Security Aspects in Networks-on-Chips: Overview and Proposals for Secure Implementations

Leandro Fiorin
ALaRI, Faculty of informatics
University of Lugano
Lugano, Switzerland
fiorin@alari.ch

Cristina Silvano, Mariagiovanna Sami
Politecnico di Milano
Dipartimento di Elettronica e Informazione
Milano, Italy
{silvano, sami}@elet.polimi.it

Abstract

Security has gained increasing relevance in the development of embedded devices. Towards the aim of a secure system at each level of the design, in this paper we address security aspects related to Networks-on-Chips (NoCs) architectures. After presenting the attacks most likely to address NoCs, we survey existing academic and industrial secure architectures relevant to our case, focusing in particular on their communication infrastructure. We outline and propose possible solutions to contrast some of the attacks described and suggest the use of the NoC as a mean to monitor and detect unexpected system behaviors [1].

1. Introduction

Networks-on-Chips (NoCs) [3], have appeared in recent years as new strategy to connect and manage the communication among the variety of intellectual property blocks required in complex System-on-Chips (SoCs).

However, the advantages introduced by the use of such a complex communication infrastructure may lead to new weaknesses in the system that should be subjected to careful studies and evaluations. While NoCs have been an emerging area of academic and research interest, security in such systems remains so far mainly unexplored. Therefore, in this work we focus on the security aspects related to the intercommunication infrastructure and we provide an outline of possible techniques that could be applied to NoCs to contribute to the overall security of the system.

First, in Section 2, we present a classification of the type of attacks addressing an embedded system, in particular focusing on the analysis of those attacks that may exploit weaknesses in the communication system. Then, in Section 3, we present an overview of th most relevant academic and industrial works related to the security aspects on a NoC and on more general SoCs. In Section 4, we introduce what we think could represent the main blocks and characteristics of an Intrusion Detection System for NoCs, and in the last Section we outline future developments along such research lines.

[1] This work has been carried out under the MEDEA+ LoMoSA+ Project and is partially funded by KTI - The Swiss Innovation Promotion Agency - Project Nr. 7945.1 NMPP-NM.

2 Security problems in NoCs architectures

As presented in [10], security attacks to an embedded system can be classified in different ways. If we consider a classification in terms of the agents used to perform the attack, they can be grouped as *software attacks*, *physical* or *invasive attacks* and *side channel attacks*.

In the first group we can include all attacks launched through software agents such as viruses, worms and trojan horses, carried out, for instance, using pitfalls in code constructs, such as in the case of buffer overflow attacks [4].

Physical attacks require physical intrusion in the embedded system at some level. They imply the use of sophisticated micro-probing techniques, involving de-packaging and reconstruction of the layout, in order to infer at various granularity the architectural structures and values on the buses and interfaces of the components.

Side channel attacks are based on information gained from the physical implementation of the system, such as power consumption, timing information, or electromagnetic leaks. They exploit the correlation between the information measured and the execution flow generating it.

The adoption in SoCs of a complex communication paradigm may introduce additional weaknesses in the system. However, as we will discuss later on, such a non-trivial layered infrastructure can be also employed to detect specific attacks.

10th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2007)
0-7695-2978-X/07 $25.00 © 2007 IEEE

2.1 Specific attacks addressing NoCs

Systems based on NoCs can be subjected to attacks addressing their specific structure. In particular, three types of attacks can be identified [7].

Denial of Service (DoS) attacks aim at lowering system performance in several ways. Among those, *Bandwidth Reduction attacks* aim at reducing the communication bandwidth through the transmission of frequent and useless packets, in order to cause high latency in the on-chip communications, up to the saturation of the network. Operation life of battery operated embedded systems is the target of *Draining* or *Sleep Deprivation attacks*. This particular type of attack can be performed through continuous sending of requests to the victim of the attack, in order to make it execute power-hungry tasks.

Extraction of secret information aims at reading sensitive data, critical instructions or information kept in areas of the memory or in configuration registers in specific targets. It can be performed exploiting buffer overflow or similar techniques addressing software weaknesses.

Hijacking attacks aim at altering the execution or configuration of the system in order to make it perform tasks set by the attacker in addition to its normal duties, such as in the case of the exploitation of buffer overflow to bypass Digital Right Management's protection in audio CODEC [5].

3 On-chip secure architectures: state of the art

This section presents an overview of the most recent research work related to the security aspects of a NoC. We analyze here as well some relevant security solutions for more general SoCs.

The paper in [8] presents a framework to secure the exchange of cryptographic keys within a NoC, addressing in particular the protection from power/EM attacks of a system containing non-secure cores as well as secure ones. The framework supports authentication, encryption, key exchange, new user's keys and public key storage, and similar procedures. No unencrypted key leaves the cores on the NoC and only secure IP cores running trusted software are supported. At the network level, security is based on symmetric key cryptography, where each secure core has its own security wrapper storing a private network key in a non-volatile memory.

Evain and Diguet [7] address security vulnerable spots introduced by the use of NoCs in SoCs. Three possible network implementations can be considered, i.e., full ASIC, full FPGA and mixed, and the system can be divided in secure and non-secure areas. Depending on the level of configurability of the system, several attack scenarios are identified. To avoid bandwidth denial, the use of low and high

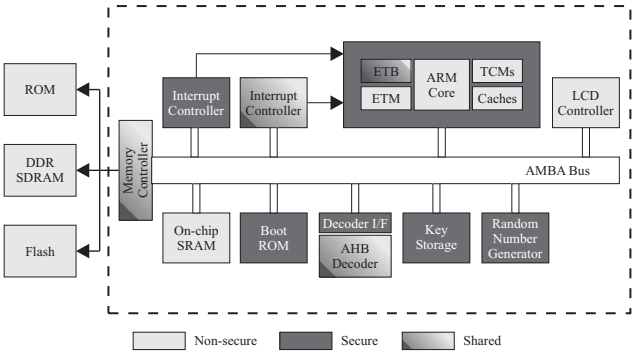

Figure 1. System example partitioning secure and non-secure area with TrustZone [2].

security virtual channels is proposed to transmit information in the secure area, giving higher priority to the information flowing in the latter. Multi-boundary filtering can be moreover considered to separate non-secure areas from secure areas. Initiators of the communication are uniquely identified through a routing techniques allowing the computation of the backward path at the target's Network Interface (NI).

Due to the increasing relevance that security aspects are gaining in SoC design, we provide hereafter an overview of more extensive and implementation-related work presented in the case of general SoC architectures.

The *ARM* approach to enabling trusted computing within the embedded world is based on the concept of the TrustZone Platform [2]. A TrustZone architecture is divided into secure and non-secure regions (see Figure 1). A Non-Secure indicator bit, 'NS', determines the security operation state of the various components and it can only be accessed through the 'Secure Monitor' processor mode, accessible only through a limited set of entry points. This mode is allowed to switch the system between secure and non-secure state, allowing a core in the secure state to gain higher levels of privilege. With regard to the interconnection system, the AMBA AXI Configurable Interconnect supports secure-aware transactions. Transactions requested by masters are monitored by a specific TrustZone controller, which is in charge to abort those considered illegal. Support to peripherals using the AMBA APB protocol is also provided.

Sonics [1] provides in its SMART Interconnect solutions an on-chip programmable security "firewall" to protect system integrity and media content processed. This is implemented through an optional access protection mechanism to designate up to 8 protection regions within the address space of specified targets. The mechanism can be dynamic and role-dependent, with sizes and locations of the protection regions programmable at run-time and with permissions defined as a function of the initiator attempting the

0-7695-2978-X/07 $25.00

SourceID.Role.MemAddr	Load/Store
0x 01 0 00XXXXXX	11
0x 2D 0 000XXXXX	11
0x 2D 1 000XXXXX	10
0x 2D 1 001XXXXX	10
0x 03 3 002XXXXX	01
0x 03 1 003XXXXX	10
0x 0F 0 003XXXXX	11
0x 0F 1 001XXXXX	10

Figure 2. Lookup table to implement memory protection

access and of its processing role. Each protection region is assigned to one of four levels of priority. In case of security violations, these are communicated to initiators and/or security controller.

4 Towards a secure on-chip communication infrastructure

In this Section, we outline some possible solutions to contrast some of the attacks described in Section 2. We believe that a complex communication infrastructure such as a NoC could support the design of a secure embedded system providing additional services to the standard communication tasks and we suggest its use as a possible mean to monitor and detect unexpected system behaviors. Therefore, as first step towards this goal, we outline several components that could constitute the building blocks of a more general Intrusion Detection System for a SoC adopting the NoC paradigm.

4.1 Address Protection Unit

Access to sensitive information in memory and configuration registers should be restricted to avoid buffer overflow or attacks aiming at stealing or modifying sensitive information. This can be achieved adopting an *Address Protection Unit* (APU) that enforces access control rules specifying how a component of the NoC can access the protected device. The APU, embedded in the target's NI, will supply services similar to those offered by a classical "firewall" in data networks. Implementations strongly depend on the protocol chosen to transmit the packets over the network, but they will be mainly based on a lookup table where to each entry is associated the access permission for a region in the address space.

Each entry in the table will be indexed by the concatenation of information transmitted in the header of the packet along with the request of access. The information univocally identifies the transaction. In Figure 2, a possible schematic implementation of an APU is shown. The identity of the initiator (SourceID), the address to which the initiator desires to access (MemAddr), the role assumed by the initiator (user, supervisor, secure mode, etc.) are used to lookup the access right (Load and/or Store) for the targeted memory block / register. For efficiency, only allowed accesses are recorded in the table. Moreover, *Don't care*s (X) are used to compact the table, grouping addresses in blocks.

4.2 Bandwidth guarantee

Quality of Service techniques can be used to reserve for each IP the desired bandwidth fraction, in order to contrast *Bandwidth Reduction attacks* due to malicious code running on a selected IP core and sending continuous and useless requests and data through the network. A possible solution to assure a fair utilization of the bandwidth can be achieved by using weighted round robin arbitration schemes [6] in each router, modified in order to be able to detect attempts to exceed the reserved quota and to signal the problem to a security manager core. Weights should be tuned in order to satisfy bandwidth requirements defined at design time, in order to avoid *false positives*. This is suitable for embedded systems, where for a specific use case the general behavior of the system in terms of data flow remains almost fixed during the entire lifetime of the device.

4.3 Security Automata

Security Automata were first proposed [9] in software engineering to model penetrations in the system as series of state changes, leading from an initial secure state to a target compromised state.

In NoCs, Security Automata could be employed for similar purposes, to monitor transactions requested by the IP cores and processed by the NI. In the hypothetical Security Automata shown in Figure 3, transactions monitored include information regarding destination and/or initiator, local address of the target IP and operation requested. Security Automata can be implemented at the initiator's or at the target's NI, depending on the scenario monitored. Behaviors likely to be monitored include known attacks carried out from/to IP cores in the system and specific and critical fixed routines, typical of the embedded system environment.

4.4 Intrusion Detection Systems

Intrusion Detection Systems (IDS), first introduced in software engineering to detect security violations [9], collect data from several monitors embedded in the system and analyze them to discover attempts of attack.

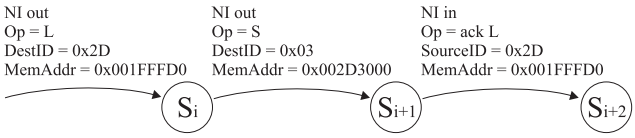

Figure 3. A hypothetical Security Automata monitoring transactions at the initiator's NI

Focusing on the communication channel, we outline here the implementation of a security framework collecting data from monitors embedded in the NIs or in routers located in critical positions in the NoC. As outlined in the previous sections, characteristics or activities likely to be monitored are:

- *Buffer occupancy:* number of elements in the buffers should be monitored in order to detect anomalous behavior of traffic in the network, caused for instance by Denial of Service attacks. Deviations from behaviors expected at design time will raise alarm signals.

- *Anomalous behavior of power manager:* NoC-based systems adopting power management blocks [11] can be subjected to *Sleep Deprivation attacks*, aiming at keeping the core active and working at full processing speed, or to *Thermal attacks*, aiming at overheating the core and causing malfunctioning in the system. Also in these cases, deviations from the statistical model of the ideal working conditions (size of queues, number of requests per time window, foreseen maximum full-speed working time, etc.) should be monitored in order to raise a warning flag of security violation.

- *Unauthorized access to secure memory locations:* an *Address Protection Unit* module can be employed to deny access to restricted parts of memory to unauthorized entities. In case of such attempts, the identity of the attacker can be easily discovered, raising an alarm and potentially isolating the "malicious" IP from the system, "sealing" the initiator's NI.

- *Violation of execution of critical routines:* the execution of critical programs, routines or known attacks can be monitored by using Security Automata. Also in these cases, unexpected behaviors will raise security alarms.

A *Security Manager Core* will take care of managing all the warning signals coming from the different monitors in the system. Type of monitor, attacker (initiator) identity, target IP are examples of the information identifying an attack. The core can be implemented in hardware, software, or a combination of the two, depending on the flexibility that the system should have. An established Trusted Computing Base, consisting of a secure kernel running on the main processor, is needed to safely configure the core, detect and elaborate security violations and react to the attacks with the appropriate countermeasures.

5 Conclusions and future work

In this paper we addressed the problem of security in embedded devices, with particular emphasis on systems adopting the Networks-on-Chips paradigm. We analyzed the state of the art of academic and industrial solutions, proposing the introduction of a series of modules that could be used to increase the level of security in NoC-based systems. The work represents a first step towards the exploitation of the intrinsic characteristics of the communication system to monitor the general behavior and detect attempts of attack. In future work, further investigation will be necessary to evaluate the right trade off between the security services provided, the performance of the system and the overhead in terms of area, energy consumption and cost.

References

[1] *SonicsMX SMART Interconnect Datasheet*. http://www.sonicsinc.com.

[2] T. Alves and D. Felton. *TrustZone: Integrated Hardware and Software Security, White Paper*. ARM, 2004.

[3] L. Benini and G. De Micheli. Networks on chips: A New SoC Paradigm. *IEEE Computer*, 2002.

[4] E. Chien and P. Szoe. *Blended Attacks Exploits, Vulnerabilities and Buffer Overflow Techniques in Computer Viruses*. Symantec White Paper, Sept. 2002.

[5] J. Coburn, S. Ravi, A. Raghunathan, and S. Chakradhar. SECA: Security-Enhanced Communication Architecture. In *Proceedings of the 2005 International Conference on CASES*, 2005.

[6] V. L. Do and K. Yun. An Efficient Frame-Based Scheduling Algorithm: Credit Round Robin. In *Workshop on HPSR*, 2003.

[7] S. Evain and J. Diguet. From NoC security analysis to design solutions. In *IEEE Workshop on Signal Processing Systems Design and Implementation*, pages 166–171, 2005.

[8] C. H. Gebotys and R. J. Gebotys. A framework for security on Noc technologies. In *Proceedings of the Annual Symposium on VLSI*, pages 113–117. IEEE Computer Society, Feb. 20-21 2003.

[9] K. Ilgun, A. Kemmerer, and P. A. Porras. State Transition Analysis: A Rule-Based Intrusion Detection Approach. *IEEE Transactions on Software Engineering*, Mar. 1995.

[10] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, and S. Ravi. Security as a New Dimension in Embedded System Design. In *Proceedings of DAC 2004*, Jun. 7-11 2004.

[11] T. Simunic, S. P. Boyd, and P. Glynn. Managing power consumption in Network on Chips. *IEEE Transactions on VLSI Systems*, 2004.

0-7695-2978-X/07 $25.00