Towards Risk Aware NoCs for Data Protection in MPSoCs

Johanna Sepulveda[1], Daniel Flórez[2], Ramon Fernandes[3], Cesar Marcon[3], Guy Gogniat[2], Georg Sigl[1,4]

[1]Institute for Security in Information Technology, Technical University of Munich, Germany
[2]Lab-STICC, South Brittany University, France
[3]FACIN - PUCRS – Av. Ipiranga 6681, 90619-900, Porto Alegre, Brazil
[4]Fraunhofer Institute for Applied and Integrated Security, Garching, Germany
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 currently exposed to attacks. Malicious applications can be downloaded at runtime to the MPSoC, infecting IP-blocks connected to a Network-on-Chip (NoC) and opening doors to perform Timing Side Channel Attacks (TSCA). By monitoring the NoC traffic, an attacker is able to infer the sensitive information, such as secret keys. Previous works have shown that NoC routing can be used to avoid attacks.**

In this paper we propose *GRaNoC*, a NoC architecture able to monitor and evaluate the risk of the communication paths inside the NoC. Sensitive traffic is exchanged to minimal low-risk paths defined at runtime. We propose five types of dead-lock free risk-aware routing algorithm and evaluate the security, performance and cost under several synthetic and SPLASH-2 benchmarks. We show that our architecture is able to guarantee secure paths during runtime while adding only low cost and performance penalties to the MPSoC.

Keywords—Security; Network-on-Chip; risk path; routing.

I. INTRODUCTION

Flexibility and high computation power have turned Multi-Processors System-on-Chip (MPSoCs) as the foreseen platform able to meet the requirements demanded by semiconductor industry. MPSoCs integrate several processing and storage Intellectual Property (IP) cores which communicate through a Network-on-Chip (NoC). By means of a set of routers and links, the NoC communicates packets between a pair of *source* IP (which injects the packet) and *destination* IP (which receives the packet). A network interface links an IP core to a router. It implements the communication protocol by packing and unpacking the data and controlling the data injection and ejection from the NoC. In order to increase the efficiency of the communication, two-level NoCs are employed. They integrate a data NoC and control NoC into commutation points (CP) for exchanging data and control packets of the MPSoC [1]. Fig. 1 presents an MPSoC with 9 IP cores interconnected through a two-level 3x3 mesh-based NoC.

The adoption of MPSoCs in the Internet-of-Things (IoT) context promises to be source of huge benefits. By interconnecting the MPSoCs through an external network, as Internet, MPSoCs are able to download programs for upgrading the firmware and executing several ever-changing applications at runtime. Each application is characterized by performance and security requirements, which must be met under tight area and power constraints [1]. In order to increase the performance of the MPSoC, applications are divided into smaller pieces of code, called tasks, and split on the shared MPSoC hardware resources. Such an approach forces the peer interaction among the IP cores. Consequently, for critical applications, sensitive data must be exchanged through the shared NoC, which increases the vulnerability of the system.

Software-based attacks can be used to extract sensitive information [2], to modify the system behavior [3] or to deny the MPSoC operation. Timing attacks are one of the most effective and dangerous security incidents at the MPSoC [4]. Shared NoCs can be exploited by an attacker in order to spy sensitive information. By using the attacker throughput variation due the traffic collision (competition for the same resources) with sensitive flows, an attacker can infer sensitive data, as shown in [5].

Previous works have shown that secure enhanced NoCs can be used to prevent and mitigate software attacks [6-8]. One of the most common techniques to implement security at NoCs is through firewalls embedded at the network interface, between the *source* IP and *destination* IP. They monitor and filter the NoC traffic according to a set of security rules. Firewalls are used to guarantee the access control [6,8] security service. Security mechanisms are controlled and configured through the Secure Manager (SM) core, a secure processor that compiles the security requirements into security rules able to be loaded into the firewalls of the system. Each time a security rule is violated, the packet is discarded and the firewall notifies the SM. Fig. 1 shows two attacks, one detected by the *source* IP firewall (A1) and the other at the *destination* IP firewall (A2). While A1 detection allows the attacker identification (source IP), A2 detection identifies a possible target of attack.

Despite the high protection derived from the firewall integration, sensitive traffic must be communicated through risky shared paths. As a complement of the firewall protection, in order to mitigate attacks, NoC

978-1-5090-2520-6/16/$31.00 ©2016 IEEE

routing can be used to secure the MPSoC. Routing defines the path that packets must follow inside the NoC. Only the works of [5,9] propose the use of adaptive routing to select disjoint paths inside the NoC in order to protect the system. However, this technique may lead to longer and probably higher risk sensitive paths. Consequently, it may degrade the performance and security of the system.

In order to overcome this drawback, in this work we propose for the first time a Global Risk aware NoC (*GRaNoC*), a secure-enhanced NoC able to protect sensitive data by using low-risk paths. The *risk* is defined in this work as the probability that a malicious process spies, denials the communication or corrupts the data in a NoC hop. The risk is measured by the amount of firewall notifications due the violation of security rules. The risk is associated to each hop of the NoC. Normally the *GRaNoC* uses the well-known XY distributed packet routing to define the path of the sensitive data. This routing technique is widely adopted at NoCs for area and power efficiency. When the risk of this path exceeds the allowed *risk threshold,* defined by the designer, a new route is set. The new sensitive path is implemented by means of source-based routing, where the source IP defines entirely the route that the packet must follow. Such information is embodied at the packet header.

In this work we propose five techniques for setting a low-risk path: deterministic, hop-based, weighted-2D and bounded. Each one presents a trade-off among the security, performance, area, and power. The contributions of this work are:

- Implementation of low-risk routing NoC based on firewall notifications.

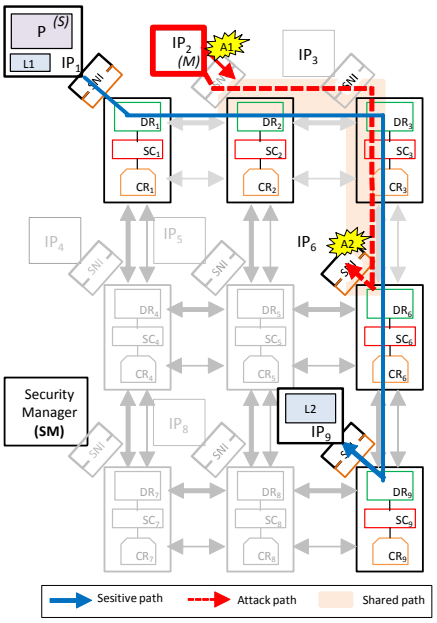

Fig. 1. Example of a two-level NoC-based MPSoC.

- Combination of packet and source routing techniques for implementing security.
- Evaluation of cost, performance, and security of five mechanisms for setting low risk paths.

This paper is divided into six Sections. Section II presents the previous works that use NoC routing for MPSoC protection. Section III describes the threat model and the risk evaluation. Section IV presents our architecture and the route configuration process. Section V reports the experimental set up and the results. Finally, conclusions are presented in Section VI.

II. Related Work

NoC-based protection architectures for MPSoCs have been shown to be an effective solution to address security of heterogeneous SoCs. Most of the NoC-based secure enhanced architectures are based on firewalls [6-8]. By comparing the transmitted data with the security rules of the system, a transaction is allowed or denied. These approaches store the security rights on the firewalls implemented as lookup tables. Firewalls are able to implement several security services [6-8]. According to the upgrading capabilities of security rules stored at the firewalls, they can be static [6] or dynamic [7,8].

Routing-based NoC protection was addressed by [5,9]. In [9] a multi-routing NoC structure is implemented. It forces the communication among a pair of source-destination IP through disjoint paths. The possible paths are stored into the network interfaces. The selection of a path is driven by the availability of the resources or by a pseudo-random generator. The work of [5] uses the adaptive west-first routing (WFR), a well-known, low-cost, and deadlock-free algorithm. It analyzes and compares the packet destination and the current router position. Despite these approaches succeed on splitting the NoC traffic, the path selection does not take into account the security metrics to define the new routes. Thus, they are prone to attacks. Sensitive information may be deviated to riskier paths. In order to overcome this difficulty, we propose in this work a *GRaNoC*, a risk-aware NoC able to route packets through low-risk paths.

III. Threat Model: Risk Paths

MPSoCs are able to support several applications which may change during runtime. In order to increase the performance of the system, applications are split into tasks and spread on the MPSoC resources [1]. Such a technique forces the peer interaction among the 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 NoC route that communicates sensitive information between two IP cores is known as *sensitive path*. Fig. 1

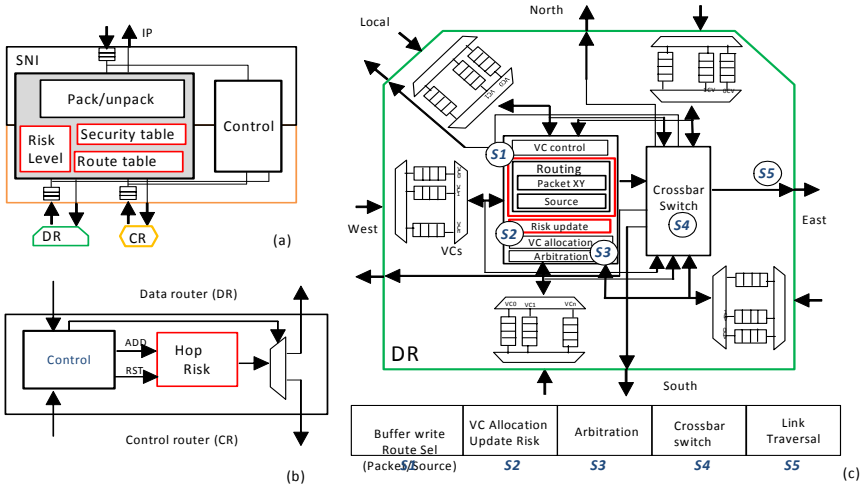

Fig. 2. Microarchitecture of: (a) Secure network interface (SNI); (b) Security control SC; (c) Data router DR.

shows a sensitive path composed by 5 routers (DR_1, DR_2, DR_3, DR_6, DR_9) from IP1 to IP9.

We consider that the attacker can infect the IPs of the MPSoC by exploiting the configurability of the processors. Code injection can be used to install malwares in the MPSoC, turning an IP malicious. However, the NoC and interfaces are considered secures. The attacker cannot modify their behavior but just. This assumption is valid, once the integration role of such components make that the interfaces and NoCs are built in-house [12]. Therefore, they are secure.

In order to perform the attack, the following preconditions are required:

- The mapping of the sensitive cores is known.
- The NoC routing algorithm is known.
- Attacker is able to infect an IP of the MPSoC.
- Attacker can control the traffic generation and monitoring of the infected IP.
- Infected IP is in the sensitive path.

We consider that sensitive (*S*) and malicious (*M*) processes are executed simultaneously at the sensitive IP core (IP_1) and infected IP core (IP2) of the MPSoC.

The attacks considered in this work are the timing attacks previously described by [5,13]. Timing attacks use the communication collision between the sensitive traffic and the attacker traffic in order to reveal a secret. Data dependences of critical applications are reflected in the sensitive traffic [14].

By continuously requesting NoC communication, the infected IP core can monitor the collisions with the sensitive packets. Such packets can be meaningless. The throughput of the infected IP reveals the access pattern and the volume of communication over the sensitive path. Considering the mesh based and XY routing NoC presented in Fig. 1, the infected IP core can be placed at IP_2. The attacker can perform the attack by exploiting the collision in a single router (single-point sensing), or through the sensitive path (multiple-point sensing). The latter attacks aim to embrace several points in the NoC in order to collect further information of the traffic of the system. The infected IP sends multiple requests to an IP close by the destination of the sensitive path [12].

The work of [13] shows that an attacker inside the sensitive path can detect the sensitive traffic with a success rate of 97%. Moreover, in the work of [11] it has been shown that by observing the traffic due 76 AES encryption, the infected core I inside the NoC-based MPSoC, is able to retrieve 12 of the 16 bytes of the secret key. Complementary brute force attack can be used to reveal the complete secret key, by exploring the remainder 2^{32} possibilities.

As a protection mechanism we consider that firewalls are placed in the secure network interface (SNI), between the IP and the router. The firewall matches the packet content regarding their content (source, destination, operation and address) and traffic characteristics (bandwidth) as in [3,7]. Fig. 1 shows the detection of a single-point (A_1 in the IP_2 network interface) and multiple-point (A_2 in the IP_6 network interface).

When an attacker performs a timing attack, the firewall can identify the increase of the bandwidth in the attacker network interface (A_1). Moreover, as frequent and usually similar request are going to be performed to the IP_6 (IP close by the sensitive destination), the firewall also can detect the attack (A_2).

Despite the firewalls can indicate the presence of an attack, the mitigation is not performed. Moreover, the bandwidth restriction of the firewall may not be enough for totally avoiding the sensing by the attacker. The work of [13] shows that the attack can succeed by alternating high data rates and low data rates. The firewall will only

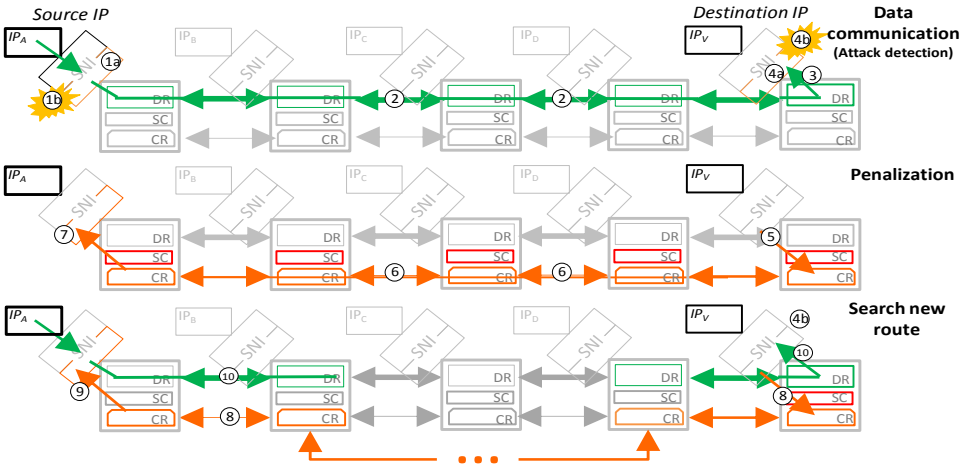

Fig. 3. Operation of the Global Risk-aware NoC.

detect the first scenario. Thus, it becomes critical to identify the sensing points and skip the infected IPs and the compromised paths. In this work we propose to consider the firewall notifications in order to avoid the risk routers and paths inside the MPSoC.

IV. Global Risk-aware NoC Architecture (*GRaNoC*)

The general overview of the *GRaNoC* architecture is shown in Fig. 1. It is composed of four main components: i) *Secure network interfaces* (SNI), which implement the communication protocol and security checking by means of firewalls; ii) *Hops*, which integrate the *Data Routers* (DR) and *Control Routers* (CR) to exchange data and control signals; and iii) *Security manager* (SM), which configures the firewalls and controls the recovery mechanism under a possible attack.

A. Secure network interfaces

The SNI microarchitecture is shown in Fig. 2(a). It is composed of five elements to implement the communication protocol and the security checking: pack/unpack, security table, route table, Risk level and control.

The communication protocol is enforced by the pack/unpack and control blocks. They wrap data as packets (at the injection to the hops), and retrieve data from packets (at ejection from the hops). Packets are composed of seven fields: i) *source*, to identify IP that inject the data; ii) *destination*, to identify the target of the communication; iii) *route,* correspond to the value stored at the route table of the source SNI. It contains the path that the packet must follow when the source routing is activated; iv) *risk*, to store the accumulated risk of the path; v) *operation*, to identify the type of packet (control, data write, data read); vi) path, intended for authentication purposes by storing the marks of each router used by the packet; and vii) *payload*, that corresponds to the data generated by the IP *source. control*). Note that non-sensitive packets do not contain the fields iii) and iv).

The communication security is enforced by means of firewall-based traffic inspection (*security table* and *risk level*). Each time a packet is injected or received, the security checking is performed. At the source IP, the access control is performed by verifying the destination and operation fields of the packet. At the destination IP, the authentication is performed by checking the source, operation and risk fields of the packet. When the security rules are violated, the firewall generates a notification, which will be communicated to the security manager (SM) through the Control NoC. This notification is used to increase the risk value: i) at source hop, when the attack is identified at the source network interface (A_1); or ii) at the hops used by the malicious packet, when the attack is identified at the destination network interface (A_2).

Each time a new application is mapped on the system, the risk level of the hops linked to the modified IPs is stored at the SM and the hop risk is restarted.

B. Hops

They perform the switching of data and control packets. Each commutation node integrates a five-stage look-ahead Data Router (DR), a Control Router (CR) to switch packets from an input to an output of the router. Also a security control (SC), shown in Fig. 2(b), is included to interface the DR and CR, as well to store the hop risk value. The microarchitecture of DR is shown in Fig. 2(c). Each DR/CR is connected through bidirectional links to a set of neighbor routers, a secure network interface and a SC, as shown in Fig.1.

Data routers integrate five main components: i) *input buffers*, organized as a set of virtual channels (VCs) which store the data at the input ports of the router. VCs are managed by the *VC control*; ii) *routing algorithm*, which selects the output port to redirect the incoming data. According to the *route* field at the packet header, the routing is performed by means of *packet routing* (implementing the XY routing) or *source routing* (using the predetermined route); iii) *arbitration* logic, that grants the utilization of the crossbar switch to one of the input VCs; iv) *crossbar switch*, which links the inputs to outputs of the router; and v) *risk update,* which stores the current *hop risk* and adds it to the value stored at the *risk* field of the packet.

The router communication is composed by 5 stages: i) *Buffer write and route selection*, to store incoming data and quantify the output port (S1); ii) *VC allocation and update risk*, to reserve the VC at the neighboring router linked to the output port and modify the risk field of the packet (S2); iii) *arbitration*, to schedule the commutation of the data (S3); iv) *crossbar switch*, to commute the data by the crossbar (S4); and v) *link traversal*, which includes the time required to reach the next input port (S5).

CR is similar to DR. The differences between these routers are the link width, the control flow (there are no VCs and VC control), small buffer depth, only packet routing and there is no *risk update*. In addition, it has the *Risk Table,* which stores values of *risk hop* of the current hop and its neighbors.

C. Security manager

The *security manager* is a light software task executed into a secure processor. It compiles the security requirements of the software tasks executed on the MPSoC and transforms them into a set of access rules, that are stored at the *security table* of the firewalls, and the accepted risk of each sensitive flow expressed as a *risk level* for each pair source/destination IP.

D. Operation of GRaNoC

The operation of the *GRaNoC* is shown in Fig. 3. It shows a sensitive communication through *GRaNoC* which includes 5 hops (from IP_A to IP_V). The operation contains 10 steps divided into four stages:

Data communication*:** The source IP_A generates the data which is wrapped as a packet and checked by the firewall of the source SNI at step (1a). If the packet is compliant with the security rules, it will be injected and transmitted through the data NoC by means of the DRs at step (2). Sensitive packets capture the hop risk of the sensitive path by upgrading the packet risk field. Packets can be commuted by distributed XY routing algorithm, when packets follow the original path, or source routing, when packets follow alternative low-risk paths. When the destination IP is reached at step (3), the packet is evaluated by the destination firewall in step (4a). It also retrieves the total risk of the sensitive path from the packet risk field and compares with the risk threshold. If the packet meets all the security requirements it is consumed at the destination. Otherwise, the penalization, at step (5), or a new route, at step (9), are started.***

***Penalization*:** Each time a packet violates a security rule, the firewall will notify the security manager of the system. An attack can be identified at the source IP, as in step (1b), or at the destination IP at step (4b). At (1b) the firewall notification is sent to the source hop and its SC will increase the hop risk value. Otherwise, at step (4b), the destination SNI injects a penalization packet at (5). It uses the packet path field to perform the backtracking of the attacker packet through the control NoC at step (6). At each hop the SC increases the hop risk value.

Search new route: When the sensitive flow exceeds the *risk threshold* at the destination SNI, a low-risk path must be set. At step (8) the destination SNI injects a *search* packet, which is routed according a searching mechanism. The new low-risk route is stored at the route table at step (7). Fig. 4 shows the communication between IP_7 (source IP) and IP_{25} (destination IP). Hop risk is represented by the circled number. At each hop, the risk field of the packet is modified by the DR. When the search packet arrives at the SNI_{25}, the value of *risk* is equal to 40 (0+5+14+8+12+1+0). Note that risk of the source and destination are not considered for the risk path calculation. This value is compared with the *risk level* allowed for this sensitive flow. Suppose it is equal to 35, then the SNI_{25} initiates the *search of the route* by means of one of the four

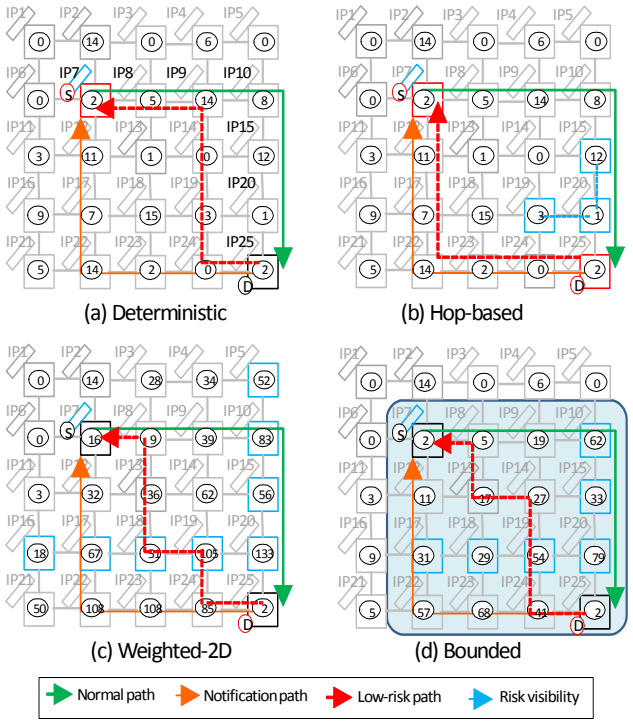

Fig. 4. Five mechanisms for setting low-risk NoC paths.

mechanisms of Fig. 4, described as follows:

i) ***Deterministic:*** It employs pre-defined alternative paths stored at design time at the route table of the source SNI. Each time a path exceeds the risk, the route is changed by the next alternative stored at the table, as shown in Fig. 4(a). This approach limits the number of alternative paths and does not guarantee the low-risk paths, but presents a low overhead in terms of the system communication and computation capabilities. The new path risk is equal to 22 (0+0+3+0+14+5+0).

ii) ***Hop-based:*** It defines the lower risk path hop by hop during the backtracking of the sensitive path from the destination IP to the target IP. In order to select the new path, two criteria are used: i) destination, the next hop must draw near to the source IP_7; and ii) risk, the neighbor hop with the lowest risk is selected. In the case the two hops have the same risk values, a random selection will be performed. Fig. 4(b) shows that at DR_{23}, the DR_{22} (risk=14) is selected when compared to the DR_{18} (risk=15). Following this mechanism, the new path risk is equal to 34 (0+0+2+14+7+11+0). Note that local decisions may trap the search packet into dangerous paths. However, the additional logic required to implement this mechanism is low.

iii) ***Weighted-2D:*** It is similar to the hop-based but it employs a risk value which corresponds to the sum of the risks of the aligned column and row of the hop weighted by the distance of the hop. Each hop integrates the risk value of column and row. Each time the risk value of a hop is updated, the new value should be broadcasted to the hops that belong to the column and row through the CRs. Fig. 4(c) shows that at DR_{18}, the DR_{13} (weighted risk=36, hop risk=1) is selected when compared to the DR_{17} (weighted risk=67, hop risk=7), differing from the hop-based selection. As a result, the new path risk is equal to 24 (0+0+3+15+1+5+0). The weights of the vertical and horizontal neighbors offer a better awareness of the possible risk at each direction. As a result, there is a higher probability to select the low risk paths.

iv) ***Bounded:*** It is similar to the weighted but includes only the columns and rows inside the area delimited by the minimal path between the source IP and destination IP. This approach eliminates the noise due IPs outside the possible routing area. In order to achieve that, the destination IP quantifies the boundaries and broadcast the information through the search packet. Fig. 4(d) shows that at DR_{19}, the DR_{14} (bounded risk=27, hop risk=0) is selected when compared to the DR_{18} (bounded risk=29, hop risk=15). Following this mechanism, the new path risk is equal to 9 (0+0+3+0+1+5+0).

Note that the example shows the existence of several route alternatives. None of these approaches guarantee that the lowest risk-path is found, once we use the minimal path restriction in this work. The integration of a mechanism that guarantees the non-minimal path and the minimal risk is out of the scope of this paper.

Low-risk route: At the arrival of the search packet to the source SNI of the sensitive data (SNI_6), the path field of the packet is retrieved. It contains the route information of the packet. This route is stored into the route table of the SNI_6. Packets whose destination is the destination IP (IP_{16}) will include the new low-risk route at the route field of the packet and will be communicated by source routing algorithm. Remaining packets will be communicated by means of XY routing.

V. EXPERIMENTAL WORK

We have modelled our architecture in SystemC-TLM and VHDL-RTL, which extend the NoC design framework presented in [11]. Blind is a modular cycle accurate simulation environment which supports a wide variety of Instruction Set Architectures (ISAs), 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 the model of our *GRaNoC* architecture we are able to model a 64-core MPSoC that is interconnected though a two-level 8x8 two-level mesh-based *GRaNoC*. The feasibility of the integration of a security countermeasure will depend on satisfying the security and performance requirements of the system. Our architecture has been evaluated under three conditions: scalability, performance and security.

The first scenario is the stand-alone low-risk route set. It measures the impact of setting a new route by using the *deterministic*, *hop-based*, *weighted-2D* and *Bounded* route mechanisms at *GRaNoC*. We compare the different approaches with a NoC with a basic firewall security and with a *GRaNoC* that implements an exhaustive route search, that is, when the risk hop value of all the possible routes at the bounded region are retrieved and evaluated. This approach will guarantee the lowest-risk paths, but can incur in high performance degradation. Fig. 5 shows the impact of the stand-alone scenario for different *GRaNoC* sizes (from 2x2 to 8x8). *Deterministic* approach presents the best scalability. Although the *dynamic low-risk* routes (*hop-based*, *weighted-2D* and *Bounded*) enhance the performance up to 75%, 70% and 64%, respectively, when compared to the exhaustive solution.

The second scenario includes the performance and cost evaluation of *GRaNoC*. Uniform traffic and SPLASH-2 programs are widely used as a benchmark for NoC-Based MPSoC. Fig. 6 shows the performance evaluation for systems under different degrees of dangerousness. The firewall detection forces the route modification in 20% of the sensitive transactions. That is, 20% of the sensitive traffic communication exceeds the *risk level*. We also

include the results for the NoC basic firewall security and exhaustive NoC configurations.

The results show that the number of times that the route is performed has a low impact on the deterministic *GRaNoC* performance. However, for weighted-2D and bounded *GRaNoCs*, which employs the risk values of several neighbors and require the broadcast of the risk values, the overhead can be significant. For the bounded *GRaNoC*, the quantification of the risk value is performed at the arrival of the search packet. In to decrease the area overhead, bounded *GRaNoC* uses smaller risk tables when compared to the weighted-2D *GRaNoC*. For bigger security areas, the remaining missed risk values must be retrieved before a route decision can be performed. The hop must request the risk value to its neighbors and wait until the response arrival. Just then, the hop risk can be quantified. Fig. 7 shows the normalized execution time of *GRaNoC* under SPLASH-2. The results show that the *GRaNoC* configuration that implements *deterministic route* achieves the best performance. Table I shows the area and power overheads of the *GRaNoC* configurations. The basic firewall security also was including for comparison purposes. Note that all the *GRaNoCs* also include the basic firewall security. Each alternative presents a trade-off among the latency, area and power overhead. While deterministic *GRaNoC* achieves the best performance, it presents a huge area and power overhead due the resources that must be integrated for storing the possible alternative paths of the sensitive traffic. The size of tables will depend on the number of destinations that require secure transactions together with the number of secure route alternatives. The overhead of the hop-based *GRaNoC* was the slightly superior to the basic firewall approach. This alternative requires only a small additional register and control to perform the route decision. Bounded *GRaNoC* results show that despite degrading the latency, is the second best alternative for saving area and power, due to the reduction of the LUT tables, when compared to the Weighted-2D approach.

The third scenario is the security evaluation. In order to test the protection of the different *GRaNoC* alternatives, four attack scenarios were implemented. The description of the attacks is given in Table II. The attacks attempt to read and write sensitive areas and perform DoS attacks at NoC paths and routers. The results present the amount of attack points avoided by the different *GRaNoCs*. The results show that despite the best performance of the deterministic approach, it results in an unsecure architecture. The deterministic GRaNoC security depends on the nature of the application and on the knowledge of the security designer. These two factors will determine the quality of the protection of the new routes.

Hop-based and weighted-2D *GRaNoCs* protect the sensitive flows in up to 84% of the cases. The local route selection of hop-based *GRaNoCs* may result in dangerous areas, where there is none low-risk hop. Regarding the weighted-2D *GRaNoC,* the security is penalized by the

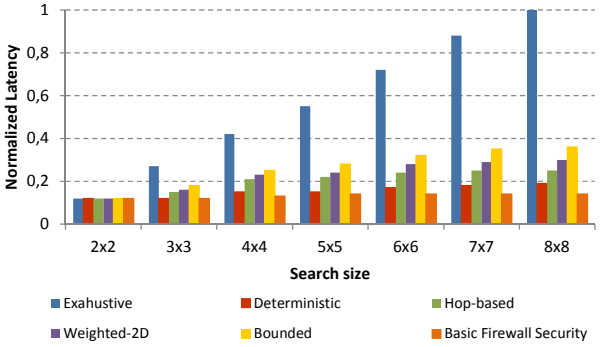

Fig. 5. Impact of the stand-alone *GRaNoC* scenaro.

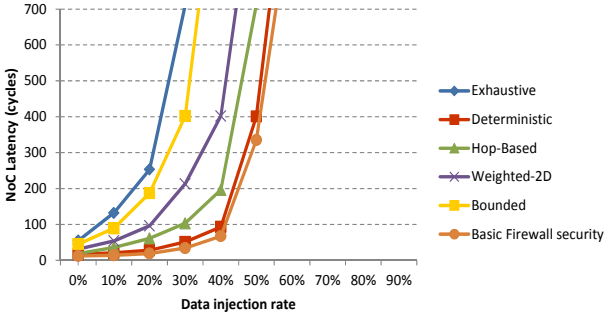

Fig. 6. Performance evaluation of the *GRaNoC* under uniform traffic (20% of new routes).

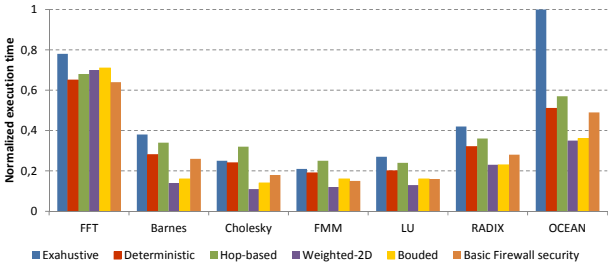

Fig. 7. Performance evaluation of the *GRaNoC* under SPALH-2.

TABLE I. OVERHEAD OF *GRANOCS* WHEN COMPARED TO THE TWO-LEVEL NOC WITHOUT SECURITY

Configuration	***Latency***	***Area***	***Power***
Basic Firewall security	5,9%	6,2%	4,2%
Deterministic	6,5%	9,6%	12,3%
Hop-based	8,3%	6,4%	5,1%
Weighted-2D	12,6%	14,3%	8,5%
Bounded	14,3%	8,6%	7,3%

inclusion of the risk values of hops (at the column or at the row) that are out of the routing area, and thus, that are never used by the packet. This noise may impact the quantification of the current risk and thus harm the routing decision. The best alternative, in terms of security is the bounded *GRaNoC*. By including only the risk values of the hops that may be used by the sensitive packet, offers the best global risk evaluation.

TABLE II. ATTACK DESCRIPTION AND SECURITY EFFICACY

Attack scenario	*Firewalls*	*Deterministic*	*Hop-based*	*Weighted-2D*	*Bounded*
Overwrite memory data to modify the system behavior by corrupting memories	100%	100%	100%	100%	100%
Read not allowed memory data	100%	100%	100%	100%	100%
Inject valid repeated packets to congest the communication and to create hot spots (DoS NoC path)	100%	64%	86%	84%	100%
Inject packets which destination is the same as the initiator to block degrade the system performance (DoS NoC router and spying)	100%	71%	87%	93%	100%

VI. CONCLUSION

GRaNoC protects sensitive paths against several types of attacks. By using the risk metric (number of attacks detected by the firewalls at each hop) *GRaNoC* is able to select low-risk paths on the network. Global risk-awareness is achieved by including the risk value of neighbor hops. We propose five mechanisms to search the low-risk paths. They can be selected statically, through predefined route tables (deterministic), or dynamically by using the risk table at each point (hop-based, weighted-2D and Bounded). Although the predefined approach achieved the best performance, the dynamic approaches provided the best protection due to the state awareness. The selection of one of these alternatives depends on the MPSoC and application requirements. As future work we plan to develop smarter routing algorithm to find the optimal low-risk path on the NoC.

ACKNOWLEDGMENT

This work was partly funded by the German Federal Ministry of Education and Research (BMBF) in the project SIBASE under grant number 01IS13020A.

REFERENCES

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

[2] J. Backer et al, On enhancing the debug architecture of a system-on-chip (SoC) to detect software attacks. In Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems. DFTS 2015.

[3] S. Lukovicn et al, Enhancing network-on-chip components to support security of processing elements. In Proceedings of the 5th Workshop on Embedded Systems Security. WESS 2010.

[4] Wang et al., Efficient timing channel protection for on-chip networks. In Proceedings of the 6th International Symposium on Networks-on-Chip. NOCS 2012.

[5] J. Sepulveda et al, NoC-Based Protection for SoC Time-Driven Attacks. In. IEEE Embedded System Letters. Vol. 7, 2015.

[6] L. Fiorin et al, A security monitoring service for NoCs. In Proceedings of the 6th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis. CODES 2008.

[7] D. Florez et al, Elastic Security Zones for NoC-Based 3D-MPSoCs. In Proceedings of the 21st IEEE International Conference on Electronics, Circuits and Systems. ICECS 2014.

[8] G. Gogniat et al, Reconfigurable Security Architecture for Disrupted Protection Zones in NoC−Based MPSoCs. In Proceedings of the 10th International Symposium on Reconfigurable Communication-centric Systems-on-Chip. ReCoSoC 2015.

[9] R. Stefan et al. Enhancing the security of Time-Divison Multiplexing Networks-on-Chip through the use of multipath routing. In Proceedings of the 4th International Workshop on Network on Chip Architectures. NoCArc 2011.

[10] J-P. Diguet et al, NoC-centric security of reconfigurable SoC. In Proceedings of the 1st International Symposium on Networks-on-Chip. NOCS 2007.

[11] J. Sepulveda et al, Side Channel Attack on NoC-based MPSoCs are practical: NoC Prime+Probe Attack. In Proceedings SBCCI 2016.

[12] D. M. Ancajas et al. Fort-NoCs: Mitigating the threat of a compromised NoC. In Proceedings of the 51st Design Automation Conference. DAC 2014.

[13] C. Reinbrecht et al. Gossip NoC - Avoiding Timing Side-Channel Attacks through Traffic Management. In Proceedings of the IEEE Computer Society Annual Symposium on VLSI. ISVLSI 2016.

[14] A. Moradi et al. One Attack to Rule them all: Collision Timing Attack versus 42 AES ASIC Cores. In IEEE Transactions on Computers, vol. 63, 2014.