Dynamic NoC Buffer Allocation for MPSoC Timing Side Channel Attack Protection

Johanna Sepúlveda[1], Daniel Flórez[2], Mathias Soeken[3], Jean-Philippe Diguet[2], Guy Gogniat[2]
[1]Institute for Security in Information Technology, Technical University of Munich, Germany
[2]Lab-STICC, South Brittany University, France
[3]Integrated Systems Laboratory, EPFL, Switzerland
johanna.sepulveda@tum.de

***Abstract*—Multi-Processors Systems-on-Chip (MPSoCs), as a key technology enabler of the new computation paradigm Internet-of-Things (IoT), are exposed to attacks. Malicious applications can be downloaded at runtime to the MPSoC, infect IPs and open doors to perform timing attacks. By monitoring the Network-on-Chip (NoC) traffic, an attacker is able to spy sensitive information such as secret keys. Previous works have shown that NoC routers can be used to avoid timing attacks. However, such approaches may lead to overall system performance degradation. In this paper we propose SER, a secure enhanced router architecture that dynamically configures the router memory space according to the communication and security properties of the traffic. Timing attacks are avoided by turning the attacker oblivious of the sensitive traffic. We evaluate the security, performance and cost of our approach. We show that our architecture is able to secure paths during runtime while adding only low cost and performance penalties to the MPSoC.**

Keywords—Network-on-Chip, timing, side channel attack

I. Introduction

Flexibility and high computation power have turned Multi-Processors Systems-on-Chip (MPSoCs) the preferred platform able to meet the requirements demanded by current semiconductor industry. MPSoCs integrate several processing and storage Intellectual Property (IP) elements which communicate through a Network-on-Chip (NoC). NoCs integrate a set of routers and links which communicate data between a pair of *source* IP (which injects the packet) and *destination* IP (which receives the packet). A network interface links an IP to a router. It implements the communication protocol by packing and unpacking the data and controlling the data injection and ejection from the NoC [1].

MPSoCs operating in the context of Internet-of-Things (IoT) are able to download programs that upgrade the firmware and execute several ever-changing applications during runtime. MPSoCs are used in a wide variety of applications. Particularly, automotive, industrial and health applications are considered critical due to the management of sensitive information and their effect on human lives [2]. In such applications, security has become a design requirement for MPSoCs. It is predicted that IoT will integrate 26 billion of interconnected devices by 2020 [2]. An attack in a single device may jeopardize all the devices interconnected within the IoT environment.

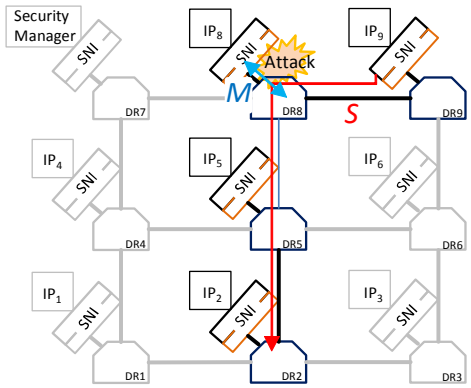

Fig. 1. Example of timing attack into an MPSoC.

MPSoCs are vulnerable and can be attacked. Downloaded malicious software can be used to infect the MPSoC IPs and extract sensitive information, modify the system behavior, or deny the MPSoC operation. Timing side channel attacks are one of the most effective and dangerous security incidents at the MPSoC. NoC traffic can be exploited by an attacker to spy sensitive information. Fig. 1 shows an example of a timing attack MPSoC with 9 IP cores interconnected through a 3x3 mesh-based NoC.

Sensitive information (*S*) is flowing through the path highlighted by a red arrow. The infected IP_8 can inject malicious traffic (*M*) and measure the throughput variation due traffic collision with the sensitive flow. The effect on the *M* throughput can be used to infer access pattern or timing behavior of the sensitive traffic. Such information can reveal a secret key.

Two of the most common techniques to protect the sensitive MPSoC traffic at NoCs are: i) encryption to encode the data [3], or ii) firewalls at the network interface to monitor and filter the NoC traffic [4]. Despite the high protection derived from these approaches, timing side channel attacks cannot be avoided. Recent works, propose the modification of the NoC router by integrating adaptive routing and arbitration algorithms, as a complement to firewall protection [5]. Despite offering protection against side channel attacks by avoiding attackers at the sensitive paths, these approaches degrade the system performance.

In this work we propose for the first time, a Secure Enhanced Router (SER) architecture that protects the MPSoC

978-1-4673-7835-2/16/$31.00 ©2016 IEEE

against timing attacks by dynamic virtual channel (VC) allocation. SER distributes the memory space of the router among input ports according to the current communication and security properties of the traffic. Dynamic allocation increases the timing side channel attacks resistance by means of performance isolation of the communication. As a result, the collision of packets at the router does not leak information to attackers. We show that our approach protects the sensitive data without degrading the system performance. We test our approach under different traffic scenarios and we compared SER with previous NoC routers that use random arbitration [5] and high priority [6] protection mechanisms. The contributions of this work are:

1. Implementation of timing side channel attacks protection by means of dynamic VC allocation.
2. Utilization of security and communication characteristics of the traffic to modify the router configuration.
3. High performance secure enhanced router.

This paper is divided into six sections. Section II presents the previous works that use NoC routers for MPSoC timing protection. Section III describes the threat model. Section IV presents the SER architecture and its functionality. Section V presents the experimental set up and the results. Section VI concludes the paper.

II. Related Work

NoC-based timing attacks have recently been addressed [5,6]. All these works propose the modification of the NoC router architecture in order to guarantee that the throughput of the attacker is independent of the sensitive traffic. In [6], Quality-of-Service mechanisms are integrated to the router for achieving the attacker performance isolation. It employs *high priority* arbitration, which assigns lower communication privileges to sensitive traffic. Despite the effectiveness of this approach, the system becomes vulnerable to Denial-of-Service (DoS) attacks, thereby preventing sensitive process communication. In order to overcome this difficulty, the authors of [5] propose the integration of *random router arbitration* and routing to schedule the packets randomly at the router for different output ports. Although this approach is effective in timing protection and allows the sensitive traffic communication, the utilization of the communication resources is still low. Our solution overcomes this drawback.

III. Threat Model

MPSoCs are able to support ever-changing applications that are stored inside the chip, stored in a near memory, or downloaded through external networks such as the Internet. To increase the system performance, downloaded programs are divided into smaller pieces of code and split on the shared MPSoC hardware resources [2]. Such a technique forces peer interaction among IP cores. Consequently, for MPSoCs that support critical information, sensitive data is exchanged among the different computation components through the shared NoC, thereby opening opportunities to attackers.

The attack considers sensitive (S) and malicious (M) processes which are executed simultaneously at the MPSoC. Fig. 1 shows *S* (e.g., a cryptographic function that employs a secret key), which is executed by IP_9. It is composed by a processor and an $L1_9$ cache memory. When the data requested by the processor at IP_9 is stored on $L1_9$, a *hit* takes place and the data is transmitted to the processor. Otherwise, a miss occurs and an access to memory $L2_2$ located at IP_2 must be performed [5]. As a result, a sensitive communication must be performed through the NoC from IP_9 to IP_2. In order to do that, the sensitive traffic must follow the deterministic path (sensitive path) determined by the well-known XY routing, which includes four routers (R9, R8, R5 and R2).

Simultaneously, M is being executed in the infected IP_8, which has been carefully selected by the attacker for being located in the sensitive path (R_8). There are several techniques that an attacker can use to tamper the software and infect an IP [7]. By using *malware* that performs *read* and *write* transactions in forbidden memory areas, an attacker may change the behavior of an IP (*victim* IP) and turn it into an *infected* IP. Moreover, buffer overflows and other similar techniques addressing software weaknesses can be exploited for such purpose [7]. An *infected* IP_8 may try to extract/infer data, modify the system behavior (by infecting other IPs) or deny the MPSoC service by means of malicious transactions. By injecting frequent and useless transactions, the infected IP_8 saturates the router R_8. Due to sharing R_8 by the malicious and the sensitive data, the throughput degradation of IP_8 can be used by the attacker to infer the access pattern of the sensitive flow (e.g, communication flow generated by a crypto-processor to a memory). This information can be used to break the cryptography or any other security mechanism [2].

Regarding the MPSoC communication structure, the network interface and NoC routers are considered secure, i.e., they are not tampered. The attacker has no access to the firewalls and cannot modify the router behavior. Once the packet is inserted into the NoC, it cannot be modified by malicious entities.

IV. Secure Enhanced Router

Our proposed SER architecture employs a single shared memory buffer to store the commuted data. At runtime, the number of VCs per input port is decided according to the communication and security requirements. By ensuring that the malicious (*M*) throughput is oblivious of sensitive (*S*) flows, SER avoids the timing attacks.

Fig. 2(a) shows the microarchitecture of a common NoC router. It commutes packets from one of the five input ports to one of the five output ports. Each port of a router is linked to neighbor router or to an IP (local port). Routers integrate four main components: i) *input buffers*, which store the data that request the communication through the router by one of the input ports. Such buffers are organized in VCs; ii) *routing algorithm*, to select the output port to be employed for redirecting the incoming data; iii) *arbitration logic*, which grants the utilization of the crossbar switch to one of the input buffers; and iv) *crossbar switch*, which links input to output ports of the router. Those components can be aligned in different pipeline ways. The example in Fig. 2(a) shows a four-

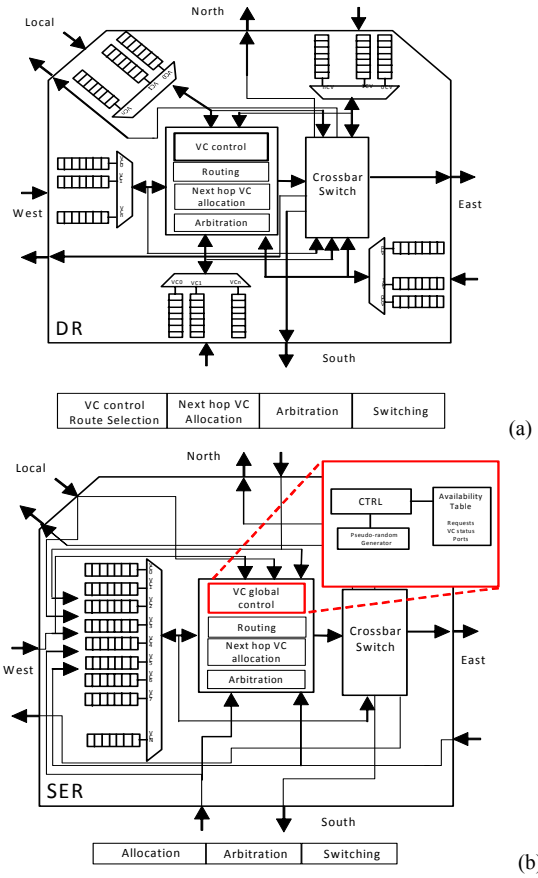

Fig. 2. Router microarchitectures (a) common router; (b) SER.

pipeline architecture composed by the stages: i) *VC control and router selection*, to store incoming data and quantify the output port; ii) *next hop allocation*, to reserve the VC at the neighboring router linked to the output port; iii) *arbitration*, to schedule the commutation of the data; and iv) *switching*, to commute the data by the crossbar.

Fig. 2(b) shows the SER microarchitecture. It differs from the common router architecture in the memory organization. The storage allocation is determined by the *global VC control.* It defines the number of VCs per input port and manages the storage process of the incoming data. *Global VC control* integrates two main components:

i) *pseudo-random number generator,* based on the implementation of [5]. It includes counters and binary logic. This configuration was selected for demonstration purposes. Larger and more complex generators can be employed (e.g., PUF-based generators based on ring oscillators [8]) to increase the randomness.

ii) *availability table*, composed by a matrix between the memory space and the input ports. Each column represents a VC and each row represents an input port. The matrix stores the status each VC (use, requested or idle).

The *global VC control* allocates a set of $b_i + k_i$ VCs for each input port i, where b_i is the minimum number of VCs per input port and $k_i = f \bmod a$ is randomly generated. The value f is the raffled number and a is the number of available VCs. VCs that

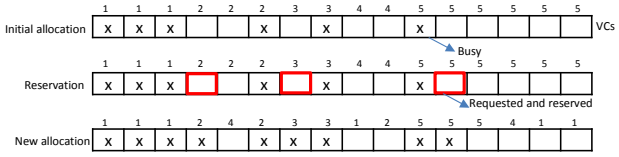

Fig. 3. Example of the Global VC control allocation.

are used or reserved are preserved. Free VCs are randomly distributed to the input ports. The sum of the number of VCs per input port is less or equal to the total amount of VCs at SER.

Fig. 3 shows an example of the allocation performed by the *global VC control*. The memory space of the example integrates 16 virtual channels distributed among the 5 ports of the router. At the initial allocation only 6 VC are used (cells with an X). SER receives the communication requests of the input ports 2, 3, and 5 and reserves VCs. By using the pseudo-random generator, the *global VC control* redistributes the free space, as shown in the last line of the figure.

SER is a three-stage pipeline router, as shown in Fig. 2(b). It adopts the *look-ahead* commutation that enhances the performance by performing the routing one hop in advance. The SER stages are described as follows:

1. *Allocation*: Includes three functionalities: i) allocation of the incoming packet in the memory space according to the *global VC control*; ii) route calculation, by reading the destination of the data, embodied at the header of the packet, the output port is defined; and iii) reservation of a VC in the next hop.
2. *Arbitration*: Defines which VC will be scheduled to the SER *crossbar switch*.
3. *Switching*: Redirects the packet from the VC to the output port defined at the first SER stage.

The SER commutation latency L_{SER} is given in (1). Where t_{AL}, t_{VC}, and t_{SW} are the times required to complete the allocation, arbitration, and switching stages, respectively. The t_{AL} is composed by the allocation time at the current hop t_{Avc} and the reservation time at the next hop t_{Rvc}, as given in (2). Note that the routing time is not considered, once it is executed simultaneously with the allocation of the incoming packet.

$$L_{SER} = t_{AL} + t_{VC} + t_{AR} + t_{SW} \quad (1)$$

$$t_{AL} = t_{Avc} + t_{Rvc} \quad (2)$$

V. Results

A. Experimental setup

SER has been modelled in SystemC-TLM and RTL-VHDL and has been integrated into the simulation environment SHOC [9]. SHOC is a modular cycle accurate simulation environment which supports a wide variety of instruction set architectures, traffic generators, and all the components required for MPSoC simulation. This environment includes libraries of MPSoC attacks and tools for power and area estimation. By integrating SER we were able

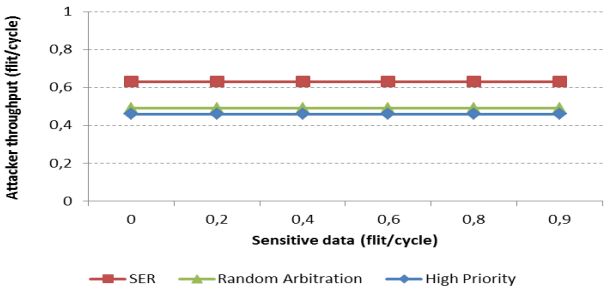

Fig. 4. Attacker traffic (*M*) throughput under different sensitive traffic (*S*) injection rates.

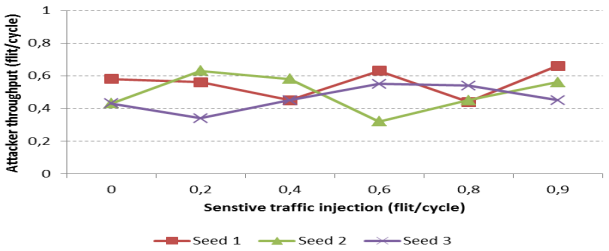

Fig. 5. SER under saturation: Attacker traffic (*M*) throughput under different sensitive traffic (*S*) injection rates.

to model an 81-cores MPSoC that is interconnected through a 9x9 mesh-based NoC. For comparison reasons, we have implemented the previous techniques for timing attacks avoidance described in Section II: *high priority* [6] and *random arbitration* [5].

B. Security and performance results

Security evaluation was performed for a single SER router. A good security countermeasure for avoiding timing attacks must guarantee the independence of the attacker traffic *M* (malicious) performance regarding the sensitive traffic *S* (sensitive). It is desirable that the security mechanism has the lowest impact over the system performance. Fig. 4 shows the throughput of an attacker *M* under different amounts of traffic *S* injection rates. It evidences that SER is able to protect against timing attacks by offering independence of *M* throughput regarding *S*. The security of the high priority and random arbitration mechanism was also tested and proven. However, SER achieves the higher throughput when compared to the previous solutions. This result is achieved by the efficient dynamic allocation of the communication resources, provided by SER.

Fig. 5 shows M throughput for different T_S rates under SER saturation, that is, when all the input ports are using the maximum bandwidth. Three different seeds for the pseudo-random generator of the *global VC control* are used. Results show that there is no correlation between *M* throughput and *S*.

Besides offering the timing attack protection, SER also is able to avoid DoS attacks. The existence of b_i guarantees that even under the increase of *M* injection rate, *S* is not prevented of being communicated. Fig. 6 shows the performance evaluation results of the SER integrated in the 9x9 NoC under uniform traffic. NoC traffic injection rate is modified by the variation of the percentage of long range-dependence traffic

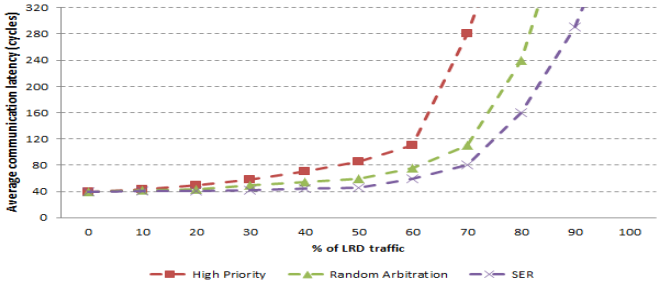

Fig. 6. Average communication latency for uniform traffic pattern.

TABLE I. Cost SoC Impact Due to Channel Protection

Mechanism	***Power***	***Area***
High priority	4%	2%
Random arbitration	9%	11%
SER	8%	9%

[5]. When this percentage is 0%, all the NoC traffic is Poisson distributed. Such traffic is characteristic of the NoC-based SoCs. Table I shows the power and area overhead of the different approaches when compared to a router without any protection. The results show that SER achieves the timing protection while introducing a low overhead.

VI. Conclusions

In this work we propose SER, a secure enhanced router able to protect the sensitive traffic from timing attacks. By dynamically allocating the number of VCs per input port, SER can guarantee the attacker is oblivious of the sensitive traffic. SER was compared with previous protection techniques and the results show that SER, besides guaranteeing the protection against timing and DoS attacks, achieves the best performance results. The better management of the SER resources and the look-ahead techniques contribute to achieve these results. Moreover, SER presents a low overhead. As future work we will further improve SER in order to reduce its cost and explore other NoC protection techniques against timing attacks.

References

[1] E. Carara et al., "Differentiated Communication Services for NoC-Based MPSoCs", In *IEEE Transactions on Computers*, vol. 63, pp. 595-608, 2014

[2] D. Evans. "The Internet of Things: How the Next Evolution of the Internet is Changing everything", *Cisco*, 2011.

[3] D. Florez et al., "Reconfigurable Security Architecture for Disrupted Protection Zones in NoC−Based MPSoCs", In Proc. *RECOSOC 2015*. pp. 1-8, 2015.

[4] D. Florez et al., "Elastic security zones for NoC-based 3D-MPSoCs", In Proc. *ICECS 2014*, pp. 506-509, 2014.

[5] J. Sepúlveda et al., "NoC-Based Protection for SoC Time-Driven Attacks", In *IEEE Embedded Systems Letters ESL*, vol.7, pp. 7-10, 2015.

[6] Y. Wang et al., "Efficient timing Channel Protection for On-Chip Networks", In Proc. *NOCS 2012,* pp. 142-151, 2012.

[7] J-P. Diguet et al., "NoC-centric security of reconfigurable SoC", In Proc. *NOCS 2007*, pp. 223 – 232, 2007.

[8] M. Hiller et al., "A New Model for Estimating Bit Error Probabilities of Ring-Oscillator PUFs", In Proc. *RECOSOC 2013,* pp. 1-8, 2013.

[9] J. Sepúlveda et al., "Implementation of QoSS (Quality-of-Security Service) for NoC-Based SoC Protection", *Transactions of Computational Science*. Ed. Springer, pp. 187-201, 2010.