SNAC: Mitigation of Snoop-Based Attacks with Multi-Tier Security in NoC Architectures

Siqin Liu
ls847719@ohio.edu
Ohio University
Athens, Ohio, USA

Saumya Chauhan
sc357320@ohio.edu
Ohio University
Athens, Ohio, USA

Avinash Karanth
ls847719@ohio.edu
Ohio University
Athens, Ohio, USA

ABSTRACT

Network-on-chips (NoCs) are crucial for multicore and manycore System-on-Chip (SoC) architectures. However, the integration of third-party Intellectual Property (IP) cores in SoCs has introduced hardware vulnerabilities. Snoop-based attacks exploit these vulnerabilities by inserting malicious Hardware Trojans into routers, allowing them to extract sensitive information as packets traverse the NoC. To address these security concerns, we propose SNAC: Mitigation of Snoop-based Attacks in NoCs. SNAC employs a three-tier architecture with increasing security levels, each with proportional power and latency overheads. The first tier introduces path randomization to prevent attackers from predicting packet routes. In the second tier, we encrypt source and destination information using lightweight backward XoR encryption. The third tier combines techniques from tiers one and two, extending obfuscation along with path randomization. SNAC was evaluated using synthetic and real-world benchmarks. Our results show that SNAC incurs dynamic power overheads of 4.2%, 3.9%, and 6.1% for Tiers 1, 2, and 3 respectively, with area overheads of 6.2%, 4.2%, and 9.2%.

CCS CONCEPTS

• **Networks → Network on chip**; • **Security and privacy → Side-channel analysis and countermeasures**.

KEYWORDS

Onion routing, Eavesdropping Security, Hardware Trojans

ACM Reference Format:
Siqin Liu, Saumya Chauhan, and Avinash Karanth. 2024. SNAC: Mitigation of Snoop-Based Attacks with Multi-Tier Security in NoC Architectures . In *Great Lakes Symposium on VLSI 2024 (GLSVLSI '24), June 12–14, 2024, Clearwater, FL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3649476.3658769

1 INTRODUCTION

System-on-chip (SoC) integrates various components onto a single chip, including cores, caches, memory, and networks. As SoC complexity increases, ensuring validation, verification, and trustworthiness becomes crucial [10]. Network-on-Chip (NoC) IPs are widely used in mobile devices, automotive systems, and general-purpose processing, leading to a surge in their usage. However, reliance on third-party IPs poses security risks, including potential malicious implants such as hardware trojans or undocumented backdoors [6].

NoC designs are increasingly complex and often concealed to protect their novelty. This complexity can conceal security vulnerabilities within dormant functions, making it difficult to detect threats using validation tools. Lightweight security measures are essential to address security concerns across various NoC application domains.

Various countermeasures have been assessed to address security vulnerabilities in NoC-based IP designs [4, 7, 11, 16]. Common approaches against snoop-based attacks in NoCs include obfuscation [15], runtime validation checks [13], packet encryption [3], and authentication (Watermaking) [2]. Encryption safeguards secure information from leakage, while authentication detects tampering with packets, including header information. However, encryption techniques remain susceptible to snooping attacks that extract secret keys through side-channel analysis. Information obfuscation can obscure packet origins and targets, making attacks more challenging [8]. However, such approaches have not been explored extensively for NoC architectures.

In this paper, we propose a multi-tiered lightweight encryption architecture called, **SNAC: Mitigation of Snoop-Based Attacks in NoCs** to extend the security of NoC architectures. Inspired by onion routing, SNAC is designed to obfuscate packet-level information to prevent any node from snooping into the packet. To thwart path-based attacks, we propose randomization that avoids trojans on specific combination of routers. As there is significant power and area overhead to implement these mitigation techniques, we propose a three-tier security architecture for NoCs that provides a design trade-off between security and power/performance overheads. In the first tier, we implement path scrambling for a source-to-destination packet transfer. In the second tier, we implement encryption of the source and destination of a packet using a lightweight backward XOR encryption scheme for each packet transfer. In the third tier, we implement both path scrambling and encryption of the source and destination of a packet for highly security-critical applications.

The major contributions of this paper are as follows:

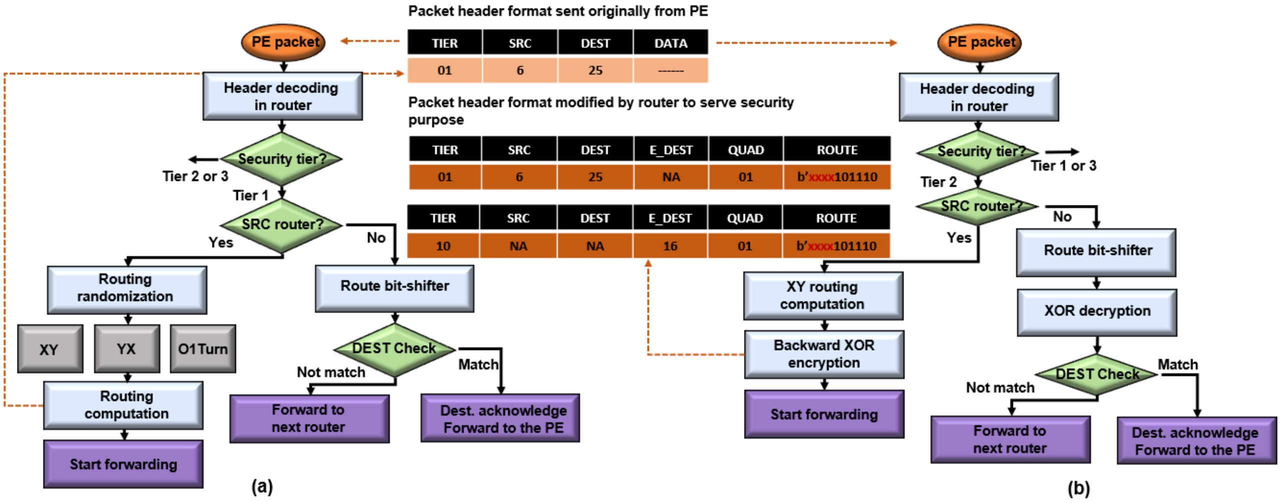
- **Lightweight encryption scheme:** We propose the use of source and destination encryption using a backward XoR encryption scheme for obfuscation. This technique utilizes the path taken by the packet to travel from the source to the destination as the key to encrypt the destination using XoR encryption. This process obviates the requirement of a global

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GLSVLSI '24, June 12–14, 2024, Clearwater, FL, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0605-9/24/06
https://doi.org/10.1145/3649476.3658769



Packet header format sent originally from PE

TIER	SRC	DEST	DATA
01	6	25	------

Packet header format modified by router to serve security purpose

TIER	SRC	DEST	E_DEST	QUAD	ROUTE
01	6	25	NA	01	b'xxxx101110

TIER	SRC	DEST	E_DEST	QUAD	ROUTE
10	NA	NA	16	01	b'xxxx101110

Figure 1: (Flowchart of Tier-based SNAC architecture. (a) SNAC-T1 implementation flowchart; (b) SNAC-T2 implementation flowchart. SNAC-T3 incorporates security measures from both tier-1 and tier-2. Tablets in the middle display the header format originally sent out from PE and after modification by security routers with different Tier implementations.

or local key in the network which reduces the susceptibilities to snooping attacks.

- **Tiered architecture to reduce overhead:** To improve performance in terms of latency, power, and area consumption, we propose different tiers that apply different levels of security. This reduces the unnecessary overhead of encrypting every packet in the network while ensuring that critical information is protected robustly.

2 SNAC MULTI-TIERED ARCHITECTURE

The three-tier security architecture is shown in Fig. 1. In the first tier, we implement, path scrambling for a source-to-destination packet transfer. The path scrambling algorithm scrambles the path taken by the packet between O1turn routing [14], X-Y routing, and Y-X routing. In the second tier, we implement encryption of the source and destination of a packet using a lightweight backward XOR encryption scheme for each packet transfer. In the third tier, we implement both path scrambling and encryption of the source and destination of a packet for highly security-critical applications.

To inter-operate with different tiers within the same NoC, the packet header contains all relevant information. Figure 1 shows the original packet header sent into the SNAC router in the first table, in which *TIER* (< 01, 10, 11>) specifies the level of security by users and *SRC* and *DEST* indicate the source and destination address. The SNAC router modifies the header information based on the selected tier by expanding more parameters, where *E_Dest* indicates the encrypted destination address using the backward XOR encryption algorithm, and *QUAD* indicates the routing quadrant (<00, 01, 10, 11>) associated with source and destination. The quadrant signifies the direction of packet traversal in the minimal rectangle of the 2D mesh from source to destination. *Route* represents the 1-bit hop-wise route established as the packet moves (<0 for x, 1 for y>).

Security Tier 1: Path Randomization: The SNAC router implements tier-1 workflow upon detecting the *TIER* value as b'01 (Fig. 1(a)). In this tier, packets are routed randomly using the 'XY', 'YX', or O1 Turn mechanisms. A randomizing function at the source router selects a routing algorithm, computes the route, and resets *QUAD* and *ROUTE*. This tier provides minimal security with lower power and no excess latency.

At each intermediate router, the header guides routing decisions to allocate appropriate virtual channels (VCs) and prevent deadlocks. Low-overhead routing involves shifting the least-significant-bit (LSB) of *ROUTE* by 1 bit for the next router. Each router checks *DEST* against its own node ID to verify destination reachability. Randomizing packet flow breaks traffic-route correlations, preventing information leakage. Spreading traffic across multiple routes makes it harder for attackers to gather side-channel information[9].

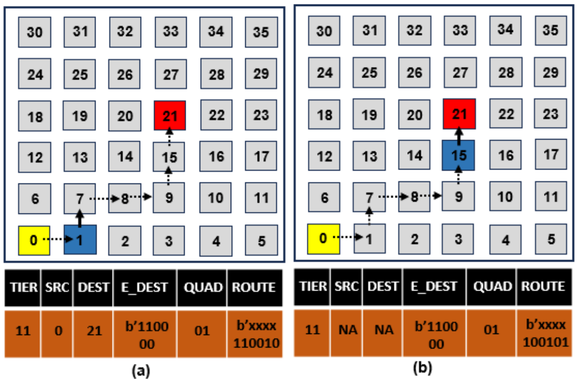
Security Tier 2: Encrypting Packet: We propose adaptive onion routing with backward XOR encryption of source and destination information. Unlike prior schemes, we obfuscate source and destination using the route as the key, ensuring no separate key transmission. To reduce routing bits, we encode quadrants (+x,+y), (+x,-y), (-x,+y), and (-x,-y) as 00, 01, 10, and 11 using the Q field in the packet header. By rotating route bits for each hop, we prevent intermediate node snooping.

To address identical route keys for same-source same-destination packets, we extend route bits with don't care bits to camouflage the route. For instance, the route key in Fig. 1 (b) is b'xxxx101110, with valid bits and random leading bits to prevent intermediate node inference. This modification hides source and destination information and generates an encrypted destination address in the header.

In SNAC-T2, backward XOR encryption involves XOR operation between original destination node address and *ROUTE* value in the header to obtain encrypted destination address. *ROUTE* bit sequence is shifted 1 bit for each intermediate router to enhance security. This dynamic key prevents HTs from stealing encryption keys through side-channel attacks[1, 5].

Security Tier 3: Encrypting Packet and Path Randomization: SNAC-T3 enhances security with combined onion routing and path randomization. This tier integrates obfuscation and path scrambling to defend against snooping attacks for critical applications, leveraging techniques from SNAC-T1 and SNAC-T2.

Path randomization in SNAC-T3 involves re-randomizing routing at each router using three routing algorithms. For example, in a 6x6 2D-mesh, packet transfer from node 0 to node 21 is illustrated in Fig. 2. The source router computes the packet's path using one of the routing algorithms, resulting in a route like 110010. To encrypt the destination based on the unique path, the route is circularly shifted bitwise, yielding 100101. This shifted route serves as the key for backward XOR encryption of the destination $110000 = 010101 \bigoplus 100101$. Source routers handle encryption, while intermediate routers decode encrypted information with their node IDs to verify destination.



TIER	SRC	DEST	E_DEST	QUAD	ROUTE
11	0	21	b'1100 00	01	b'xxxx 110010

TIER	SRC	DEST	E_DEST	QUAD	ROUTE
11	NA	NA	b'1100 00	01	b'xxxx 100101

Figure 2: An example of the SNAC packet encryption scheme applied to a 6 × 6 mesh network for packets sent from source 0 to destination 21.

3 PERFORMANCE EVALUATION

3.1 Simulation Environment

The evaluation of the proposed SNAC design was conducted on 16 CPUs in a 4×4 and 64 CPUs with 2-level cache in an 8×8 2D-mesh network respectively using cycle-accurate NoC simulator called NetSim. The NoC microarchitecture was implemented based on an open-sourced RTL [12] using the 5-stage router pipeline router design (route compute + virtual channel allocation + switch allocation, switch traversal, and link traversal) and 4 virtual channel buffers per input port.

We synthesize the design with the TSMC 45nm library, using a 1 GHz clock frequency operating at 1V in Synopsys Design Compiler. Our results of power and area overhead for the 4×4 2D-mesh baseline are shown in Table. 1 for SNAC security hierarchies. The most critical path lies in the input buffer with the longest propagation delay for all architectures, which is restricted by the 1 GHz operating frequency. To simulate real network traffic, benchmarks are collected from a subset of PARSEC and Splash-2 traces using the Sniper simulator for the following applications: PARSEC (Canneal, Dedup, and Ferret) and Splash-2 (Barnes, Cholesky, FFT). Due to space limitations, only six application trace results are presented here. The application traces which contain packet information, injection/ejection events, and clock time stamps, are executed in the NoC simulator to analyze our framework. We then compare the performance of the proposed SNAC architecture to the 4×4 and 8×8 mesh networks in terms of power consumption, end-to-end latency, and energy cost.

Table 1: Area and power breakdown of SNAC router microarchitecture at 45nm technology node compared with the baseline design.

Architectures	Major Components	Area (mm^2)	Power (mW)
Baseline	5-stage pipe.	26352.7	12.17
OCRA-T1	Additional 3:1 multiplexer 6-bit comparator Overheads	27996.6 6.2%	12.76 4.8%
OCRA-T2	Additional 6-bit comparator 10-bit comparator Overheads	27472.5 4.2%	12.59 3.5%
OCRA-T3	Additional 3:1 multiplexer 6/10-bit comparator Overheads	28776.5 9.2%	13.2 8.4%

3.2 Simulation Results

Overheads Analysis: We evaluate the overheads of the proposed SNAC architecture in terms of timing, chip area, and power as shown in Table 1. Specifically, the timing overhead is truncated as the longest propagation delay of the critical path. The chip area and static power consumption are evaluated using Synopsys Design Compiler software with 45nm technology. The baseline router microarchitecture is obtained to be of a total $26352.7um^2$ area and $1.27mW$ power while SNAC-T1, SNAC-T2, and SNAC-T3 incur 6.2%, 4.2%, and 9.2% power overheads, and 4.8%, 3.5%, and 8.4% area overheads, respectively.

NoC power analysis: We evaluate static and dynamic power consumption of SNAC with all three security tiers, that is, SNAC-T1, SNAC-T2, and SNAC-T3 in Fig. 3(a). We first model the static power of all components with Synopsys Design Compiler as discussed in Table. 1. Afterward, the captured power values are incorporated into the cycle-accurate NoC simulator to obtain accurate dynamic power simulation. As seen in Fig. 3(a), SNAC-T1 averagely consumes 4.2%, SNAC-T2 averagely consumes 3.9%, and SNAC-T3 averagely consumes 6.1% more power when compared to the baseline 4×4 2D-mesh router architecture. The majority of the excess power is attributed to the routing computation stage and the increased input buffer to accommodate the security control specified in the package header. SNAC-T2 incurs less power overhead than SNAC-T1 because the XOR encryption in SNAC-T2 is completed in the source router. Along the package traversal, all the intermediate router only takes a few comparators to do the encrypted destination checking. It is also important to note that for the real traffic simulation, SNAC-T1 and SNAC-T3 demonstrate significant dynamic power reduction than that in the RTL simulation phase as shown in Table. 1. This benefit originates in the proposed path randomization that amortizes the cost among all routers in the network.

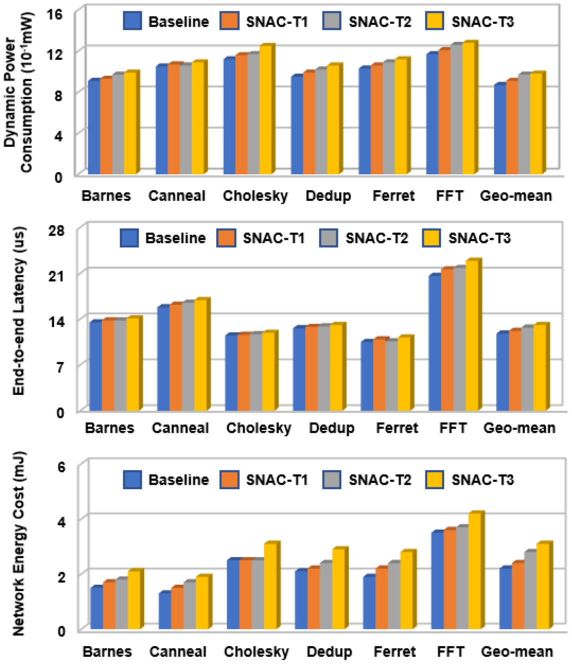


Figure 3: a)Power, (b)latency, and (c)energy performance of SNAC routers with different security hierarchies.

Execution Latency: The timing overheads of SNAC security tiers is introduced by the multiple routing algorithms and src/dest encryption. The execution time with the SNAC architecture is measured as the average end-to-end packet latency of real traffic on the 4×4 mesh network as shown in Fig. 3(b). The graph shows an overall end-to-end packet latency increase of 1.7% for SNAC-T1, 1.9% for SNAC-T2, and 2.4% for SNAC-T3 as compared to that for the baseline router. This increase in overhead is marginal for SNAC-T1 and SNAC-T2 and is acceptable for providing adequate security measures to the security criticality of those tiers. The higher increase in the Tier 3 execution time is due to the multiple layers of path randomization and encryption/decryption being applied in SNAC-T3. We expect that SNAC-T3 will be applied only on a few select applications and therefore, most of the operation will be in either SNAC-T1 or SNAC-T2.

Energy Consumption: We define energy computation as: $Energy = (P_{static} + P_{dynamic} \times T_{exec})$. P_{static} and $P_{dynamic}$ are static and dynamic power consumption, respectively. T_{exec} is the execution time of each benchmark application. Fig. 3c shows the energy cost of all applications on SNAC hierarchies and the baseline. It shows that SNAC-T1 incurs an average 6.3% energy cost increase, with 6.5% and 7.1% increases for SNAC-T2 and SNAC-T3, respectively.

4 CONCLUSIONS

In this paper, we proposed to secure NoCs against snooping and eavesdropping attacks using onion routing and path randomization in a tier-based secure architecture framework to reduce energy and power consumption. To thwart path-based attacks, we propose randomization that avoids trojans on specific combinations of routers. With significant power and area overhead of these mitigation techniques, we propose a threetier security architecture for NoCs that provides a design trade-off between security and power/performance overheads. In our analysis, we show that SNAC provides security against snooping eavesdropping attacks, and shows advantages against state-of-the-art security mechanisms in terms of execution time, latency, and energy consumption.

ACKNOWLEDGMENTS

This research was partially supported by NSF grants CCF-1703013, CCF-1901192, CCF-1936794, CCF-2324645, and CCF-2311544. We sincerely thank the anonymous reviewers for their excellent feedback.

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*. 1–6.

[2] Subodha Charles, Vincent Bindschaedler, and Prabhat Mishra. 2022. Digital watermarking for detecting malicious intellectual property cores in noc architectures. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 30, 7 (2022), 952–965.

[3] 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.

[4] Subodha Charles, Yangdi Lyu, and Prabhat Mishra. 2020. Real-time detection and localization of distributed DoS attacks in NoC-based SoCs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 12 (2020), 4510–4523.

[5] 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.

[6] Subodha Charles and Prabhat Mishra. 2021. A survey of network-on-chip security attacks and countermeasures. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–36.

[7] Abhijitt Dhavlle, M Meraj Ahmed, Naseef Mansoor, Kanad Basu, Amlan Ganguly, and Sai Manoj PD. 2023. Defense against On-Chip Trojans Enabling Traffic Analysis Attacks based on Machine Learning and Data Augmentation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2023).

[8] David Goldschlag, Michael Reed, and Paul Syverson. 1999. Onion routing. *Commun. ACM* 42, 2 (1999), 39–41.

[9] Leandro Soares Indrusiak, James Harbin, and Martha Johanna Sepulveda. 2017. Side-channel attack resilience through route randomisation in secure real-time networks-on-chip. In *2017 12th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*. IEEE, 1–8.

[10] Prabhat Mishra and Rajat Subhra Chakraborty. 2018. Tutorial T2B: Hardware Intellectual Property (IP) Security and Trust: Challenges and Solutions. In *2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID)*. IEEE, xxxix–xl.

[11] Atul Prasad Deb Nath, Srivalli Boddupalli, Swarup Bhunia, and Sandip Ray. 2020. Resilient system-on-chip designs with NoC fabrics. *IEEE Transactions on Information Forensics and Security* 15 (2020), 2808–2823.

[12] Michael K Papamichael and James C Hoe. 2015. The CONNECT network-on-chip generator. *Computer* 48, 12 (2015), 72–79.

[13] Ritesh Parikh and Valeria Bertacco. 2011. Formally enhanced runtime verification to ensure noc functional correctness. In *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*. 410–419.

[14] Daeho Seo, Akif Ali, Won-Taek Lim, and Nauman Rafique. 2005. Near-optimal worst-case throughput routing for two-dimensional mesh networks. In *32nd International Symposium on Computer Architecture (ISCA'05)*. IEEE, 432–443.

[15] Johanna Sepúlveda, Andreas Zankl, Daniel Flórez, and Georg Sigl. 2017. Towards protected MPSoC communication for information protection against a malicious NoC. *Procedia computer science* 108 (2017), 1103–1112.

[16] Ahmed Shalaby, Yaswanth Tavva, Trevor E Carlson, and Li-Shiuan Peh. 2021. Sentry-NoC: A statically-scheduled NoC for secure SoCs. In *Proceedings of the 15th IEEE/ACM International Symposium on Networks-on-Chip*. 67–74.