generated and affected in a synthetic traffic patterns are tabulated in Table 1.

Table 1. HT Effect on Packets passing through it

Traffic	No of Pkts	Through HT router	Total Affected
Uniform Random	82622	26876	4265
Shuffle	82487	24742	4431
Bit Complement	84260	42698	7273
Bit Reverse	82482	25768	5447
Transpose	82482	23428	3221

4.2 Detection Model using key authentication

To detect the packets which are affected by trojans, we make use of key based authentication technique. This is a two way process i.e., first, the key is generated and stored, then the stored key is used for verification. As the trojan considered here causes misrouting of packets, the key is used to check if the packet is reaching proper intended upstream routers. The key

authentication process is depicted in Figure 4, where DA is Destination router Address, NA is Next router Address and CA is Current router Address. The key is generated just after the routing unit using destination address and next router address obtained through the conventional routing technique. The key verification is done twice. Initially, before the packet entering routing unit the verification unit uses key value, current address and destination address to validate. Second verification is done before switch traversal using key, destination address and next router address. If the key verification unit's output is 0 then, the packet is in correct router and is not misrouted. If the verification output is 1 then, it is evident that the packet is affected by HT as it is misrouted and has reached the wrong router other than the expected. This indicates to counter the trojan effect which is done in mitigation section.

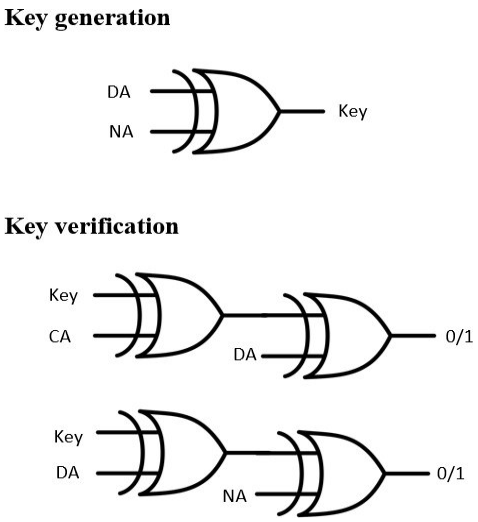

Figure 4. Key generation and verification

The key value is stored in head flit as an extra field from the CMD field (refer Figure 2). Size of the key will be equal to the size of router ID/address. The key field is dynamically updated every time the packet enters new router after routing unit. This key authentication not only detects the packet reaching wrong router but also detects if the trojan changes destination address in head flit after key generation.

4.3 Mitigation model using bit-Obfuscation

In NoC, even if a packet is misrouted and reaches a router where it should not have reached, the employed routing algorithm will never be able to detect it and will continue routing the packet according to protocol. Our proposed HT threat model exploits this feature of the routing algorithm enabling misrouting, information leakage and dropping packets to cause packet re-transmission. To identify HT infected packets, we propose key based authentication. In order to avoid the trojan triggering condition and to

mitigate the HT effect on packets we propose Bit-Obfuscation technique, which is employed in every router. Obfuscating the packets will shuffle or complement critical bit fields which will obscure the packet fields making them less sensitive to the trojan and pass through the HT affected router without triggering. Key based authentication is an additional feature of protection that makes use of a unique key to detect and mitigate trojans. The overall process of trojan attack, proposed detection and mitigation technique is depicted in Figure 5. When the packet enters the router, first the key is authenticated before the routing unit. If the authentication is successful, the obfuscated fields in head flit is de-obfuscated and proper routing is carried out. If the authentication fails, the critical fields in the head flit is not de-obfuscated and there by preventing the packet from trojan. Therefore, to carryout any meaningful attack, the trojan has to target a particular field in a particular way. Here, the Trojan will not be able to access the required fields but it will be unaware of the obfuscation and key authentication technique, thus failing to trigger and cause misrouting, leak information & drop packets to cause re-transmission.

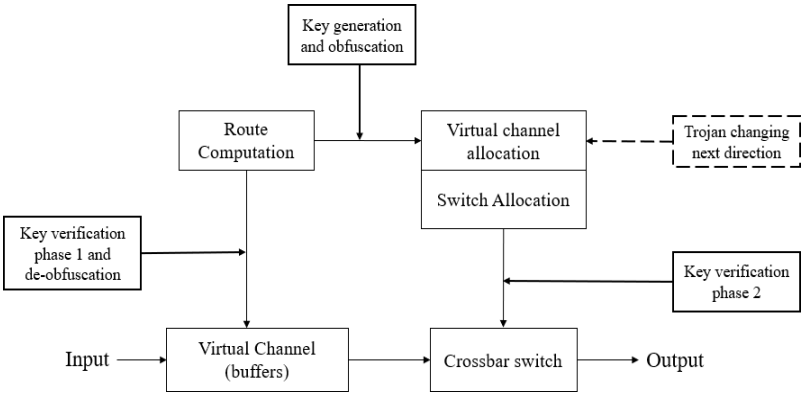

Figure 5. Working of Obfuscation and Logic key encryption technique

Let us consider Figure 3, here when the packets enter router 7 from router 3, key authentication at router 7 will be successful as it is the correct router for packet as per conventional routing protocol (green path). Next, the packet is de-obfuscated and passed through the routing unit. In route computation, the key is generated for the next router i.e., router 11 and the packet is obfuscated (refer Figure 5). Since the trojan in router 7 meets the triggering conditions, it gets activated and misroutes packet to router 6. When the packet reaches router 6, key authentication will be unsuccessful and the obfuscated packets are passed to next blocks without de-obfuscating. Since the packet fields are obscure, the trojan in router 6 does not meet the specific triggering condition to get activated. Therefore, the trojan effect in router 6 i.e., information leakage or dropping is foiled. To avoid misrouting beforehand in router 7 with some extra overhead, we also verify the key with the next destination address before processing the packet to crossbar switch or port traversal. If the key authentication is successful, packet is sent directly to the next router. If the key doesn't match, rerouting computation is

performed to route the packets to correct router. Therefore, the trojan effect in router 7 i.e. misrouting is foiled. Because the packet fields are obfuscated and 2-step key verification is performed at the router's itself, the next router direction is not modified. Hence, misrouting, information leakage & packet dropping trojans are detected and mitigated from the proposed techniques efficiently.

5. EXPERIMENT AND ANALYSIS

In this section, we provide the results of the implemented HT model and discuss the attack in terms of throughput, latency, area and power. We evaluate our proposed detection and mitigation technique to assess the effectiveness of our proposed technique.

The Gem5 simulator is used to evaluate the performance of 4 x 4 Mesh NoC based SoC with 16 cores. The interconnect was built on top of "Garnet2.0" model which is coherent with Gem5 [24]. Each router has 5 connections, 4 connections to its neighbours (Top, Right, Bottom, Left) and a done connection to the local core. The boundary routers will have only 2 or 3 connections to their neighbour along with local core connection. Every router can be identified by its coordinates (x,y). All the packets are traversed through this connection between routers and cores. The conventional routing algorithm employed here is XY routing. XY is a minimal deterministic routing which will first route the packet in 'X' direction (Horizontal) and then when the packet reaches the column same as destination packet will be routed in 'Y' direction (vertical).

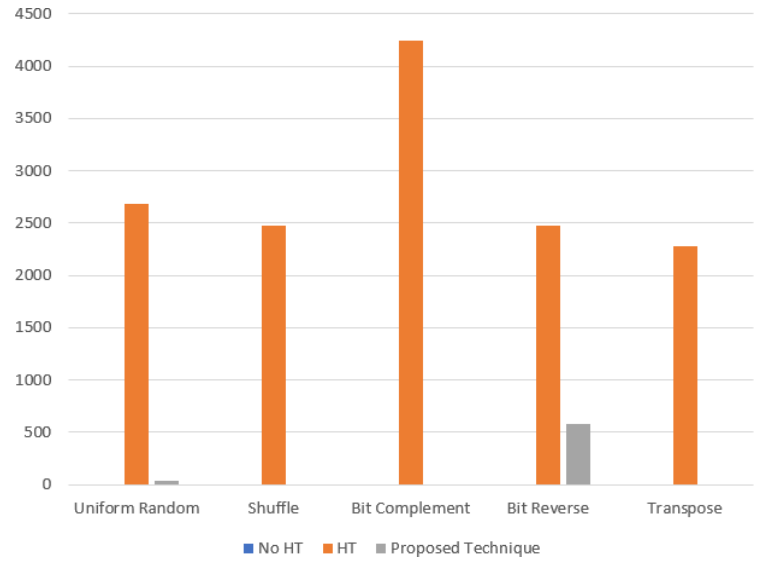

Figure 6. Number of packets re-transmitted for different workloads

The system was tested using synthetic traffic patterns (uniform random, shuffle, bit complement, bit reversal and transpose) and 8 combinations of benchmark (lbm, cactusADM, leslie3d, soplex, omnetpp, hmmer, sphinx, gromacs) into 5 different workloads as tabulated in Table: 2.

The benchmark combinations for workloads where chosen such that it has different Misses Per Kilo Instructions(MPKI) characteristics from High to Low. We considered different Flit Injection Rates (FIR) ranging from 0.01 to 0.15 in steps of 0.5 flits per cycle per node. We have simulated both synthetic traffic and benchmark workloads for 1 million cycles considering the first 10k cycle period as warm-up time. The average packet latency, number of re-transmissions and throughput is explained in further sub sections.

5.1 Packet re-transmission

Whenever the packet was dropped by trojan, network after some threshold time re-transmits the packet to destination again. This increases number of packets in the network which in turn affects traffic and packet latency. From Figure 6, it can be noticed that, in the presence of HT, around 2000 to 4000 packets were re-transmitted during the simulation period and when our proposed technique was employed re-transmission of packets were reduced to around 50 to 500 packets. For the benchmark workloads, the HT was re-transmitting around 1500 to 4000 packets and re-transmission was brought down to around 20 to 100 packets using our proposed detection and mitigation technique.

Table 2. Workload benchmarks

Workload	SPEC CPU 2006 Benchmarks
WL1	lbm (4) cactusADM (4) leslie3d (4) soplex (4)
WL2	omnetpp (4) hmmer (4) sphinx (4) gromacs (4)
WL3	lbm (2) cactusADM (2) leslie3d (2) soplex (2) omnetpp (2) hmmer (2) sphinx (2) gromacs (2)
WL4	lbm (3) cactusADM (3) leslie3d (3) soplex (3) omnetpp (1) hmmer (1) sphinx (1) gromacs (1)
WL5	lbm (1) cactusADM (1) leslie3d (1) soplex (1) omnetpp (3) hmmer (3) sphinx (3) gromacs (3)

5.2 Average packet latency

The average packet latency indicates how much time delay a packet is taking to reach the destination from its origin on an average. Figure 7 depicts the graph of average packet latency v/s different injection ratios for uniform random synthetic traffic pattern. As we can see from graph, the HT affect has increased the average packet latency by 30% on an average. By using our proposed detection and mitigation technique the average packet latency will only have 2-3% greater than the actual latency. The average packet latency for different synthetic traffic pattern is depicted in Figure 8. On an average the packet latency is increased to 35% and with our proposed technique it is brought downto 4-5%. The same process was carried out for benchmark workloads whose analysis is shown in Figure 9. With varied MPKI value workload, the HT affect was consistently maintained and had increased the

latency by 31% compared to baseline architecture. Our proposed technique efficiently prevented the HT effect and with packet latency of around 1.5%.

5.2 Throughput

Throughput gives the network's efficiency which is nothing but the rate of information delivered through the network. Figure 10 shows throughput analysis of the system. Presence of trojan reduced the throughput of system to 96-97%, because the proposed trojan was condition specific assailant and hard to detect. If the throughput is reduced by more than 10% the trojan will be evident and easily detectable. This reduction in throughput was handled efficiently with our proposed detection and mitigation technique with a throughput of around 99-99.5%.

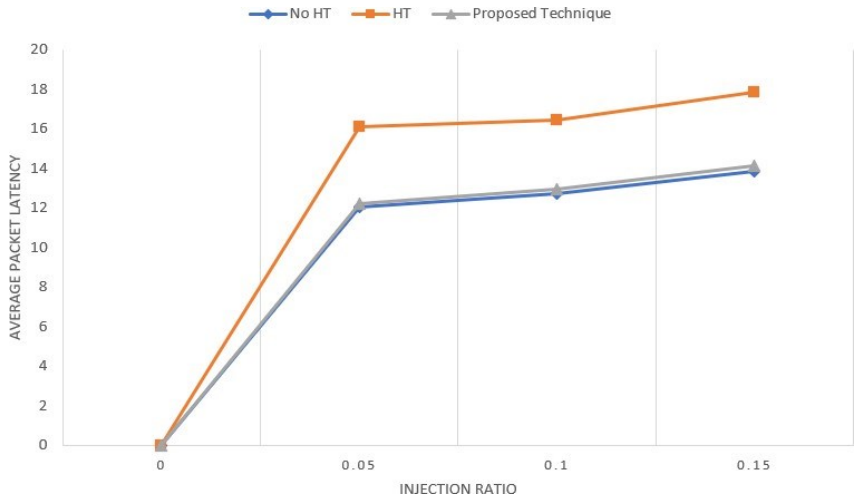

Figure 7. Average Packet Latency for different injection ratios for uniform random traffic

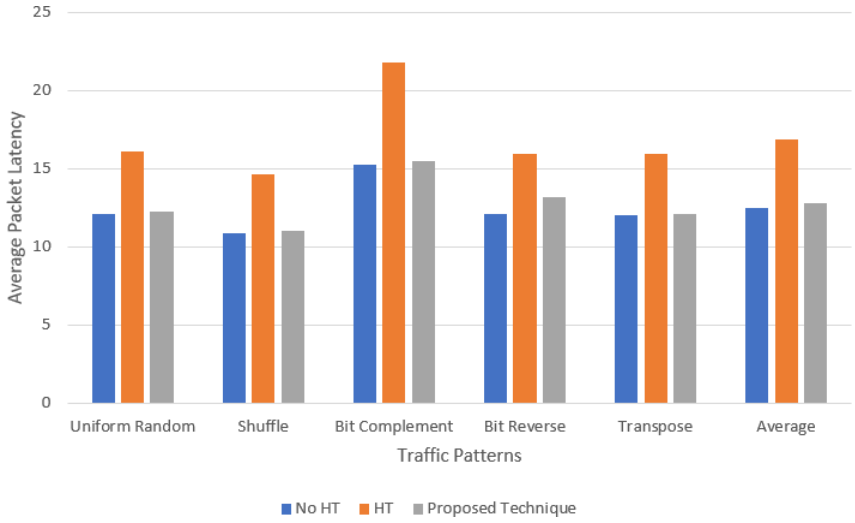

Figure 8. Average Packet Latency for different synthetic traffic patterns

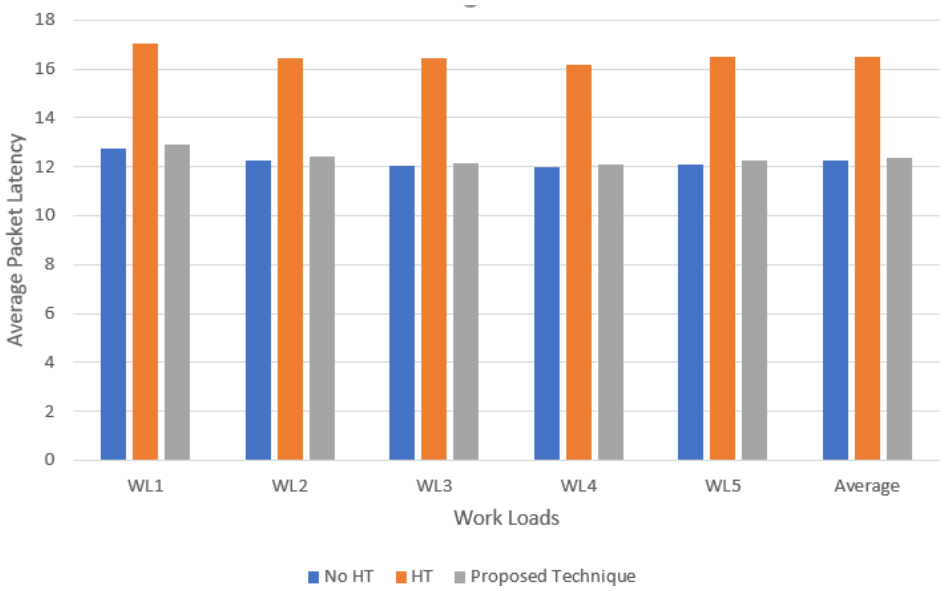

Figure 9. Average Packet Latency for different workloads

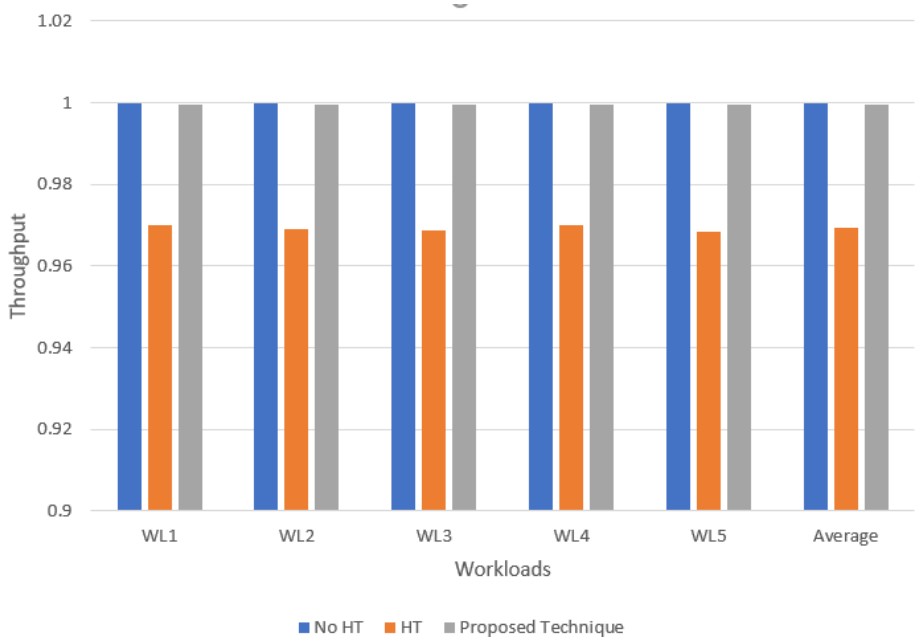

Figure 10. Throughput for different workloads

5.3 Analysis of circuit metrics

Proposed technique was implemented in Verilog HDL and synthesized using Cadence genus tool with 90nm technology. The proposed method achieves a higher HT detection rate and reduces the probability of HT attack successfully by obfuscation and key based authentication techniques. As shown in Table 3, the HT has an extra area overhead of 0.08% with power consumption of 0.06% which is very negligible and can easily bypass without

getting caught. Our proposed detection and mitigation technique has a 2% area and 8.6% power overhead, which is almost negligible in comparison to the baseline router. The code for baseline architecture was obtained using ProNoC [25]-[28].

Table 3. Area and Power overhead

Design for comparison	Area (nm)	Power(nW)
Baseline	60669.319 (100%)	1662359.869 (100%)
Hardware Trojan	60720.789 (100.08%)	1663498.506 (100.06%)
Proposed Technique	61818.293 (102%)	1806789.372 (108.6%)

When compared to the existing work related to the HT we considered, our method performs much better than the secure trojan aware routing [12]. The existing work has an increase in average packet latency of 7% compared to baseline, whereas our proposed technique is only increased by 1.5%. Also, in case of throughput, area and power consumption our proposed technique outperforms the existing work.

6. CONCLUSION

In this paper, we model Hardware Trojans(HT) that performs misrouting, information leakage ad packet drop causing re-transmission. This HT affected 10-15% of packets passing through trojan infected router degrading the performance of overall system in terms of packet latency and throughput. The proposed obfuscation and key based authentication technique is designed for run-time HT detection and mitigation. It achieves a higher HT detection rate and effectively mitigates all the infected packets from the trojan attack. The HT detection and mitigation technique was evaluated on 4x4 NoC. The simulation results shows that, due to the presence of trojan there is reduction in the throughput to 96% with an average packet latency increased by 35%. Proposed technique achieves the throughput of 99% almost equal to the baseline architecture with an average packet latency of only 1.5-2% increased in comparision to the baseline NoC. Proposed detection and mitigation technique adds 2% area overhead with 8.6% power consumption which is very negligible in comparison with the baseline NoC router.

REFERENCES

[1] Rajesh, J. S., Koushik Chakraborty and Sanghamitra Roy. **Hardware trojan attacks in soc and noc**. *In The Hardware Trojan War*, pp. 55-74. Springer, Cham, 2018.

[2] Mishra, Prabhat, and Subodha Charles, eds. **Network-on-Chip Security and Privacy**. *Springer Nature*, 2021.
[3] Hoefflinger, Bernd. **ITRS: The international technology roadmap for semiconductors**. *In Chips 2020*, pp. 161-174. Springer, Berlin, Heidelberg, 2011.
[4] Swarbrick, Ian, et al. **Network-on-chip programmable platform in VersalTM ACAP architecture.** *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. 2019.
[5] Charles, Subodha, and Prabhat Mishra. **A survey of network-on-chip security attacks and countermeasures.** *ACM Computing Surveys (CSUR) 54*, no. 5 (2021): 1-36.
[6] Jindal, Neetu, et al. **Enhancing network-on-chip performance by reusing trace buffers.** *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 39.4* (2019): 922-935.
[7] Diallo, Papa Issa, Seyed-Hosein Attarzadeh-Niaki, Francesco Robino, Ingo Sander, Joel Champeau, and Johnny Oberg. **A formal, model-driven design flow for system simulation and multi-core implementation.** *In 10th IEEE International Symposium on Industrial Embedded Systems (SIES)*, pp. 1-10. IEEE, 2015.
[8] Abramovici, Miron and Paul Bradley. **Integrated circuit security: new threats and solutions.** *In Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*, pp. 1-3. 2009.
[9] Bhunia, Swarup, Michael S. Hsiao, Mainak Banga and Seetharam Narasimhan. **Hardware Trojan attacks: Threat analysis and countermeasures.** *Proceedings of the IEEE 102*, no. 8 (2014): 1229-1247.
[10] Liakos, Konstantinos G., et al. **Conventional and machine learning approaches as countermeasures against hardware trojan attacks.** *Microprocessors and Microsystems* 79, 2020.
[11] Frey, Jonathan and Qiaoyan Yu. **A hardened network-on-chip design using runtime hardware Trojan mitigation methods.** *Integration* 56: 15-31, 2017.
[12] Manju, R., Abhijit Das, John Jose, and Prabhat Mishra. **SECTAR: Secure NoC using Trojan Aware Routing.** *In 2020 14th IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pp. 1-8. IEEE, 2020.
[13] Hussain, Mubashir. **Runtime Detection of Hardware Trojan in Untrusted Network-on-Chip.** *Diss.* UNSW Sydney, 2018.
[14] Harttung, Julian, et al. **Lightweight authenticated encryption for network-on-chip communications.** *Proceedings of the 2019 on Great Lakes Symposium on VLSI*. 2019.
[15] JYV, Manoj Kumar, Ayas Kanta Swain, Sudeendra Kumar, Sauvagya Ranjan Sahoo and Kamalakanta Mahapatra. **Run time mitigation of performance degradation hardware trojan attacks in network on**

chip. *In 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 738-743. IEEE, 2018.

[16] Boraten, Travis and Avinash Karanth Kodi. **Mitigation of denial of service attack with hardware trojans in noc architectures.** *In 2016 IEEE international parallel and distributed processing symposium (IPDPS)*, pp. 1091-1100. IEEE, 2016.

[17] Raparti, Venkata Yaswanth, and Sudeep Pasricha. **Lightweight mitigation of hardware Trojan attacks in NoC-based manycore computing.** *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2019.

[18] Frey, Jonathan and Qiaoyan Yu. **Exploiting state obfuscation to detect hardware trojans in NoC network interfaces.** *In 2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1-4. IEEE, 2015.

[19] Shalaby, Ahmed, et al. **Sentry-NoC: a statically-scheduled NoC for secure SoCs.** *2021 15th IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*. IEEE, 2021.

[20] Daoud, Luka, and Nader Rafla. **Routing aware and runtime detection for infected network-on-chip routers.** *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2018.

[21] Kumar, JYV Manoj, Ayas Kanta Swain, and Kamalakanta Mahapatra. **Fortified-NoC: A Robust Approach for Trojan-Resilient Network-on-Chips to Fortify Multicore-Based Consumer Electronics.** *IEEE Transactions on Consumer Electronics* 68.1 (2021): 57-68.

[22] Bahrebar, Poona, and Dirk Stroobandt. **Abacus turn model-based routing for NoC interconnects with switch or link failures.** *Microprocessors and Microsystems* 59 (2018): 69-91.

[23] Azar, Kimia Zamiri, et al. **{COMA}: Communication and Obfuscation Management Architecture.** *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)*. 2019.

[24] Power, J. Hestness, M. S. Orr, M. D. Hill and D. A. Wood, **gem5-gpu: A Heterogeneous CPU-GPU Simulator** *In IEEE Computer Architecture Letters*, vol. 14, no. 1, pp. 34-36, 1 Jan.-June 2015.

[25] Alireza Monemi, Jia Wei Tang, Maurizio Palesi, and Muhammad N Marsono. **ProNoC: A low latency network-on-chip based many-core system-on-chip prototyping platform**. *Microprocessors and Microsystems*, 54:60–74, 2017.

[26] Alireza Monemi, Chia Yee Ooi, Muhammad Nadzir Marsono, and Maurizio Palesi. **Improved flow control for minimal fully adaptive routing in 2D mesh NoC**. *In Proceedings of the 9th International Workshop on Network on Chip Architectures*, NoCArc'16, pages 9–14. ACM, 2016.

[27] Alireza Monemi, Chia Yee Ooi, and Muhammad Nadzir Marsono. **Low latency networkon-chip router microarchitecture using request**

masking technique. *International Journal of Reconfigurable Computing*, 2015:2, 2015

[28] Alireza Monemi, Chia Yee Ooi, Maurizio Palesi, and Muhammad Nadzir Marsono. **Low latency network-on-chip router using static straight allocator**. *In Proceedings of 3rd International Conference on Information Technology, Computer and Electrical Engineering*, ICITACEE'16. IEEE, 2016.