This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Multiradix Trivium Implementations for Low-Power IoT Hardware

J. M. Mora-Gutiérrez, C. J. Jiménez-Fernández, and M. Valencia-Barrero

***Abstract*— The integration of lightweight symmetric encryption is becoming increasingly widespread in very low-power Internet of Things applications, with the rapid emergence of very low energy block and stream ciphers in portable and wireless systems. Trivium is one of the lightweight stream ciphers shortlisted for the hardware profile of the eSTREAM project. This paper describes low-power multiradix Trivium implementations based on the use of parallelization techniques to reduce dynamic power consumption. The low-power Trivium designs were implemented and characterized in TSMC 90 nm to compare area resources and power reduction. The implementation results show that our proposed designs offer dynamic power savings of 31%–45% with radix-1 and radix-2 when compared with the standard Trivium, and 15% with radix-8. There is no improvement, however, with radix-16.**

***Index Terms*— Application-specific integrated circuit (ASIC) implementation, Internet of Things (IoT) hardware, lightweight cryptography, low power, stream ciphers, trivium.**

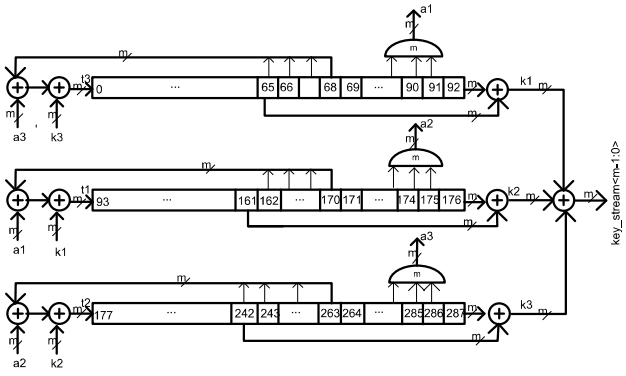

Fig. 1. Schematic of the radix-*m* Trivium cipher.

I. Introduction

Privacy, data protection, and information security are key requirements for Internet of Things (IoT) hardware applications. However, many of the communications tasks that take place in IoT applications are between battery-based portable devices with limited computing resources, in which the mechanisms of encryption and authentication are severely constrained. Logically, therefore, one of the main priorities in IoT designs is to choose lightweight cryptographic algorithms with low power consumption (lightweight cryptography) [1]–[5] and apply techniques that reduce dynamic power consumption [6]. Stream ciphers, such as Grain, Mickey, and Trivium, featuring algorithms specially designed for optimal performance in hardware cryptography, were identified and published in the European initiative known as the *eSTREAM* (*ECRYPT* Stream Cipher) project [7]–[9]. We selected the Trivium stream cipher, which was specified in ISO/IEC 29192-3 as a keystream generator for lightweight stream ciphers [10]–[12] very well suited to this type of systems.

From the theoretical perspective of design techniques for power reduction, this paper focuses on improving dynamic consumption at a logical level. The parallelization technique [13] is a good option for reducing power consumption in Trivium, because this cipher's internal architecture based on shift registers lends itself well to parallelization. In previous works [14], we proposed two different parallelization methods exclusively for radix-1 Trivium: the mixed parallel low power (MPLP) option, where parallelization is applied to flip-flops unaffected by nonlinear feedback paths, and the full parallel low power (FPLP) option, where parallelization is applied to all the flip-flops in the Trivium cipher even though this implies redesigning nonlinear feedback paths.

Manuscript received December 30, 2016; revised April 17, 2017; accepted July 20, 2017. This work was supported by the Spanish Government projects: CESAR under Grant TEC2013-45523-R, INTERVALO under Grant TEC2016-80549-R, and LACRE under Grant CSIC 201550E039. *(Corresponding author: J. M. Mora-Gutiérrez.)*

The authors are with the Instituto de Microelectrónica de Sevilla, CSIC–University of Seville, 41092 Seville, Spain (e-mail: jmiguel@imse-cnm.csic.es; cjesus@imse-cnm.csic.es; manolov@dte.us.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2017.2736063

This paper extends the results obtained in those studies in two ways. First, it focuses on multiradix Trivium hardware implementations for low-power IoT applications. Second, the results presented here were obtained from postlayout data, which considers the influence of clock trees, and this, as will be seen, has major repercussions. The objective is to propose an application-specific integrated circuit implementation of low-power multiradix Trivium designs based on standard cell libraries in CMOS technologies, in order to compare area resources and power consumption. Twelve versions of Trivium (four for each standard, MPLP, and FPLP versions) were designed in a 90-nm technology to obtain place-and-route parasitic parameters from the layout level. For this purpose, a detailed study of dynamic power consumption was carried out, extracting full timing and postlayout parasitic parameters.

The contents of this paper are organized as follows. Section II briefly describes the multiradix Trivium algorithm and its hardware implementation. In Section III, two low-power multiradix architectures are presented along with power reduction results for a 90-nm CMOS technology process. Section IV compares the results obtained for the MPLP and FPLP Trivium options, and finally, some conclusions are presented in Section V.

II. Standard Multiradix Trivium Hardware Implementation

Trivium is a synchronous stream cipher designed to generate up to 264 bits of pseudorandom key stream from an 80-bit secret key (KEY) and an 80-bit initialization vector (IV). It was first proposed by De Canniere and Preneel [15].

The cipher's architecture is based on a 288-bit cyclic shift register (also called an internal state register), with combinational logic (AND and XOR gates) providing its nonlinear feedback. Implementations of the Trivium algorithm comprise three shift registers of different lengths [14], [15]. The number of output bits generated per clock cycle is called radix. This paper describes implementations of the Trivium algorithm that generate radix-1, radix-2, radix-8, and radix-16. These multiple radices are generated using the same internal state register but shifting it one or more bits to the right depending on the radix (1, 2, 8, or 16 bits) as shown in the schematic of the radix-*m* Trivium stream cipher in Fig. 1.

1063-8210 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

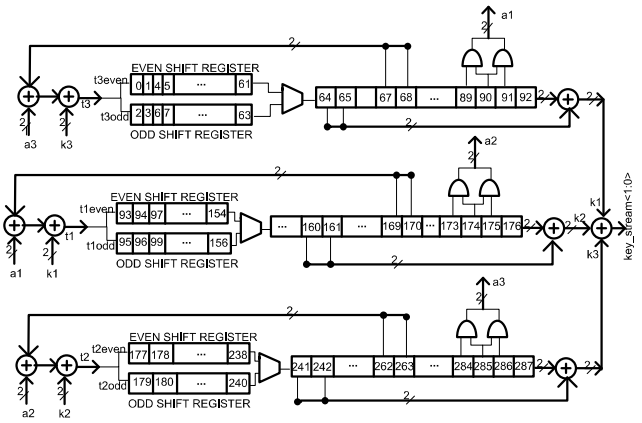

Fig. 2. Schematic for radix-2 MPLP.

The length of each internal shift register is not the same; the first register has 93 bits, the second 84 bits, and the third 111 bits. Each register is shifted serially with the inclusion of m bits depending on the number of output bits, m. Multiradix key stream bits are then generated every clock cycle. The bits to be introduced in Trivium's shift registers are generated by XOR gates. With this architecture and mode of operation, most of the power consumption is attributable to the flip-flops in the cyclic shift registers.

To reduce dynamic power consumption, we, therefore, focused on decreasing the switching activity by applying parallelization techniques, maintaining the same input frequency and supply voltage [13]. This technique basically splits the shift register into an odd and even shift registers with half bits. Dynamic power consumption depends directly on switching activity factor, which represents the average fraction of clock cycles in which a signal transition occurs, clock frequency, supply voltage, and output capacitance [6]. The greater the logic transitions in the cell output, the greater the switching power.

The Trivium implementations were described in *VHDL* code, synthesized with *Design Vision* (*Synopsys*), and verified using the *ModelSim (Mentor Graphics)* simulation environment, with the same test vectors and using the same key and IV as those presented in the Trivium reference files [7].

III. Low-Power Multiradix Trivium Hardware Implementation

The parallelization technique cannot be applied directly to all the state register flip-flops in the Trivium stream cipher, because the outputs of some of them are involved in combinational operations.

We propose two new low-power multiradix Trivium implementations using logic parallelization techniques: MPLP and FPLP. In MPLP, parallelization is applied to flip-flops unaffected by nonlinear feedback paths, i.e., the less significant bits from each shift register; 196 out of 288 bits in the state register for radix-1 and radix-2, 144 out of 288 for radix-8, and 96 out of 288 for radix-16.

Parallelization requires a slight hardware modification in each shift register shown in Fig. 1. A schematic representation of the radix-2 MPLP Trivium is shown in Fig. 2. The first shift register contains bits 0–63, the second, bits 93–156, and the third, bits 177–240. With the parallelization technique, the bits in each shift register not involved in feedback or combinational operations (LSB bits) are divided into two separate shift registers denominated odd and even shift registers. Each of these has half-bits, so the total length of the shift register remains the same.

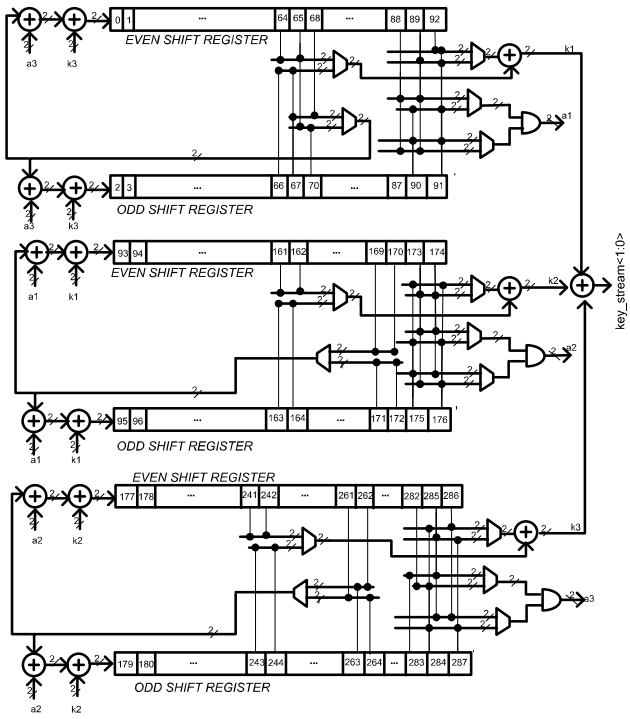

Fig. 3. Schematic for radix-2 FPLP.

In multiradix MPLP Trivium, 1, 2, 8, or 16 bits are shifted to the right in each cycle, depending on the radix. The required modification involves generating a half frequency clock and adding a multiplexer in the least significant bit of each shift register. The *KEY* and *IV* are loaded in parallel, the even registers being loaded with the rising edge and the odd registers with the falling edge of a clock with half the frequency of the external clock.

If the parallelization technique were applied to all the Trivium stream cipher state register flip-flops, the reduction in power consumption would be bigger, because is where most of the power consumption is dissipated. To achieve this, the second low power implementation, named FPLP Trivium, was designed. In this implementation, all the bits in the Trivium state register were parallelized. This required further modification to the standard version. Each Trivium shift register in Fig. 1 was split into two half-length shift registers denominated odd and even. Fig. 3 shows a schematic of the radix-2 FPLP Trivium. The length of each shift register is shown in Fig. 3 inside the odd and even registers. In radix-2 Trivium, 2 bits are shifted to the right in each cycle.

As mentioned in Section II, generation of the input bits in each of the shift registers and the key stream depends on the bits stored in different positions in the shift registers. It is, therefore, necessary to introduce extra logic to select the bits from the correct shift register, depending on whether the clock cycle was odd or even. On one clock edge, the bit to be retrieved will be in the even register, and on the following clock edge, the bit to be retrieved will be in the odd register. This implies the use of multiplexers which, using the clock as the selection signal, select the bit to be retrieved from the corresponding shift register, as shown in Fig. 3. In the first clock cycles, both the secret key and the *IV* are loaded in parallel, as described earlier for the MPLP implementation.

A. Area, Cell, and Timing Implementation Report

The area and timing report provided by the *Design Vision* synthesis tool (*Synopsys*) for the standard, MPLP, and FPLP multiradix

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
MULTIRADIX TRIVIUM POSTSYNTHESIS REPORT

Trivium	**Library: TSMC 90 nm**			
	GE NAND21 2.82μm²	*Number Logic Cells*	*Area Non Comb. μm²*	*Area Comb. μm²*
Trivium_x1	2254	625	4390	1971
Trivium_x1mp	2339	665	4474	2128
Trivium_x1fp	2392	620	4633	2117
Trivium_x2	2278	631	4387	2041
Trivium_x2mp	2355	646	4545	2101
Trivium_x2fp	2444	656	4629	2269
Trivium_x8	2427	723	4368	2484
Trivium_x8mp	2505	791	4512	2559
Trivium_x8fp	2733	873	4737	2976
Trivium_x16	2631	855	4410	3016
Trivium_x16mp	2706	979	4494	3143
Trivium_x16fp	3186	1147	5063	3929

TABLE II
PERCENTAGES OF AREA AND NUMBER OF LOGICAL CELLS, WITH REFERENCE TO RADIX-1 VERSION

Trivium	**Library: TSMC 90 nm**			
	Total Area	*Logic Cells*	*Area Non Comb. Cells*	*Area Comb. Cells*
Trivium_x1	-	-	-	-
Trivium_x1mp	-	-	-	-
Trivium_x1fp	-	-	-	-
Trivium_x2	1%	1%	0%	4%
Trivium_x2mp	1%	-3%	2%	-1%
Trivium_x2fp	2%	6%	0%	7%
Trivium_x8	8%	16%	-1%	26%
Trivium_x8mp	7%	19%	1%	20%
Trivium_x8fp	14%	41%	2%	41%
Trivium_x16	17%	37%	0%	53%
Trivium_x16mp	16%	47%	0%	48%
Trivium_x16fp	33%	85%	9%	86%

*$(\% = \%(T_{standard} - T_{FPLP-MPLP})/T_{standard})$.

Triviums is shown in Table I. The first column shows the gate equivalent number or GE (area of a two-input NAND gate in TSMC 90-nm technology, 2.82 μm^2). Furthermore, Table I shows the details about the number of logical cells and the areas of sequential (noncombinational) logic and combinational logic in each of the Trivium ciphers designed after the synthesis. The increments (%) in each measurement compared with the corresponding 1-bit Trivium version are shown in Table II. To better appreciate the details of the synthesis tool's report, the data are also represented graphically in Fig. 4.

The results show that the resources consumed increase significantly as the number of output bits rises. For example, the increase in cells in the standard 16-bit version was 37% higher than in the 1-bit version (see Table II). Something similar occurred in the MPLP version (with an increase of 47%) and the FPLP version (with an increase of 85%). Similar increases also occurred, although to a lesser extent, in the total area (17% in the standard version, 16% in MPLP, and 33% in FPLP). With respect to delay, the implementations successfully overcame the restriction imposed on the clock (50 MHz) although the multiplexer delay slowed low power implementations down slightly in comparison with the standard implementation.

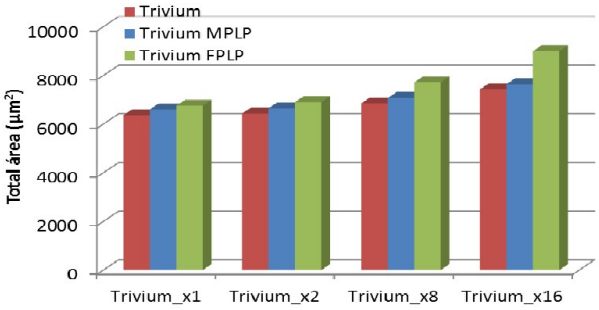

Fig. 4. Multiradix Trivium area report.

Area and cells were evaluated by the synthesis tool (Synopsys) using information from the manufacturer's technological library (TSMC 90 nm). The FPLP Trivium ciphers have more logical cells and more combinational area than the other Trivium cipher proposals. The noncombinational area is similar in all versions except for the 16-bit FPLP version (9% larger), due to the increase in the number of flip-flops in some odd and even parallel registers. Of particular interest are the increases in the combinational cell area shown in the eight and sixteen bit FPLP versions (41% and 86%, respectively). These are due to the large number of bits that have to be feedback, which raises the number of combinational cells considerably.

The low power version, Trivium_x1fp, uses less standard cells than the other versions, because the *Synopsys* synthesis tool chooses the most appropriate cells for each design. However, despite having less logic cells, the total area is bigger than that of Trivium_x1mp and Trivium_x1. The Synopsys synthesis of the MPLP Triviums, the version that evolved best with the number of output bits, generate designs with fewer combinational cells than the FPLP versions. However, its area in multibit cases ("radix-2," "radix-8," or "radix-16") increased more than in the standard Trivium version.

As it is summarized in Table II, the MPLP and FPLP Trivium proposals maintain a good relationship in terms of their numbers of cells compared with the standard version in the radix-1 and radix-2 implementations. Their areas are very similar when implementations with the same radix are compared, although at radix-8 and radix-16 the difference between them becomes greater, especially in the radix-16 FPLP version, where there is a clear difference in resources and area.

The differences between the standard and low power versions in terms of total area are also shown graphically in Fig. 4, with the same conclusions. The FPLP versions have always more area than the others and for radix-8 and radix-16 the difference is greater.

B. Power Consumption Implementation Report

Power reports provided by the *Encounter RTL-GDSII* tool for the multiradix standard and MPLP versions of Trivium are shown in Table III. All the power results were obtained, including activity files obtained from post place-and-route simulations with full timing and parasitic parameters. We used the worst case corner for the simulations. The static power was rather small and its effect was minimal. In this technology, static power is much lower than dynamic power and usually accounts for less than 3% of dynamic

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
MULTIRADIX TRIVIUM POSTLAYOUT REPORT

VDD=1,08 V-T=125 ºC-50MHz-worst case timing			
Trivium	*Power consumption*		
	Dynamic		*Static*
	μW	%*	*μW*
Trivium_x1	330.6	-	2.9
Trivium_x1mp	227.0	31.3	3.0
Trivium_x1fp	179.8	45.6	4.6
Trivium_x2	339.5	-	3.0
Trivium_x2mp	231.7	31.8	3.0
Trivium_x2fp	197.4	41.9	4.7
Trivium_x8	395.6	-	3.1
Trivium_x8mp	330.1	16.6	3.1
Trivium_x8fp	334.8	15.4	6.4
Trivium_x16	418.5	-	3.1
Trivium_x16mp	423.7	-1.2	3.3
Trivium_x16fp	512.18	-22.4	6.6

*$(\% = \%(T_{standard} - T_{FPLP-MPLP})/T_{standard})$.

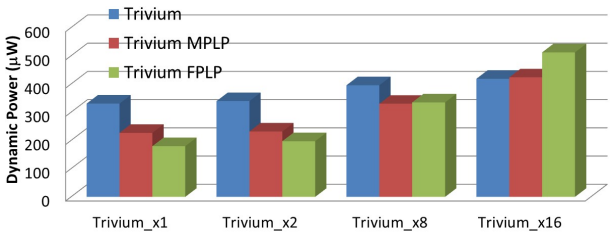

Fig. 5. Multiradix Trivium power report.

power consumption. When the power consumption of the three Trivium implementations was compared, it was noticed that the low power implementations have lower dynamic power consumption than the standard Trivium up to radix-16, where there is no improvement due to the increase in combinational logic and, for the MPLP version, the progressive decrease in bit numbers in the parallelized registers: from 196 bits in radix-1 and radix-2 to 144 and 96 bits in radix-8 and radix-16, respectively.

As was expected, power consumption was found to increase with the level of parallelization. However, the Trivium versions proposed to reduce power consumption proved effective up until radix-8 (although they were unsuccessful for radix-16). The MPLP and FPLP proposals showed the significant improvements of about 31%–45% for radix-1 and radix-2 versions and a significant improvement (15%–16%) in radix-8 version. The FPLP radix-1 and radix-2 versions offered the best power reduction: more than 45% less than the standard version. The dynamic power differences between the standard and low power versions are also shown graphically in Fig. 5, with the same conclusions.

The FPLP version can be seen to improve the power consumption of all the standard versions of Trivium up to the radix-16 version, where there is no improvement. It is important to note that, in the radix-1, the FPLP version almost halves the consumption of the standard Trivium. In fact, in comparison with the MPLP version, FPLP has much better values than MPLP for the radix-1 and radix-2 versions. For radix-8, both are very similar.

Finally, Fig. 6 shows a bar diagram with the percentages of improvement in power consumption. The percentages are positive when power consumption is lower than in the standard version and

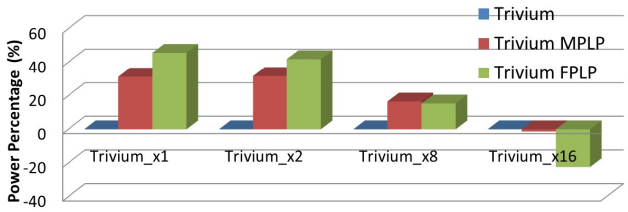

Fig. 6. Multiradix Trivium power percentage.

negative when it is not. Obviously, the improvement value for the standard version itself is 0. Fig. 6 clearly shows the lower power consumption of the MPLP and FPLP versions, except in the case of 16 bits, where they consume more.

IV. MULTIRADIX TRIVIUM RESULTS COMPARISON

When the power consumption of the three Trivium implementations is compared, it can be seen that the FPLP implementation offers the best results in power reduction for radix-1 and radix-2 (see Table III and Fig. 6). The power consumption of the MPLP Trivium is lower than that of the standard implementation but slightly worse than that of the FPLP version. The disadvantage of FPLP is that it has more cell area than MPLP, although the slight overhead in area is less than 7% in radix-1 and radix-2 and 21% in radix-16. The MPLP version has no cell area penalties in comparison with the standard version (3%–4%, as shown in Table I), and is, therefore, still a good option for radix-1 and radix-2.

The results clearly show that the reduction in dynamic power consumption in the FPLP Trivium implementation is the best in radix-1 and radix-2 (about 45%), while the MPLP implementation offers a power reduction of about 31% for the same radix. However, for radix-8, the power consumption of both low power versions is very similar.

V. CONCLUSION

In this paper, we have proposed new low-power multiradix Trivium designs: MPLP and FPLP. Both are based on logic parallelization techniques. Twelve versions of Trivium (radix-1, radix-2, radix-8, and radix-16 for each standard, MPLP, and FPLP proposal) were designed using semicustom design methodologies in a 90-nm technology. All the versions of Trivium were characterized for area and dynamic power consumption with postsynthesis and postlayout data.

In the cases of radix-1 and radix-2 bits, the two versions implemented, MPLP and FPLP, greatly reduced the power consumption of the standard version of the Trivium, while, for radix-8, the reduction was significant but considerably lower. The FPLP version raised the power consumption improvement percentage achieved by MPLP by between 10 and 15 points for radix-1 and radix-2. For radix-8, the reduction achieved by both versions was similar. In the case of radix-16, neither of the low power versions reduced power consumption. The area penalty and cell numbers obtained with this technique were very low (less than 13% for FPLP radix-8), while the improvement in dynamic power consumption was quite high (more than 15% for FPLP radix-8). The FPLP Trivium version offers greater power reduction than the MPLP Trivium, but also produces a slight increase in the complexity of the implementation and the logic used, which in turn increases the final area.

Hence, MPLP and FPLP implementations are presented as very competitive options for incorporating hardware security solutions for very low-power IoT applications.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

REFERENCES

[1] C. Manifavas, G. Hatzivasilis, K. Fysarakis, and Y. Papaefstathiou, "A survey of lightweight stream ciphers for embedded systems," *Secur. Commun. Netw.*, vol. 9, pp. 1226–1246, 2015. doi: 10.1002/sec.1399.

[2] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel, "A survey of lightweight-cryptography implementations," *IEEE Des. Test. Comput.*, vol. 24, no. 6, pp. 522–533, Nov. 2007.

[3] M. Kocheta, N. Sujatha, K. Sivakanya, R. Srikanth, S. Shetty, and P. V. A. Mohan, "A review of some recent stream ciphers," in *Proc. Int. Conf. Circuits, Controls Commun. (CCUBE)*, Dec. 2013, pp. 1–6.

[4] J. Gong, G. Chen, L. Li, and J. Li, "A secure authentication protocol for RFID based on Trivium," in *Proc. Int. Conf. Comput. Sci. Service Syst. (CSSS)*, Jun. 2011, pp. 107–109, doi: 10.1109/CSSS.2011.5974817.

[5] B. Preneel, C. Paar, and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. Berlin, Germany: Springer-Verlag, 2010, doi: 10.1007/978-3-642-04101-3.

[6] A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design*. Boston, MA, USA: Kluwer, 1995, doi: 10.1007/978-1-4615-2325-3.

[7] *eSTREAM: The ECRYPT Stream Cipher Project*. Accessed: Jul. 2017. [Online]. Available: http://www.ecrypt.eu.org/stream/

[8] M. Robshaw and O. Billet, Eds., *New Stream Cipher Designs: The eSTREAM Finalists*. Berlin, Germany: Springer-Verlag, 2008, doi: 10.1007/978-3-540-68351-3.

[9] T. Good and M. Benaissa, "Hardware results for selected stream cipher candidates," in *Proc. Art Stream Ciphers Workshop (SASC)*, 2007, pp. 191–204.

[10] K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha. (2016). "DRAFT report on lightweight cryptography," NISTIR, Tech. Rep. 8114. [Online]. Available: https://doi.org/10.6028/NIST.IR.8114

[11] J. Hosseinzadeh and M. Hosseinzadeh, "A comprehensive survey on evaluation of lightweight symmetric ciphers: Hardware and software implementation," *Adv. Comput. Sci., Int. J.*, vol. 5, no. 4, pp. 31–41, Jul. 2016.

[12] J. H. Kong, L.-M. Ang, and K. P. Seng, "A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments," *J. Netw. Comput. Appl.*, vol. 49, p. 15–50, Mar. 2015, doi: 10.1016/j.jnca.2014.09.006.

[13] T. Schneider *et al*, "Low-voltage low-power parallelized logic modules," in *Proc. PATMOS*, Oldenburg, Germany, Oct. 1995, paper S4.2.

[14] J. M. Mora-Gutiérrez, C. J. Jiménez-Fernández, and M. Valencia-Barrero, "Trivium hardware implementations for power reduction," *Int. J. Circuit Theory Appl.*, vol. 45, no. 2, pp. 188–198, Nov. 2016, doi: 10.1002/cta.2281.

[15] C. De Cannière, "Trivium: A stream cipher construction inspired by block cipher design principles," in *Proc. Int. Conf. Inf. Secur.*, 2006, pp. 171–186, doi: 10.1007/11836810_13.