2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

NoC Security through Encryption: Mitigating Threats from Compromised Networks

Hengtai Zhao[0009-0008-3053-1685] and Guozhi Song

Tiangong University, Xiqing Tianjin 300380, China
songguozhi@tiangong.edu.cn

Abstract. Network on chip (NoC) is a communication architecture that uses on-chip communication protocols and infrastructure to connect various components within a system on chip (SoC). NoC has replaced the traditional bus-based communication architecture with a network-based approach. It typically consists of routers and links that connect processing cores, memory blocks, and other IP blocks within the chip. The high scalability of NoC allows for the integration of many processing components or IP cores on a single chip. The increase in SoC complexity has led to an increase in IP utilization from third-party suppliers. Attack NoC by implanting hardware trojans (HT) into the IP from an unreliable third-party IP provider. Since it is directly related to various aspects of the chip, it will become a prime target for data leakage and other security attacks. To enhance the anonymity of critical security information in the NoC, we considered packet encryption and routing methods that explore path diversity. We designed and implemented a new NoC architecture and proposed the use of the Ascon encryption algorithm and secure anonymous routing for safeguarding secure packets. Secondly, we also conducted experimental evaluations of the improved NoC architecture using the gem5 simulation tool. Through these evaluations, we verified the performance of the architecture under different traffic patterns. The results demonstrate that while ensuring low network latency, it can effectively prevent packets from being maliciously redirected and reduce the average hop count increased by hardware Trojan attacks.

Keywords: Network-on-Chip, eavesdropping, anonymous routing, Ascon.

1 Introduction

Over the past few decades, the semiconductor industry has made significant strides in the design and fabrication of integrated circuits (ICs). However, the considerable expense associated with constructing and maintaining manufacturing facilities, or foundries, has prompted SoC design firms to seek external production resources. The outsourcing of integrated circuit fabrication presents a formidable threat to critical infrastructure, as malevolent actors may exploit these processes by activating hardware trojans to circumvent security protocols. The malicious modifications introduced by these untrustworthy manufacturing sites in their designs can leak any confidential information in the security system to opponents [1].

NoC is extensively employed in the architecture of multi-core on-chip systems to fulfil their communication requisites. NoC has attracted significant attention from both aggressors and defenders alike. The increasing usage of NoC, coupled with its distributed nature on chips, makes it a potential focus of security attacks. By virtue of its central position within SoC and its interconnections with various components, NoC can serve as an effective medium for implementing security measures to safeguard SoC against potential threats [2].

The necessity of routers in NoC to efficiently access source/destination addresses in packet headers restricts the application of data encryption methods solely to the data payload. Indeed, encrypting header information may adversely impact the performance of Multi-Processor System-on-Chip (MPSoC), as each router would be required to decrypt, process, and re-encrypt packet headers. Consequently, various security breaches exploiting the source/destination of sensitive data packets may still facilitate the theft of confidential data, exacerbating security concerns. This underscores the imperative of integrating security considerations into the design of SoC communication architectures. Eavesdropping attacks, also known as snooping or sniffing, entail attackers passively monitoring on-chip communication to pilfer sensitive information clandestinely over an extended period without detection.

In this paper, we propose and evaluate two encryption-based network-on-chip (NoC) architectures to mitigate attacks that could cause information leakage, such as eavesdropping, and to enhance the security of NoCs. The main contributions of this paper can be summarized as follows.

(1) We provide an illustrative example of a threat model derived from a 4×4 NoC design and discuss the impacts of this model.
(2) We modify the NI/router structure in NoC to support encryption algorithms and secure routing algorithms.
(3) We use the Ascon encryption algorithm to encrypt the information carried by key data packets in NoC.
(4) Performance testing and security evaluation of the NoC based on the Ascon encryption algorithm and anonymous routing were conducted using the gem5 simulator.

The remainder of this paper is structured as follows. Section II provides an overview of the current background and related work in this field. Section III introduces the attack model adopted in this study. Section IV elaborates on the proposed NoC framework. Section V describes the Ascon and secure routing algorithms. Section VI presents our experimental results. Finally, Section VII provides a comprehensive summary of the paper.

2 Background and related work

In this section, we will introduce the unique security challenges faced by NoC based SoCs and discuss previous work on lightweight encryption and anonymous routing.

2.1 Background

In the field of computer networks and other related areas, the general issue of securing interconnected systems has been well-researched. However, implementing security features incurs area, power, and performance overhead. While complex security countermeasures can be afforded in computer networks, the resource-constrained nature of embedded and IoT (Internet of Things) devices presents additional, unique challenges [3]. These challenges include: (1) the complexity of SoC design makes exhaustive security verification an unattainable task, as most IPs are black boxes provided by vendors that do not disclose design details to maintain a competitive edge in the market; (2) in the design of NoC-based SoCs, electrical communication is widely used, and emerging NoCs also support chip-level photonics (optical NoCs) and wireless communication (wireless NoCs). Therefore, ensuring secure communication in NoCs must not only address security issues over wires but also consider new challenges posed by data transmission through photonic waveguides and wireless channels; and (3) when enabling communication between IPs, NoCs must meet a variety of requirements, including security, privacy, energy efficiency, domain-specific needs, and real-time constraints, making it difficult to balance conflicting demands such as security and energy efficiency.

2.2 Related work

In recent years, various measures have been introduced to safeguard confidential information within NoC architectures and ensure secure communication in SoC. These measures encompass the adoption of lightweight, confidential, and anonymous routing[4,5], the implementation of a blend of symmetric and asymmetric encryption techniques, the proposal of hybrid encryption framework[6], the utilization of tunnel-based network interfaces alongside AES encryption[7], the deployment of extended DyXY (E-DyXY) routing[8], authentication-based encryption communication protocols[9], and the development of NoC data encryption framework based on optical encryption devices (LED) algorithm[10]. Although these methods can protect security and prevent confidential information leakage, packets still face more complex attacks. Complex attacks that combine multiple erroneous behaviors can lead to greater persecution of NoC data and a higher probability of hitting, such as encryption analysis attacks.

Charles et al. proposed a lightweight encryption and anonymous routing protocol in [4] for IP core communication in NoC, using a novel secret sharing mechanism to eliminate the main overhead associated with traditional confidential and anonymous routing protocols. Each hop changes the packet, and only the complete packet is built at the destination. Compared with traditional anonymous routing methods such as onion routing, it achieves excellent performance, but modifying the packet operation at each hop still brings significant performance impact.

In [6], a hybrid encryption framework is proposed by combining the advantages of symmetric and asymmetric encryption to achieve a balance between security and efficient data transmission in NoC. The session key is generated using the Elliptic Curve Diffie Hellman (ECDH) algorithm, and an improved mCrypton encryption technique

is proposed. The session key is encrypted using Elliptic Curve Encryption (ECC). This method is effective in alleviating the limitations of traditional encryption algorithms. Sepúlveda et al. proposed in [7] to implement tunnel communication in secure network interfaces, encapsulating information in encrypted/hashed data packets to prevent data leakage to malicious NoCs. Adopting the anti-pattern of Advanced Encryption Standard (AES-CTR) to dynamically generate a new key KIV for each encryption/decryption. AES-CTR has high parallelism and achieves high throughput, but the cost is that it will bring higher area and power overhead to NoC.

Sankar et al. [8] considered a HT model that could leak packets received from expected sources to a serial application running on a separate core. To steal packet information, malicious applications engage in passive sniffing attacks with compromised routers. The E-DyXY routing algorithm is proposed to address this threat model, which introduces path diversity between nodes to reduce the source prediction accuracy of HT. This algorithm considers all anonymous features and reduces the impact of HT on source prediction by exploring path diversity. This method is designed for a special HT model, and the security and performance issues brought by its application to other threat models still need to be studied.

Haase et al. proposed a secure communication protocol based on RM in [9], which has a recovery mechanism and can establish secure end-to-end communication between NoC nodes. Use lightweight symmetric block cipher PRINCE for AE on network adapters and dedicated packet structures and use improved OkamotoTanaka protocol for key distribution. Ayachi et al. proposed a NoC data encryption framework based on the Light Encryption Device (LED) algorithm in [10]. The main advantage of the algorithm is that it reduces the implementation area and achieves high speed while reducing power consumption. By using a larger input/output FIFO, data waiting is avoided, and processing time is accelerated. This LED block cipher has the characteristics of a small implementation area, fast processing speed, and high security performance.

From the discussion, it is evident that there exist numerous solutions addressing information leakage in NoC. This article uses the lightweight encryption algorithm Ascon and secure routing, which are more secure and suitable for resource constrained environments, to avoid security information leakage in NoC. Additionally, we compare it with the traditional encryption algorithm AES to analyze the advantages and disadvantages of applying the Ascon algorithm in NoC.

3 Threat model

In this study, the HT threat model utilized is situated within the NoC router and is triggered during the routing calculation phase. Upon activation, the HT initiates malicious behaviour by altering the legitimate output port, thereby inducing a change in the routing path. Subsequently, as the data traverses through a compromised IP along the modified path, the malicious IP clandestinely captures and stores the security data encapsulated within the security data packet, consequently resulting in information leakage. Furthermore, this HT-induced behaviour may precipitate undesired service delays and instances of denial of service. NoC routers harboring such HT may erroneously route

critical data packets, thereby impairing the performance of delay-sensitive applications. Notably, this form of HT can be introduced into NoC IPs at any stage throughout the IC lifecycle, including specification, design, and manufacturing phases[11].

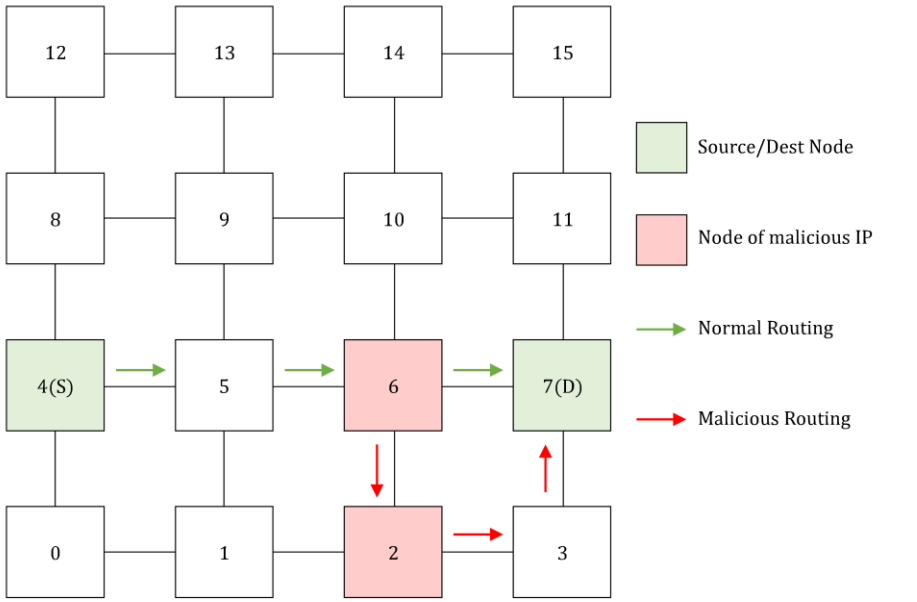

Fig. 1. An example of a threat model consisting of HT-infected routers and malicious tasks.

Figure 1 illustrates a conceptual model of a compromised NoC that leads to an information leakage threat. According to this example, an HT-infected node (node 6, highlighted in red in Figure 1) snoops on secure information exchanged between the source and destination nodes (S and D nodes, highlighted in green in Figure 1). The router in the malicious node may replicate the contents of the data packets and send them to another node running malware (node 2, highlighted in red in Figure 1) for further processing/analysis to uncover sensitive information. The impact of this threat can also be observed in terms of service latency. Due to routing errors caused by the malicious node, the captured information can only reach its destination after traversing additional hops, which results in delays and inevitably consumes more resources for packet transmission and reception.

To avoid easy detection and ensure that it does not affect all or most of the packets passing through it, the hardware Trojan only interferes with 10-15% of the packets in the infected router. From the perspective of the entire network, the proportion of affected packets accounts for only 0.5-1% of the total generated packets. This effectively reduces the risk of detection while still posing a threat to network performance and data security. Under synthetic traffic patterns, the number of packets generated in the network and the proportion of affected packets have been statistically analyzed and presented in Table 1.

Table 1. Table captions should be placed above the tables.

Traffic pattern	No of pkts	Through HT router	Total affected
Uniform random	82622	26876	4265
Bit complement	84260	42698	7273
Bit reverse	82482	25768	5447

The impact of this threat can also be observed in terms of network throughput. Due to routing errors caused by malicious nodes, the captured information can only reach the destination node by traversing additional hops, which leads to task delays and ultimately results in an inevitable reduction in throughput.

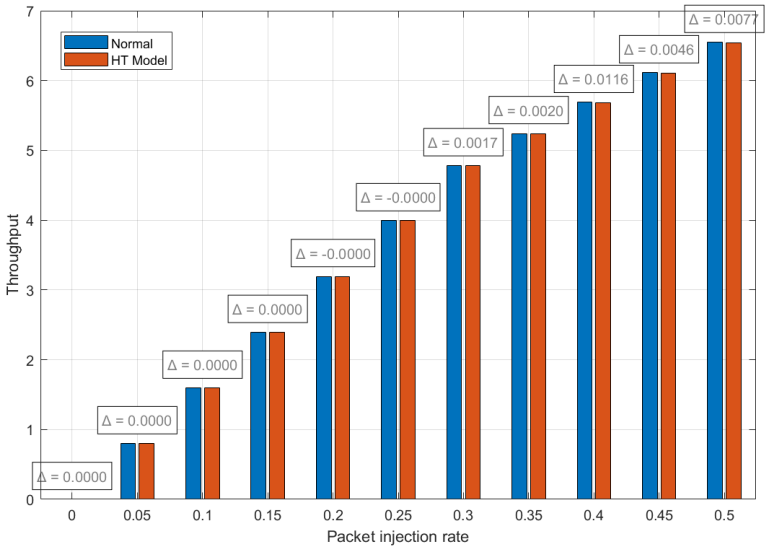

Fig. 2. Throughput of two types of NoC at different injection rates

Figure 2 illustrates the relationship between the hardware Trojan and NoC throughput. As shown in the figure, when the packet injection rate is less than 0.3 packets per cycle, the network throughput does not suffer significant losses. Only when the injection rate exceeds 0.3 packets per cycle does the system throughput decrease by no more than 1%. This is because the proposed Trojan can only attack under specific conditions, making it difficult to detect. If the throughput reduction were too large, it would become noticeable. Given these results, the threat model is stealthy, and such NoCs could pose a significant threat to SoCs in the near future without being detected.

4 NoC Framework

In order to achieve secure communication in NoC while avoiding excessive router latency, the core idea of this article revolves around implementing additional

modifications to the design of network interfaces (NIs). Figure 3 illustrates the NI and router architecture of NoC used in this article. In the ensuing discussion, we will provide specific explanations of the significant changes in NoC framework.

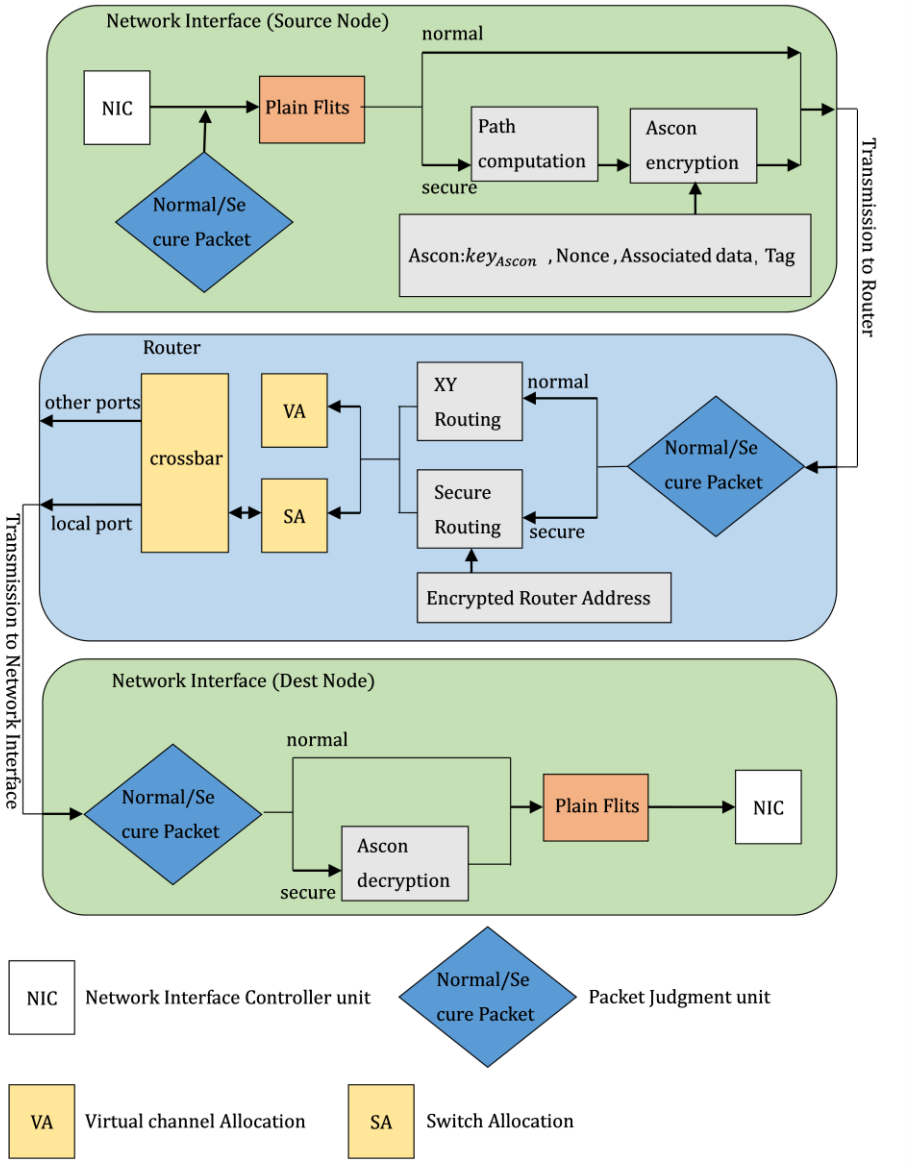

Fig. 3. NoC's network interface and router architecture

Network interface. The primary function of the network interface is to receive messages from the cache controller and convert them into flits. Each cache controller is connected

to a Network Interface Controller Unit (NIC), which utilizes message queues to handle incoming messages. Each message is converted into a unicast message and then de-composed into fixed-length flits based on the supported size of the outgoing link. In this article, we first add a judgment unit in the process of de-composing packets into flits to determine whether the packet is secure. If deemed secure, subsequent secure path calculations and Ascon data encryption operations will be performed. After the secure path calculation is completed, we will embed the calculated E_D, D_1, S_1, D_2, S_2, D_3 into the flit for subsequent secure routing. In the case of Ascon encryption, key_{Ascon}, nonce, associated data and Tag are required. After encryption, the key and other information also need to be embedded in flit for subsequent decryption. It is worth mentioning that the specific operational procedures of the encryption algorithms will be elaborated in detail in Section 5. Upon completing the operations, the NIC will arrange these flits for transmission based on the availability of the next hop buffer of the output link, and select the outgoing link based on the receiver, routing strategy, and message type. Additionally, we set up a packet type determination unit in the network interface that receives flits from the router. Upon identification of a secure packet, the received flits are decrypted, converted into a consistent message, and sent to the cache controller.

Router. Each network interface is connected to one or more local routers, which may be interconnected via external links. Once a flit is arranged, it will be transmitted through these external links, eventually reaching the router after a certain delay. Garnet routers execute the following operations: (1) Buffer Write (BW), (2) Routing Compute (RC), (3) Switch Allocation (SA), (4) VC Selection (VS), (5) Switch Traversal (ST), and (6) Link Traversal (LT). As shown in Figure 4, upon arrival, the flit is first placed in the input buffer queue. The buffered flit is then processed by the routing computation unit to determine its output port. Before this, we first identify its packet type, if it is a normal data packet, the XY routing algorithm is used, while a secure data packet employs the secure routing algorithm. The router contains multiple input buffer queues that compete for output links and the next hop's virtual channels (VCs). This is accomplished during the VC allocation and switch arbitration phases. Once the flit is selected for transmission, the crossbar switch directs the flit to the output link.

5 Lightweight encryption and secure routing algorithms

5.1 Ascon

The password suite Ascon [12] provides authentication encryption with associated data (AEAD) and hashing functionality. This suite consists of authenticated passwords Ascon-128 and Ascon-128a, which were selected as the primary choice for lightweight authentication encryption in the final combination of the CAESAR competition. The Ascon suite, especially the underlying 320-bit arrangement, was designed with these challenges of modern information infrastructure in mind. Ascon is considered highly secure and robust in practice, occupying a very small area in hardware, while providing good performance in both software and hardware. To provide these attributes, the main

components of Ascon are inspired by standardized and thoroughly analyzed primitives. This article is also the first attempt at the Ascon encryption algorithm in the field of NoC.

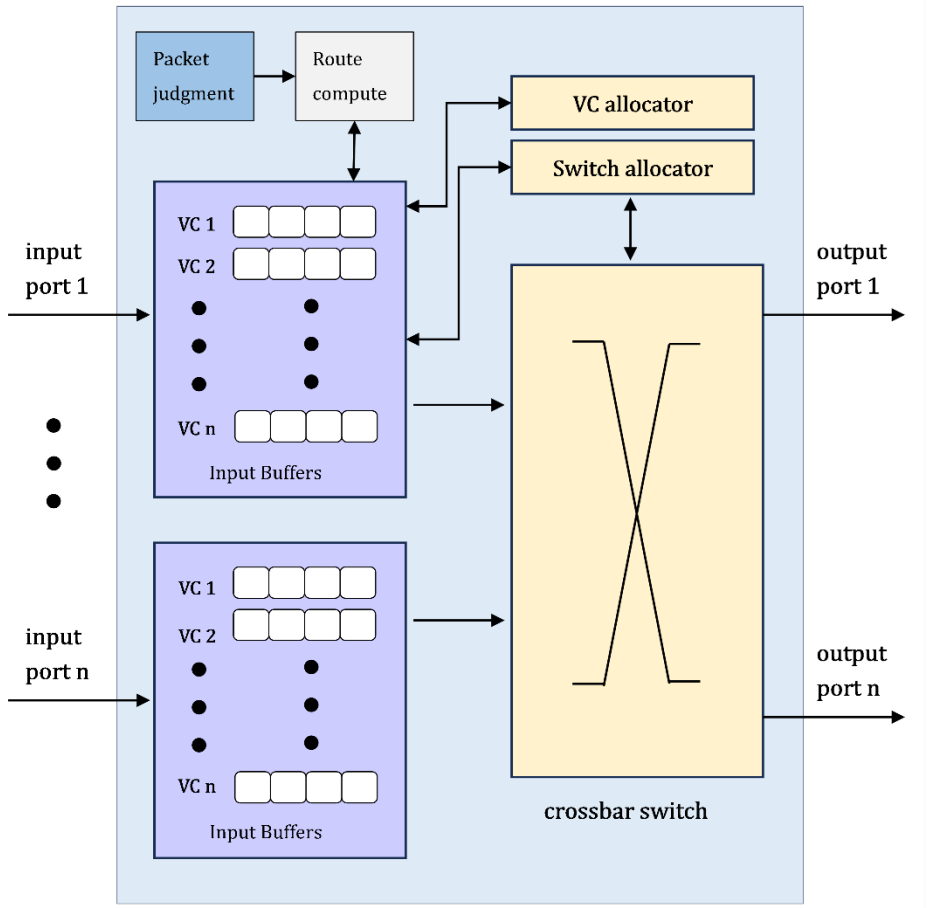

Fig. 4. Router Microarchitecture.

Passwords must withstand real-world threats. Therefore, the arrangement and the authenticated encryption mode of Ascon aim to provide robustness against certain implementation errors and attacks and promote effective protected implementations. For example, even if the attacker manages to recover the internal state during data processing (e.g. due to side channel attacks), this will not directly lead to the recovery of the key or the construction of forgery without significant computation. In addition to increasing the robustness of any implementation, this also allows for more effective prevention of side channel attacks. Due to the low S-box of Ascon, shielding implementation only incurs relatively small hardware and software overhead, making it feasible to implement protection on restricted devices.

For Ascon's authentication encryption design, family members are parameterized by key length K < 160-bit, rate (data block size) r, and internal integers a and b. Each design specifies the authentication encryption algorithm $\varepsilon_{k,r,a,b}$ and a decryption algorithm $D_{k,r,a,b}$. The authenticated encryption process $\varepsilon_{k,r,a,b}$ takes a k-bit secret key K, a 128-bit nonce (public information number) N, any length of associated data A, and any length of string plaintext P as inputs. The output it generates consists of authenticated ciphertext C, which is exactly the same length as plaintext P, plus a 128-bit authentication label T. It authenticates the relevant data and encrypted messages:

$$E_{k,r,a,b}(K, N, A, P) = (C, T) \tag{1}$$

The decryption authentication process $D_{k,r,a,b}$ takes the key K, random number N, associated data A, ciphertext C, and label T as inputs. If the label verification is correct, the plaintext P is output; If label verification fails, output error $\perp$:

$$D_{k,r,a,b}(K, N, A, C, T) \in \{P, \perp\} \tag{2}$$

Ascon's operation mode for authentication encryption is based on duplex modes such as MonkeyDuplex [13], but uses stronger keying initialization and keying termination functions. The encryption operation is shown in Figure 5.

5.2 Secure routing

The concept of secure routing is to hide the source/destination addresses in the packet. Instead, the approximate path and turning information required by the packet are embedded in the header of the secure packet. Secure routing randomly selects a routing scenario for each secure packet, such as XY, YX, or XYX. The shortest path for the selected strategy is then calculated and embedded into the header flit of the secure packet. The generation of path information does not reveal the source/destination addresses of the secure packet, as the header of the secure packet only contains two possible turning instructions that the packet might use. Secure routing provides the following three routing scenarios for secure packets: (1) an XY path toward the destination, with the header embedding one real and one misleading turn; (2) a YX path toward the destination, with the header embedding one real and one misleading turn; and (3) an XYX path, with the header embedding two actual turns. The goal of having multiple routing scenarios is to further obfuscate the attacker's ability to gather information about the packet's source and destination.

From the attacker's perspective, each local router performs one of the following four operations when processing a secure packet: (1) forwarding the packet in the received direction; (2) redirecting the packet from the X direction to the Y direction; (3) redirecting the packet from the Y direction to the X direction; or (4) ejecting the packet from the network upon reaching the destination.

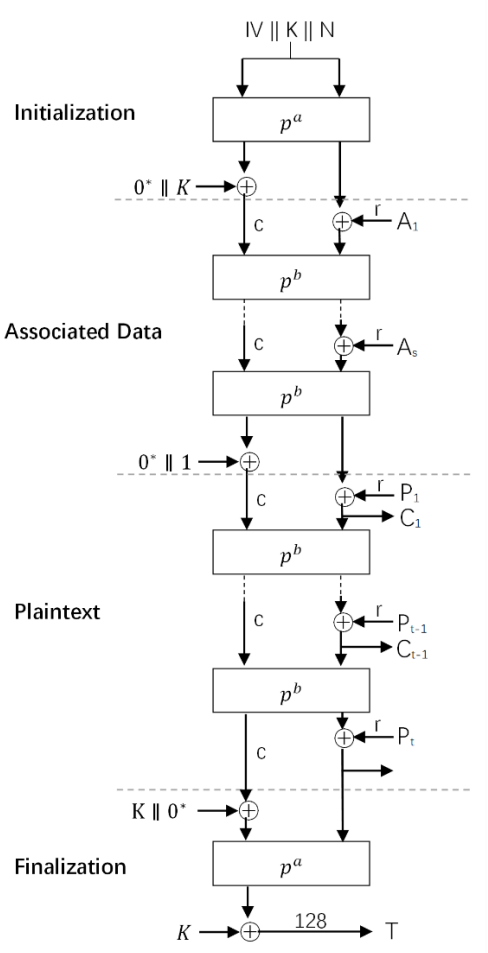

Fig. 5. Ascon encryption process.

The information embedded in the header of a secure packet is illustrated in Figure 6. The first field (Type) indicates the type of the packet: secure or normal (non-secure). The key_{Ascon} field carries the key used for Ascon encryption and decryption. As mentioned earlier, the Ascon algorithm also requires a Nonce and a Tag for encryption/decryption. The E_D field carries the encrypted version of the destination address for anonymous routing (which will be explained in detail later). The D_1 to D_3 fields carry the traversal directions of the packet, which can be one of the following four options: X^+, X^-, Y^+, and Y^-. The S_1 and S_2 fields represent the number of steps required for the packet to reach its first and second turns, respectively. The $E_k(body)$ field is the encrypted Body flit. Finally, the packet terminates with a Tail flit transmitted in plaintext.

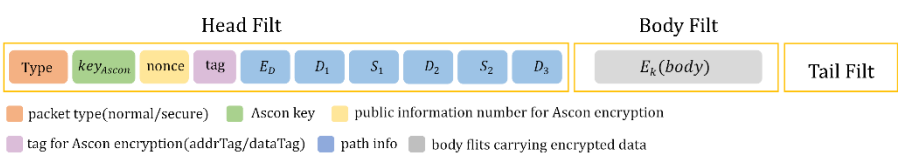

Fig. 6. Improved secure packet format based on Ascon.

Algorithm 1 describes the secure anonymous routing scheme. The basic idea is that a secure packet is ejected from the network only if the encrypted destination field matches the encrypted address of the current router. Otherwise, S_1 is checked, and if it is not zero, it is decremented by one, and the packet moves according to the command in D_1. When S_1 becomes zero, the condition for S_2 is rechecked and similarly decremented. Finally, if none of the above conditions are met, the packet is forwarded according to D_3 until it reaches the destination. For normal packets, the XY routing is used to allow them to share virtual channels with secure packets.

Algorithm 1 Secure Routing

Input: Strides S_1, S_2, Directions D_1, D_2, D_3;
Input: Encrypted Destination Addtress E_D;
Input: Encrypted Router Address E_R;
Output: Seleted Output Port $OutPort$;

if $E_D == E_R$ **then**
  $OutPort = Local$;
else
  if $S_1 \neq 0$ **then**
    $S_1 \leftarrow S_1 - 1$;
    $OutPort = D_1$;
  else if $S_2 \neq 0$ **then**
    $S_2 \leftarrow S_2 - 1$;
    $OutPort = D_2$;
  else
    $OutPort = D_3$;
  end if
end if
return

5.3 Secure path computation

The improved secure packet will adopt a dedicated secure routing mechanism. In this paper, we propose a secure path calculation method combined with the secure routing algorithm described in the previous section to achieve anonymous communication for secure packets. As shown in Algorithm 2, the algorithm first assigns the Ascon-

encrypted destination address to E_D. Next, the algorithm calculates the horizontal and vertical distances between the current node and the target node.

Algorithm 2 secure path computation

Input: Source and 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;
$E_D \leftarrow \text{Ascon}(P_d)$;
$\Delta X = P_d.X - P_s.X; and \Delta Y = P_d.Y - P_s.Y$;
Goto L1, L2, or L3 with a probability distribution of P_{L_1}, P_{L_2} and P_{L_3} s.t. $P_{L_1}+P_{L_2}+P_{L_3} = 1$;
L1:(an XY path)
$S_1 \leftarrow \text{abs}(\Delta X)$;
$S_2 \leftarrow \text{Rand}(n); and D_3 \leftarrow \text{Rand}(X^+, X^-)$;
if $\Delta X \geq 0$ **then**
    $D_1 \leftarrow X^-$;
else
    $D_1 \leftarrow X^+$;
end if
if $\Delta Y \geq 0$ **then**
    $D_2 \leftarrow Y^-$;
else
    $D_2 \leftarrow Y^+$;
end if
Return;
L2:(a YX path)
$S_1 \leftarrow \text{abs}(\Delta Y)$;
$S_2 \leftarrow \text{Rand}(n); and D_3 \leftarrow \text{Rand}(Y^+, Y^-)$;
if $\Delta Y \geq 0$ **then**
    $D_1 \leftarrow Y^-$;
else
    $D_1 \leftarrow Y^+$;
end if
if $\Delta X \geq 0$ **then**
    $D_2 \leftarrow X^-$;
else
    $D_2 \leftarrow X^+$;
end if
Return;
L3:(an XYX path)
$m \leftarrow \text{RandBetween}(0, \Delta X - 1); and S_1 \leftarrow \text{abs}(\Delta X) - m$;
$S_2 \leftarrow \text{abs}(\Delta Y)$;
if $\Delta X \geq 0$ **then**
    $D_1, D_3 \leftarrow X^-$;
else
    $D_1, D_3 \leftarrow X^+$;
end if
if $\Delta Y \geq 0$ **then**
    $D_2 \leftarrow Y^-$;
else
    $D_2 \leftarrow Y^+$;
end if
Return;

Subsequently, the algorithm randomly selects one of the three routing scenarios L_1, L_2, or L_3 to generate the path. The paths generated by L_1 and L_2 include one real turn and one misleading turn, while the path generated by L_3 contains two real turns.

6 Experimental results

6.1 Experimental setup

When configuring experimental settings, we utilized an open-source community-supported computer architecture simulator, which is a thoroughly explored architecture model [14]. We modeled a 4×4 mesh NoC using "HeteroGarnet" Internet model [15], combined with the gem5 full system simulator. HeteroGarnet improves upon the widely-popular Garnet 2.0 network model by enabling accurate simulation of emerging interconnect systems. It has provided a cycle-accurate microarchitectural implementation of a NoC router. The implementation of NoC model adopts X-Y routing with wormhole switching. we tested the model using the garnet standalone synthetic traffic injector which is built on top of the garnet standalone cache coherence protocol. Default HeteroGarnet implementation is used for packet formats. For each flit, 128-bit are allocated. The control packet is represented by 1 flit and the data packets are by 5 flits. In the control packet, 64-bit are allocated for the payload (address) while data packets have a payload of 512-bit. We have modified the Gem5 simulator to support two types of data packets (secure and non-secure packets). Secure packets are routed based on secure routing algorithms [16]. Non-secure packets are routed through the XY routing algorithm, allowing normal and secure packets to share virtual channels. On a 4×4 mesh network, simulations were conducted for 100,000 cycles using synthetic traffic patterns (uniform random, tornado, bit complement, bit reverse, and bit rotation) with varying injection rates, as well as the Splash-2 benchmark (FFT), under a 30% proportion of secure packets.

The specific experimental configuration parameters are shown in Table 2. An encryption module based on the Ascon algorithm was used to encrypt the data blocks of secure packets.

Table 2. Garnet experimental parameters.

Parameter	value
Network	Garnet
Network topology	4×4 Mesh
Virtual channels	4
Link width(bits)	128
Routing algorithm	XY and Secure Routing
Traffic	Synthetic traffic and Splash-2
Injection rate	0-0.5
Simulation cycles	100000
Packet type	Normal and 30% secure packets

6.2 Performance evaluation

The experimental results for the NoC with the secure architecture are as follows: First, the anonymous routing algorithm can bypass HT nodes with information-stealing capabilities by using alternative routing schemes. For secure packets encountering malicious modifications to the next routing direction, the encrypted source/destination addresses are ultimately used to ensure the successful transmission of secure packets. This mechanism relies on encryption and decryption operations at the source and destination nodes, so implementing such a security mechanism inevitably incurs some performance overhead.

To quantify this overhead, we measured the average network latency under a uniform random synthetic traffic pattern on a 4×4 network structure and compared it with the baseline NoC and the NoC with an eavesdropping model. In the experiment, we defined 30% of the packets as secure packets and transmitted them using the proposed routing method. The experimental results for the average packet latency are shown in Figure 7. From the results, unless the packet generation rate is very high (exceeding 0.25 packets per cycle), the performance overhead of the proposed secure system is acceptable, especially considering its significant security improvements.

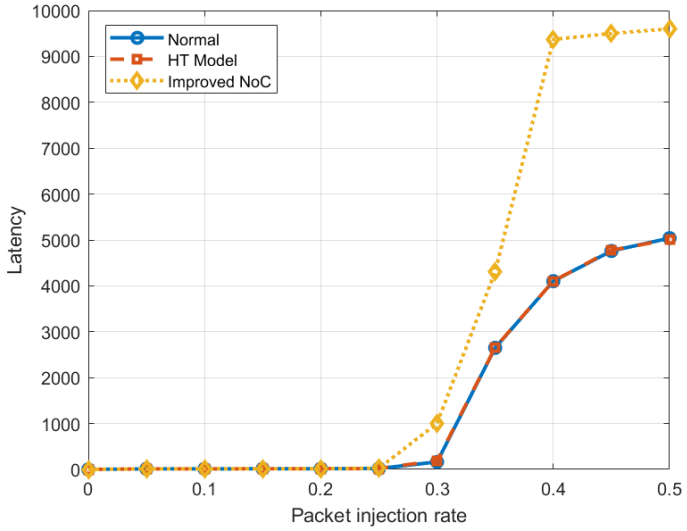

Fig. 7. Comparison of average delay of NoC under different packet injection rates.

Additionally, to test whether the improved secure NoC architecture can operate smoothly under different traffic conditions, we conducted experiments under five different synthetic traffic patterns and FFT application traffic. As shown in Figure 8, the secure NoC operates normally under five different synthetic traffic patterns within the injection rate range of 0.01 to 0.03, with acceptable latency. Table 3 presents the experimental data of the proposed secure NoC architecture when executing the FFT task from Splash-2. From the table, it can be observed that the architecture also functions properly under real application traffic, with normal average latency and hop count data.

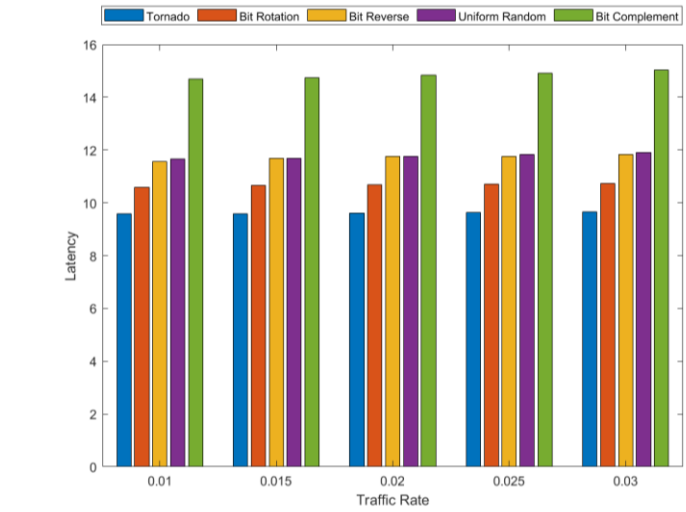

Fig. 8. latency results of secure packets in five different traffic modes.

latency	hops	computation time	Transpose Time
6027.953562	2.214369	513	75

6.3 Security analysis

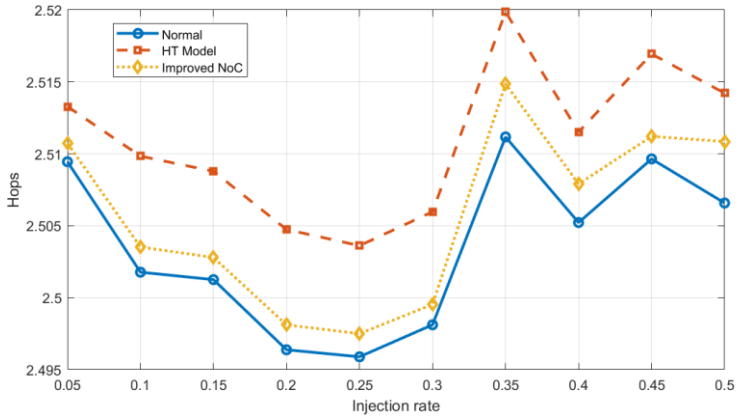

Fig. 9. Comparison of hops of NoC at different injection rates.

In this section, we will examine whether the improved secure NoC can successfully prevent the leakage of secure packets from the perspective of average hop count. In the HT eavesdropping model, the hardware Trojan modifies the transmission path of packets, redirecting them from the original optimal path to malicious nodes. This redirection

typically causes packets to traverse more routers before reaching their destination, thereby increasing the average hop count. In contrast, the proposed secure NoC architecture uses anonymous routing and encrypts the source/destination addresses of secure packets to prevent the Trojan from tampering with their paths, ensuring that secure packets can reach their destination normally.

As shown in Figure 9, the average hop count of the NoC with HT is consistently higher than that of the baseline NoC (with an average increase of 0.293%). This also proves that the eavesdropping model designed in this paper can be triggered normally. At the same time, by observing the data of the yellow dashed line, we can see that in the presence of HT, the improved NoC architecture successfully reduces the average hop count of the network (by 0.207%), indicating that the architecture can prevent HT from tampering with the transmission paths of secure packets.

7 Conclusion

In this paper, we investigate a secure NoC architecture based on the Ascon encryption algorithm and improved anonymous routing to counter hardware Trojan (HT) eavesdropping attacks. Hardware Trojans can tamper with packet paths, redirecting them to malicious nodes, leading to data leakage or performance degradation. To address this, we propose a secure packet format and path calculation method, which hides the source/destination addresses of packets using Ascon encryption and enhances security through randomized routing mechanisms (e.g., XY, YX, and XYX paths). In the Garnet simulation environment, we tested the performance under various traffic patterns (such as uniform random and tornado) on a 4×4 Mesh topology. Experimental results demonstrate that the architecture ensures low network latency while effectively preventing packets from being maliciously redirected and reducing the average hop count increased by the HT model (by approximately 0.207%). Additionally, Ascon, as a lightweight encryption algorithm, balances security and low resource consumption, making it suitable for NoC security protection. Overall, our proposed solution performs excellently in enhancing NoC security, particularly demonstrating strong defense capabilities against hardware Trojan attacks.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Jain, A., Zhou, Z., Guin, U. (2021, May). Survey of recent developments for hardware trojan detection. In 2021 ieee international symposium on circuits and systems (iscas) (pp. 1-5). IEEE.
2. Charles, S., Mishra, P. (2021). A survey of network-on-chip security attacks and countermeasures. ACM Computing Surveys (CSUR), 54(**5**), 1-36.

3. Bui, T., Rao, S. P., Antikainen, M., Bojan, V. M., Aura, T. (2018). {Man-in-the-Machine}: Exploiting {Ill-Secured} Communication Inside the Computer. In 27th USENIX security symposium (USENIX Security 18) (pp. 1511-1525).
4. Charles, S., Mishra, P. (2023). Lightweight Encryption and Anonymous Routing in NoC based SoCs. arXiv preprint arXiv:2302.06118..
5. Sarihi, A., Patooghy, A., Hasanzadeh, M., Abdelrehim, M., Badawy, A. H. A. (2021, October). Securing network-on-chips via novel anonymous routing. In Proceedings of the 15th IEEE/ACM International Symposium on Networks-on-Chip (pp. 29-34).
6. Thejaswini, P., Sahana, A. R., Shankar Singh, C. (2023, October). Strengthening NoC Security: Leveraging Hybrid Encryption for Data Packet Protection. In TENCON 2023-2023 IEEE Region 10 Conference (TENCON) (pp. 738-743). IEEE.
7. Sepúlveda, J., Zankl, A., Flórez, D., Sigl, G. (2017). Towards protected MPSoC communication for information protection against a malicious NoC. Procedia computer science, 108, 1103-1112.
8. Sankar, S., Jose, J., Gupta, R., Nandi, S. (2023, December). Enhancing Anonymity in NoC Communication to Counter Traffic Profiling by Hardware Trojans. In 2023 IEEE 16th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC) (pp. 560-567). IEEE.
9. Haase, J., Jaster, S., Franz, E., Göhringer, D. (2022, July). Secure communication protocol for network-on-chip with authenticated encryption and recovery mechanism. In 2022 IEEE 33rd International Conference on Application-specific Systems, Architectures and Processors (ASAP) (pp. 156-160). IEEE.
10. Ayachi, R., Mhaouch, A., Ben Abdelali, A. (2021). Lightweight Cryptography for Network-on-Chip Data Encryption. Security and Communication Networks, 2021(**1**), 9943713.
11. Rajan, M., Das, A., Jose, J., Mishra, P. (2021). Trojan aware network-on-chip routing. In Network-on-chip security and privacy (pp. 277-307). Cham: Springer International Publishing.
12. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M. (2021). Ascon v1. 2: Lightweight authenticated encryption and hashing. Journal of Cryptology, 34, 1-42.
13. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G. (2012). Permutation-based encryption, authentication and authenticated encryption. Directions in Authenticated Ciphers, 159-170..
14. Lowe-Power, J., Ahmad, A. M., Akram, A., Alian, M., Amslinger, R., Andreozzi, M., ... Zulian, É. F. (2020). The gem5 simulator: Version 20.0+. arXiv preprint arXiv:2007.03152..
15. Bharadwaj, S., Yin, J., Beckmann, B., Krishna, T. (2020, July). Kite: A family of heterogeneous interposer topologies enabled via accurate interconnect modeling. In 2020 57th ACM/IEEE Design Automation Conference (DAC) (pp. 1-6). IEEE..