Securing Network-on-Chips via Novel Anonymous Routing

Amin Sarihi
sarihi@nmsu.edu
New Mexico State University
Las Cruces, NM, USA

Ahmad Patooghy
apatooghy@ncat.edu
North Carolina A&T State University
Greensboro, NC, USA

Mahdi Hasanzadeh
m.hasanzadehh@gmail.com
Independent Scholar
Tehran, Iran

Mostafa Abdelrehim
mabdelrehim@csub.edu
California State University
Bakersfield, CA, USA

Abdel-Hameed A. Badawy
badawy@nmsu.edu
New Mexico State University
Las Cruces, NM, USA

ABSTRACT

Network-on-Chip (NoC) is widely used as an efficient communication architecture in multi-core and many-core System-on-Chips (SoCs). However, the shared communication resources in NoCs, e.g., channels, buffers, and routers might be used to conduct attacks compromising the security of NoC-based SoCs. Almost all of the proposed encryption-based protection methods in the literature need to leave some parts of the packet unencrypted to allow the routers to process/forward packets accordingly. This uncovers the source/destination information of the packet to malicious routers, which can be used in various attacks. In this paper, we propose the idea of secure anonymous routing with minimal hardware overhead to hide the source/destination information while exchanging secure information over the network. The proposed method uses a novel source-routing algorithm that works with encrypted destination addresses and prevents malicious routers from discovering the source/destination of secure packets. To support our proposal, we have designed and implemented a new NoC architecture that works with encrypted addresses. The conducted hardware evaluations show that the proposed security solution combats the security threats at an affordable cost of 1% area and 10% power overheads chip-wide.

KEYWORDS

Network-on-Chips, System-on-Chips, Security, Hardware Trojan, Anonymous Routing, Source Routing

1 INTRODUCTION

Today's IC fabrication paradigm allows multiple companies to contribute to the design, design integration, fabrication, testing, and packaging of a chip [14]. In this process, third-party intellectual properties (3PIPs) are widely used to reduce the design cost and minimize time to market [2, 14]. Multi-processor system-on-chips (MPSoCs) are one of the example cases in which 3PIP processing cores, memory modules, and I/Os offered by different providers, are integrated to construct the final design. Although outsourcing will reduce the design/manufacturing time and cost, it is vulnerable to new security risks such as hardware Trojans (HTs). An HT is any unwanted modifications or functionalities trying to achieve or facilitate malicious goals in the final product [19].

As most of the modern MPSoCs use an on-chip network at their backbone communication architecture, the industry has already started offering Network-on-Chip (NoC) IPs (for instance, FlexNoC IP from Arteris company [1, 12]). The main idea of on-chip networks, which is resource sharing to boost resource utilization, might complicate the security challenges of modern MPSoCs. More specifically, several on-chip routers contribute to forwarding a packet that might not be directly related to them. The benefit gained by resource sharing in NoC fabrics may be counter-productive when data integrity and confidentiality are taken into account. As NoC routers demand easy access to the source/destination addresses in the header flits, the application of data encryption methods is only limited to the data flits of packets. Indeed, the encryption of the header information will adversely impact the performance of the MPSoC as each router will have to decrypt, process, and encrypt the packet's header. As a result, a wide range of security attacks that depend on the source/destination pairs of sensitive packets are feasible. These attacks could lead to stealing sensitive data and exacerbating security. This emphasizes the need to consider security in the design of the communication fabric of MPSoCs.

To address the mentioned issue, we propose and evaluate an encryption-based NoC architecture in this paper. To facilitate the routing of encrypted packets, a novel path computation method, which computes the path at the source node, is also proposed. The pre-computed path embedded into the header of a secure packet does not reveal any information about the source/destination, yet it is enough to guide NoC routers to route the packet. Upon arrival at the destination node, the encrypted destination address embedded in every secure packet will match the router's pre-encrypted destination, and the packet is ejected to the local core. Later in the paper, we discuss how the proposed anonymous routing helps sensitive data from being leaked to any malicious node/router in the network. The contributions of this paper are as follows:

- Presenting an efficient router architecture capable of supporting traditional routing for non-secure packets along with source routing for secure data transmission.
- Boosting the security of NoCs by offering a feasible way of encrypting the entire packet to hide the source/destination of sensitive information.

NOCS '21, October 14–15, 2021, Virtual Event, USA
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-9083-5/21/10…$15.00
https://doi.org/10.1145/3479876.3481592

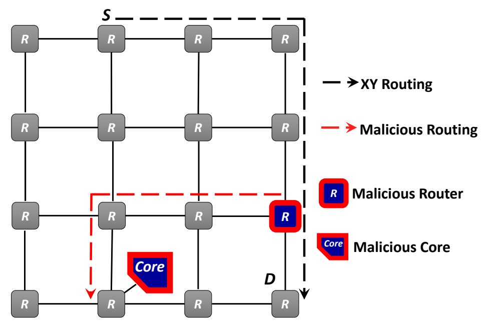


Figure 1: An example of the target threat model consisting of an HT-infected router and a malicious task.

The rest of this paper is organized as follows. Section 2 presents the necessary background and the addressed security threats. Section 3 discusses the proposed security system, including the architecture and the routing method. Simulation experiments and the obtained results are presented and discussed in Section 4, and finally, Section 5 concludes the paper.

2 THREAT MODELS OF NOCS

In SoCs, cores are connected to the on-chip network using their local network interfaces (NIs) and routers. Cores and NIs deal with the local data only. Routers, on the other hand, are in charge of forwarding both the local and global data. As a result, adversaries prefer routers to host their HTs to wreak havoc on security. Generally, security goals are summarized as confidentiality, integrity, and availability, and the adversary's goal is to compromise at least one of these goals through a plethora of attacks [18]. The widely addressed attacks applicable to NoCs consist of: denial of service attacks (packet flooding [13], packet dropping [9]), packet tampering [3, 5], packet duplication [3, 6, 7], and eavesdropping [6, 7].

An example of the addressed threat model in this paper is depicted in Figure 1. Per this threat, an HT-infected router (the red highlighted router in Figure 1) sniffs secure information being exchanged between a source/destination pair (S and D pair in Figure 1). The malicious router may copy the packet contents and send them to a malicious core (the red highlighted core shown) that runs rogue software for further processing/analysis to discover sensitive information. Even though the sender might encrypt the sensitive data before injecting it into the network, cryptanalysis attacks can still be launched to infer secret encryption keys. Timing attacks are one example of cryptanalysis in which an adversary makes intentional collisions between its own data and the sensitive data on a specific path to obtain valuable information about the timing, and the volume of the sensitive information [15, 16]. Other studies such as [6, 7] mention linear/differential cryptanalysis in their threat models as well to crack the key. It is worth mentioning that data corruption attacks are not the target of our study.

3 PROPOSED SECURITY SYSTEM

In the proposed security system, packets are divided into secure and non-secure packets. Secure packets have high-security requirements, i.e., their information must be encrypted, and neither the packet contents nor their source/destination addresses should be disclosed to an unauthorized party. Conversely, non-secure packets are sent in plaintext since they do not contain sensitive information. Hence, two different routing mechanisms (one for each packet type) are employed that will be later described in this section.

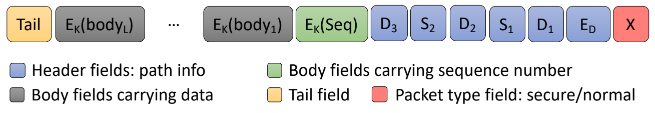

Tail
E_K(body_L)
...
E_K(body_1)
E_K(Seq)
D3
S2
D2
S1
D1
E_D
X
Header fields: path info
Body fields carrying data
Body fields carrying sequence number
Tail field
Packet type field: secure/normal

Figure 2: Format of a secure packet in the proposed anonymous routing.

Algorithm 1 : Secure Path Computation

```
Input: Source & Destination Address Plaintext $P_s, P_d$;
Output: Encrypted Destination Address $E_D$;
Output: Strides $S_1, S_2$, Directions $D_1, D_2, D_3$;
Output: Packet type $X_{bit}$;
1:  $X_{bit} \leftarrow 1$;
2:  $E_D \leftarrow HB\text{-}2(P_d)$;
3:  $\Delta X = P_d.X - P_s.X$;
4:  $\Delta Y = P_d.Y - P_s.Y$;
5:  Set $Next_{VC} = 1$;            # Initial VC for secure packets
6:  Randomly GOTO L1 or L2;
    L1:                             # Path with real and misleading turns
7:  $S_1 \leftarrow abs(\Delta X)$;
8:  $S_2 \leftarrow Rand(n)$;           # To mislead the attacker
9:  $D_3 \leftarrow Rand(X^+, X^-)$;    # To mislead the attacker
10: if ($\Delta X \geq 0$) then
11:     $D_1 \leftarrow X^-$;
12: else
13:     $D_1 \leftarrow X^+$;
14: end if
15: if ($\Delta Y \geq 0$) then
16:     $D_2 \leftarrow Y^-$;
17: else
18:     $D_2 \leftarrow Y^+$;
19: end if
20: Return;
    L2:                             # Path contains two real turns
21: $m \leftarrow RandBetween(0, \Delta X - 1)$;
22: $S_1 \leftarrow abs(\Delta X) - m$;    # Take $m$ hops once returned to $X$
23: $S_2 \leftarrow abs(\Delta Y)$;
24: if ($\Delta X \geq 0$) then
25:     $D_1, D_3 \leftarrow X^-$;
26: else
27:     $D_1, D_3 \leftarrow X^+$;
28: end if
29: if ($\Delta Y \geq 0$) then
30:     $D_2 \leftarrow Y^-$;
31: else
32:     $D_2 \leftarrow Y^+$;
33: end if
34: Return;
```

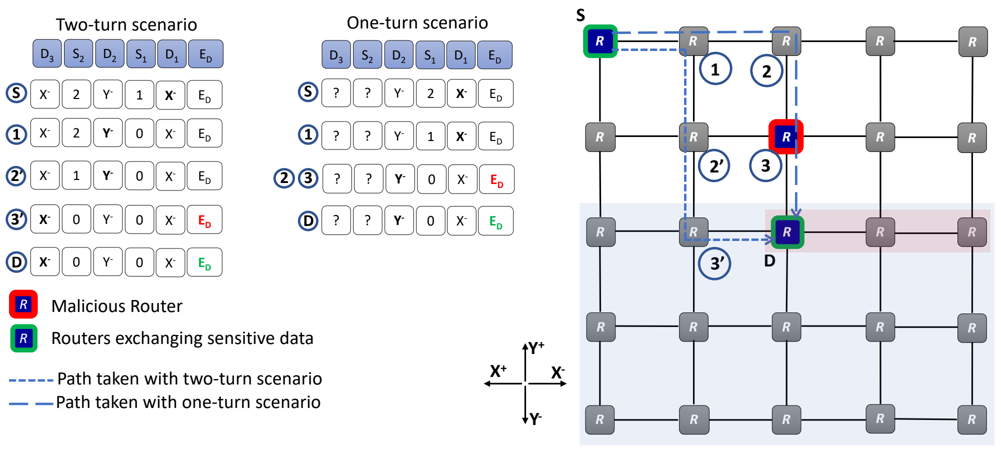

Two-turn scenario
One-turn scenario
S
D
Malicious Router
Routers exchanging sensitive data
Path taken with two-turn scenario
Path taken with one-turn scenario
Y+
Y-
X+
X-

Figure 3: Example cases of routing a secure packet in the network with one or two legitimate turns embedded in the packet. The secure packet carries information that includes when and where to turn. From the attacker's point of view, all routers in red or blue shaded regions are potential destinations of a two-turn or one-turn packet, respectively.

3.1 Secure Anonymous Routing

The idea behind secure source-routing is to exclude source/destination addresses in the packet. Instead, the approximate route and turns a packet would need are embedded in secure packets. The secure routing, which is based on the source XYX routing method, either offers a 1- or 2-turn path for secure packets. Although the header of a secure packet carries the information of both turns, in some cases, the second turn will never be used. Indeed, we have devised two scenarios for source XYX routing: 1) a path with one misleading turn and 2) a path with two real turns. The four cases could happen to a secure packet are i) the packet is being forwarded in the direction from which it was received, ii) the packet is redirected from X (horizontal) direction to Y (vertical) direction, iii) the packet makes the second turn, i.e., turns from Y to X, or iv) the packet is being ejected from the network upon reaching the destination.

Information embedded in the header of secure packets (format is shown in Figure 2) is as follows. The first field (X_{bit}) is a single bit which indicates the packet type: secure or normal (non-secure). The E_D field carries the encrypted version of the destination address for anonymous routing (details will be explained later in this section). The packet traversal directions can be found in D_1 and $D2$ fields, and they can be one of these four options: $X^+, X^-, Y^+,$ and Y^-. S_1, S_2, and S_3 represent the number of strides (hops) the packet has to traverse in the X, Y, and X directions, respectively. The next field is the packet's encrypted sequence number, $E_K(Seq)$. The fields E_D, $E_K(Seq)$, and $E_K(body_i)$ are encrypted using a pre-shared key K using the symmetric-key encryption scheme. The encryption adopted is Hummingbird-2 (HB-2) which will be discussed later in Section 3.3.1. This key needs to be exchanged in prior between an *initiator* and a *recipient* node. Body flits are then encrypted using the same key. Lastly, the packet is terminated with a tail flit transmitted in plaintext.

A secure packet will be ejected from the network only if its encrypted destination field is equal to the encrypted address of the router.[1] Otherwise, S_1 field will be checked, and in case it is non-zero, it will be decremented by 1, and the packet will be moved as D_1 commands. After S_1 becomes 0, the same condition will be rechecked for S_2, and it will be decremented by 1 as well. At last, if neither of the above conditions is met, the packet will be forwarded according to D_3 until it reaches its destination. As for the non-secure packets, we use the well-known dimension order (XY) routing to allow them to share virtual channels with secure packets.

To enable both routing configurations, packet headers must support both methods. Hence, a secure header generator is needed in the NI, and it is described as in Algorithm 1. It first sets the packet type; then assigns the encrypted destination address to the E_D field. The horizontal and vertical differences between the current node and the destination nodes are computed in lines 3-4. The algorithm randomly chooses between $L1$ and $L2$ subroutines to generate the path. $L1$ generates a path that contains a real and a misleading turn, whereas $L2$ generates a path with two real turns. As can be noticed, in $L1$ subroutine (lines 7-20), S_1, D_1, and D_2 are computed deterministically based on the source/destination coordinates; however, S_2 and D_3 are chosen randomly to mislead the attacker about the packet's destination. Conversely, in $L2$ (lines 21-33), S_1, S_2, D_1, D_2, D_3 are computed with some levels of adaptivity. This is done through a random selection of the variable m which denotes the numbers of hops to be taken in the first and last X movements. So, in the case of having a malicious router in the path, there is always a high chance to bypass it.

Two examples (Figure 3) are provided for each of the routing scenarios. In the one-turn routing scenario, S_2 and D_3 fields are randomly assigned to hide the real destination. The packet will be ejected to the local port at the destination node. A misleading path could either make a turn at the destination or either go further in the Y direction and turn. Although the packet will eventually reach its destination node, from the malicious router's (may be any of nodes 1, 2, 2′, 3) point of view, numerous nodes can be the ultimate destination of the packet (the area highlighted in blue). To further enhance the path diversity, two-turn routing was leveraged as well, in which all the header fields are pre-assigned deterministically. In this scenario, the secure packet is able to detour the malicious node (given such information is available) by making the first turn at node 1 even before reaching the destination column. It makes a second turn at node 3′ and then continues moving in the X direction. The packet will eventually exit the network when the comparison of E_D and E_R is true; however, the dummy path (the red region) will mislead the intermediate node 3′ if it is a malicious one.

[1]Since the destination address, E_D, is encrypted, every router must also have its encrypted address, E_R, ready for comparison. This way, there is no need to decrypt the field E_D to ensure security.

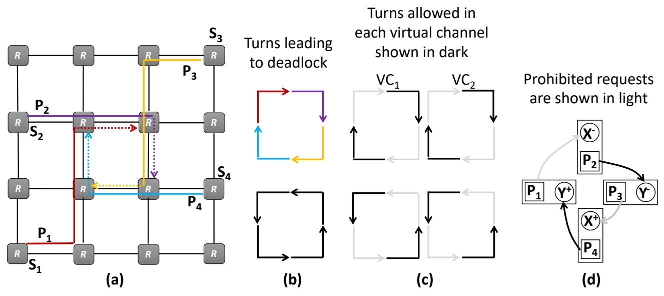

S3
P3
P2
S2
S4
P4
P1
S1
(a)
Turns leading to deadlock
(b)
Turns allowed in each virtual channel shown in dark
VC1
VC2
(c)
Prohibited requests are shown in light
X-
P2
P1
Y+
P3
Y-
X+
P4
(d)

Figure 4: In (a) is a group of secure packets in a deadlock. The cyclic dependency is shown in color in (b). The cycles can be avoided using a different VC for the last X turns of secure packets so that two turns of each cycle are assigned to each VC (c) and (d).

3.2 Deadlock Freedom

3.2.1 Deadlock problem: Applying XYX routing for secure packets in its raw form might create a cyclic dependency between NoC buffers. Figure 4.a shows an example case of four secure packets waiting to access their required buffers. The clockwise cycle shown in Figure 4.b depicts the cyclic dependency between packets of this example that never ends. The color code used for illustrating clockwise dependency in Figure 4.b helps to relate packets to turns. The same situation may happen for counter-clockwise turns, which is shown in black in Figure 4.b.

3.2.2 Deadlock prevention: we have devised two virtual channels, i.e., VC_1 and VC_2 with the following allocation policies to prevent deadlock. Both VCs are available for normal packets at the source; however, once a normal packet is assigned to a VC, the packet will use the same VC untill it reaches its destination. Normal packets are assigned to VCs based on the traffic condition at the source node. For the secure packets, they start with VC_1 and continue their path untill their second turn (if the secure packet is of 2-turn packets). For making the second turn, i.e., a turn from Y to X, a secure packet has to switch over VC_2. In other words, if a secure packet is managed to take XYX path, the packet switches over VC_2 when the packet returns to the direction X where the actual destination is located.

Turns allowed in each VC are shown in Figure 4.c. These are in fact the turns allowed in XY routing with no cyclic dependency in either VC_1 or VC_2; thus there is no deadlock within normal packets. From the deadlock point of view, secure packets also follow XY routing in each VC with the point that if a secure packet is currently using VC_2, it is not allowed to request/use VC_1 anymore. So there is no deadlock between secure packets as well. Notable to mention that switching over VC_2 by a secure packet is technically a cascaded routing that allows the packet to make the YX turn (at the time of switching over VC_2) with no deadlock [8]. The dependency analysis shown in 4.d shows the deadlock freedom for VC_1. The same analysis is applicable to VC_2 as well. The two points mentioned prove the deadlock-freedom of the proposed routing.

3.3 Proposed NoC Architecture

To protect our countermeasure from reverse-engineering, we have embedded most of the proposed security system within the in-house built NI. Figure 5 depicts the proposed architecture for the NI and router to support source-routing, encryption, and the key exchange. Neither the router's key ram table nor the secure source routing module contains sensitive information in the router.

3.3.1 Encryption. We use a lightweight encryption algorithm called Hummingbird-2 (HB-2) [10] in counter mode to encrypt both the destination address and body flits. The HB-2 has been proven to be secure against DPA attacks [10]. The block size is 16 bits, and the key size is 128 bits. An Initialization Vector (IV) is encrypted with a shared key K and then XORed with a plaintext block to obtain the ciphertext. In each encryption session, IV is incremented by one, and due to the diffusion and confusion properties, the block cipher output will have a completely different outcome. As shown in Figure 5, a single encryption module has been designed in such a way that it can encrypt the destination address and the body flits. Moreover, the router's own address is encrypted with the same key K and stored in the router key ram table. When a secure packet enters one of the input E, W, N, and S buffers, their encrypted destination will be compared with the contents of this table and processed accordingly. After body flits are decrypted, they are ejected to the NI through the local egress port.

3.3.2 Key Exchange. A node involved in a secure connection can be either an *initiator* or *recipient*. After an *initiator* sends a key exchange request to a *recipient*, both parties compute the shared key and then the new key will be written in their key ram tables in the NI. As depicted in Figure 5 (NI key ram table), *recipients* are shown with R and *initiators* are shown with an I. Since *recipient* is the ultimate destination, the router address will only be encrypted in this node and updated in the router's key ram table. The session counter defines how long a key should be kept in the key ram table (shown in the session counter column of the key ram table in the NI). We set the session expiration to be twice the worst-case delay of the network to ensure that encrypted packets have enough time to arrive at the destination. To securely exchange keys, we use the Diffie-Hellman key exchange mechanism [17].

3.3.3 Routing Algorithms. As previously explained, normal (non-secure) packets utilize the XY routing and the secure packet take a source-computed XYX path. However, to hide the destination address, secure packets may or may not need to take all three directions i.e., X, Y, and again X in their XYX path. The path for secure packets is computed in the path-computation unit at the NI for secure packets following Algorithm 1. This unit obtains the required encryption key from the key ram table and embedded the required fields of a secure packet (as shown in Figure 2) to enable anonymous routing of secure packets. We have added the necessary hardware to the NI and router logic to enable them generate and forward both secure and non-secure packets. When a router receives a secure packet, it prioritizes them over non-secure packets to minimize their time-span in the network. The prioritization impacts the virtual channel as well as the switch allocation stages on the router's pipeline by assigning resources to secure packets first.

4 EXPERIMENTAL EVALUATIONS

We have conducted two types of experiments to study the chip-wide and router-wide overheads of the proposed security system. In the

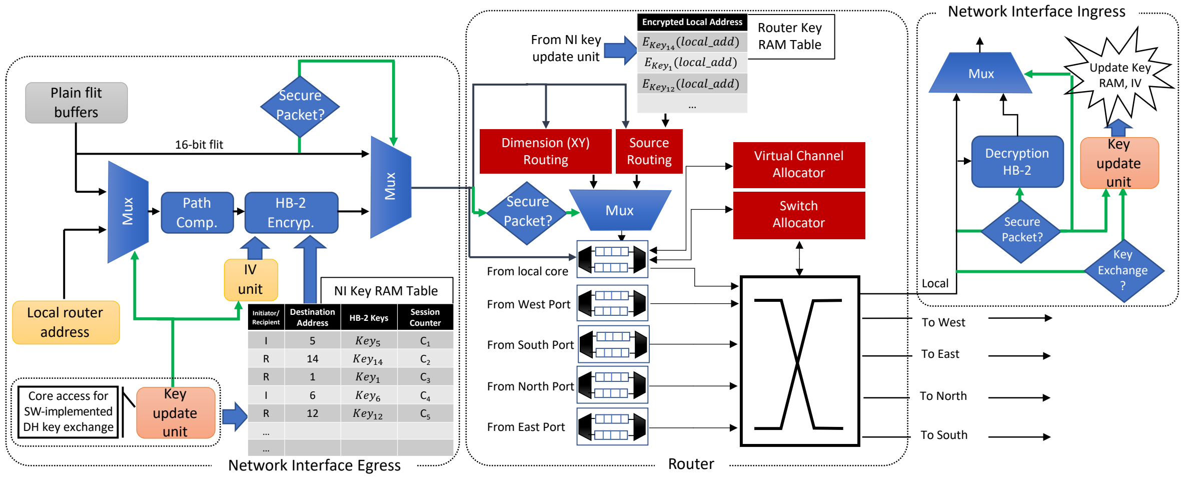


Figure 5: Proposed architectures for the network interface and the router to support anonymous routing.

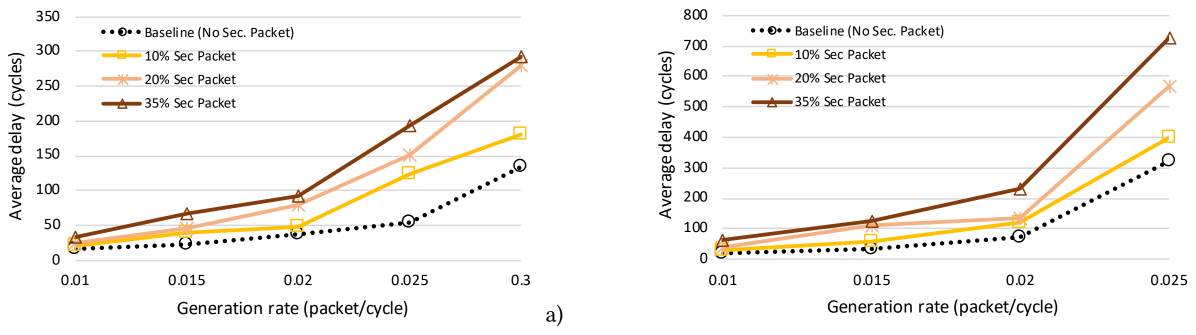


Figure 6: Average delay results under secure packets ratios of 10%, 20%, and 35% for a 6×6 (a) and an 8×8 (b) NoCs.

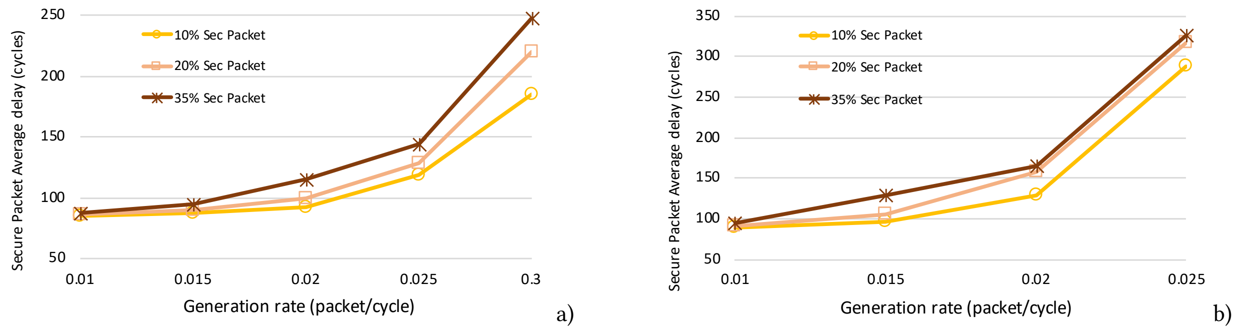


Figure 7: The network delay observed under secure packets ratios of 10%, 20%, and 35% for a 6×6 (a) and an 8×8 (b) networks.

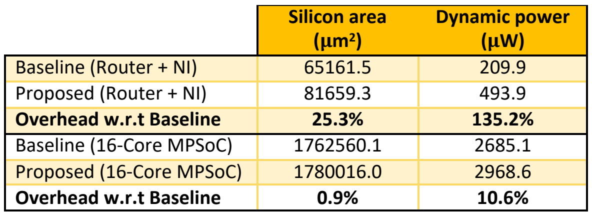
former (see Section 4.1), we have used Noxim [4] simulator to simulate the performance behavior of a NoC-based MPSoC equipped with our proposed method. We have implemented and synthesized the proposed router using Verilog to estimate its hardware and latency overheads with respect to the baseline router (Section 4.2).

4.1 Chip-Wide Evaluations

The simulator was modified to support two types of packets (secure and non-secure packets). Secure packets are routed based on a pre-computed path (see our explanations in Section 3) embedded in the packet's header. The non-secure packets are routed using the XY routing algorithm to allow the routers share virtual channels between normal and secure packets. Simulations are conducted for 50000 cycles with 10% warm-up time on 8×8 and 6×6 networks under various rates of uniform traffic and secure packets ratio. Packets consist of eight flits in all simulations. Based on our hardware implementation, the HB-2 encryption module requires 20 cycles to encrypt a 16-bit data block. The Network delay of secure packets is computed with respect to their length and the mentioned encryption delay. The key exchange delay is not accounted for in our simulations since key exchange happens once per session, and only a fraction of the total packets needs encryption.

Figures 6.a and 6.b, respectively, report the average network delay with respect to the baseline NoC. Here we define 10%, 20%

Table 1: Area and dynamic power consumption overheads of the proposed security system as compared to the baseline router and the baseline 16-core MPSoC.



	Silicon area (µm²)	Dynamic power (µW)
Baseline (Router + NI)	65161.5	209.9
Proposed (Router + NI)	81659.3	493.9
Overhead w.r.t Baseline	**25.3%**	**135.2%**
Baseline (16-Core MPSoC)	1762560.1	2685.1
Proposed (16-Core MPSoC)	1780016.0	2968.6
Overhead w.r.t Baseline	**0.9%**	**10.6%**

and 35% of packets as secure packets so that the network routes them using the proposed source routing method. As we see in the results, unless the packet generation rate is very high (above 0.02 packets/cycle) or the ratio of the secure packet is high (35%), the performance overhead of the proposed security system is affordable considering its remarkable security improvements. The same pattern can be seen for both sizes of the network.

For the next experiment, we measure the average network delay only for the secure packets. This might be of interest since secure packets use a different routing method. Figures 7.a and 7.b respectively report the average network delay observed by secure packets. Since higher priority is given to secure packets, the delay imposed on them is very low, i.e., the average network delay of a secure packet is kept close to that of the baseline network, while the delay overhead is mostly incurred to normal packets. This can be observed by comparing corresponding parts of Figures 6 and 7.

4.2 HDL Evaluations

We have implemented the proposed security system in Verilog HDL to verify its functionality and estimate the hardware overhead. We have then synthesized an MPSoC with 16 32-bit MIPS processors interconnected as a 4×4 mesh NoC [11]. The MPSoC was designed in Verilog and synthesized using 45*nm* technology in Synopsys Design Compiler tool for ASIC platforms. The baseline NoC has four virtual channels per each physical channel with a depth of eight buffers and a data width of 32 bits.

The synthesis results of the 16-core prototypical MPSoC are reported in Table 1. As the table shows, chip-wide silicon area of the proposed security system is negligible (less than 1%). The chip-wide power overhead is reported 10.6% assuming that the ratio of secure packets is 10%. Obviously increasing the ratio of secure packets will produce more signal activities at the secure modules and consumes more dynamic power. Further improvements in power consumption can be also obtained by applying routine dynamic power reduction methods like clock-gating. The area and power consumption overheads of the proposed network interface along with the router are 25.3% and 135.2% respectively. Based on our synthesis results for each separate module, the HB-2 module is the main contributor to the power overhead of the design. The HB-2 module accounts for 80% (out of 135.2%) of dynamic power consumption overhead.

5 CONCLUSIONS

This paper proposed a novel security solution for NoC-based MPSoCs to protect sensitive data from being attacked by cryptanalysis and timing attacks. The proposed security system consists of a novel source-routing method along with encryption modules. We could 100% hide the source/destination of secure packets so that no intermediate routers is able to discover this information. This not only enhances data privacy, but also prevents a variety of attacks since the attacker will no longer have ample information to compromise security. One significant achievement of the proposed system is its very low performance overhead that makes the proposed solution a feasible one for MPSoCs.

REFERENCES

[1] Dean Michael Ancajas, Koushik Chakraborty, and Sanghamitra Roy. 2014. Fort-NoCs: Mitigating the Threat of a Compromised NoC. In *Proceedings of the 51st Annual Design Automation Conference* (San Francisco, CA, USA) *(DAC '14)*. 1–6.

[2] Swarup Bhunia, Michael S Hsiao, Mainak Banga, and Seetharam Narasimhan. 2014. Hardware Trojan attacks: Threat analysis and countermeasures. *Proc. IEEE* 102, 8 (2014), 1229–1247.

[3] Travis Boraten and Avinash Karanth Kodi. 2016. Packet security with path sensitization for NoCs. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1136–1139.

[4] Vincenzo Catania, Andrea Mineo, Salvatore Monteleone, Maurizio Palesi, and Davide Patti. 2016. Improving the energy efficiency of wireless Network on Chip architectures through online selective buffers and receivers shutdown. In *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*. 668–673.

[5] Subodha Charles, Megan Logan, and Prabhat Mishra. 2020. Lightweight anonymous routing in noc based socs. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 334–337.

[6] Subodha Charles and Prabhat Mishra. 2020. Lightweight and Trust-Aware Routing in NoC-Based SoCs. In *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 160–167.

[7] Subodha Charles and Prabhat Mishra. 2020. Securing network-on-chip using incremental cryptography. In *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 168–175.

[8] Kun-Chih Chen, Shu-Yen Lin, Hui-Shun Hung, and An-Yeu Andy Wu. 2013. Topology-Aware Adaptive Routing for Nonstationary Irregular Mesh in Throttled 3D NoC Systems. *IEEE Transactions on Parallel and Distributed Systems* 24, 10 (2013), 2109–2120. https://doi.org/10.1109/TPDS.2012.291

[9] Luka Daoud and Nader Rafla. 2019. Analysis of black hole router attack in network-on-chip. In *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 69–72.

[10] Daniel Engels, Markku-Juhani O Saarinen, Peter Schweitzer, and Eric M Smith. 2011. The Hummingbird-2 lightweight authenticated encryption algorithm. In *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer, 19–31.

[11] Michel A. Kinsy, Michael Pellauer, and Srinivas Devadas. 2011. Heracles: Fully Synthesizable Parameterized MIPS-Based Multicore System. In *2011 21st International Conference on Field Programmable Logic and Applications*. 356–362.

[12] Jean-Jacques Lecler and Gilles Baillieu. 2011. Application driven network-on-chip architecture exploration & refinement for a complex SoC. *Design Automation for Embedded Systems* 15, 2 (2011), 133–158.

[13] Kyle Madden, Jim Harkin, Liam McDaid, and Chris Nugent. 2018. Adding security to networks-on-chip using neural networks. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1299–1306.

[14] Debdeep Mukhopadhyay and Rajat Subhra Chakraborty. 2014. *Hardware security: design, threats, and safeguards.* CRC Press.

[15] Cezar Reinbrecht, Altamiro Susin, Lilian Bossuet, and Johanna Sepúlveda. 2016. Gossip noc–avoiding timing side-channel attacks through traffic management. In *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 601–606.

[16] Cezar Reinbrecht, Altamiro Susin, Lilian Bossuet, Georg Sigl, and Johanna Sepúlveda. 2017. Timing attack on NoC-based systems: Prime+ Probe attack and NoC-based protection. *Microprocessors and Microsystems* 52 (2017), 556–565.

[17] Eric Rescorla et al. 1999. *Diffie-hellman key agreement method.* Technical Report. RFC 2631, June.

[18] Amin Sarihi, Ahmad Patooghy, Ahmed Khalid, Mahdi Hasanzadeh, Mostafa Said, and Abdel-Hameed A. Badawy. 2021. A Survey on the Security of Wired, Wireless, and 3D Network-on-Chips. *IEEE Access* 9 (2021), 107625–107656.

[19] Jie Zhang, Feng Yuan, and Qiang Xu. 2014. Detrust: Defeating hardware trust verification with stealthy implicitly-triggered hardware trojans. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 153–166.