Implementation of SIMON Block Cipher to Provide Confidentiality to a Network-on-Chip

Antonio F. Mellies Neto, Eduardo A. da Silva, Fabricio Bortoluzzi, and Cesar A. Zeferino
Laboratory of Embedded and Distributed Systems
University of Vale do Itajaí, Itajaí – BRAZIL
antonio.mellies@edu.univali.br,{eas, fb, zeferino}@univali.br

***Abstract*—The increasing amount of cores in the design of computer processors brought the need to develop scalable communication channels, while preserving high data rates between its components. The natural solution to address this problem was found by applying concepts associated with traditional computer networks, inside the chip, resulting in the advent of networks-on-chip. As consequence, networks-on-chip inherits similar vulnerabilities to those found in common networks, such as the lack of confidentiality. This work presents a solution to provide confidentiality by using SIMON block cipher algorithm performing in SoCIN network-on-chip. As result, it was observed an increase of latency in intra-chip communications due to the need for more clock cycles to complete each message exchange, justified as an acceptable trade-off by the layer of security provided.**

***Index Terms*—Networks-on-chip, encryption, confidentiality**

I. Introduction

The increasing amount of energy and limitations in performance in single-core processors carried out to the development of multi-core processors [1]. The ever-shortening dimensions and distance between transistors to few nanometers made viable the integration of entire computational systems inside a single silicon chip. These systems are entitled Systems-on-Chip (SoCs).

The need for ever increasing performance caused the building of SoCs with a large number of cores, allowing one chip to integrate from dozens to hundreds of these interconnected. This unified integration on a chip, named Multiprocessor System-on-Chip (MPSoC) permits the addition of components that enhance performance, such as embedded memory and special purpose processors to cope with video, audio, communications, and others [2].

In this scenario, it is paramount to provide solutions of scale to the implicit bottleneck imposed by direct communication between cores and tasks among the several units made available. The ideal communication infrastructure must provide high levels of communication rates and parallelism. The natural solution to this matter lies within intra-chip networks (NoCs – Networks-on-Chip). They follow the growth and sustain scale as other cores and elements are added to the chip. [3].

Landwehr [4] established that a system can be considered secure if it is able to preserve confidentiality, integrity, and availability. Security mechanisms are designed to enforce or comply with one or more specific needs, such as the protection against physical destruction, data exposure, corruption or modification, theft or any form of unauthorized access. The same applies to the design of NoCs. Traffic generated within any given NoC is, essentially, clear text. Anyone able to extract a portion of this traffic and interpret its header and body will be able to compromise confidentiality and, eventually, integrity.

This work proposes the use of SIMON block cipher algorithm in SoCIN as an effort to provide proper confidentiality, with the corresponding measurement of the correctness operation, impact on performance, and any other characteristic that may arise from its integration. SIMON is, among several other mechanisms, a suite of block ciphers made available by the National Security Agency – NSA of the United States of America, in 2013. The main goals of SIMON are: a) to be as lightweight as possible, and b) to be suitable for hardware implementation [5].

To run the experiment, we opted to use a 2D version of the SoCIN NoC [6], which is a configurable and *de facto* standard NoC for research purposes within the activities of the Laboratory of Embedded and Distributed Systems at the University of Vale do Itajaí (UNIVALI).

This paper is structured as follows. In Section II we present related research about security applied to NoCs. Section III describes our method and decisons taken to parametrize the NoC environment. Section IV describes how the simulation took place, followed by a discution on the results in Section V. Section VI presents our conclusions and future works.

II. Related Work

The use of related techniques to address security in NoCs was already discussed in other works. Gebotys and Gebotys [7] were the first to present a solution using symmetric cryptography, focusing on avoiding eavesdropping on packets inside the network. Baron [2] proposed a mechanism to provide resilience against DoS (Denial-of-Service) attacks in SoCIN. Silva [8] implemented a solution based on AES (Advanced Encryption Standard) to provide both confidentiality and authenticity.

SIMON centric solutions were implemented outside the context of NoCs by Wetzels and Bokslag [9] for Xilinx Spartan-6, for a collection of architectures and combinational circuits. Costa et al. [10] compared the performance of SIMON and SPECK algorithms in FPGA (Field-Programmable Gate

Array). Sopran and Melo [11] also evaluated performance and cost of SIMON in FPGA.

This work is part of a broader research that aims to provide solutions for the most common security concerns of data exposure when being transmitted inside a Network-on-Chip.

III. System Architecture

A. Topology

The network topology used to the development of this work was a 2D, 3×3 mesh, represented in Fig. 1. It illustrates a grid where each CPU core has a network interface (NI) to provide communication between the core and the router.

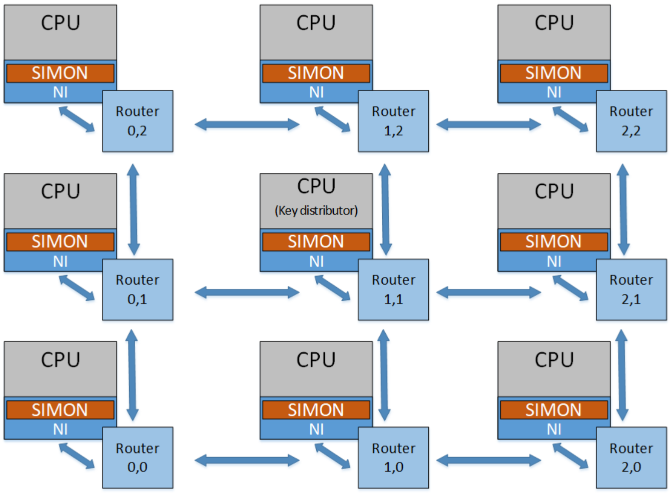


Fig. 1. Network topology for development.

Each network interface received a pair of encrypting and decrypting functions of SIMON algorithm. The router located at position [1,1] was locally connected to the key distributor. It is responsible to deliver keys to the cores involved in each message exchange.

B. Communication Flow

In order to establish a communication, both cores are required to obtain an encryption key first. Then, the key distributor answers with a key that must be used in that transmission. Fig. 2 depicts an example in which Core A asks for a key. The Key distributor then takes two actions: (*i*) answers Core A with a key; and (*ii*) sends a message to Core B containing the same key, which it will use to decrypt the message.

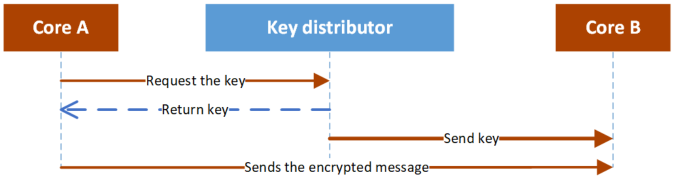


Fig. 2. Communication flow.

We assume enough fulfillment of the confidentiality premise because every transmission from/to the Key distributor is encrypted with the use of a pre-shared key exclusive, different and hardcoded to each core. This approach prevents any node in the path to be able to decrypt and interpret any part of the payload.

C. SIMON

SIMON offers several choices on the block size used in its cryptographic algorithm. We decided to apply the SIMON32/64 variant, meaning that the largest input block is 32-bit long and the correspondent key is 64-bit long. So, in practice, we are limited to messages no longer than 32 bits.

D. Packet Format

Each core creates its own packets, taking into account the set of parameters given at the beginning of the simulation. Fig. 3 shows its structure. Each packet must have a header and a tail flit, which represent the begin and the end of the packet, respectively. The number of flits that compose the payload is adjustable as one of the initial parameters.

The header is composed of the following fields: a flit type, an information flag and the addresses of the sender and receiver nodes. The information flag is useful to extend header and to allow new kinds of data. To specify the type of one flit, it is necessary to fill the two most significant bits with a BOP (Begin-of-Packet) and an EOP (End-of-Packet) bits. Header length is filled in the third bit to inform if the next flit is or is not part of the header. Sender and receiver's addresses are filled in the next set of 16 bits of the flit. It was not necessary to modify this structure to develop our solution.

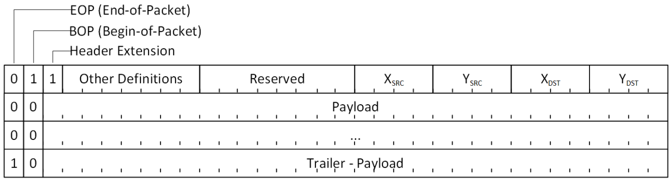


Fig. 3. Packet format.

When a core needs to send an encrypted packet, it is necessary to obtain an encryption key from the key distributor first. Such request is accomplished by sending the proper packet to the distributor, as Fig. 4 depicts. This packet is arranged with a header flit and a body flit, in which it is written the receiver's address, followed by an end packet.

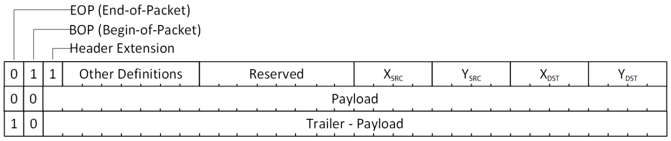
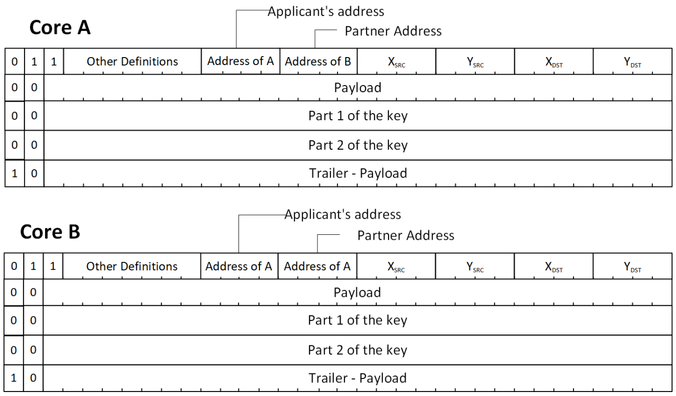


Fig. 4. Key request packet format.

The key distributor creates the packet, identified in the first flit, containing the header, the payload, and the receiver address. Fig. 5 illustrates an example of a packet sent by the key distributor to both endpoints of communication. In the example, Core A asked for a key from the distributor to establish a communication with Core B. The payload is filled solely with the required key for encryption, because the receiver address is already written in the header. SIMON instances in both sender and receiver will look for this information in order to decide where to store keys for encryption and decryption.



Core A

Applicant's address
Partner Address

0	1	1	Other Definitions	Address of A	Address of B	X_{SRC}	Y_{SRC}	X_{DST}	Y_{DST}
0	0	Payload							
0	0	Part 1 of the key							
0	0	Part 2 of the key							
1	0	Trailer - Payload							

Core B

Applicant's address
Partner Address

0	1	1	Other Definitions	Address of A	Address of A	X_{SRC}	Y_{SRC}	X_{DST}	Y_{DST}
0	0	Payload							
0	0	Part 1 of the key							
0	0	Part 2 of the key							
1	0	Trailer - Payload							

Fig. 5. Format of packages sent by the key distributor.

IV. Experimental Setup

The simulation was configured to generate packets exclusively between two cores. In order to evaluate any impact on the placement of the key distributor, we decided to place it in three different positions, Fig 6 depicts.

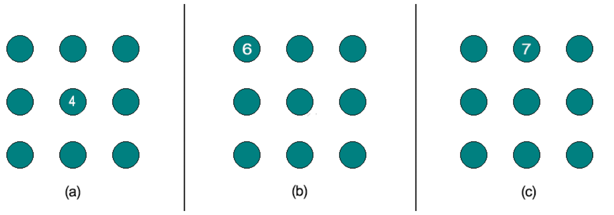


Fig. 6. Distributor position on tests: (a) centralized; (b) at a corner; (c) between two corners.

The choice of which cores should act as a sender or receiver was assorted. This allowed measuring different distances in terms of hops needed to be covered by packets. Fig 7 exhibits all possible distances given the 2D Mesh of 3×3.

Simulations were conducted varying the number of packets from 1 to 3. We consider a communication as finished only after the last packet is entirely delivered to the receiver.

V. Results

In order to measure the impact caused by the overhead of cryptography, we used the NoC-based simulation environment RedScarf [12]. In Redscarf, it is possible to specify every condition needed to replicate the behavior of an NoC. Once fed with the proper parameters, the simulation takes place and a graphical result is generated, alongside with all detailed, round based, raw data about what happened inside the NoC.

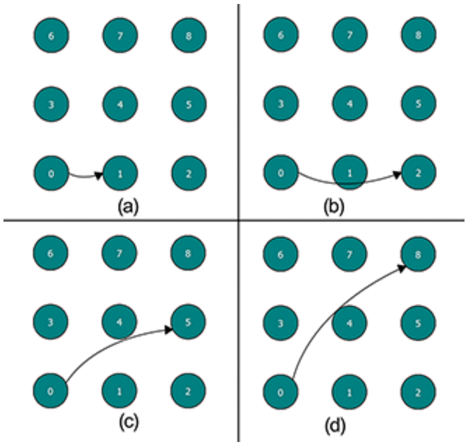


Fig. 7. Possible distances: (a) one hop; (b) two hops; (c) three hops;(d) four hops.

Results reported in this work are for packets exchanged between only two cores each time. These packets were sent according to four possible distances in the network (as of Fig 7). Core 0 was the sender in every case.

We set parameters to establish a baseline latency, without and with cryptography. Waveforms were generated for the same purpose, in order to measure the number of cycles imposed by both versions of the NoC. The topologies chosen were 2D Mesh and 2D Torus, essentially because they support the 3×3 network model proposed.

When positioned in the center of the network, the key distributor added an average latency of 134% over the same communication without encryption. The routing algorithm imposed no further significant latency in this scenario, essentially because the key distributor was placed in the center of the network.

When positioned in one of the corners, the key distributor was measured both when sending one packet and when sending two packets. Destinations were always four hops away. 2D Mesh simulations showed an increase of 42% of extra initial latency when SIMON was enabled. 2D -Torus added 62% under the same conditions.

When the key distributor was set between two corners of the network, results showed consistency in relation to the average latency in communications that involve more than one hop. When sending only one packet, and compared to communications that require more hops, we noticed an increase in latency of 12.5% in 2D Torus and 9.7% in the 2D Mesh.

There was no significant impact in the baseline average latency among the different mesh topologies, the positioning of the key distributor, and the number of packets sent, except for 2D Mesh, which imposed a 9.3% higher latency.

When enabling cryptography to transmit one packet, latency increased 115.8% on 2D Mesh and 125.5% on 2D Torus.

When sending three packets, results were 132.5% and 157% respectively.

As shown in Fig 8, waveform analysis demonstrated that unencrypted data transmission started in the 13th clock cycle and ended in the 15th. Receiving started in clock cycle number 19 and ended at the 21st cycle.

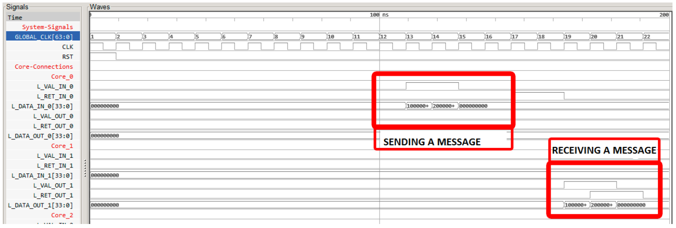


Fig. 8. Waveform analysis for one communication between a pair of cores in clear-text

With SIMON enabled, the sender needs to obtain a cipher key prior to sending the packet. Then, it is necessary to wait for the key, sent by the key distributor to both terminals. As in the example, this task finishes only at clock cycle 49, demanding 23 more cycles to complete. This means an increase in latency by 176%. This dialog is presented in Fig 9.

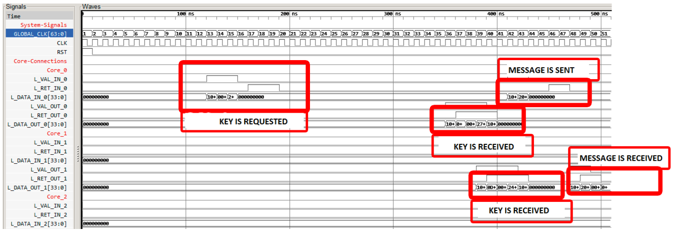


Fig. 9. Waveform analysis for one communication between a pair of cores encrypted with SIMON

VI. Conclusions

This work presented a proposal to use SIMON block cipher algorithm to provide a basic layer of encryption in data exchange for SoCIN. We argue the need for security mechanisms to be applied to NoCs with emphasis on a solution able to cause as low as possible impact in performance on SoCIN NoC.

The results obtained from simulations allowed precise measurement of extra latency imposed due to additional clock cycles to operate encryption and decryption of data. This impact, while not desired, is considered acceptable when the need for confidentiality surpasses the need for performance.

For future works, we foresee the need to investigate SIMON's behavior in more complex topologies, such as the 3D Mesh. Also, we recommend the research towards making SoCIN resistant to other kinds of attack.

References

[1] N. Jerger and L. Peh, "On-chip networks-synthesis lecture on computer architecture," *Morgan & Claypool Publishers*, 2009.

[2] S. Baron, M. S. Wangham, and C. A. Zeferino, "Security mechanisms to improve the availability of a Network-on-Chip," in *2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS)*, Dec 2013, pp. 609–612.

[3] E. A. Carara, "Serviços de comunicação diferenciados em sistemas multiprocessados em chip baseados em redes intra-chip," 2011. [Online]. Available: http://hdl.handle.net/10923/1593

[4] C. E. Landwehr, "Computer security," *International Journal of Information Security*, vol. 1, no. 1, pp. 3–13, 2001.

[5] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK families of lightweight block ciphers. cryptology eprint archive, report 2013/404, 2013."

[6] C. A. Zeferino and A. A. Susin, "SoCIN: a parametric and scalable Network-on-Chip," in *16th Symposium on Integrated Circuits and Systems Design, 2003. SBCCI 2003. Proceedings.*, Sept 2003, pp. 169–174.

[7] C. H. Gebotys and R. J. Gebotys, "A framework for security on NoC technologies," in *VLSI, 2003. Proceedings. IEEE Computer Society Annual Symposium on*. IEEE, 2003, pp. 113–117.

[8] M. R. Silva and C. A. Zeferino, "Confidentiality and authenticity in a platform based on Network-on-Chip," in *2017 VII Brazilian Symposium on Computing Systems Engineering (SBESC)*, Nov 2017, pp. 225–230.

[9] J. Wetzels and W. Bokslag, "Simple SIMON: FPGA implementations of the SIMON 64/128 block cipher," *arXiv preprint arXiv:1507.06368*, 2015.

[10] C. Costa, F. D. Roberto, Pereira, E. D. Moreno, and F. M. O. Tachibana, "Análise de metodologias de implementação e desempenho em FPGA dos algoritmos criptográficos leves Simon e Speck," in *Anais do WoCCES 2016 Workshop de Comunicação de Sistemas Embarcados Críticos*. SBC, 2016, pp. 1–11.

[11] R. Sopran, D. R. Melo, C. A. Zeferino, and E. A. Bezerra, "Análise comparativa do custo e do desempenho de um algoritmo de criptografia para sistemas embarcados explorando o particionamento hardware/software," *Anais do Computer on the Beach*, pp. 259–268, 2017.

[12] E. A. da Silva, D. Menegasso, S. V. Jr., and C. A. Zeferino, "Redscarf: A user-friendly multi-platform network-on-chip simulator," in *2017 VII Brazilian Symposium on Computing Systems Engineering (SBESC)*, Nov 2017, pp. 71–78.