Hierarchical NoC-based security for MP-SoC dynamic protection

Johanna Sepulveda[1], Guy Gogniat[2], Cesar Pedraza[3], Ricardo Pires[1], Wang Jiang Chau[1] and Marius Strum[1]
[1] Microelectronics Laboratory LME, University of São Paulo, São Paulo, Brazil
[2]Information and Communication Science and Technology Laboratory Lab-STICC, Université Bretagne Sud, France
[3]Telecommunications Engineering Faculty, Santo Tomás University, Colombia
jsepulveda,jcwang,strum@lme.usp.br, guy.gogniat@univ-ubs.fr, cesarpedraza@usantotomas.edu.co

Abstract— **MPSoCs are able to support multiple applications on the same chip. This flexibility offered by the MPSoC also represents a vulnerability, turning the MPSoC security specially challenging. The goal of the designers is to provide MPSoC protection that meets the performance and security requirements of all the applications. The Network-on-chip (NoC) interconnection structure can be used to efficiently overcome the present MPSoC vulnerabilities. In this paper, we present the implementation of a hierarchical security NoC-based architecture to detect and prevent a wide range of MPSoC attacks. We integrate agile and dynamic security firewalls into the NoC in order to detect attacks based on different security rules. It uses the QoSS (Quality of Security Service) concept. It takes into account the tradeoff between security and performance. We evaluate the effectiveness of our approach over several MPSoCs attack scenarios and estimate their impact on the overall performance. We show that our architecture can perform a fast detection of a wide range of attacks and a fast configuration of the different security policies for several MPSoC applications.**

Keywords-security; network-on-chip; Multi-Processor SoC; QoSS (Quality-of-Security-Service)

I. INTRODUCTION

MPSoCs (Multi processor System-on-Chip) have been proposed as a promising architecture choice to overcome the new challenging application requirements. A MPSoC integrates multiple programmable processor cores, specialized memories and other intellectual property (IP) components into a single chip [1]. The MPSoCs platform allows the execution of several applications in the same structure. Each application supported by the MPSoC is characterized by different sets of security rules, called security policy. The set of applications can be mapped dynamically at the MPSoC. Therefore, there is not a single and static security requirement, but a set of ever changing security policies that must be satisfied. The security of the MPSoCs must be able to have different levels and be capable of being changeable through the operation time. MPSoC can be attacked via hardware/software [3]. Software attacks are responsible for 80% of the security incidents [3]. All software attacks starts with an abnormal communication. In this paper we address protection of the MPSoC against the software attacks by the implementation of the security at the NoC-based communication structure. In order to support the MPSoC high communication requirements the Network-on-Chip (NoC) is employed [4-7]. A NoC is an integrated network that uses routers to allow the communication among the computation structure components. The data flow through the NoC as packets. The NoC may contribute to the overall security of the system, providing the ideal mean for monitoring systems behavior and detecting specific attacks [7]. The communication structure is becoming the heart of the MPSoC [6]. It has a significant impact on the overall MPSoC performance. To make feasible the MPSoC protection by NoCs, the security must be customized, in order to provide a better trade-off between the system performance and the security. Our work proposes the implementation of QoSS (Quality of security service) to overcome present SoC vulnerabilities. QoSS is a novel concept for data protection that introduces security as a dimension of QoS (Quality-of-Service). QoSS uses a Network-on-Chip (NoC) to provide predictable security levels of the system by adding functionality to the routers of the network and consequently changing some local configuration parameters or modifying the network interfaces. Our hierarchical approach distributes the security policy management by partitioning the NoC topology into different *security zones* (*low NoC*), ruled by a *local* security policy. Different security zones are connected through a global interconnect (*high NoC*), ruled by a *global* security policy. Our approach provides an effective way to handle security policy changes and improves the overall system performance. Each zone integrates a set of mechanisms capable of being configured according to the QoSS needs of each application. We show that our architecture can perform a fast detection of a wide range of attacks and a fast configuration of the different security policies for several MPSoC applications. We also show that the penalties due to the integration of the dynamic NoC-based security architecture are limited to a fraction of time and space of the system. For the best of our knowledge, it is the first attempt to implement layered security architecture able to handle different application security policies at the MPSoC. The different levels of security of each *security zone* arise from the configuration of the parameters of the NoC security mechanisms. Two techniques are employed in order to achieve an efficient configuration: 1- the security mechanisms are implemented hierarchically therefore avoiding the NoC interruption; and 2- QoS (Quality-of-service) mechanisms are employed to provide predictable penalties while the network interfaces are modified. The experiments were performed using a SystemC-TLM timed simulation framework. It automatically carries out performance evaluations for a wide variety of MPSoC scenarios. The remaining text is organized as follows: Section 2 presents an

overview of previous NoC security works. Section 3 presents the security concerns in MPSoC. Section 4 presents the QoSS concept and our security mechanisms. Our architecture is described in Section 5. The experiments and its results are described in Section 6. Finally we present our conclusions in Section 7.

II. PREVIOUS WORKS

Security integration at the NoC level was addressed in the works of [7-8]. [7-9] integrate a table at the network interface containing the access control rules of each IP. They specify how a component of the NoC can access the protected device. The packets that do not satisfy the access control rules are discarded. The purpose of [8] is to prevent attacks through the verification of the source and size of the packets. The packets that do not obey the communication rules are discarded. This work also integrates a secure network manager component to monitor the NoC behavior. The purpose is to prevent four common NoC attacks: denial-of-service (DoS), draining, extraction of secret information and modification. These previous works show the main role of the NoC in the security of the system. It is a powerful structure for the surveillance and detection of the SoC attacks. However, the adoption of these previous works to address the MPSoC security challenges present three main limitations. 1) they implement a single NoC security level for the entire SoC; 2) they implement static security policies; and 3) they are not appropriate for multithread systems. In our previous work [4] we developed a dynamical NoC-based protection for SoCs. However it uses central control components that present a huge impact on the overall area due to the link overhead. The purpose of the present work is to overcome these limitations by using a hierarchical NoC-based implementation and the QoSS concept.

III. MPSOC SECURITY CHALLENGES

Current ubiquitous computing and flexibility in MPSoC design trends promote the resource sharing and upgrading capabilities that integrates the MPSoC onto an aggressive world. Many SoCs interact with other electronic devices, in many cases wirelessly. By interacting with other digital devices, a SoC may receive viruses (or other similar malicious pieces of code). MPSoC attacks exploit different system vulnerabilities. It is known that previous MPSoC attacks have succeeded. An attack can be defined as any unauthorized attempt to access or to use the system resources [3]. In order to prevent and to mitigate attacks, security services can be implemented at the NoC. The main function of security services is to protect network resources and data exchanged by means of communication management [8,9]. There are six security services [9]: 1) Confidentiality: ensures the data secrecy; 2) Integrity: assures that data are identically maintained during any operation; 3) Authentication: validates the sender IP integrity; 4) Access Control: allows or denies the use of a particular resource; 5) Availability: ensures the use of the network resources; and 6) Non-repudiation: maintains evidence of NoC communication events. MPSoC security policy must be upgraded in response to the execution of a new application. We consider, for example, that there is a MPSoC designed to support three applications (*A1*, *A2* and *A3*). Initially, the set of applications *A1* and *A2* are being executed in the MPSoC when, at some instant, the application *A3* must be executed. The tasks of *A3* are mapped onto the components of the MPSoC. The task that performs critical functions and the sensitive information (i.e. a ciphering key or personal data) of *A3* are mapped together with the tasks of *A1* and *A2*. The tasks of *A3* will be unprotected, if the security policy, that defines the access control rights, is not upgraded for the new scenario. An attacker can take advantage of this security hole and use the rights over *A1* and *A2* to expose/modify the sensitive information of *A3* or/and avoid the utilization of the MPSoC resources by the tasks of *A3* by keeping busy the component with a never-ending *A1*/*A2* task execution.

IV. QUALITY-OF-SECURITY-SERVICE (QOSS)

The traffic of a single embedded application may integrate several flows, each of which is characterized by different security requirements. The QoSS (Quality-of-Security-Service) concept allows differentiated treatment different data exchange events carried out through the NoC. The QoSS can be implemented by adding functionality to NoC components. Different security levels are implemented through security mechanisms. The advantage of the QoSS inclusion is the customization of the security, arising from the adoption of different *security levels;* it enhances the efficiency of the resources utilization, allows better system control and increases the system flexibility. Each level represents a security choice according to the security policy. The security choices represent a special configuration of the *security mechanisms*. The security mechanisms use the embodied information within the packet to allow or block a transaction. For comparison reasons, in this paper we address access control and authentication services. They are implemented as *firewalls* at the NoC interface. Each firewall stores the security policy information of each resource in a *security table*. Each time a transaction takes place, the corresponding security data embodied in the packet is checked against the security table information. The rights change according to the applications that are mapped on the MPSoC at that time. Unauthorized packets are discarded. It should be noted that our method can support any number of security levels.

A. Access control

Our access control service implements four security levels (L0-L3), which arise from the combination of three security mechanisms that verifies: source (SV), operation (OV) and role (RV). Table I shows the different access control levels. A higher security level checks more information embodied at the packet.

B. Authentication

Our authentication service implements four security levels (L0-L3), which arise from the combination of three security mechanisms that verifies: Source (SV), path (PV) and code (CV). Table I shows the different authentication levels. The *security table* stores the correct signature and code of a trusty packet.

TABLE I. ACCESS CONTROL AND AUTHENTICATION LEVELS

	Access Control			Authentication		
Level	SV	OV	RV	SV	PV	CV
L0						
L1	x			x		
L2	x	x		x	x	
L3	x	x	x	x	x	x

V. OUR APPROACH

Our architecture integrates four key components: *low/high NoCs, policy keeper*, *configuration control, monitors* and *security mechanisms*.

A. Description

Low/high NoCs*:* The hierarchical NoC is divided into low and high NoCs. The IP cores of the MPSoC are organized into independent clusters forming *low* NoCs. The *high* NoC integrates the traffic coming from these different clusters. Each cluster constitutes a *security zone*, composed by the IPs with similar security characteristics. The security policy, that rules the interaction among all the MPSoC components, is divided into local and global policies. The IPs of a security zone are ruled by the local security policy. The inter-cluster communication is ruled by the global security policy. Such hierarchical approach provides 3 advantages: 1) facilitates the security management of the MPSoC; 2) uses smaller security tables; and 3) improves the system performance. Fig 1. shows our architecture using four security zones (Zone I to Zone IV).

Policy keeper: It is a safe component that stores a thread-oriented policy (local and global) representation. It integrates the information of the MPSoC thread scheduler, the security zone and the access rules (rights). The local security policy configures the *low* NoC security mechanisms of each *security zone* and the global security policy configures the security mechanisms embodied at the *high* NoC.

Configuration Control: Coordinates and configurates the security mechanisms of the MPSoC according to the *local* and *global* security policies of each application. It uses the NoC interface ID source/destination information embodied in the *policy keeper* component to block the communication and start the reconfiguration process of the security mechanisms embodied at the *high NoC* and subsequently at the *low NoC*.

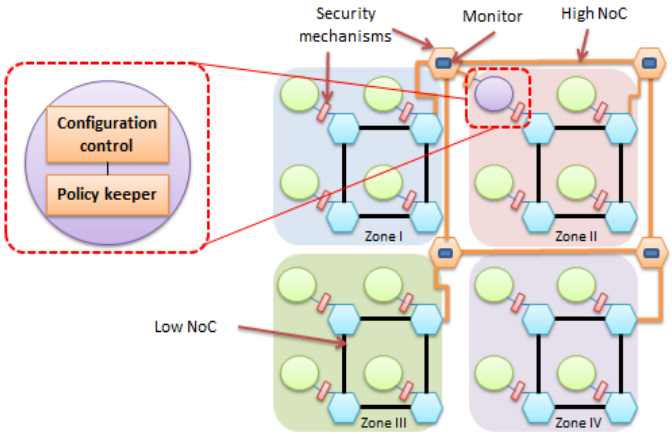

Fig 1. Hierarchical architecture for dynamic protection.

Security Mechanisms: The NoC firewalls are implemented at the network interfaces of *low NoC* and *high NoC* (See Section IV). Our approach supports the multithreading characteristic of the MPSoC. Note that our architecture is also feasible for different security mechanisms whose security characteristics are capable of being changed throughout the operation time.

Monitors: They are integrated at the *high NoC* routers and are permanently auditing the communication behavior of the MPSoC. They monitor the NoC activity in order to determine the completion of the communication among the different master/slave pairs of the MPSoC. The monitor also has the ability to record and report on the security mechanism configuration at any moment.

B. Operation

When the security policy of the MPSoC must be updated, the *configuration control process* starts three procedures: 1) *look up policy keeper*; 2) *block, look up monitors and global/local configuration* and; 3) *unblock*. At the *look up policy keeper* the configuration controller downloads the proper *local* and *global* security policies, stored in the *policy keeper* component. It uses the MPSoC tasks mapping information to identify which *security mechanism* and which *security zones* must be modified and its suitable configuration parameters. The *block* procedure interrupts the injection of packets whose final destination is the processing component linked to the security zone that is going to be reconfigured (called *target interface* for explication purposes). Such packets are stored onto the interface linked to the master processing component. The reconfiguration manager also starts a QoS (Quality-of-Service) mechanism that modifies the arbitration of the routers and rises up the priority of the communication of the packets whose final destination is the component linked to the *target interface*. Once the communication of all the packets flowing to the target interface is finished, the reconfiguration of the *target interface* can be started. In order to configure the hierarchical NoC, the *configuration controller* sends the new information that must be stored at the *security tables* of the firewalls. The *high NoC configuration* modifies the security tables at the routers. The *low NoC configuration* modifies the security tables at the network interfaces of the selected security zone. Note that as the *security mechanisms* are implemented at the NoC interface, the communication is not interrupted at the NoC during the reconfiguration. The NoC routers continue forwarding the communication of the remaining packets through the network. This characteristic of our architecture can reduce the latency penalties due to the reconfiguration. When the reconfiguration is finished, the final *unblock procedure* is started. The *configuration control* frees the injection of the packets that were being blocked during the reconfiguration. The normal NoC operation is achieved when all the target interfaces are reconfigured.

VI. RESULTS

We have developed the SystemC-TLM cycle-accurate model of thisour architecture. Its evaluation was performed by

the SystemC-TLM framework developed in [4]. We employ a 4x4 mesh based NoC characterized by a XY routing scheme, round-robin (RR) arbiter and FIFO memory organization. The proposed solution has been verified against three types of attack scenarios: 1) extraction; 2) modification; and 3) DoS. The performance evaluation of our approach was based on MiBench benchmark suite [11]. We select three applications: auto/industrial, consumer electronics and telecommunication. Each application is characterized by a security policy that establishes different levels of authentication and access control security mechanisms. Our experimental work considers all the possible mapping combinations resulting for the execution of these applications on the MPSoC. Our MPSoC traffic is composed by a set of heterogeneous tasks arriving to different rates. We inject a percentage of Long-Range-Dependece (LRD) traffic at the MPSoC. Such traffic is typical for the three selected MiBench applications [10]. Each traffic pattern is composed of five flit size packets of three types: real-time, write or read and signaling, characterized by a different generation rate. The inter-cluster communication varies from 20% to 50% of the overall traffic. We compared the communication performance of our hierarchical NoC-based dynamic security architecture against the simple NoC-based dynamic security [4] and the best-effort NoC (without security). Our approach was able to detect all these attacks. Figures 2-3 show the latency and power distributions over all the components of our hierarchical architecture. They show that the security interfaces and the policy keeper are the components that consume more time and power, respectively. The results of the comparison among the best effort NoC (without security), our previous work [4] and the present layered security architecture are shown at the Fig 4 and Table II. It shows that our hierarchical approach performs better than the simple dynamic security architecture for all the percentage of LRD traffic. Table II show the implementation penalties for both approaches when compared to the best effort NoC. It shows that our approach always achieve the best results.

VII. CONCLUSIONS

In this work we proposed the implementation of a hierarchical NoC security implementation able to support dynamic protection for MPSoC. We implement two security services: access control and authentication. We adopt the QoSS concept that allows the implementation of different security levels. Results show that the inclusion of security issues in the hierarchic NoC performs better that the simple NoC architecture. The inclusion of QoSS concept allows the designer to customize the MPSoC protection in order to satisfy both, security and performance requirements. Currently we are implementing cryptographic techniques to our hierarchic NoC in order to guarantee its security. As a future work, we will study different techniques that allow an improvement in the implementation of the proposed security mechanisms.

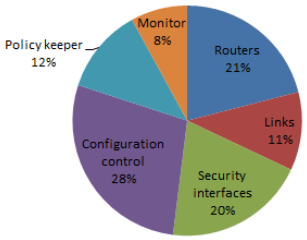

Fig 2. Latency distribution in our architecture.

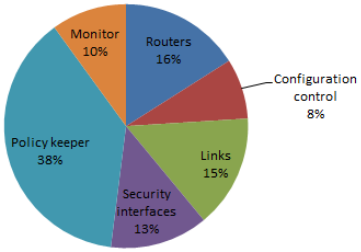

Fig 3. Power consumption distribution in our architecture.

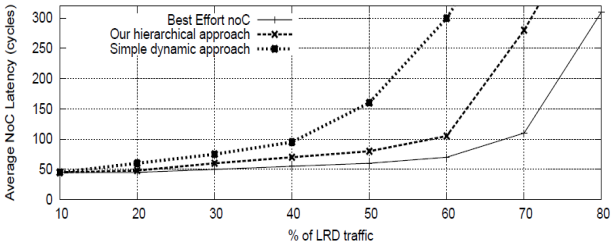

Fig 4. Average NoC latency

TABLE II. IMPLEMENTATION PENALTIES

Parameter	Dynamical approach	Our hierarchical approach
Latency increment	4.1%	3.8%
power increment	19.6%	7.6%
area increment	26.7%	5.2%

REFERENCES

[1] M. Loghi, F. Angiolini, D. Bertozzi, L. Benini, R. Zafalon.: Analyzing On-ChipCommunication in a MPSoC Environment. Design, Automation and Test in Europe Conference and Exhibition (DATE 2006).

[2] Benini L.: Application Specific NoC Design. Design, Automation and Test in Europe Conference and Exhibition (DATE 2006).

[3] Kocher P., Lee R., McGraw G., Raghunathan A., Ravi S.: Security as a New Dimension in Embedded System Design. Design Automation Conference (DAC2004). 2004.

[4] J. Sepulveda, G. Gogniat, J.C Wang, M. Strum. Dynamic NoC-Based architecture for MPSoC security implementation. In Proc. 24th symposium on integrated circuits and systems. SBCCI 2011.

[5] Gebotys C., Zhang Y.: Security wrappers and power analysis for SoC technologies. CODES 2003.

[6] Ogras U., Hu J., Marculescu R.: Communication-centric SoC design for nanoscale domain. IEEE IASS 2005.

[7] Fiorin L., Silvano C., Sami M.: Security Aspects in Networks-on-Chips: Overview and Proposals for Secure Implementations. In Proc. 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools. 2007.

[8] Evain S., Diguet J.: From NoC security analysis to design solutions. Design, Automation and Test in Europe Conference and Exhibition (DATE 2006).

[9] Fiorin L., Lukovic S., Palermo G.: Implementation of a Reconfigurable Data Protection Module for NoC-based MPSoCs. In Proc. IEEE Parallel and distributed processing. 2008.

[10] MiBench Version 1.0. http://www.eecs.umich.edu/mibench/