5G Security: FPGA Implementation of SNOW-V Stream Cipher

Lampros Pyrgas[1,2], Paris Kitsos[1,2]
[1] Industrial Systems Institute of "Athena" RIC in ICT and Knowledge Technologies, Patras, Greece
[2] ECSA Lab., Electrical and Computer Engineering Department, University of the Peloponnese, Greece
{pyrgas, pkitsos} @isi.gr

2021 24th Euromicro Conference on Digital System Design (DSD) | 978-1-6654-2703-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/DSD53832.2021.00064

***Abstract*—In this paper, a very compact architecture the newest member of the SNOW family of stream ciphers, called SNOW-V, is presented. The proposed architecture has a 128-bit datapath and is pipelined in key areas in order to achieve the maximum possible frequency while using only a small number of hardware resources. The design was coded using the VERILOG hardware description language and the BASYS3 board (Artix 7 XC7A35T) was the target of the hardware implementation. The proposed implementation utilizes only 2109 FPGA LUTs and 1352 FFs and reaches a data throughput of 2.6 Gbps at 224 MHz clock frequency.**

Keywords— SNOW-V stream cipher, FPGA, Compact architecture, Internet of Things security, 5G Security.

I. INTRODUCTION

The 5G wireless networks that are on the horizon are expected to provide high data rates, low latency and better Quality of Service (QoS) [1]. However, with this advent of the 5G wireless networks, the need for security and privacy will be even greater than before. Organizations and associations, like the Next Generation Mobile Networks (NGMN) [2], are already trying to categorize the threats and provide technical solutions for higher security. Scientific interest is continuously increasing and works that analyze the threats and propose solutions are added in literature [3].

Stream ciphers have always been used as a solution to security problems in wireless networks of previous generations. In 4G systems, one of the best possible choices was the 128-bit key SNOW 3G primitive [4]. However, with the coming of 5G networks algorithms like SNOW 3G have to face challenges and adapt.

One major challenge for security in 5G wireless networks is that many of the nodes of these networks will be virtualized, reducing the available specialized hardware that can be used for the implementation of cryptographic algorithms. Moreover, in order to increase security the 3GPP standardization organization is moving towards algorithms with 256-bit key lengths [5]. These two challenges can result in the exclusion of many of the currently used algorithms.

In order to surpass the above challenges, the SNOW 3G was recently revised to SNOW-V stream cipher [6]. SNOW-V keeps the general structure of SNOW 3G (it consists of a linear feedback shift register (LFSR) part and a Finite State Machine (FSM) part), but the LFSR part's internal structure is different and now operates 8 times faster than the FSM part. Moreover, a masking process has been added during the initialization phase in order to protect against attacks. The new SNOW-V stream cipher offers the same level of security as the AES-256 algorithm but faster encryption time.

In this paper, a hardware implementation of the SNOW-V stream cipher is presented. This architecture operates in a 128-bit datapath and is pipelined in key areas in order to reach the maximum possible frequency. In order to reduce the required hardware resources, many architectural design optimizations are used in our paper. The proposed design can reach a clock frequency up to 224 MHz, needs 11 clock cycles per round reaching a throughput more than 2.6 Gbps while it utilizes only 2109 LUTs and 1352 FFs on the XC7A35T FPGA.

The paper is organized as follows: in Section II the specifications of the SNOW-V stream cipher are presented. In Section III, the proposed architecture is presented and is explained in detail. The implementation's results are compared with results from implementations of other ciphers in Section IV. Finally, Section V concludes the paper.

II. SNOW-V STREAM CIPHER SPECIFICATIONS

The SNOW-V stream cipher is a new member in the well-known SNOW family of stream ciphers [6]. Its components are similar to the ones in the other members of its family. The algorithm has a 256-bit key and a 128-bit initialization vector (IV) as inputs and a 128-bit output.

SNOW-V consists mainly of two parts, an LFSR part and an FSM part. The LFSR part has two shift registers feeding into each other and the FSM part has two instances of the AES encryption round function [7] and three 128-bit registers.

The LFSR part consists of two LFSRs, LFSR-A and LFSR-B, of length 16 and with a cell size of 16-bit. However, their elements are generated by different polynomials in $F_{2^{16}}$. Specifically, the elements of LFSR-A are generated by the polynomial:

$$g^A(x) = x^{16} + x^{15} + x^{12} + x^{11} + x^8 + x^3 + x^2 + x + 1$$

while the elements of LFSR-B are generated by the polynomial:

$$g^B(x) = x^{16} + x^{15} + x^{14} + x^{11} + x^8 + x^6 + x^5 + x + 1$$

In each LFSR update step, 256 of the total 512 bits are updated. This is done through a series of XORings and data shifts. Finally, the 128 MSBs of LFSR-B, denoted as T1, and

This work was financially supported by EPAnEK with national and EU funds in the context of the research project "I3T" - MIS5002434.

978-1-6654-2703-6/21/$31.00 ©2021 IEEE
DOI 10.1109/DSD53832.2021.00064

the 128 LSBs of LFSR-A, denoted as T2, are passed to the FSM part as inputs.

The FSM part consists of three 128-bit registers (R1, R2, R3) with two AES encryption round functions between them, two XORings, two components that consists of four parallel additions modulo 2^{32} and one byte-oriented permutation, denoted σ, and given by:

$$\sigma = [\ 0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15\ ].$$

The registers and the output are updated after a full AES encryption round.

Finally, an initialization step is required before the main execution process of the stream cipher. During this step, the input key's 128 MSBs are stored in the 128 MSBs of LFSR-B while the input key's 128 LSBs are stored in the 128 MSBs of LFSR-A. The initialization vector (IV) is stored in the 128 LSBs of LFSR-A. The initialization runs for 16 steps. The cipher is updated in the same way as during its main execution with the exception that the final output is XORed with the 128 MSBs of LFSR-A in every step. Moreover, during the last two steps of the initialization phase, the key is XORed with the contents of register R1.

III. Proposed SNOW-V Architecture

The proposed architecture for the SNOW-V stream cipher consists of three parts, the LFSR part where the data of the internal state are stored, the FSM part where the encryption's computations are performed, and the Control part where the signals that control the computational flow are generated. The proposed architecture is shown in Fig. 1.

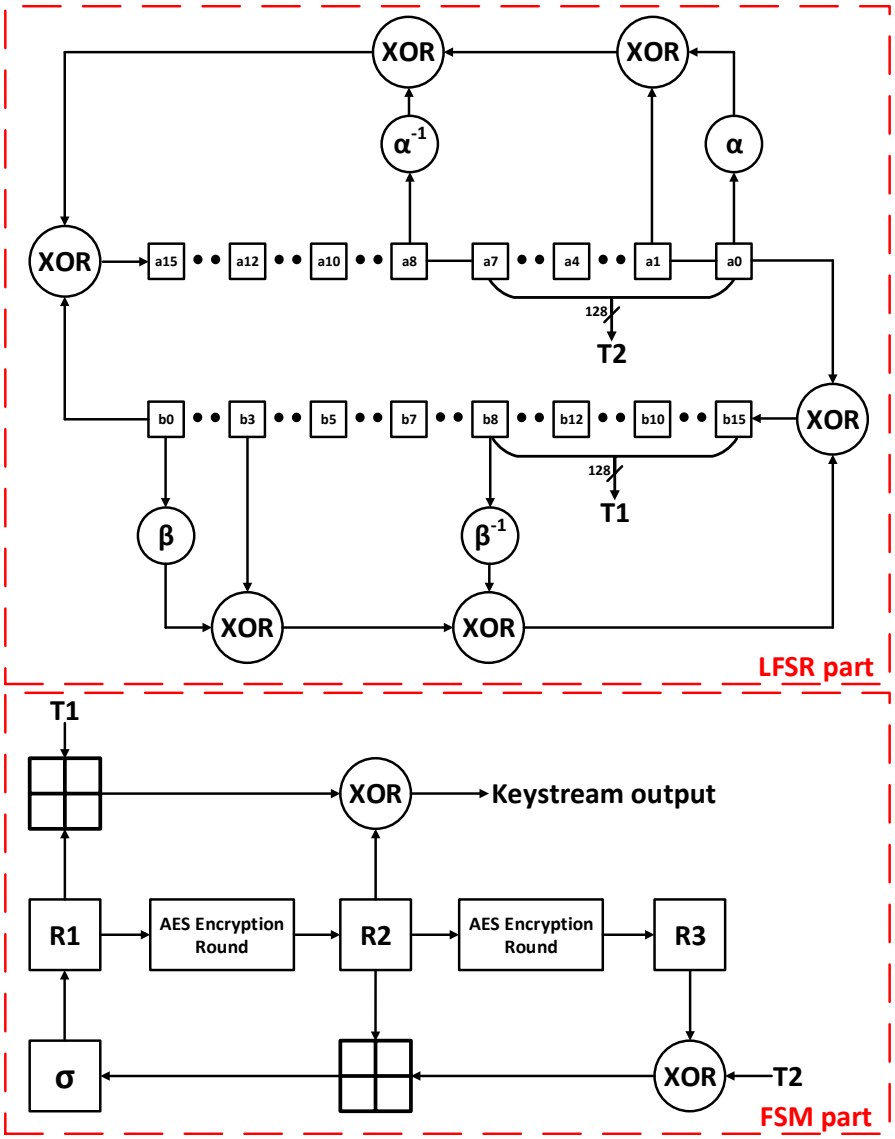

Fig. 1. IArchitecture of SNOW-V based on the specifiactions in [6].

The Control part contains two counters. The first counter counts the number of steps that are passed during the initialization phase, while the second counter counts how many clock cycles have passed during the current round. These counters generate the control signals for a series of multiplexers that are responsible for selecting the appropriate data and driving them to the correct component.

A. LFSR architecture and update

As already stated the LFSR part mainly consists of two LFSRs, denoted LFSR-A and LFSR-B. During the initialization phase, which lasts for 16 rounds, the input Key and the initialization vector IV are driven and stored in the two LFSRs in their appropriate positions according to the algorithm's specifications. The contents of the two LFSRs are updated for the rest of the algorithm's execution through a combination of XORs and shifts as shown in Fig. 1. In each round the contents are updated 8 times.

Specifically in the subunits α and β, the input data are first shifted to the left by one and then are XORed with the coefficients 0x990f and 0xc963 respectively. If the MSB of the initial input value is '1' then the computed XORed value is outputted, while if it is '0' then the shifted value is outputted instead. The internal architecture of this unit is shown in Fig. 2.

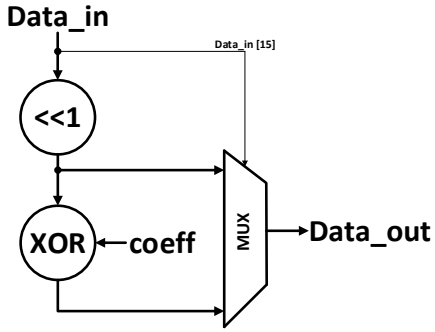

Fig. 2. Internal architecture of units α and β

In the subunits α^{-1} and β^{-1}, the input data are first shifted to the right by one and then are XORed with the coefficients 0xcc87 and 0xe4b1 respectively. If the LSB of the initial input value is '1' then the computed XORed value is outputted, while if it is '0' then the shifted value is outputted instead. The internal architecture of this unit is shown in Fig. 3.

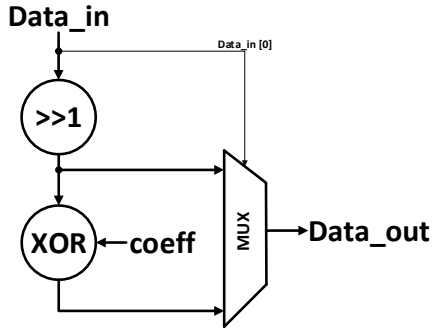

Fig. 3. Internal architecture of units α^{-1} and β^{-1}

During the initialization step only, the 128 MSBs of LFSR-A are XORed with the algorithm's output. As already stated, the 128 MSBs of LFSR-B, denoted as T1, and the 128 LSBs of LFSR-A, denoted as T2, are passed to the FSM part as inputs.

B. FSM architecture and update

The FSM part is based on two AES encryption round functions, with round keys set to zero, which are implemented based on [8], but have pipeline registers between their subparts.

Because the round keys are set to zero the AddRoundKey subpart is completely removed leading to fewer hardware resource requirements and better operation frequency than a full AES round.

Three 128-bit registers, denoted R1, R2, R3, are used in order to store the input and output data for and from the round functions. On each round, the input T1 is added with the contents of register R1 through four parallel 32-bit additions. The 32-bit parts are added with carry that does not, however, carry from a lower part to the next. The result is then XORed with the contents of register R2 and the final result is outputted. The T2 input is XORed with the contents of register R3 and the result is then added with the contents of register R2 through four parallel 32-bit additions. The result of this addition is then passed through the permutation σ and the output is stored in register R1. During the last two steps of the initialization step, the initial Key is XORed with the contents of register R1.

IV. FPGA Implementation And Results

The proposed architecture of the SNOW-V stream cipher was synthesized and implemented in Xilinx's Vivado 2020.1 through the use of the VERILOG hardware description language. The hardware implementation was placed, routed and evaluated our experiments the BASYS3 (ARTIX XC7A35T FPGA) board was used. To test the correct operation of the proposed SNOW-V stream cipher implementation we used the test vectors provided by [6]. The implementation results of the proposed architecture are depicted in TABLE I.

TABLE I. FPGA Implementation Results.

FPGA Device	XC7A35T (BASYS3)
Registers (FFs)	1352
LUT	2109
#Slices	530
Freq (MHz)	224
Initialization Latency (# Clock cycles)	176
Round's latency(# Clock cycles)	11
Throughput (Mbps)	2606

As it can be seen from TABLE I, the proposed architecture of SNOW-V stream cipher, utilizes only a small portion of the FPGA's resources, specifically 1352 FFs and 2109 LUTs that occupy 530 slices. It has a 128-bit datapath and requires only 11 clock cycles for the completion of a round. However, by

pipelining the design, only the first round has this 11 clock cycles latency. It requires a total of 176 clock cycles for its initialization process. The maximum supported clock frequency is 224 MHz, which leads to a maximum throughput equal to 2606 Mbps.

In TABLE II, we present some comparisons between the proposed SNOW-V implementation and previous FPGA designs of other ciphers.

TABLE II. COMPARISONS WITH OTHER ARCHITECTURES.

Cipher	Freq. (MHz)	Throughput (Mbps)	# Registers	# LUTs	#Slices
AES-128 [9]	192	1290	830	1417	431
AES-128 [10]	250	2900	-	-	1220
ZUC [11]	38	1216	-	-	1147
SNOW 3G [11]	104	3328	-	-	3559
Prop. SNOW-V	224	2606	1352	2109	530

In [9] an AES-128 architecture is implemented that utilizes 830 FFs and 1417 LUTs in 431 Slices. It achieves a max frequency of 192 MHz and reaches a throughput of 1290 Mbps. Compared to our SNOW-V implementation, this AES-128 implementation utilizes less FFs and LUTs and slightly less Slices but has a lower frequency and achieves half the throughput of our implementation.

In [10] an AES-128 architecture is presented that utilizes 1220 Slices. It achieves a max frequency of 250 MHz reaching a throughput of 2900 Mbps. This implementation has slightly higher frequency and throughput when compared to our SNOW-V implementation but it utilizes more than double the Slices.

The ZUC architecture that is proposed in [11] uses 1147 Slices and has a frequency of 38 MHz leading to a throughput of 1216 Mbps. This implementation has lower frequency, half the throughput and it utilizes double the Slices when compared to our SNOW-V implementation.

Again in [11], a SNOW 3G architecture is implemented that has a 104 MHz max frequency and a throughput of 3328 Mbps while it utilizes 3559 Slices. Compared to our SNOW-V implementation, this SNOW 3G implementation utilizes a lot more Slices, has a lower frequency but it achieves a better throughput than our implementation.

Finally, in [12] a series of ASIC implementations are presented for the SNOW-V stream cipher. The most compact utilizes 4776 gate equivalents of silicon area and reaches a throughput of more than one Tbps. These results are in line with the theoretical analysis in [6].

According to TABLE's II comparisons, the proposed SNOW-V implementation compares favorably with all the other implementations when all the comparison metrics are taken are considered. By taking into account the FPGA that was used as the target of our implementation, it can be seen that the proposed SNOW-V implementation is a primate candidate for 5G security applications.

V. CONCLUSIONS

A new implementation of the SNOW-V stream cipher, suitable for industry applications that require high speed encryption in preparation for the future 5G mobile communication system. The proposed architecture has a 128-bit datapath. The proposed designed was implemented in the Artix 7 XC7A35T FPGA It achieves a throughput up to 45 Mbps at a max clock frequency of 224 MHz with each round requiring 11 clock cycles. Finally, it utilizes only 2109 FPGA LUTs and 1352 FFs.

ACKNOWLEDGMENT

We acknowledge support of this work by the project "I3T - Innovative Application of Industrial Internet of Things (IIoT) in Smart Environments" (MIS 5002434) which is implemented under the "Action for the Strategic Development on the Research and Technological Sector", funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

REFERENCES

[1] M. Agiwal, A. Roy, N. Saxena, "Next Generation 5G Wireless Networks: A Comprehensive Survey", IEEE Communications Surveys Tutorials, vol. 18, no. 3, pp. 1617–1655, thirdquarter 2016.

[2] N. Alliance, "NGMN 5G white paper," Next Generation Mobile Networks, White paper, 2015.

[3] I. Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, A. Gurtov, "5G security: Analysis of threats and solutions," 2017 IEEE Conference on Standards for Communications and Networking (CSCN), Helsinki, Finland, 2017, pp. 193-199, doi: 10.1109/CSCN.2017.8088621.

[4] SAGE, "Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2" Version 1.1, ETSI/SAGE, 2006. https://www.gsma.com/aboutus/wp-content/uploads/2014/12/snow3gspec.pdf.

[5] 3rd Generation Partnership Project (3GPP), 3GPP TR 33.841 version 16.1.0 - 3rd Generation Partnershop Project, Technical Specification Group Services and System Aspects, Security aspects, "Study on the support of 256-bit algorithms for 5G" (Release 16), March 2019.

[6] P. Ekdahl, T. Johansson, A. Maximov, J. Yang, "A new SNOW stream cipher called SNOW-V", IACR Transactions on Symmetric Cryptology, 2019(3), pp. 1-42. https://doi.org/10.13154/tosc.v2019.i3.1-42

[7] J. Daemen and V. Rijmen, "AES Proposal: Rijndael", AES algorithm submission, September 3, 1999.

[8] https://opencores.org/project,aes_core.

[9] L. Daoud, F. Hussein, N. Rafla, "Optimization of Advanced Encryption Standard (AES) Using Vivado High Level Synthesis (HLS)", CATA, 2019.

[10] P. Bulens, F.-X. Standaert, J.-J. Quisquater, P. Pellegrin, G. Rouvroy, "Implementation of the AES-128 on Virtex-5 FPGAs", In Progress in Cryptology–AFRICACRYPT 2008, S. Vaudenay, Ed., Springer: Berlin/Heidelberg, Germany, 2008, pp. 16–26.

[11] P. Kitsos, N. Sklavos, G. Provelengios, A. N. Skodras, "FPGA-based performance analysis of stream ciphers ZUC, Snow3g, Grain V1, Mickey V2, Trivium and E0", Microprocessors and Microsystems, Volume 37, Issue 2, 2013, pp. 235-245, ISSN 0141-9331, https://doi.org/10.1016/j.micpro.2012.09.007.

[12] A. Caforio, F. Balli, S. Banik, "Melting SNOW-V: improved lightweight architectures", Journal of Cryptographic Engineering (2020), https://doi.org/10.1007/s13389-020-00251-6.