Analysis and design of FPGA-based hardware accelerators oriented to security applications in IoT

Yubal B. Alfaro, Pedro P. Carballo, Antonio Núñez
IUMA, Institute for Applied Microelectronics, University of Las Palmas de Gran Canaria, Spain
{ybarrios,carballo,nunez}@iuma.ulpgc.es

***Abstract*— this work comprises the design of a Fog node and its implementation on a programmable FPGA-based MPSoC for secure communications in an IoT environment. The design has been made keeping in mind the Fog Computing latency and power consumption requirements. This system is comprised of an IP block based on the Simon algorithm that decrypts the Ethernet frames received by the Fog node and a second IP, a Counting Bloom Filter that analyses the Ethernet packet header at the network and transport level layer to detect potential threats that can jeopardize the system. Finally, we implemented the complete system over a Xilinx Zynq XC7Z020-CLG484-1 EPP, obtaining a maximum work frequency of 150 MHz and a throughput of 200 Mbps. Low power optimization strategies are applied, making the fog node ideal to operate in these environments. The node itself consumes 19.83% of the available slices of the device.**

Keywords- Fog computing, IoT, security, FPGA, Simon, Bloom Filter.

I. INTRODUCTION

Nowadays, the number of devices connected to the Internet is constantly growing at an annual rate of 8%. Much of this growth is due to the rise of the Internet of Things, whose architecture is based on the constant connection to the network of many devices and sensors [1]. It is estimated that by 2020 the number of devices connected to the Internet will be almost 50 billion, which is 6 per person [2].

The large volume of data generated is interesting to cybercriminals, who have more sources of personal information. In fact, this type of criminal activity constitutes 67% of all cyber threats worldwide, above cyber spying or hacktivism [3]. A relevant case is the DDoS-type cyber-attack suffered by the Dyn server (which provides service to Spotify or Twitter, among others) in October 2016 by the massive sending of Ethernet packets from IoT devices, mainly IP cameras, rendering the server unusable [4].

A new technology like IoT needs improvements regarding the connectivity, latency and security techniques. In this way, Cisco introduces in 2016 the Fog Computing (Fig. 1) concept [5], based in move cloud resources to a distributed layer near the endpoints, with the goal to reduce the latency of critical information analysis and to provide a dedicated security policy to all IoT devices [6].

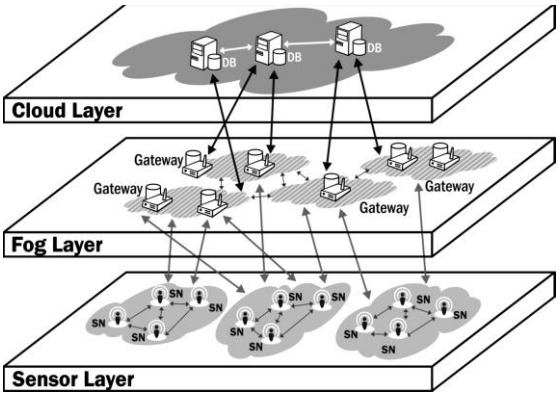

Fig. 1. Fog Computing Model.

Due to its recent development, Fog Computing suffers security issues, which requires the development of systems that ensure the privacy of the transmitted information.

II. PROPOSED ARCHITECTURE

Keeping this goal in mind, we a design a Fog nod with an authentication phase that analyses the header of the Ethernet packets received and another phase that decrypts the payload (Fig. 2).

The filters inspect the header at a network and transport level of the OSI model, focusing on the most used protocols at these levels in IoT architecture (IPv6 and UDP, respectively). For this purpose, we use a Counting Bloom Filter-based structure, a statistical method that estimates if an element is included in a specific group (Fig. 3).

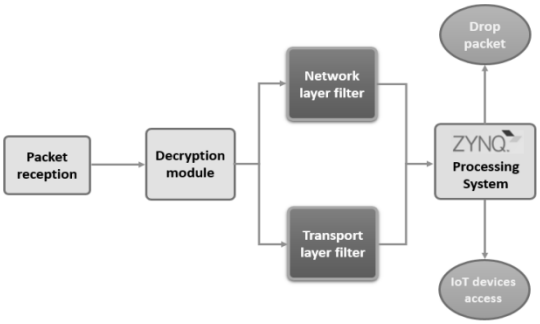

Fig. 2. System architecture.

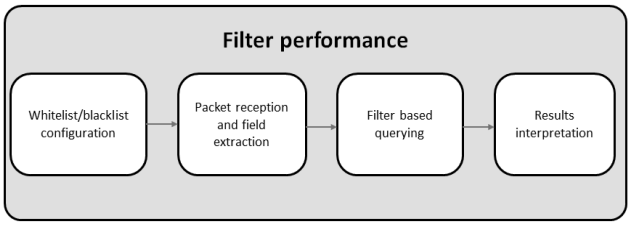

Fig. 3. Filter phase performance.

The main functions of the filter are element insertion, elimination and query. The main disadvantage of this structure is the existence of false positives, although its value is normally less than the 0.1% and proportional to the numbers of elements that the filter includes and the size of the array *m* [7]. For the query method, it is necessary to extract previously the relevant header fields to compare their values with the mapped elements that exist in the filter. If the filters return a positive response (0 value), which means that the element has been found, the connection is blocked.

The decryption module is based on the Simon lightweight block filter, developed by the NSA in 2013 with the aim of creating secure systems for IoT applications with a minimum memory footprint and a reduced power consumption [8]. It is based on the Feistel networks, an encryption methodology that applies binary operations at each round between an input word and a symmetric key defined previously. Being this process reversible, deciphering the data it is only necessary to know the unique key and begin to iterate the information from the last word.

The configuration used in this work is a 64 bits word size with a key comprised by a 32-bit words array with size 42. To finish the algorithm with enough robustness it is necessary to apply 42 rounds to every input word. To apply the main algorithm, the header section of the packet must be extracted as to avoid its encryption. Packet headers must not be encrypted as the packet would not reach its destination if the routing information is stored in it in a cyphered manner. Afterwards, a key expansion phase that generates the entire array values providing only the first three words. Once the main function is executed, the resulting message should be copied on the AXI4-Stream output interface of the IP block to send it to the processing system (Fig. 4). The PS, the main module of the system, will carry out on that message the action it deems appropriate.

It should be noted that the decryption stage of the received payload will only be carried out in the case that neither filter returns a positive result on the header analysis. Otherwise, the package is discarded.

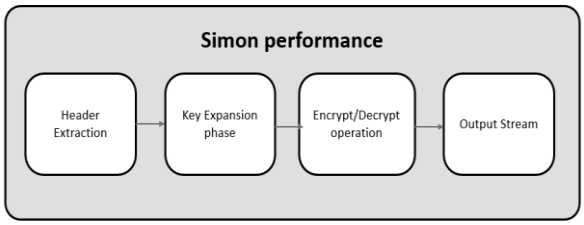

Fig. 4. Encryption module performance.

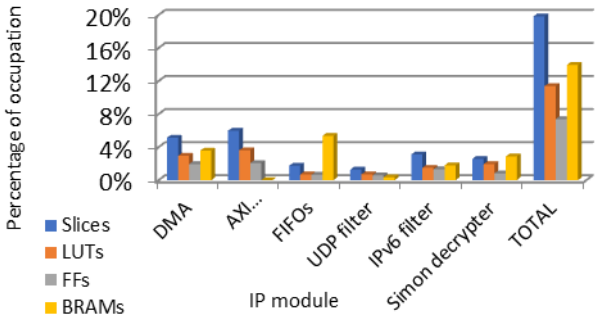

Fig. 5. System utilization.

The mentioned IP blocks have been designed following a High-Level Synthesis methodology that allows modeling the functionality with a high level of abstraction using the C/C++ programming language. The complete architecture includes, in addition to the aforementioned custom modules, a DMA (*Direct Access Memory*) to transfer the packets stored in the off-chip memory to the rest of the system, an AXI4-Stream Broadcaster to triplicate the information with the goal that every IP block receives the data at the same time, and a series of modules destined to manage the communications over the AMBA AXI4 infrastructure.

III. Implementation Results

The proposed hardware architecture is written in C/C++ using a High-Level Synthesis design methodology. The design software tools used are Vivado HLS and Vivado Design Suite.

The equivalent RTL description generated by the High-Level Synthesis tool is implemented over a Xilinx Zynq XC7Z020-CLG484-1 EPP device, which includes two ARM Cortex-A9 general purpose microprocessors and a Kintex-7 FPGA. An Avnet ZedBoard™ Zynq®-7000 ARM/FPGA SoC has been used as a prototype board. To reduce the power consumption of the system, a low power strategy has been applied during the implementation stage on Vivado.

The prototype is able to run at a maximum frequency of 150 MHz if it executes its complete performance. The system latency for its complete performance is 8.558 µs, obtaining a throughput of 187 Mbps. If at least one filter provides a positive response for the header analysis, the decryption stage is omitted and the system reaches a throughput of 1.2 Gbps with a latency of 1.334 µs. Taking into account that the total latency of the system in its purely software implementation is 2.067 ms, we obtain a speedup of x241.

The systems' power consumption estimation is about 1.814 W. By applying low power optimization strategies, the BRAM power consumption difference varies between 44 to 60 mW (Fig. 6).

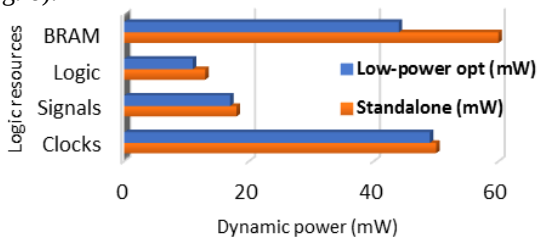

Fig. 6. Power consumption strategies.

In relation to the programmable logic resources utilization (Fig. 5), our design consumes 11.38% of the Slice LUTs and 7.39 % of the Slice Flip Flops (FF) on the Xilinx Zynq XC7Z020-CLG484-1 EPP device. A rough estimation shows that the FPGA is approximately utilized up to 19.82%.

Comparing the filter designed in this work with others available in the state-of-the-art (Table 1), we can conclude that our filter presents a good trade-off between throughput reached and efficiency due to it analyses the two critical levels of the OSI model, giving to the endpoint an extra protection against potential threats.

Table 1. Comparison between the filter and similar works.

Work	Device	Memory structure	OSI levels	Frequency (MHz)	Throughput (Gbps)
Loinig *et al.* [9]	Virtex-4	RAM-based	2, 3	125.0	1.00
Rohrbeck *et al.* [10]	Virtex-5	RAM-based	2,3, 4	142.9	4.57
Yu,Cong *et al.* [11]	Virtex-5	Bloom Filters	7	272.0	8.70
This work	Zynq Z7020	Counting Bloom Filters	3, 4	229.0	1.20

In relation to the decryption IP, it has been possible to define the ratio *throughput/slices* to compare different papers about Simon and AES implementations over FPGA with the Simon IP module designed in this work (Table 2). In this sense, we can conclude that our design obtains the best ratio.

All the works included in the comparison have been modeled in an HDL (*Hardware Design Level*) language, so one would expect them to be more efficient than a high-level description. However, this comparison demonstrates the quality of a hardware-friendly C/C ++ model, as well as the efficiency of current high-level synthesis tools.

Table 2. Comparison between the decryption stage and similar works.

Work	Device	Slices	Throughput (Mbps)	Ratio
Gulcan *et al.* [12]	Spartan-6	6%	3.60	0.100
Wetzels *et al.* [13]	Spartan-6	43.27%	479.30	0.163
Jararweh *et al.* [14]	Virtex-5	98.07%	254.00	0.083
This work	Zynq Z7020	19.82%	221.00	0.594

IV. Conclusions

In this paper, we present a Fog node solution comprised by an authentication and by an encryption stage, that are implemented on a FPGA-based MPSoC, achieving a latency of 8.558 µs and a speedup of x241 compared to its software solution. Also, it accomplishes the specified requirements in terms of logic and power consumption.

Finally, it should be noted that much of the work done has been dedicated to the exhaustive study of efficient algorithmic encryption solutions as well as reduced utilization and power consumption solutions, due to the inability to integrate AES in an embedded electronic system oriented to low power as a result of the high-energy consumption derived from its activity. In this way, the use of the Simon algorithm is a significant advance, since it has been shown to offer good results with a minimum memory footprint and reduced power consumption, especially when compared to AES solutions.

References

[1] T. J. Barnett, A. Sumits, S. Jain, and U. Andra, "Cisco Visual Networking Index (VNI): Global Mobile Data Traffic Forecast Update, 2016-2021," 2015.

[2] D. Evans, "The Internet of Things - How the Next Evolution of the Internet is Changing Everything," *CISCO white Pap.*, no. April, pp. 1–11, 2011.

[3] A. Bannister, "The numbers behind the inexorable rise of cyber threats," *IFSEC Global*, 2017. [Online]. Available: https://www.ifsecglobal.com/numbers-inexorable-rise-cyber-threats/.

[4] Dyn Inc., "Dyn Statement on 10/21/2016 DDoS Attack." 2016.

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," *Proc. first Ed. MCC Work. Mob. cloud Comput.*, pp. 13–16, 2012.

[6] B. Negash, A. M. Rahmani, P. Liljeberg, and A. Jantsch, "Fog Computing Fundamentals in The Internet-of-Things," in *Fog Computing in the Internet of Things. Intelligence at the Edge*, 2017, pp. 1–11.

[7] S. Tarkoma, C. E. Rothenberg, and E. Lagerspetz, "Theory and Practice of Bloom Filters for Distributed Systems," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 1, pp. 131–155, 2012.

[8] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "Simon and Speck: Block Ciphers for the Internet of Things," 2015.

[9] J. Loinig, J. Wolkerstorfer, and A. Szekely, "Packet Filtering in Gigabit Networks Using FPGAs," pp. 1–8, 2007.

[10] J. Rohrbeck, V. Altmann, S. Pfeiffer, D. Timmermann, M. Ninnemann, and R. Maik, "Secure Access Node : an FPGA-based Security Architecture for Access Networks," *ICIMP 2011 Sixth Int. Conf. Internet Monit. Prot. Secur.*, pp. 54–57, 2011.

[11] H. Yu, R. Cong, L. Chen, and Z. Lei, "Blocking pornographic, illegal websites by internet host domain using FPGA and bloom filter," *Proc. - 2010 2nd IEEE Int. Conf. Netw. Infrastruct. Digit. Content, IC-NIDC 2010*, pp. 619–623, 2010.

[12] A. Aysu, E. Gulcan, and P. Schaumont, "SIMON says: Break area records of block ciphers on FPGAs," *IEEE Embed. Syst. Lett.*, vol. 6, no. 2, pp. 37–40, 2014.

[13] J. Wetzels and W. Bokslag, "Simple SIMON FPGA implementations of the SIMON64/128 Block Cipher," *arXiv Prepr. arXiv1507.06368*, 2015.

[14] Y. Jararweh, L. Tawalbeh, H. Tawalbeh, and A. Moh'd, "28 Nanometers FPGAs Support for High Throughput and Low Power Cryptographic Applications," *J. Adv. Inf. Technol.*, vol. 4, no. 2, pp. 84–90, 2013.