Distributed security for communications and memories in a multiprocessor architecture

Pascal Cotret, Jérémie Crenne, Guy Gogniat, Jean-Philippe Diguet, Lubos Gaspar, Guillaume Duc

2011 IEEE International Parallel & Distributed Processing Symposium

Distributed security for communications and memories in a multiprocessor architecture

Pascal Cotret*, Jérémie Crenne*, Guy Gogniat*, Jean-Philippe Diguet*, Lubos Gaspar† and Guillaume Duc‡
*Laboratoire Lab-STICC, Université de Bretagne-Sud, Lorient (France)
name.surname@univ-ubs.fr
†Laboratoire Hubert-Curien, Université Jean-Monnet, Saint-Etienne (France)
lubos.gaspar@univ-st-etienne.fr
‡Département COMELEC, Télécom ParisTech, Paris (France)
guillaume.duc@telecom-paristech.fr

***Abstract*—The need for security in embedded systems has strongly increased since several years. Nowadays, it is possible to integrate several processors in a single chip. The design of such multiprocessor systems-on-chip (MPSoC) must be done with a lot of care as the execution of applications may lead to potential vulnerabilities such as revelation of critical data and private information. Thus it becomes mandatory to deal with security issues all along the design cycle of the MPSoC in order to guarantee a global protection. Among the critical points, the protection of the communications is very sensible as most of the data are exchanged through the communication architecture of the system. This paper targets this point and proposes a solution with distributed enhancements to secure data exchanges and to monitor communications within a MPSoC. In order to validate our contribution, a case study based on a generic multiprocessor architecture is considered.**

***Keywords*-cryptography, security, communication architecture, monitoring, firewalls**

I. Introduction

Nowadays, computers, smartphones, set-top boxes and other electronic embedded systems are part of our daily life. They contain critical information such as passwords and personal data that must be protected against potential attackers. These devices are characterized by several processing elements embedded in a single chip in order to perform advanced computations. The security mechanisms built in these systems must be efficient and evolutive in order to be up-to-date with the most recent standards of security protocols and algorithms.

As these systems have a complex architecture, an extensive analysis of their structure and their functionalities must be carried out in order to guarantee an efficient protection. Furthermore, a tradeoff between the latency and area overheads of the protection mechanisms is mandatory. This is a critical issue as the final system must be competitive in terms of development and manufacturing costs but also resistant to attackers who want to retrieve sensitive data. Reconfigurable technologies such as FPGAs are widely used to build such complex embedded systems as they can integrate processors, memories, dedicated IPs. All these components are relying on a communication architecture where all data are exchanged. Thus, it is of paramount importance to protect communications in order to keep secret and private information within the system. This is the goal of this work and we propose a distributed solution based on firewalls between IPs and the communication architecture.

II. Related work

Several works have targeted the protection of communication architectures within an embedded system. Coburn et al. [1] propose a solution dedicated to buses. Their technique is based on security elements (SEI, Security Enforcement Interface) implemented in each interface between IPs and the bus. Each SEI computes information from the data handled by its associated IP and sends it to a global manager (SEM, Security Enforcement Module). The SEM manages the security of the system and controls all SEIs in order to apply the right level of protection. Other works are devoted to NoC (Network on Chip). Evain et al. [2] propose a solution similar to [1] where controls are done in each interface. A global manager gathers all information from the interfaces and deal with the security policy. Another approach uses the interfaces as filters splitting the IPs address map into zones with specific security policies [3]. This interface called *Adress Protection Unit*, verifies if the requested address is allowed or not. An extension to this work adds probes within the interface structure to refine the protection mechanisms [4].

In our work we target a bus-based system where a limited number of IPs are connected together [5] [6]. Compared to previous efforts we propose to use a distributed solution to secure the communications in a MPSoC architecture. The main idea is that each interface manages the security policy corresponding to its associated IP.

III. Threat model

When dealing with security, one of the first tasks that must be addressed is to define the threat model as it has a direct impact on the countermeasures that must be defined.

1530-2075/11 $26.00 © 2011 IEEE
DOI 10.1109/IPDPS.2011.158

In this section we propose the generic model of attacks that we consider in this work.

A. Attacker's goals and possibilities

We consider that potential attackers have several goals:

- Processor hijacking: running a malicious source code on a processor to misbehave the whole embedded system.
- Extraction of secret information: cryptographic keys, security parameters, private information...
- Denial of service (DoS): cancelling out security services to stop the system, disabling communications, injecting dummy data to create overwhelming traffic.

B. Attack vectors

In order to achieve his goals an attacker has several solutions that must be clearly analyzed. One important point is to define how an attacker can have access to the system. In our case, we consider logical attacks and do not target fault injection or side-channel attacks.

We consider the FPGA as secure so the only way for an attacker to tamper with the system is through the external bus and the external memory. Indeed the considered architecture is composed of an FPGA connected to an external memory that contains data and code. The FPGA embeds several processors, internal memories and dedicated IPs. When critical systems are targeted, external memory is encrypted and authenticated. In that case, it is impossible to tamper with the external memory. However such a solution has a very high cost, so many systems do not provide a uniform protection but allow some parts of the memory to be unprotected or only ciphered. The unprotected memory corresponds to a non sensitive part of the system, however an attacker can take benefit of this non protected area to introduce his attack within the system. When the memory is only ciphered it is more difficult for an attacker but he can still target a DoS attack by randomly changing some data.

In a more general way, when targeting the external bus, an attacker can perform replay, relocation and spoofing attacks. This model is widely used and covers major threats on the external bus. When targeting the external memory, an attacker can modify some code or data that will be executed by one of the processors within the system. This execution may lead to get access to sensitive data or to obtain abnormal behavior of the system. One way to track such attacks is to check all communications within the system to detect any unauthorized access to the communication architecture. This is the contribution that we propose.

C. Security features

In order to cover the considered threat model and to obtain an efficient solution, we target three security features. We believe it is possible to secure a system by monitoring the communications in order to check if any abnormal or unauthorized access to the communication architecture is performed:

- If an error is detected (we will see how later in the paper), the system must react as fast as possible. This is important to be able to stop an attack before it succeeds to extract or modify some data.
- Attacks must not affect the global behavior of the system. If an attack is detected, the goal is to limit its impact to the IP that launches the attack. For that purpose, the attack must not reach the communication architecture but be stopped in the interface associated with the infected IP.
- The security enhancements should achieve a good area/latency tradeoff. This is very important for embedded systems where constraints are tight in terms of area, power and energy.

IV. Security enhancements for a multiprocessor architecture

The considered system as mentioned previously is composed of processors, internal memories, dedicated IPs embedded within an FPGA and connected to an external memory. All the resources within the FPGA are connected to a bus. We propose to add a specific interface to each resource in order to build a secure gateway to the bus. That way, we can monitor all communications before they reach the bus and propagate within the system. Figure 1 shows a system with two resources and an external memory. Each resource is connected to a specific interface called Local Firewall. The external memory is also connected to a specific interface called Local Ciphering Firewall. These firewalls handle the security policy within the system as explained in the following.

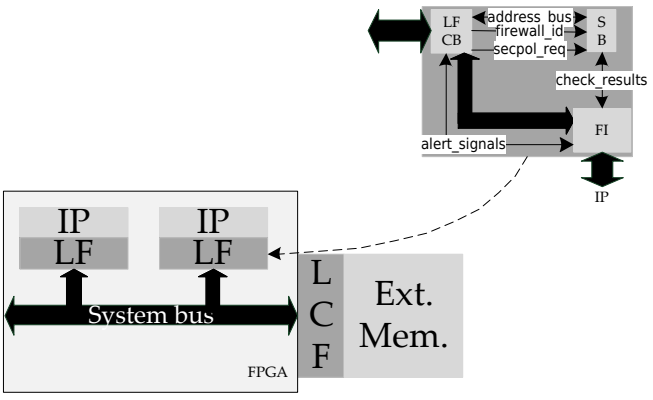

Figure 1. Embedded distributes architecture with security enhancements

A. Security policy

A Security Policy (also known as SP) is a set of parameters that aims to protect the system against the considered threat model. Each resource (processor, dedicated IP) has a

specific SP which defines the associated security rules. We consider the following parameters:

- **SP Identifier (SPI)**. Identifier of the security policy.
- **Read/Write Access rules (RWA)**. Three choices are available for data accesses: read-only, write-only or read/write. This information is used in order to check if the current access to the bus by an IP is authorized.
- **Allowed Data Format (ADF)**. There can be several data lengths allowed for a given security policy (for instance, 8 up to 32 bits). This information is used in order to check if the accessed data has the correct format. An unauthorized format may overwrite some protected data in the target IP.
- **Confidentiality and Integrity Modes (CM and IM)**. For each option, it is possible to execute or to bypass confidentiality (block cipher) and integrity (hash trees) modules. These parameters are only available for the Local Ciphering Firewall that is connected to the external memory. Indeed we consider that all internal communications are not encrypted as the Local Firewalls protect them against unauthorized access.
- **Cryptographic Key (CK)**. This parameter corresponds to the key used by the block cipher module. This parameter is only available for the Local Ciphering Firewall that is connected to the external memory.

These parameters allow us to cover the threat model presented previously. Indeed replay, relocation and spoofing attacks are managed by the Local Ciphering Firewall using ciphering and integrity. Time stamp tags are also used to monitor the access time to the external memory (replay attacks). Memory addresses are controlled to protect the system against relocation attacks. Protection within the FPGA is managed by the Local Firewalls that monitor all accesses to the bus preventing any unauthorized or abnormal access to the internal bus. This technique allows to detect any tampering within the system.

B. Features of Firewalls

Considering the system presented in Figure 1, and as presented before there are two categories of Firewalls: Local Firewalls are located at the interface between an IP and the bus, Local Ciphering Firewall is connected between the internal bus and the external memory.

1) Local Firewall: Local Firewalls monitor the communications using the security parameters presented above. For a write operation, before reaching the bus all data are checked. If the security rules are respected the data can be sent to the bus. For a read operation, all data are checked before reaching the IP. As for the write operation, if the security rules are respected the IP can receive the data. In case there is a violation of one of the security rules, the data is discarded.

The Security Policies (SP) associated to a Local Firewall are stored in on-chip memories: these memories (called Configuration Memories) are considered as trusted units and do not need to be ciphered. Each Local Firewall is composed of the following resources:

LF Communication Block (LFCB): It is responsible for receiving and transmitting bus signals through a dedicated interface. It launches the security rules checking by triggering the `secpol_req` signal.

Security Builder (SB): When the `secpol_req` signal is received by SB, it reads the associated SP from the Configuration Memory. Then, SP parameters (security rules) are sent to specific checking modules that are embedded in the SB resource.

Firewall Interface (FI): This block achieves the communication datapath between the bus and the IP. It takes into account some alert signals to allow (or not) data block to be transmitted.

2) Local Ciphering Firewall: Local Ciphering Firewall (LCF) monitors the exchanges between internal IPs and the external memory. The main feature of LCF is the protection of the external memory in terms of confidentiality and integrity (each feature has its own module in the LCF internal architecture). The architecture of the Local Ciphering Firewall is similar to the LF one except the ciphering and integrity modules that are not available in Local Firewall.

Confidentiality Core: This module is responsible for ciphering operations. This core is based on a AES (Advanced Encryption Standard) algorithm with 128-bits key.

Integrity Core: This module is based on hash-trees.

Figure 1 shows the connection between the IPs and the Local Firewall and the Local Ciphering Firewall. The internal architecture of the Local Firewall is also detailed. We can see the various resources used to manage the security within the interface.

V. Implementation results

The considered architecture has been implemented on a ML605 development board from Xilinx (with a XC6VLX240T1136-1 FPGA). This device has around 240,000 logic cells and 15 Mb RAM blocks. In order to evaluate the performances of the proposed solution, a study in terms of consumed area and observed latency on the FPGA is proposed.

This work discusses the implementation results of our security enhancements within a multiprocessor architecture. This system contains 3 MicroBlaze softcore microprocessors, One internal shared memory (BRAM blocks), one external memory (DDR RAM) and one dedicated IP.

A. Synthesis results

In order to estimate the area overhead added by the firewalls, we compare the area occupied by the system without and with firewalls. Results are given in Table I.

The overheads given in the second line, show the costs in terms of area due to the firewalls for the considered

		Slice Regs	Slice LUTs	F. used LUT FFs pairs	# BRAMs
Generic w/o firewalls		12,895	11,474	15,473	53
Generic w/ firewalls		15,833 +13.43%	19,554 +34.40%	21,530 +26.50%	63 +18.87%
Local Firewall	**SB**	0	393	393	0
	CC	436	986	344	10
	IC	1,224	1,404	1,704	0
Local Firewall		8	403	403	0

Table I
SYNTHESIS RESULTS OF THE MULTIPROCESSOR SYSTEM

case study. We can observe an overhead of 13.43% for the slice registers. This cost is not negligible however as expected most of the area is devoted to the confidentiality and Integrity Cores (about 90% of Local Ciphering Firewall area). The interesting point is that the cost of Local Firewalls is limited. The cost of firewalls is also related to the number of security rules that must be monitor. A more aggresive security policy will lead to a larger cost in terms of area. This point will be further analyzed in future work.

Table II shows the overhead in terms of latency due to the various modules within the firewalls. We can observe that the security rules checking requires 12 cycles, the ciphering operation 11 cycles and the integrity checking 20 cycles. The impact of the protection mechanisms on the global execution time depends on the percentage of computation time versus communication time. Furthermore the latency overhead is also impacted by the percentage of internal communication versus external communication (i.e. communication with the external memory). Indeed external communications have a larger overhead due to the cryptography resources so promoting internal computation and communication will improve the overall performance.

		Nb. of clk cycles	Throughput (Mb/s)
(LF/LCF)	**SB**	12	-
	CC	11	450
	IC	20	131

Table II
LATENCY RESULTS OF THE FIREWALLS

The implementations results underline some interesting points:

- Most of the controls are done locally within the firewalls: it implies a low latency overhead for the communication.
- The implementation of a firewall at each IP interface leads to a limited area overhead. This is not the case for the area of the Local Ciphering Firewall that is rather high due the confidentiality and integrity cores.

VI. CONCLUSION AND PERSPECTIVES

This paper provides a solution to secure communications within a multiprocessor architecture. This work offers distributed security enhancements that embed protection mechanisms where existing solutions are mainly centralized. Our solution is a layer above the communication protocol as our security enhancements do not modify the communication protocol between the processors and the other parts of the systems. The application designer does not have to deal with the security mechanisms as they are embedded within the hardware architecture. The latency and area costs will be further analyzed in our future work in order to provide an extensive discussion of costs in relation with security rules and security policies. We also plan to integrate reconfiguration of security services (i.e. modification of security policies) to counter some attacks against the system. In this work, policies are defined using the address spaces, it can be interesting to study the adaptation to thread-specific security where each thread has its own security level.

ACKNOWLEDGMENT

The work presented in this paper was realized in the frame of the SecReSoC project number ANR-09-SEGI-013, supported by a grant of the French National Research Agency (ANR).

REFERENCES

[1] J. Coburn, S. Ravi, A. Raghunathan, and S. Chakradhar, "Seca: security-enhanced communication architecture," in *CASES '05: Proceedings of the 2005 international conference on Compilers, architectures and synthesis for embedded systems*. New York, NY, USA: ACM, 2005, pp. 78–89.

[2] J.-P. Diguet, S. Evain, R. Vaslin, G. Gogniat, and E. Juin, "Noc-centric security of reconfigurable soc," in *NOCS '07: Proceedings of the First International Symposium on Networks-on-Chip*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 223–232.

[3] L. Fiorin, C. Silvano, and M. Sami, "Security aspects in networks-on-chips: Overview and proposals for secure implementations," *Digital Systems Design, Euromicro Symposium on*, vol. 0, pp. 539–542, 2007.

[4] L. Fiorin, G. Palermo, and C. Silvano, "A security monitoring service for nocs," in *CODES+ISSS '08: Proceedings of the 6th IEEE/ACM/IFIP international conference on Hardware/Software codesign and system synthesis*. New York, NY, USA: ACM, 2008, pp. 197–202.

[5] S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, "Security in embedded systems: Design challenges," *ACM Trans. Embed. Comput. Syst.*, vol. 3, no. 3, pp. 461–491, 2004.

[6] A. Mandal, S. Mandal, A. Tripathy, N. Gupta, and R. Mahapatra, "A bio-inspired framework for secure system on chip," in *1st Workshop on SoC Architecture, Accelerators and Workloads*, 2010.