Fortified-NoC: A Robust Approach for Trojan-Resilient Network-on-Chips to Fortify Multicore based Consumer Electronics

Manoj Kumar JYV, Ayas Kanta Swain, Member, IEEE, KamalaKanta Mahapatra, Senior Member, IEEE, and Saraju P. Mohanty, Senior Member, IEEE.

***Abstract*—Consumer electronics hardware is designed and manufactured following a global supply chain which opens doors to their security challenges. Even with the advanced methods of formal verification and coverage analysis, there is still a chance of hiding malicious hardware/software which can degrade performance, leak data, or even stop functionalities. In this work, we present security solution of Network on Chip (NoC) based Multiprocessor System-on-Chip (MPSoC) consumer electronics systems such as set-top boxes and autonomous vehicles. Most of the existing methods targeted for such systems focus on protection in Network Interfaces (NI) and other software solutions rather than routers against Hardware Trojans (HT) which can be embedded in NoC by a rogue designer. In this work, we propose a 3-tier methodology leading to "Fortified-NoC" to secure the data and resources against different kinds of threats. A Trojan cognizant routing algorithm (TCRA) is proposed which limits the HTs to a particular router that contains them. Data shuffling with Trojan detectability is also used to mislead and identify the HTs. We validated the proposed approach using various experiments. Our proposed method is capable of mitigating the Trojan attacks such as data leakage, performance degradation, denial of service and live locking of data packets at the cost of a little latency and some extra hardware. It is able to recover more than 80% of lost packets, improve the throughput by 1.3× against performance degrading Trojan attacks.**

***Index terms*—** Consumer Electronics, Electronic Systems, Hardware Trojan, Hardware Security, Multiprocessor System on Chip, Network-on-chip, Live Lock, Denial of service

I. Introduction

All consumer electronics systems such as smart vehicles, home automation system, smart TV, and set-top-box use Intellectual Property (IP) cores to meet the cost and time-to-market demand [1]–[3]. The communication of IP blocks using bus architecture limits the scalability of the system and puts restrictions on meeting the targets of the consumer market [4]. Multiprocessor System-on-a-Chip (MPSoC) is key workhorse of sophisticated generic platform for the embedded real-time systems in different domains, such as industrial control, and consumer-electronics applications (see Fig. 1). The major challenges that are addressed by MPSoCs are cognitive complexity, robustness and energy efficiency. MPSoC with time-triggered Network-on-Chip (NoC) are deployed to bring the efficiency to the brown goods including television sets, audio equipment, and similar household appliances [5]. A HDTV decoder SoC multi-processor platform with system general-purpose processor, audio processor, and video processor is a best example that illustrates how MPSoCs are becoming trivial and universal in Consumer electronic appliances [6], [7]. Ride safety system-on-chips (SoCs) are being developed to addresses the complexity of autonomous driving [8].

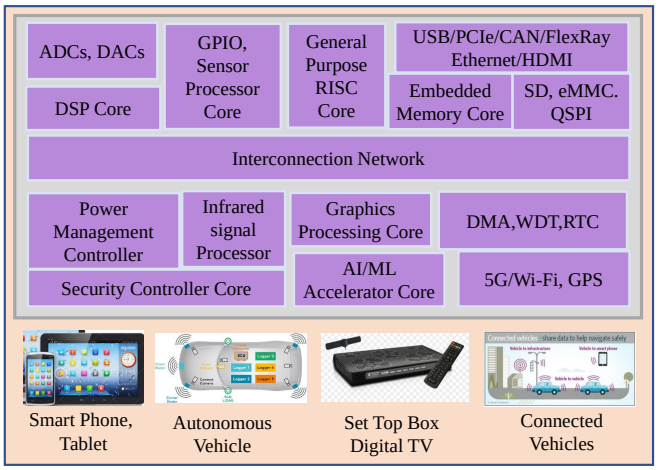


Fig. 1. Multiprocessor System-on-a-Chip (MPSoC) for CE devices.

Latest advancements in technology especially in the Internet of Things and Artificial Intelligence come at the cost of huge hardware. This is unavoidable in order to achieve high speed data processing and parallel computing. So, this accumulated requirement of hardware makes it very difficult to interconnect all the modules on a single chip. There is also a limitation on the number of interconnect layers in a chip [9]. A feasible solution is to use NoC to connect the different modules, processors, and memories [10] resulting in MPSoC design. NoC is inevitable for the future IC's with the pace of today's scaling and future requirement of heavy-duty digital logic driven computing systems. Routing data packets rather than routing of interconnects is a good choice [11].

The Security-by-Design (SbD) paradigm has been advocated for consumer electronics to ensure that their security requirements are considered right from the beginning of the

Manoj Kumar JYV is with the Dept., of Electronics and Communication Engineering, NIT Rourkela, India, E-mail: manojkumarjyv@gmail.com.

Ayas Kanta Swain is with the Dept., of Electronics and Communication Engineering, NIT Rourkela, India, E-mail: swaina@nitrkl.ac.in.

KamalaKanta Mahapatra is with the Dept., of Electronics and Communication Engineering, NIT Rourkela, India, E-mail: kkm@nitrkl.ac.in.

Saraju. P. Mohanty is with the Dept. of Computer Science and Engineering, University of North Texas, E-mail: saraju.mohanty@unt.edu.

design phase so that retrofitting at the later stage is not needed [12]. It was reported that the "back door Trojan purposely implanted within a commercially of the self (COTS) microprocessor can have serious consequence [13]. For example, the Trojan had temporarily blocked the radar by sending a preprogrammed code to those chips and thus disrupted the chips' function [13]. It has been reported that the Quantum program of US National Security Agency (NSA) directly implants HT circuitry into USB communication protocol or USB port [14].

This article follows that vision to develop security-integrated NoC (called "Fortified-NoC") based MPSoC to ensure that CE systems are fully secured right from their design phase. A Smart TV is a such CE device that comprises of either a television set with integrated internet capabilities or a set-top box for television that offers advanced computing ability and connectivity than a basic TV set. There are evidences that a Smart TV is vulnerable to attacks [15]. Some serious security bugs have been discovered, and some successful attempts to run malicious code to get unauthorized access were documented. There is evidence that it is possible to gain root access to the device, install malicious software, access and modify configuration information through remote control, and modify files on TV and attached USB drives, access camera and microphone [15]. There have also been concerns that hackers may be able to remotely turn on the microphone or webcam on a smart TV, being able to eavesdrop on private conversations. A future-generation smart TV architecture consists of multicore RISC processors, graphic processor, memory system, TV modulator, peripherals, Input/Outputs. The new IP offerings include image processing, overlay and picture-in-picture (PIP) functions. To accelerate the productivity, all these IP s can be acquired from 3rd party and can be integrated in-house. The Smart TVs with beyond 4K resolutions, requires greater bandwidth efficiency because it has to process more information per pixel. To support a scalable system and to enhance the bandwidth requirement, NoC can be used as the interconnection medium between the IP cores. Hence adversaries can introduce the malicious hardware into the 3rd party IPs and interconnections that may create security harm in Smart TV [16]. Hence a research on robust approach for Trojan-Resilient Network-on-Chips to fortify Consumer Electronics devices, such as Smart TV is highly essential.

In global design and manufacturing of CE systems, one design house acquires IP cores from other vendors to overcome the designing cost and technological difficulties in the ever decreasing time-to-market of consumer electronics [17], [18]. NoCs are also being made as IPs. With more use of 3rd party IP cores, comes more security threats [18]. Introduction of Hardware Trojans into different parts of a chip including NoC by a rogue engineer for 3rd party's interest has become a covert practice [19]–[21]. These malicious designs are capable of accessing and leaking data and deteriorating the network performance. So NoCs must be protected at any cost since all the data transfer between different modules will flow through NoC and an HT inserted in it can affect the functionality of any module by illegally accessing the packets addressed to them and snooping the data content.

The rest of the paper is organized as follows: Section II discuss related works on this area of research. The contributions of the current paper is presented in Section III. Hardware Trojan insertion and attack scenario presented in Sections IV. Proposed method to fortify NoC is listed in Section V. The experimental setup is described in Section VI. Mitigated results are discussed in section VII. Finally, the conclusions and future directions of this research is stated in Section VIII.

II. Related Prior Works

CE products have been heavily using digital computing paradigm and have largely amalgamated with the industries which is increasingly referred to as the consumerization of information technology. Research on the Security of CE systems and products is essential and hence very active. Physical Unclonable Function (PUF) based hardware-assisted security for smart healthcare has been designed [22]. RFID tag has been design for smart healthcare and smart home applications [23]. Privacy methods for smartwatch which is used for smart healthcare is presented in [24]. It has been demonstrated that security can be compromised when data errors can occur during message transfer in mobile devices [25]. For example, a parity error in the transferred byte instigates the receiver to transmit a parity error signal and an I/O transmission protocol is designed for processing parity errors with software. Secure firmware upgrade method has been presented for embedded systems which can be part of CE devices such as smart home, smartphones, smartcards and smart TV [26]. Several solutions have been presented in the literature for the security of smart cars [27]. Security methods for smart personal assistant system has been introduced [28].

Research is also in full swing for the security of the various components of the CE devices. PUF based security of NoCs applied to consumer electronics has been proposed [21]. An AES based security framework to counter non-secure IP core's unauthorized access has been proposed [29]. Fort NoCs with a 3-layer protection mechanism is proposed against hardware Trojans embedded in compromised IP cores [19]. Monitoring based countermeasures against unauthorised access to routing tables and an attack of misrouting are proposed in [30]. A PUF based run time mitigation method is proposed in [31]. P-Sec, a packet validation technique is addressed to suppress the attacks within a compromised NoC [32]. They also contributed significant mitigation model over a target activated sequential payload (TSAP) which embeds false data into flit by inspecting them [33]. A secured router architecture (SeRA) is proposed to limit the illegal packet request attack (IPRA) which are triggered when the core is in idle state [20]. A state obfuscation method is proposed to Detect Hardware Trojans present in NoC Network Interfaces (NI), but the method is sensitive to Trojans present in router [34]. So, most of the existing mechanisms deal with data leakage, and Hardware Trojans present in IP cores and NIs and some methods don't detect Trojans rather they directly mitigate effects of HTs.

The "Fortified-NoC" work of the current paper advances the state-of-art to integrate security features right at the early design phase of components which make consumer electronics system.

III. Contributions of the Current Paper

A. Research Questions Addressed in The Current Paper

The research questions addressed in the papers are:

- What are the various Trojans that can have adverse effect on the performance of NoC?
- How the proposed method with data encryption and obfuscation approach can help in mitigating the Trojan effects in NoC?
- How a cognizant routing can further reduce the blocking of resources by Live Lock Trojan?

B. Challenges in Solving The Problem

The challenges involves in solving the problem are:

- Standard available encryption algorithms like AES [35] demand huge amount of hardware and each Hardware added to a soft or firm NoC IPs create challenges during the PNR stage.
- Any combinatorial logic added in the router can have an impact on the critical timing path that restricts the maximum usable packet transfer rate [36].
- Developing a universal strategy to counter variety of Trojan attacks is always a challenge.

C. Novel Contributions of the Current Paper

The following are major contributions of the current paper:

- A novel multilayer low-overhead protection scheme that addresses the HTs inserted in router components as well as third party IP cores by a dishonest designer in the design team or by the outsourced integration firm.
- A novel error detection scheme that will detect the feasible single or multi bit changes up to very high extent in the data due to HT's dirty deeds.
- A novel fast TCRA is proposed in this work which comes into play upon detection of bit changes by the Trojans. The proposed algorithm is developed by the phenomena of isolation of links of the HT affected router. This clearly restricts the HT inserted in a particular router to access the pass-by packets and hence HT's potency is nullified completely outside that router.
- A novel cryptographic encoding scheme for restraining of payload data from attack logic blocks.
- A novel method for realization of data leakage and live locking of packets even with corner router.

IV. Hardware Trojan in Network-on-Chip Hardware - A Detailed Perspective

A. NoC Architecture - A Brief Overview

NoC consists of connected tiles, network interface (NI), router, and local processing core (See Fig. 2(a)). The topology decides the number of input/output links in a router. A popular topology for an on-chip design is 2D Mesh topology. The primary components of a router are input and output buffers, crossbar switch, and switch control mechanism. Switch control is mainly a route computation module and a control logic to control the routing, switching activities and allocation of virtual channels used in the router design.

The transfer of data among different nodes takes place in form of data packets of different sizes depending on the network parameters and protocols used. Each data packet is further divided into small quantities called flow control units (flits) (See Fig. 2(b)). A packet is comprised of many number of flits and the first flit is routed based on the routing algorithm implemented and the remaining flits follow the same path. The packet is comprised of three types of flits – Head flit, Body flit, and Tail flit. Apart from the data field, other fields are essential for proper and complete transfer of packets from source to destination. The special bit fields Tr, and NF are not members of a general flit structure. They are added as part of security development and their purpose is explained in Section V.

B. Trojan Inclusion is Possible in Any Stage of CE Hardware

HTs can be inserted at different levels of CE component hardware design like architecture level, logic level, and silicon stage. Each stage is having its own level of difficulty to insert HTs. HTs can be inserted at any place in a router or NI or within a link. In a router, most probable places of Trojan inclusion are in buffers as shown in Fig. 2(c), crossbar, and in control logic. Buffers are large in size to hold large number of bits. So, keeping the extra tenacious hardware blocks within buffer also helps the Trojan's presence neglected in side channel analysis.

C. HT Model and Attack Scenario

The primary goals of placing HTs in NoCs can be broadly classified into leaking data to local or other desired cores, performance limiting, and Denial of service etc. In general, HTs are designed in several components, each one is having its own purpose and these designs are not always active.

The HTs inserted in the design to carry out attacks. HTs are typically having two paths in it. One is normal path and the other one is Trojan included path as shown in Fig. 3. When Trojan is inactive, normal path is selected; otherwise second path is selected. The Trojan deals with the design, data, and control modifications to harm the CE systems. Upon arrival of a trigger condition, the Trojan detector detects the condition and activates the modification-hardware and initiates for payload delivery.

D. Analysis of Trojan Attacks

1) Performance Degrading Trojans: This kind of Trojans aim different essential bit fields like head bit, tail bit, address fields and packet length field to cause disorder and muddling in the network. This leads to deterioration of NoC's performance.

The following are some Trojans that causes major attacks. *Head Bit Trojan*: The Head Bit Trojan (HBT) attacks the head bit and changes it, thereby virtually prevent the route calculation module from accessing the essential bit fields to find out routing path. This leads to packet loss and resource wastage. It can be easily recovered by using a single bit ECC.

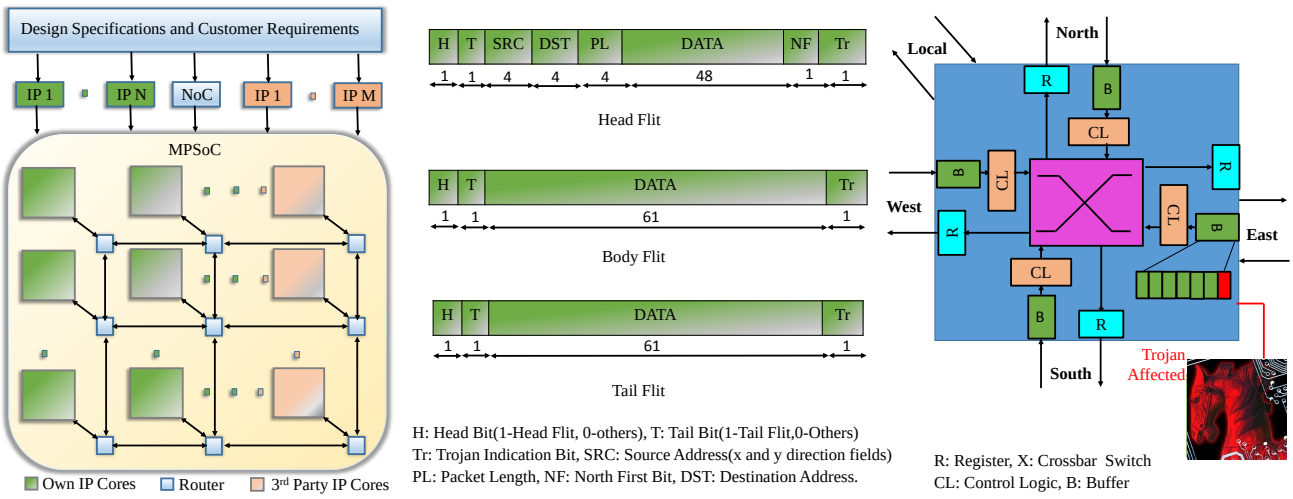


Fig. 2. Different Elements of NoC: (a) NoC Formation (b) Packet Format (c)Hardware Trojan Insertion Scenario.

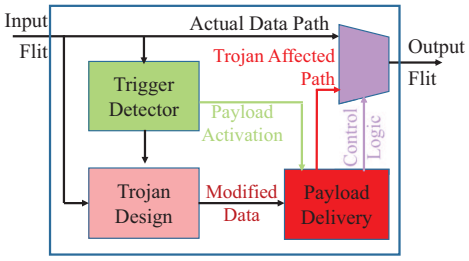


Fig. 3. A Hardware Trojan Model.

Destination Address Trojan: In Destination Address Trojan (DAT), the packets that flow through Trojan inserted router are modified in their destination addresses and scattered among other nodes rather than destined ones. If the trigger condition is set in such a way that packets of different nodes frequently get blocked and diverted, there will be a denial-of-service for different nodes and a request of retransmission is necessary to get the missing packets frequently. This also results in wastage of resources like links and buffers in other routers as well.

Packet Length Trojan: Packet Length Trojan (PLT) attacks the packet length field of head flit, making it more or less than the actual and thereby causes disruption in normal operation. If packet length is changed, the receiving node checks this number and verifies the received flits. If number of flits starting from head flit to tail flit is matching with the packet length field then that set of flits is treated as a packet. If the number in packet length field is more (say 7) and received flits are less (say 5), then the receiver will wait for a tail flit to occur at the number indicated by packet length field (7) but there will be no such flit coming and the receiver node either has to abandon the packet or wait for a next tail flit to make a packet. If PL field is changed to less and after receiving that many flits, receiving node packs up the flits into a packet and remaining flits are lost forever. This results in poor performance of NoC.

2) Data Leakage Trojan: In NoC, the delivery of the packets to right destination depends on destination address field and efficient routing. If the destination address field is compromised, one can misroute the packets to other destination such as local node or a specific other node leading to data leakage. The packets can be directly leaked or a copy of them can be generated and then leaked. Data Leakage Trojan (DLT) can be placed in a router connected to the untrusted 3rd party core and with the help of a predesigned malicious receiver in it, data can be snooped, processed, retransmitted to another module. These aim at leaking data either in the form of bit stream or as a side channel signatures such as electromagnetic radiation As an example, we placed a Trojan in router 2, 2 (x, y directions) which is designed to demonstrate the leakage of packets from node 0,2 to local core (See Fig. 4).

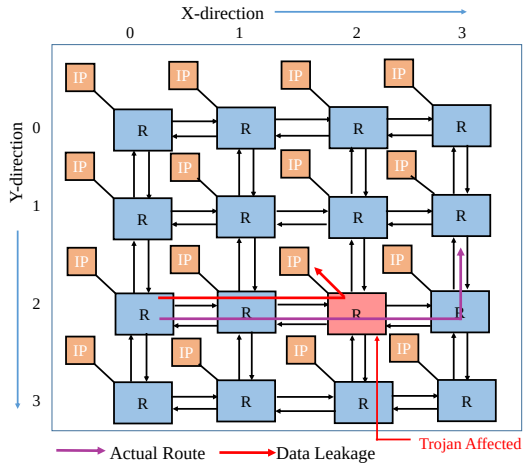


Fig. 4. Data leakage to local (desired) node scenario.

3) Live Lock Trojan: Live Lock Trojan (LLT) causes live locking of packets. A Trojan inserted in router 15 which attacks the route calculation module's output and also modifies tail bit of the packet is shown in Fig. 5. As head and body flits are not modified, the router has to wait for the tail flit to complete the transfer of a packet. In bit complement traffic, the node 12 sends packets to node 3, node 15 sends packets to

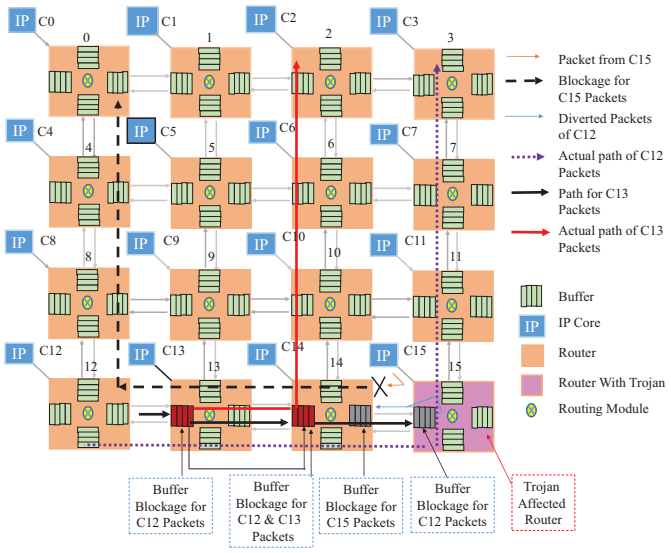


Fig. 5. Live lock of flits due to buffer blockage by Live Lock Trojan.

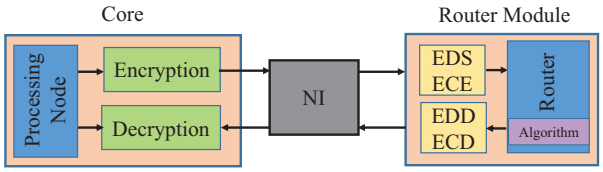
node 0, and node 13 sends packets to node 2. LLT is designed to be activated after a predefined number of occurrences of trigger condition. To do this, a counter can be designed which increments when the specified condition is met. When it gets activated, with further trigger conditions it diverts the packets coming from router 14 from going towards router 11 back to router 14 by changing the output direction from North to West as given in the figure. Flits are stored in East input buffer of router 14 and routed back to East i.e. router 15. As soon as the number of flits that are affected increases, there will be a clogging in buffers. The saturation of East input buffer blocks packets originally coming from core 15 which are destined to core 0 and at the same time encumbrance of West input buffer won't allow packets from core 13 once buffer-full of flits are stored in West input buffer of router 14 and in turn this leads to buffer blockage in router 13 as well. Finally, the packets from cores 12,13, and 15 are struck due to repeated flow of flits in a loop between routers 14 and 15 leading to live-locking of flits. The route can't be changed by the router for the remaining flits other than head flit and there is even no identification of end flit to treat a particular set of flits of a packet as tail bit is also manipulated. The proposed method identifies the modifications in tail bit of tail flit and TCRA comes into effect and packets are re-routed through router 10.

V. Our Proposed Methods to Fortify NoC

The proposed method can be broadly divided into 3 layers: (1) Data Encryption, (2) Obfuscation by bit shifting and recovery, and (3) Mitigation with TCRA with Trojan detection. The overall the method is depicted in the Fig. 6 Core-NI-Router diagram. Hardware corresponding to data encryption is setup in the processing node or IP core. The remaining shuffling and ECC along with TCRA are added to Router.

A. Cryptography based Security of Packets

Once the data packet leaves the source core, it travels through several hardware blocks i.e., routers to reach its



EDS: Error Detectable Shuffler, ECE: Error Correction Encoder
EDD: Error Detectable De-shuffler, ECD: Error Correction Decoder
NI: Network Interface,

Fig. 6. Illustration of the proposed method.

destination. Meanwhile, the HTs planted in respective sections can take down the packets and modify the data to perform a specific function or to command the other IP cores or even steel the data. Hence data encryption or cipher is a good option within own/trusted IP cores. The encryption involves a private key to encode and decode or a proper way/sequence of data modification process. With these details only, one can decipher the coded data when the amount of data is huge. The existing high standard algorithms demand a large amount of hardware and some initial processing delay. We propose to use P-box and S-box which are two of the main blocks in AES algorithm. The permutation box permutes the flit bits in a unique way in every secure core independently and the substitution box performs a predefined direct substitution of the data words. So, encryption here is carried out in two steps: (1) 64-bit data is permuted independently and uniquely in all trusted cores and (2) by dividing the 64 bits into 4 bit sets and concurrent substitution for each set is done individually as described in Fig. 7. With a given input say x, the S-box design generates output y, and the function gives transformation relation between I/O pair. As more than one S-boxes are used, we can have different functions in different modules. One should have knowledge about all the functions to decode the original data. The S-box is instantiated by avoiding fixed points and opposite points.

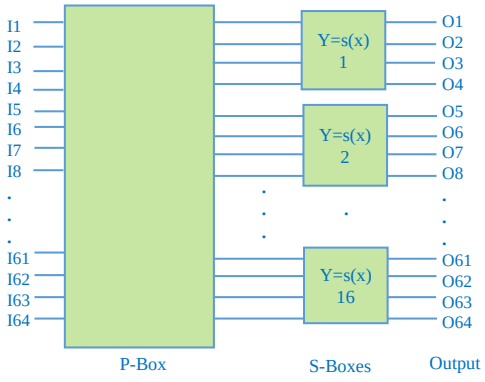


Fig. 7. P-box and S-box implementation.

B. Error Detectable Shuffling

Bit shuffling is a technique to make the flit content enigmatic and Trojans can't easily perform their assigned tasks. Bit shifting is deliberate misplacement of flit content by

shifting/shuffling different fields among others and themselves as well. So, whatever the standard description about flit field identity, is worthless when the flit is highly shuffled. In hardware terminology, shifting/shuffling is selection of different patterns of input data to output lines as shown in Fig. 8. We propose to shuffle the crucial bit fields like Head bit, Tail bit, Addresses, and Packet Length fields (e.g. a total of 14 bits in).

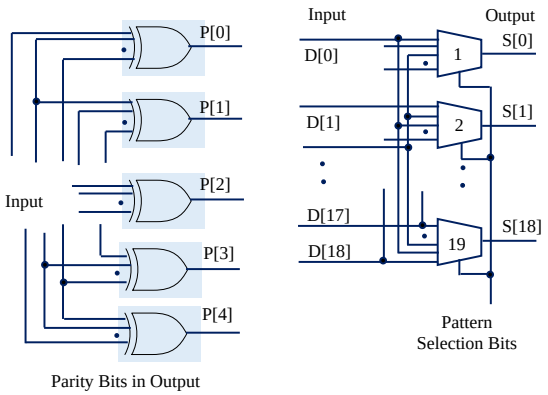


Fig. 8. Hamming code parity generator and Error Detectable Shuffler

We present an ECC based scheme to detect the single and multiple bit changes in the data to identify unwanted malicious modifications in the data to know the presence of Trojan design. Hamming code is a good choice for ECC because it not only detects multiple errors but also capable of correcting a single bit error. Before bit shuffling, we add hamming parity bits to the data content. This extra overhead is necessary to detect the Trojan and can be accommodated in the front part of payload bits. Fig. 2(b) shows a 48-bit payload (data) in the head flit and 61-bit payload in remaining flits. Hence cores must insert required number of dummy bits so that they can be replaced by these parity bits.

There is no efficient code that detects all multiple errors by using less overhead than in hamming code parity bit pack. Because some particular combination of changes to data gives no parity error and the changes go undetected. For example, change of bit positions 16,17,18,19 in a [32], [37] hamming code gives no error. We propose an Error Detectable Shuffling (EDS) which shuffles the bits in a specific way rather than randomly so that the hamming code will be able to detect changes in most number of bits.

The placement of the shuffling and hamming block is shown in router's architecture. The same operation but in reverse manner is performed at the ends of the router.

C. Trojan Cognizant Routing Algorithm (TCRA)

Fig. 9 shows the novel NoC architecture of router including additional blocks to be placed in router as proposed in this work. Tr and NF bits are used in the proposed routing algorithm. Tr bit stands for Trojan present and NF stands for North First. When ED detects an error in any flit, the Tr bit gets updated by TE block incorporated in Security Decoder. EC block takes care of single bit corrections which is a part of hamming decoder. The updated Tr bit will be verified by Tr checker in the arbiter. The arbiter also includes 4 single bit Trojan direction indication registers (DIR namely NEWS resisters) that are updated by Tr detection and will be used for identifying the Trojan inserted router.

The TCRA is developed in two phases. One is X-first when NF is 0 and Y-first otherwise. The algorithm focuses on moving towards the destination in one particular direction (based on the NF bit) until it comes in same row or column with the destination addressed node.

Algorithm 1 Proposed Trojan Cognizant Routing Algorithm (TCRA).

- **INPUTS:**
  - NF North first flag
  - N, W, E, S Direction registers
  - LX, LY coordinates of the local router
  - DX, DY coordinates of the destination router
- **OUTPUTS:** direction to route flits

If $(DX == LX)$
  If $(DY == LY)$ Return LOCAL;
  Else If $(DY < LY)$
    If $(N == 0)$ Return NORTH;
    Else If $(LY - DY == 1)$ Return NORTH;
    Else If $(LX == 0)$ Return EAST;NF=1;
    Else Return WEST; NF=1;
  Else If $(S == 0)$ Return SOUTH;
  Else If $(DY - LY == 1)$Return SOUTH;
  Else If $(LX == 0)$ Return EAST;NF=1;
  Else Return WEST; NF=1;
Else If $(DX > LX)$
  If $(E == 0)$ Return EAST;
  Else If $(DX - LX == 1)$ Return EAST;
  Else If $(DY == LY)$
    If $(LY == 0)$ Return SOUTH;
    Else Return NORTH;
  Else If $(DY > LY)$ Return SOUTH;
  Else Return NORTH;
Else If $(W == 0)$ Return WEST;
Else If $(LX - DX == 1)$ Return WEST;
Else If $(DY == LY)$
  If $(LY == 0)$ Return SOUTH;
  Else Return NORTH;
Else If $(DY > LY)$ Return SOUTH;
Else Return NORTH;

Then TCRA covers the other dimension path to finally reach the destination. If it finds the Trojan with the help of registers it avoids that router when it encounters it on its way. The algorithm is capable of back-forwarding of packets if necessary in order to avoid the Trojan-inserted router. This is very helpful when the packet reaches a corner with two sides blocked and one side Trojan affected router being there. Thus, the proposed algorithm blocks the access of packets that flow in the network by the Trojan.

Flit flow in router: Before flit leaves the core, it first undergoes encryption (and decryption while returning) and then leaves for router. As soon as the flit enters the router, it

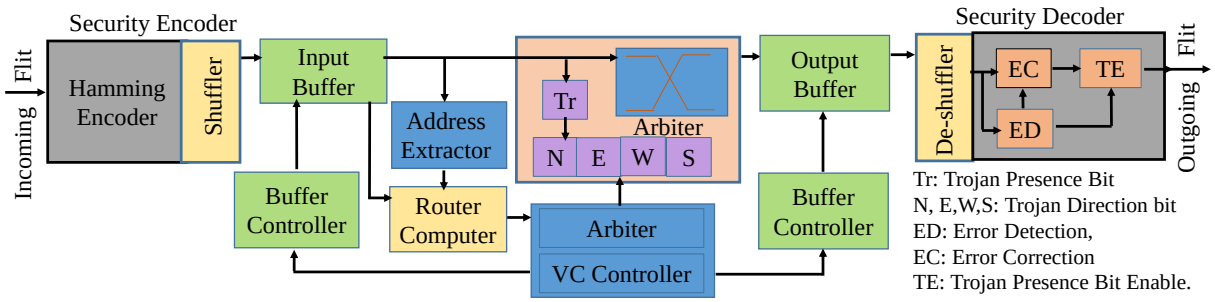


Fig. 9. Modified router architecture with security hardware blocks.

first passes through the hamming encoder, and parity bits for Trojan detection are added in the payload. It then undergoes shuffling in EDS before stored in input buffer. Depending on the arbitration, when it's time comes, the stored flit passes through the arbiter. Here if Tr bit doesn't indicate the Trojan presence, normal XY (or default) routing algorithm is used otherwise it uses the proposed algorithm for routing. Based on the Tr bit, one of the NEWS registers is updated. Then the route computer uses this information to avoid choosing the direction indicated by the NEWS register set if possible. This is how the Trojan-present router gets avoided by the router.

Depending upon the routing, the flit will be pushed into one of the five (4 directions and 1 local core) available output buffers. Before the flit leaves the arbiter, its Tr bit gets reset and Trojan detection from this bit is limited to present router only. After output buffer, it goes through Security Decoder. Here, ECC is applied, error detection and single bit error correction are performed by ED and EC blocks respectively. If an error is detected, Tr bit in that flit is set to 1 by TE block and flit leaves the present router for another selected router or local core. Here Tr bit is updated at the end of the router and it is utilized in next router to identify the router which is Trojan-inserted one.

VI. Experimental Verification of Fortified-NoC

A. NoC setup and Simulation Parameters

We used widely accepted NoCTweak network simulator [38] to verify the proposed work. It is a parameterised simulator based on SystemC and provide a wide variety of network parameters to estimate network performance. The simulation is carried out with 64-bit flit length, each packet is having 5 flits. Input FIFO is of 8-bit depth and wormhole pipelining is implemented by setting inter-arrival time distribution of packets to uniform mode. We performed simulation with 5 runs by varying the flit injection rate from 0.2 flits/packets/node in steps of 0.4. The simulation carried for 100K cycles and first 20% of it is exempted in calculations to allow the network to get into steady state traffic.

We setup a network with 16 nodes in a 4×4 mesh topology and we divided the network into a group of clans (See Fig. 10). We inserted Trojan in the 11th tile $(x = 2, y = 2)$ which carries attacks on critical bit fields and also performs data leakage to the local node. With these Trojans, we performed the evaluation of our method against performance degradation and data leakage Trojan. We also demonstrated live locking of packets with in a router by implementing Trojan in a corner router to even satisfy the worst-case scenario.

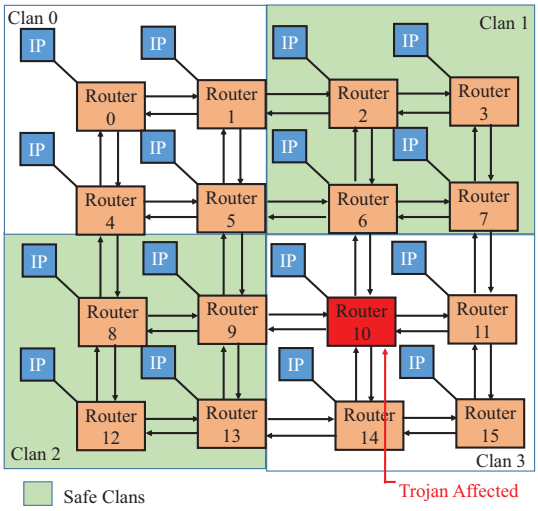


Fig. 10. NoC Setup with secure and non-secure clans.

B. Related Work considered for Comparison

It can be noted that in the existing works, the Trojan type, attack type, placement, trigger everything varies from one's work to other. Hence a fair direct comparison with other works is not possible. All have their proposed methodology to tackle a particular Trojan and performance results along with area overhead are given as a result validation. We consider the work in [31] as the baseline for comparison with the proposed method, which is compared other existing works [36], [39], [40] and the authors have shown that their work is better and efficient in mitigating the Trojan attacks in terms of total number of packets delivered, traffic distribution, effective average latency and link availability. Apart from performance degradation Trojan, mitigation of two new hardware Trojans such as data leakage and Live lock Trojan is addressed in this work. To explain the attack scenario, Bit-compliment type of traffic is used which is symmetric in all directions with respect to centre of the network so that we can easily track the packets.

C. Measurements and Estimates

The metrics that considered to evaluate the performance of Fortified-NoC are the following:

1) Latency: Latency is measured as cycles that a packet takes to reach its destination while going through the network:

$$\text{Avg. Latency} = \frac{\text{Total latency or latency of all packets}}{\text{Total received packets}}. \quad (1)$$

Low latency indicates a smooth flow of packets and best utilisation of resources, and in another hand, high latency indicates high traffic and waiting due to insufficient resources.

2) Throughput: It is defined as the rate at which network can successfully accept and deliver the injected packet:

$$\text{Avg. Throughput} = \frac{\text{(Total received packets)*(Packetlength)}}{\text{(x-dim * y-dim)*(Total sim. cycles)}}. \quad (2)$$

It is a measure of networks ability to transfer the data at most possible rate so that cores can communicate and process data at a faster rate.

3) Free link availability: It estimates the percentage of the free links available for a particular configuration of NoC:

$$\text{Link traffic} = \frac{\text{(Total traffic) * (Packet length)}}{\text{(Active links available) * (Active sim. cycles)}}. \quad (3)$$

If more free links are available, the ability of the network to handle high amount of data transfer will be more. Link availability (%) == 100 (For link traffic = 1).

4) Total packets received: A direct indication of missing packets which need to be delivered to destined nodes. Low count of total received packets indicates packet misrouting, packet loss, packet corruption etc.

5) Traffic: Number of packets that are entering a particular router keeping it busy. This distribution indicates traffic congestion and critical routes.

VII. Results and discussion

A. Performance Degrading Trojans

1) Average latency:: Fig. 11(a) shows that HBT causing an additional latency of 5.67% and our method is causing an additional delay of 11.51% which is unavoidable due to specialized routing. Fig. 11(b) shows the DAT scenario which indicates almost same latency in all cases but as the traffic rate increases, the Trojan affected scenario is having a tremendous increase in latency which could be due to unauthorised use of critical resources like links which are not meant to be. The permutation method and proposed method both are able to restrict this sharp rise. Fig. 11(c) is to show latency variations in PLT scenario. Trojan is responsible for an increase of 5.7% in latency and the mitigation methods are not performing to control this rise as expected.

2) Average throughput:: In Fig. 12(a) we can see a decline of 25% in throughput when HBT brings its action and both the mitigation methods are capable of retrieving the original standards. When DAT delivers the attack, there is no much change in throughput because this attack only misplaces the delivery. As shown in Fig. 12(b), the permutation method gives a bad performance and our method is good in this case. PLT reduces the throughput by 25% and the permutation method is good enough in getting it high and our method is able to get it high by more than 99% when compared to the loss due to Trojan as shown in Fig. 12(c).

3) Total packets distributed:: Fig. 13(a) shows that the HBT is not allowing proper communication between source to destination. There is a dip of 27.25% when Trojan attacks head bit. The low packet count is an indication of lost or diverted or dropped packets. The mitigation methods are good enough to recover the lost packets to the cent. As discussed earlier, when DAT attacks and changes the original bit field, there should be no packet loss except at very high intake rate of flits and is indicated in Fig. 13(b). But the permutation method is not retaining this property instead there is a decrease of 8.88% in packet count. The proposed method is able to retain this. There is a 27.63% reduction in packets received in PLT case as presented in Fig. 13(c).

The permutation method is able to recover 89.04% of lost packets whereas proposed method is capable of recovering 85.79%.

4) Traffic distribution: Traffic distribution shows how traffic patterns are being formed for a particular configuration of the network and how much traffic is there at each router. Fig. 14(a) indicates a disruption in traffic which means packets are diverted when DAT strikes the target and distribution of proposed method clearly shows the efficiency of routing algorithm isolating the Trojan affected router. Fig. 14(b) is for PLT situation.

5) Node wise packet reception: Fig. 15(a) gives a clear indication of diversion in packets delivered. It also shows that the Proposed method is not capable of retrieving the packets when destination bit field is attacked. The proposed method is far better in improving the condition as indicated in the figure.

The PLT attack result in packets delivery is shown in Fig. 15(b) and the permutation method fails to stop it where as proposed method gets succeeded in defending the Trojan.

B. Data Leakage Trojan

Fig. 16(a) gives packet distribution which shows missing packets at node 3,2 and an increase in packet count at node 2,2. The permutation method also failed because it alone couldn't make the case perfect because there is a loss in packets in tile 2,1. As our method aims at isolating that malicious router apart from the data shuffling, leakage is avoided and the original packet distribution is restored. Fig. 16(b) is traffic distribution which shows missing traffic in nodes 3,1 and 3,2 due to leakage and with our method, packets are successfully delivered to destined node. As there are no other calamities such as packet loss, waiting, and resource blockage, the other parameters of the network such as latency and throughput are unaffected. Apart from this, encryption of data in the core also makes any successful attacks of data leakage meaningless.

C. Live Lock Trojan

Fig. 17(a) shows the packet reduction in several nodes as discussed and this is a kind of denial of service for those nodes who lost their packets. The Permutation method can

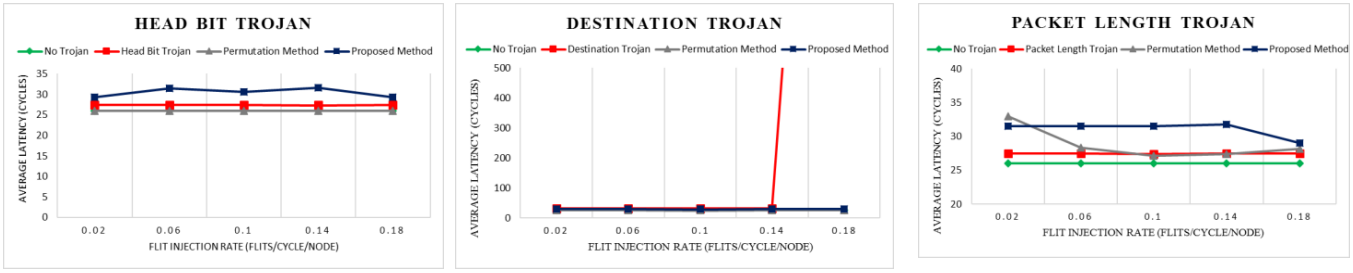


Fig. 11. Average latency of(a)Head Bit Trojan (b)Destination Address Trojan c) Packet Length Trojan

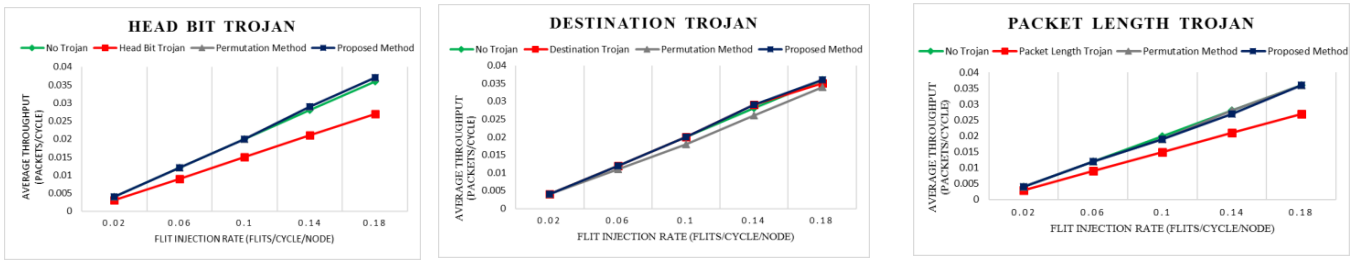


Fig. 12. Average throughput (a)Head Bit Trojan (b)Destination Address Trojan (c)Packet Length Trojan

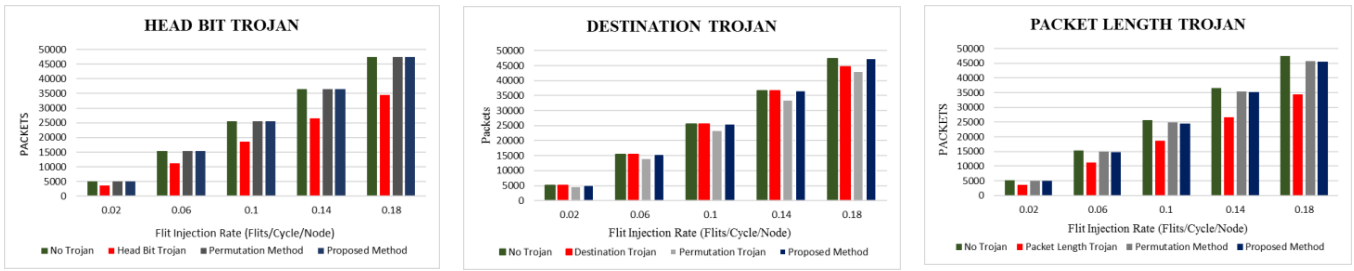


Fig. 13. Total packets received (a)Head Bit Trojan (b)Destination Address Trojan (c)Packet length Trojan.

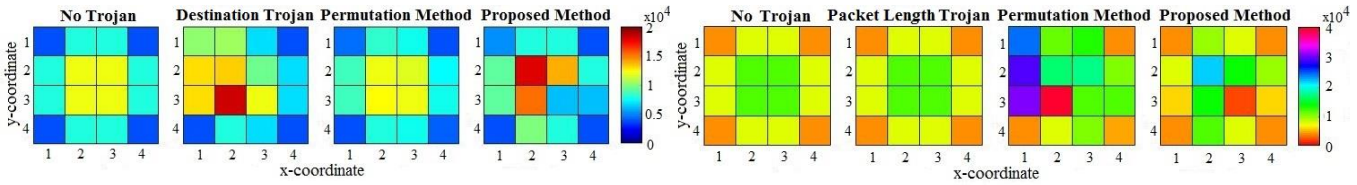


Fig. 14. Traffic pattern of (a)Destination Address Trojan scenario (b) Packet Length Trojan scenario

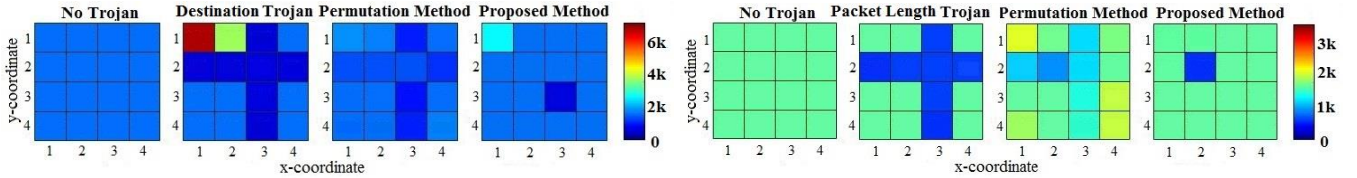


Fig. 15. Node wise packet received in (a)Destination Address Trojan scenario (b) Packet Length Trojan scenario

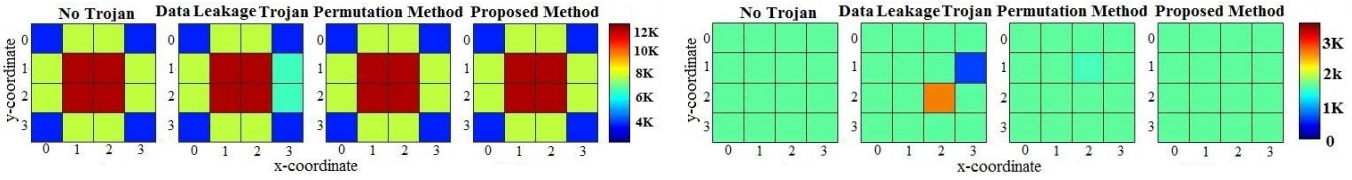


Fig. 16. (a)Traffic pattern in Data Leakage Trojan scenario (b)Node wise packets received in Data Leakage Trojan Scenario

only save packets destined to the local router but not do anything to mitigate the route change by Trojan. The proposed method is able to handle this well and prevent the Trojan from getting succeeded. Fig. 17(b) shows node wise packets received in Live Lock Trojan. Fig.18(a) indicates that the latency when an attack happens is increased by 70% on an average over the simulations samples. The base method and proposed method both are able to lower it back to the normal scenario. Throughput is reduced by 15% and the base method is unable to recover it while the proposed method is recovered the throughput fully and is shown Fig. 18(b). Fig. 18(c) shows the total received packet count is reduced by 16.79% and the permutation method is able to recover only 7.35% while our method is able to recover it completely.

D. Overall Link Availability

Table I gives some idea regarding the free link availability in percentage. The values in no Trojan scenario is the ideal case with optimum and normal link usage. If the link availability is increased, that means packets are not transferred properly and is an indication of packet loss. The simulation runs taken for evaluating link availability is 0.1 flits/cycle/node. In the HBT case, the rise in value can be compensated to original value by both the methods. In DAT scenario, the reduction in link availability is due to the dispersion of packets throughout the network and the link availability is improved with the proposed method. When PLT attacks the flits, there is no change even with Trojan presence but the permutation method is having less free links than the proposed method. It indicates that the traffic handling is better in the proposed method. In DLT, leakage has no effect on link availability. In LLT, the maintenance of free links is same as in normal scenario which is not in the other case as can be seen in the Table I.

TABLE I
OVERALL FREE LINK AVAILABILITY

Trojan Type	No Trojan Case	Trojan Case	Permutation Method	Proposed Method
HBT	87	89	87	87
DAT	87	85	87	86
PLT	87	87	74	86
DLT	87	87	87	87
LLT	87	88	89	87

E. Hardware and Overhead Costs

We implemented a 4×4 NoC using Verilog HDL with flit width of 64 and buffer depth of 8 to completely match the simulation setup used in this work. We used Design Vision from Synopsys with 90nm technology for synthesis. Table II gives block wise hardware cost in terms of area when compared with the actual router design.

TABLE II
HARDWARE COSTS.

Module	Units Used	Area Overhead (%)
Shuffler	5	1.278
De-Shuffler	5	1.278
Hamming coder	5	0.954
Hamming Decoder	5	4.584
Address Extractor	5	0.458
Encryptor	1	0.219
Decryptor	1	0.218
TCRA	5	0.512
Total Cost	-	9.501

The extra bits NF and Tr are overhead bits which take one-bit space each from data field. The Tr bit is present in all the flit so it gives a flit overhead of 1.58% which is affordable and slightly more for overall packet due to presence of an extra bit NF in head flit.

VIII. CONCLUSIONS AND FUTURE RESEARCH

As the consumer electronics and semiconductor industry is moving towards MPSoC configurations, the tendency of security challenges is increasing rapidly. As NoC is an important module in SoCs, it needs attention in developing new measures or revamp the existing methods to suit NoC architectures to curb the security challenges especially hardware Trojan attacks. The proposed method exploits the need for security arrangements in the router rather than NI in which promising safeguarding methods are already available. The proposed method is able to save all packets which are not associated with Trojan affected router. Even for the packets generated or received in that router, the other tier of security, TDS and ECC are good enough to obfuscate the flit data and prevent successful attacks. Apart from these, data encryption also assures data protection against snooping, leakage, processing, and stealing before the packets enter the router.

The experimental results show that our methodology can tackle a wide range of HTs which include not only performance degradation, denial of service or data leakage but also

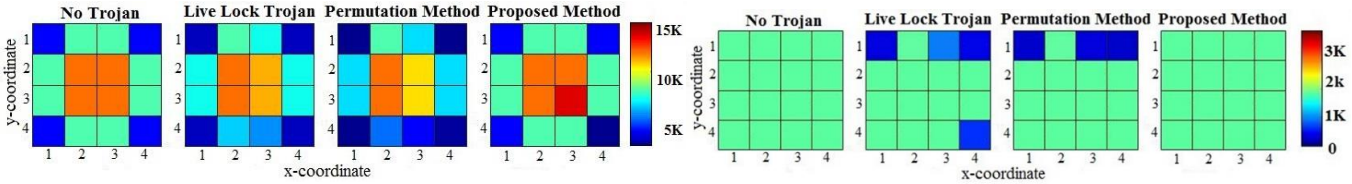


Fig. 17. (a)Traffic pattern of Live Lock Trojan scenario (b) Node wise packets received in Live Lock Trojan scenario.

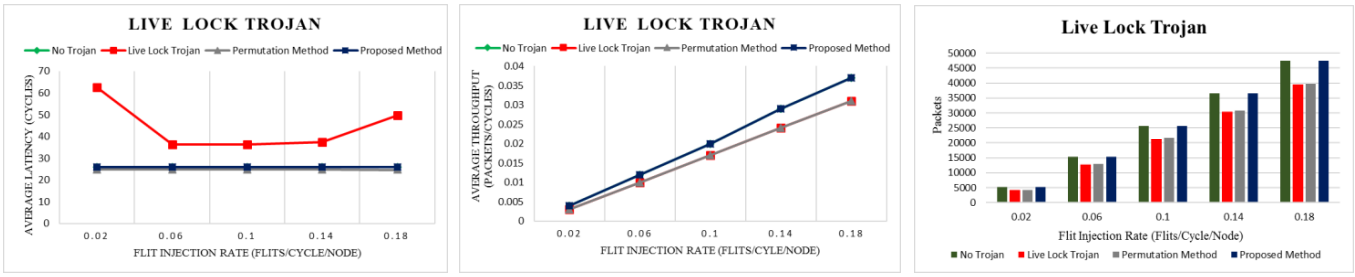


Fig. 18. (a)Average latency for Live Lock Trojan (b) Average throughput for Live Lock Trojan (c)Total packets received for Live Lock Trojan.

unusual attacks like live locking of packets etc. Our method outweighs the other method considered for comparison which is better than many proposed methods, in many aspects like average throughput, packet delivery, and free link availability. Even though One complete Trojan hardware can deliver the targets, our method is capable of dealing multiple (and individual) Trojans placed in different routers.

Our future research in this area include security against HT attacks such as illegal packet requests, packet flushing, packet duplication etc and also against Hardware Trojans inserted with an IP. We will also validate the proposed method in embedded system applications such as smart-TV SoC.

ACKNOWLEDGMENT

Our cordial thanks to Ministry of Electronics and Information Technology, Government of India for supporting us by providing high-end VLSI design tools for research purposes. We also thank the developers of Open Source NoCTweak simulation package.

REFERENCES

[1] E. Choi and S. Chang, "A Consumer Tracking Estimator for Vehicles in GPS-free Environments," *IEEE Trans. Consum. Electron.*, vol. 63, no. 4, pp. 450–458, November 2017.

[2] J. Kim, E. Jung, Y. Lee, and W. Ryu, "Home Appliance Control Framework based on Smart TV Set-Top Box," *IEEE Trans. Consum. Electron.*, vol. 61, no. 3, pp. 279–285, August 2015.

[3] T. Perumal, A. R. Ramli, and C. Y. Leong, "Interoperability Framework for Smart Home Systems," *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, pp. 1607–1611, November 2011.

[4] A. K. Swain, A. K. Rajput, and K. K. Mahapatra, "Network on Chip for Consumer Electronics Devices: An Architectural and Performance Exploration of Synchronous and Asynchronous Network-on-Chip-Based Systems," *IEEE Consum. Electron. Mag.*, vol. 8, no. 3, pp. 50–54, May 2019.

[5] R. Obermaisser, H. Kopetz, and C. Paukovits, "A Cross-Domain Multiprocessor System-on-a-Chip for Embedded Real-Time Systems," *IEEE Trans. Indust. Info.*, vol. 6, no. 4, pp. 548–567, Nov 2010.

[6] Z. Tan, S. Zheng, P. Liu, and G. Lin, "An Implementation of Open Source Operating System on Multiprocessor System-on-a-Chip," *IEEE Trans. Consum. Electron.*, vol. 52, no. 3, pp. 1118–1123, August 2006.

[7] M. Pastrnak, P. H. n. De With, and J. V. Meerbergen, "QoS Concept for Scalable MPEG-4 Video Object Decoding on Multimedia (NoC) chips," *IEEE Trans. Consum. Electron.*, vol. 52, no. 4, pp. 1418–1426, Nov 2006.

[8] "Paves the Way to Autonomy with Scalable, Customizable and Power Efficient Autonomous Driving Platform and Qualcomm Snapdragon Ride Autonomous Stack," https://www.qualcomm.com/news/releases/2020/01/06/qualcomm-accelerates-autonomous-driving-new-platform-qualcomm -snapdragon, Jan. 2020.

[9] K. C. S. P. Kapur and S. Souri, "Performance Limitations of Metal Interconnects and Possible Alternatives," in *Proc. Intl. Sympo. ULSI Process Inte. III*, vol. 2003, 2003, p. 194.

[10] S. Li, K. Lim, P. Faraboschi, J. Chang, P. Ranganathan, and N. P. Jouppi, "System-level Integrated Server Architectures for Scale-out Datacenters," in *Proc. 44th IEEE/ACM Intl. Sympo. Microarch.*, 2011, pp. 260–271.

[11] W. C. Tsai, Y. C. Lan, Y. H. Hu, and S. J. Chen, "Networks on Chips: Structure and Design Methodologies," *Jo. Elect. Comp. Eng.*, vol. 2012, no. 4, pp. 450–458, 2012.

[12] S. P. Mohanty, "Security and Privacy by Design is Key in the Internet of Everything (IoE) Era," *IEEE Consum. Electron. Mag.*, vol. 9, no. 2, pp. 4–5, Mar 2020.

[13] S. Adee, "The hunt for the kill switch," *IEEE Spectrum*, vol. 45, no. 5, pp. 34–39, 2008.

[14] "The Quantum Program of NSA,," Available on-line: https ://www.nytimes.com/2014/01/15/us/nsa-effort-spries-open-computers-not-connected-to-internet.html.

[15] C. M. J. S. Marcus Niemietz, Juraj Somorovsky, "Not so smart: On smart tv apps," in *International Workshop on Secure Internet of Things*, 2015.

[16] "FlexNoC Interconnect IP," https://www.arteris.com/flexnoc, Last visited on Nov. 2021.

[17] P. Coussy, A. Baganne, and E. Martin, "IP Cores Integration in DSP System-on-Chip Designs," in *Proc. Eur. Signal Proc. Conf.*, 2002, pp. 1–4.

[18] R. JayashankaraShridevi, D. M. Ancajas, K. Chakraborty, and S. Roy, "Security Measures Against a Rogue Network-on-Chip," *J. Hardware Syst. Secu,*, vol. 1, no. 2, pp. 173–187, May 2017.

[19] D. M. Ancajas, K. Chakraborty, and S. Roy, "Fort-NoCs: Mitigating the Threat of a Compromised NoC," in *Proc. 51st Annu. Des. Automation Conf.*, 2014, pp. 1–6.

[20] N. Prasad, R. Karmakar, S. Chattopadhyay, and I. Chakrabarti, "Runtime

Mitigation of Illegal Packet Request Attacks in Networks-on-Chip," in *Proc. IEEE Int. Symp. on Circ. and Sys.*, 2017, pp. 1–4.
[21] S. K. Kumar, N. Satheesh, A. Mahapatra, S. Sahoo, and K. K. Mahapatra, "Physical Unclonable Functions for On-Chip Instrumentation: Enhancing the Security of the Internal Joint Test Action Group Network," *IEEE Consum. Electron. Mag.*, vol. 8, no. 4, pp. 62–66, July 2019.
[22] V. P. Yanambaka, S. P. Mohanty, E. Kougianos, and D. Puthal, "PMsec: Physical Unclonable Function-Based Robust and Lightweight Authentication in the Internet of Medical Things," *IEEE Trans. Consu. Elect.*, vol. 65, no. 3, pp. 388–397, Aug 2019.
[23] S. Lee, C. Tsou, and P. Huang, "Ultra-High-Frequency Radio-Frequency-Identification Baseband Processor Design for Bio-Signal Acquisition and Wireless Transmission in Healthcare System," *IEEE Trans. Consu. Elect.*, vol. 66, no. 1, pp. 77–86, Feb 2020.
[24] J. W. Kim, J. H. Lim, S. M. Moon, and B. Jang, "Collecting Health Lifelog Data From Smartwatch Users in a Privacy-Preserving Manner," *IEEE Trans. Consu. Elect.*, vol. 65, no. 3, pp. 369–378, Aug 2019.
[25] H. Ju, Y. Kim, Y. Jeon, and J. Kim, "Implementation of a Hardware Security Chip for Mobile Devices," *IEEE Trans. Consum. Electron.*, vol. 61, no. 4, pp. 500–506, Nov 2015.
[26] B. Choi, S. Lee, J. Na, and J. Lee, "Secure Firmware Validation and Update for Consumer Devices in Home Networking," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 1, pp. 39–44, Feb 2016.
[27] H. Thapliyal, S. P. Mohanty, and S. Prowell, "Emerging Paradigms in Vehicular Cybersecurity," *IEEE Consum. Electron. Mag.*, vol. 8, no. 6, pp. 81–83, Nov 2019.
[28] E. Alepis and C. Patsakis, "Monkey Says, Monkey Does: Security and Privacy on Voice Assistants," *IEEE Access*, vol. 5, pp. 17 841–17 851, 2017.
[29] K. Sajeesh and H. K. Kapoor, "An Authenticated Encryption based Security Framework for NoC Architectures," in *Proc. Int. Symp. on Elect. Sys. Des.*, 2011, pp. 134–139.
[30] A. K. Biswas, S. Nandy, and R. Narayan, "Router Attack toward NoC-enabled MPSoC and Monitoring Countermeasures Against such Threat," *Circuits, Syst., and Sig. Processing*, vol. 34, no. 10, pp. 3241–3290, February 2015.
[31] J. Frey and Q. Yu, "A Hardened Network-on-Chip Design using Runtime Hardware Trojan Mitigation Methods," *Integration*, vol. 56, pp. 15–31, January 2017.
[32] T. Boraten and A. K. Kodi, "Packet Security with Path Sensitization for NoCs," in *Proc. Design, Auto. & Test in Europe Conf. & Exhibition*, 2016, pp. 1136–1139.
[33] T. Boraten and A. Kodi, "Mitigation of Denial of Service Attack with Hardware Trojans in NoC Architectures," in *Proc. IEEE Intl. Parallel Distributed Processing Sympo.*, 2016, pp. 1091–1100.
[34] J. Frey and Q. Yu, "Exploiting State Obfuscation to Detect Hardware Trojans in NoC Network Interfaces," in *Proc. IEEE Int. Midwest Symp. Circuits and Syst.*, 2015, pp. 1–4.
[35] R. D. Bajaj and U. Gokhale, "AES Algorithm for Encryption," *Int. J. Latest Res. in Eng. Tech.*, vol. 2, no. 05, pp. 63–68, 2016.
[36] Y. Wang and G. E. Suh, "Efficient Timing Channel Protection for On-chip Networks," in *Proc. IEEE/ACM Int. Symp. on Networks-on-Chip*, 2012, pp. 142–151.
[37] S. Narasimhan, W. Yueh, X. Wang, S. Mukhopadhyay, and S. Bhunia, "Improving IC Security Against Trojan Attacks through Integration of Security Monitors," *IEEE Des. & Test of Computers*, vol. 29, no. 5, pp. 37–46, Oct. 2012.
[38] A. T. Tran and B. Baas, *NoCTweak: a Highly Parameterizable Simulator for Early Exploration of Performance and Energy of Networks-on-Chip*, VLSI Computation Lab, ECE Department, UC Davis, July 2012.
[39] S. Baron, M. S. Wangham, and C. A. Zeferino, "Security Mechanisms to Improve the Availability of a Network-on-Chip," in *Proc. IEEE Int. Conf. on Elect., Circuits, and Sys.*, 2013, pp. 609–612.
[40] L. Fiorin, G. Palermo, S. Lukovic, V. Catalano, and C. Silvano, "Secure Memory Accesses on Networks-on-Chip," *IEEE Trans. Comp.*, vol. 57, no. 9, pp. 1216–1229, Sep. 2008.

Manoj Kumar JYV received Bachelor's degree in Electronics and Communications Engineering from Gayatri Vidhya Parishad College of Engineering (A), Visakhapatnam, in 2016, Master's degree in VLSI Design and Embedded Systems from National Institute of Technology, Rourkela, in 2018. He is currently with Cadence Design Systems (India) Pvt. Ltd. His research interests include Hardware Security, System on Chip and Low power design.

Ayas Kanta Swain received the bachelor's degree (Honors) in electrical engineering from Indira Gandhi Institute of Technology, Sarang, from Utkal University, odisha, in 2001, the master's in research degree in Electronics and Communication Engineering from the National Institute of Technology, Rourkela, in 2009, and continuing his Ph.D. degree in Electronics and Communication Engineering from the National Institute of Technology, Rourkela. He is an Assistant Professor with NIT Rourkela. His research are includes System on Chip Design and Embedded Systems.

Kamala Kanta Mahapatra is a professor in Electronics and Communication Engineering Department of National Institute of Technology, Rourkela, since February 2004. He obtained his BTech degree from Regional Engineering College (now National Institute of Technology), Calicut, Kerala, India, in 1985, MTech from Regional Engineering College (now National Institute of Technology), Rourkela, in 1989, and PhD from Indian Institute of Technology, Kanpur, India, in 1999.He is a fellow of the Institution of Engineers (India) in ECE Division. He has published several research papers in national and international journals. His research interests include embedded computing systems, VLSI design, electronic circuits, and industrial electronics.

Saraju P. Mohanty (SM'08) received the bachelor's degree (Honors) in electrical engineering from the Orissa University of Agriculture and Technology, Bhubaneswar, in 1995, the master's degree in Systems Science and Automation from the Indian Institute of Science, Bengaluru, in 1999, and the Ph.D. degree in Computer Science and Engineering from the University of South Florida, Tampa, in 2003. He is a Professor with the University of North Texas. His research is in "Smart Electronic Systems" which has been funded by National Science Foundations (NSF), Semiconductor Research Corporation (SRC), U.S. Air Force, IUSSTF, and Mission Innovation. He has authored 400 research articles, 4 books, and invented 7 granted/pending patents. His Google Scholar h-index is 45 and i10-index is 172 with 8100 citations. He is a recipient of 13 best paper awards, Fulbright Specialist Award in 2020, IEEE Consumer Electronics Society Outstanding Service Award in 2020, the IEEE-CS-TCVLSI Distinguished Leadership Award in 2018, and the PROSE Award for Best Textbook in Physical Sciences and Mathematics category in 2016. He has delivered 12 keynotes and served on 12 panels at various International Conferences. He has been the Editor-in-Chief (EiC) of the IEEE Consumer Electronics Magazine (MCE) during 2016–2021 and serves on the editorial board of 6 journals/transactions.