Exploring the Energy Consumption of Lightweight Blockciphers in FPGA

Subhadeep Banik
DTU Compute
Technical University of Denmark
2800 Kgs. Lyngby
Email: subb@dtu.dk

Andrey Bogdanov
DTU Compute
Technical University of Denmark
2800 Kgs. Lyngby
Email: anbog@dtu.dk

Francesco Regazzoni
ALaRI - USI
via Buffi, 13
6900, Lugano, Switzerland
Email: regazzoni@alari.ch

***Abstract*—Internet of things and cyber-physical systems requiring security functionality has pushed for the design of a number of block ciphers and hash functions specifically developed for being implemented in resource constrained devices. Initially the optimization was mainly on area and power consumption, but, nowadays the attention is more on the energy consumption.**

In this paper, for the first time, we look at energy consumption of lightweight block ciphers implemented in reconfigurable devices, and we analyze the effects that round unrolling might have on the energy consumed during the encryption. Concentrating on applications that require a number of parallel encryptions, we instantiate several designs on the target FPGA and we analyze how the energy consumption varies in each algorithm when changing the amount of unrolled rounds.

Our results, obtained on the Xc6slx45t device of the Spartan6 family, demonstrate that Present is the most energy efficient algorithm and that the relation between the energy consumption and the number of unrolled rounds measured on FPGA is similar to the one measured on dedicated hardware.

I. Introduction

Electronic applications are pervading our lives and they will certainly do even more in the future. Cars currently include an huge number of processors and dedicated devices, while airplanes and trains heavily rely on electronic components. Smart devices are used to control our homes helping improve our comfort and minimizing the energy consumption. Wearable devices are used to monitor our body parameters for supporting a healthy lifestyle and to prevent and predict diseases. Finally, smart grids are operated and controlled by electronic components which, in real time, attempt to estimate the energy demand for optimizing its production and the distribution. Several of these electronic components also integrate actuators, forming the so called cyber-physical systems (CPS).

An appealing target for several of these applications is reconfigurable hardware. FPGAs combine the performance benefits of hardware with the flexibility of software (most FPGAs can, in fact, be easily reprogrammed) and are realized using state of the art technological processes, which are not always easily accessible even when building dedicated ASICs. Reconfigurable devices are also very suitable for low production volumes and their use can significantly reduce the time to market. Finally, in the last few years, FPGAs have reached the size and the capability levels to store a complete system-on-chip, thus becoming an attractive alternative to dedicated ASICs.

Several of the applications mentioned above need to have security capabilities. Commands issued to the actuator part of cyber-physical systems need to be authenticated and guaranteed, as these systems are often used in critical environments. Data collected and analyzed by wearable devices are often extremely sensitive, as they might include health related information. Thus it has to be ensured that only legitimate people have access to them.

Security in reconfigurable devices has been extensively explored by the scientific community. Works proposed so far range from low cost implementation of standard algorithms [1], [2] to high performance devices dedicated to cryptanalysis [3]. Researchers have also dedicated significant amount of effort in realizing FPGA designs which are robust against physical attacks, focusing in particular on power analysis [4], [5], [6] (the standard platform for analyzing the robustness of a design against power analysis is based on reconfigurable hardware [7]).

Despite these efforts, however, the problem of energy efficient cryptography, which is becoming more and more popular, is, on reconfigurable devices, largely unexplored. This fact is particularly surprising, as in FPGAs, contrary to ASICs, it is not possible to apply smart techniques for energy minimization such as clock gating and power gating. Nevertheless, the variety of applications where FPGAs are used so far would significantly benefit from energy efficient designs.

In this paper, for the first time, we tackle this problem and we focus on a number of lightweight ciphers, as they are the ones more likely to be implemented in CPSs and in the devices which populate the future internet of things. We explore their energy efficiency in FPGAs by instantiating in each device, several copies of each cipher, mimicking a real case in which FPGAs are used as central node and need several encryptions to communicate with a number of connected cyber-physical devices.

The rest of the paper is organized as follows. Section II presents the problem of low energy encryption in general and discusses how the problem of low energy is addressed in FPGAs. Section III summarizes the main features of the considered algorithms. Section IV presents the setup we have used for the energy estimation. Section V reports the results of our experiments and concludes with some related discussion.

II. Related Works

Design of components which optimize the energy consumption is a well known problem for electronic designers. Energy aware design for reconfigurable hardware is however not very common. Works carried out in the past have addressed mainly the internal structures of FPGAs and tools used for synthesis, placement and routing. Kusse and Rabaey [8] proposed an FPGA module capable of reducing the energy by one order of magnitude compared to implementations existing at the time of publication. However, these approaches for reducing

978-1-4673-9406-2/15/$31.00 ©2015 IEEE

energy can be applied only by FPGA manufacturers and not by regular designers.

Gayasen et al. [9] proposed dividing the FPGA fabric into small regions and employing switching on/off the power supply of each region using a sleep transistor in order to conserve leakage energy. Reported results showed a significant reduction in the leakage power consumption. However, as in the case of the work of Kusse and Rabaey, the application of the techniques proposed in this work requires competency and access to low level details that are usually outside the scope and ability of regular designers.

Specifically for security applications, Goodman et al. [10] presented an implementation of a suite of cryptographic primitives over the integers modulo N, binary Galois fields and nonsupersingular elliptic curves over GF(2^n), showing performance comparable to the ones over dedicated hardware at the time of publication. The work represented an initial step towards the direction of low energy designs for FPGA. However, it remained an isolated attempt.

The energy efficiency of lightweight block ciphers was instead well studied for ASIC. Kerckhof *et al.* [11] presented a comprehensive study comparing a number of algorithms using different metrics such as area, throughput, power, and energy, and applying state of the art techniques for reducing power consumption such as voltage scaling. However, the evaluation reported in the paper is at very high level and concentrates only on a specific implementation, without considering the effects on energy consumption of different design choices, such as size of the datapath, amount of serialization, or effects of architectural optimization applied at each level of the algorithm.

Batina *et al.* [12] explored area, power, and energy consumption of several recently-developed lightweight block ciphers and compared it with the AES algorithm, considering also possible optimizations for the non linear transformation. However, no possible optimization was considered for other transformations, and effects of other design choices, such as serialization. A comparison of the energy consumptions of fully and partially unrolled circuits with respect to the latency in the circuit was also carried out in [13], [14].

III. Overview of considered algorithms

In this paper, we have considered seven block ciphers of different flavors to make a fair evaluation of the energy consumption figures. We have considered both SPN and Feistel architectures, and ciphers with both algebraically simple and complicated round functions. In the following part of this section we highlight the characteristics of the ciphers evaluated in this paper.

1) AES 128: The Advanced Encryption Standard [15] is arguably the most popular and widely studied block cipher. It has a simple SPN type round function. We analyze AES 128 which supports 128-bit blocks and Keys. Each round consists of the following operations: SubBytes, ShiftRows, MixColumn, and AddRoundKey. The operation SubBytes applies the 8-bit AES S-Box to each byte of the state. The permutation layer consists of ShiftRows and MixColumn operations. The ShiftRows operation simultaneously rotates the i^{th} row of the state by i bytes. The MixColumn operation multiplies each column of the state by an MDS matrix over $GF(2^8)$.
2) Present: Present [16] is a 64-bit block cipher which has an SPN type round function. It has recently been adopted as a standard in ISO/IEC 29192-2. The cipher specifications allow for both 80 and 128-bit Key, but we will only focus on the 80-bit version in this work. The only non-linear component in the round function is the 4-bit S-box, which is applied in parallel to each of the sixteen nibbles of the 64-bit state after the RoundKey addition. Thereafter the state-bits are rearranged by a permutation layer (in hardware this achieved at zero cost to energy and gate area by simple crossing of wires).
3) LED 128: LED [17] is a 64-bit block cipher with an SPN type round function. It allows for Keys of size 64, 80, 128 bits. We will concentrate on the 128-bit version. Its design is very much like AES, with each round consisting of an AddConstant, SubBytes, ShiftRows and MixColumn operation. It uses the same S-box as Present and the same ShiftRows operation as AES 128. Although it has no KeySchedule operation, the most significant bits and the least significant bits of the Key are alternately added to the state after every 4 rounds.
4) Prince: Prince [18] is a 64-bit block cipher with an SPN type round function. It allows for a 128-bit Key but does not use any KeyScheduling logic. Prince is based on the FX construction: The 128-bit Key is divided into the most and least significant 8-byte blocks k_0, k_1 and a key k_0' is computed from them by a simple rotate and add operation. k_0 and k_0' are used as whitening keys, and k_1 is used as the RoundKey in every round. The cipher uses three types of Round functions: Forward, Middle and Inverse. The Forward round consists of a SubBytes, MixColumn and addition of a Round constant and RoundKey. The Middle round consists of a SubBytes, MixColumn and Inverse SubBytes layer. The Inverse rounds are structurally and functionally the opposite of the Forward round. As a result, the Prince encryption operation is basically an involution.
5) Twine: Twine [19] is a 64-bit block cipher with a Type 2 Feistel round function. Here too the cipher allows 80 and 128-bit Keys, but we will focus on the 80-bit version. The state is divided into 8 bytes. The round function consists of adding a nibble of the RoundKey to the most significant nibble of each byte of the state, passing it through a 4-bit S-box and then adding the result to the least significant nibble of the state byte. Thereafter the bytes are rearranged through a permutation layer.
6) Piccolo: Piccolo [20] is a 64-bit block cipher with support for 80 and 128 bit keys, but we will focus on the 80-bit version. It has a Feistel round function which first divides the current state into 2 blocks of 32 bits each. The most significant 16 bits of each block is passed through an F-function and added to the least significant 16 bits and a portion of the RoundKey. The bytes are then shuffled by a permutation layer. The F-function applies a 4-bit S-box to each of its 4 input nibbles, passes the output through a linear MixColumn layer, and performs another round of substitution by applying 4 S-boxes in parallel to each nibble.
7) Simon 64/96 : The Simon and Speck family of block ciphers is a family of lightweight ciphers proposed in [21], with support for various block and key sizes. In this work, we will concentrate on Simon 64/96, which has a 64-bit block length, 96-bit key and a Feistel type round function. Unlike most other lightweight ciphers, Simon does not employ any

Substitution table but generates non-linearity by employing bitwise and operations. The round function first divides the current state into two four byte blocks x_H and x_L and adds together the following values $(x_H \lll 1) \cdot (x_H \lll 8)$, $x_H \lll 2$, x_L and the current RoundKey. The result of this operation and x_H are taken as the new state for the next round.

For the purpose of evaluation, for each cipher, we choose the keysize which is likely to yield the most energy efficient configuration for that particular cipher. For example LED allows keys of size $64, 80, 128$. We chose to evaluate the version of LED with 128 bit key because **a)** due to a complicated round key selection process, the 128 bit version is more energy efficient than the 80 bit version, **b)** we left out the version with 64 bit key, because 64-bit security is no longer considered secure for most applications. Similarly we chose the version of Present with 80 bit key rather than the 128 bit version, since the 80 bit version offers more energy efficiency.

IV. Experimental Setup

As far as power/energy consumption is concerned, in general FPGAs have a wide range of very different internal architectures that result in non-linear relationship with ASICs. Usually a minimum of two voltages are needed to power FPGAs: one for the core (1.0V to 2.5V typ.) and one for the "I/Os" (3.3V typ.). Many FPGAs also require a third low-noise, low-ripple voltage to provide power to the auxiliary circuits. Typical voltages are 2.5V or 3.3V depending on the individual FPGA family. Operating current for each of these voltages is also not fixed and depends upon many application-related factors, such as FPGA speed, capacity utilization, and the like.

In [22], a comprehensive comparison of lightweight metrics like area and power consumption between FPGAs and ASICs built with the same 90 nm CMOS technology was done. For circuits containing only look-up table-based logic and flip-flops, the ratio of silicon area required to implement them in FPGAs and ASICs was found to be on average 35. The dynamic power consumption ratio between FPGAs and ASICs was approximately 14 times. The corresponding ratio of the static power consumptions varied from 87 to 5.4, and so the authors could not provide any meaningful comparison of the static powers consumed in the 2 cases.

The issue of static power is a significant one, especially when considering the case of implementing lightweight block ciphers on FPGA devices. Most lightweight block ciphers would consume around 1% of the total logic resources in a standard FPGA device. However, the Xilinx Power Analyzer reports the static power consumed in the entire device (i.e. even in the unused part), which generally is much higher than the total dynamic power consumed in the single instance of the block cipher. In order to get a meaningful comparison of the static and dynamic power consumption, we implemented multiple instances of the given block cipher so that around $(2/3)^{rd}$ of the total resources of the device are utilized. The power/energy consumed in each instance of the cipher is then computed as a ratio of the total power/energy consumed and the number of implemented instances of the cipher. This scenario finds practical use in the internet of things, where a central node, implemented using FPGA, is connected with a number of other devices. In this scenario, each device will have its data encrypted and sent to the central node, and it is very likely that several devices will use the same algorithm. To increase the performance, in the central node, several implementations of the same algorithm are instantiated. In this way, the central node can encrypt and decrypt data coming from several nodes in parallel. A similar scenario is the one of a central node connected to several cyber-physical systems in a smart grid.

For the experiments presented in paper, we have implemented designs including multiple instances of each of the seven considered lightweight block ciphers. Our target platform was the Xc6slx45t device of the Spartan 6 family. For the experimental evaluation the following design flow was adhered to:

1) The design was first implemented in the RTL level. A functional simulation of was done using Mentorgraphics ModelSim SE software.
2) The design was synthesized, mapped, placed and routed using the Xilinx ISE design tools.
3) A timing simulation on the placed and routed design was done using the ISIM simulator using an external testbench. A switching activity (SAIF) file, recording the 0-1/1-0 transition statistics of all the nets, was also created in the process.
4) The Xilinx Power Analyzer software was used to estimate the average power consumption of the device using the switching activity information in the SAIF file.
5) Energy was then calculated as the product of the average power and the total time taken for one encryption.

The implementations were simulated at the same operating frequency of 10 MHz. This frequency was selected since the clock period was sufficiently larger than the critical paths of the 4-round unrolled implementations of all the ciphers.

V. Results

In this section we present and discuss the results of our experiments. Several techniques were designed specifically for cryptographic algorithms allowing us to implement them in low energy fashion. Some of them were specifically tailored for the target algorithm, some others were more generic and could be applied to a range of them. Some others were generic and could be applied to all the algorithms. Among the latter, a technique which could be very effective in reducing the energy consumption of block cipher consists in unrolling several rounds and computing them in a single clock cycle. The optimal amount of unrolled rounds depends from a large number of parameters, and can significantly vary from one algorithm to the other.

As this technique appears to be one of the most suitable for FPGAs, we explored if and how the amount of unrolled rounds would affect the energy consumption of lightweight ciphers implemented in reconfigurable hardware. To carry out this exploration, we unrolled up to four rounds for all the ciphers except Prince, due to its non-linear structure.

Table I reports the overall result of our experiments. In the last column, it is possible to see the the energy consumed by each instance. The number of instances which lead to the minimum energy consumption are indicated in bold. Some algorithms have a minimum energy consumption when two rounds are instantiated (Simon 64/96 and Present), while for the others (AES 128, LED 128, Twine, Piccolo), there is a direct variation between the increase of the unrolled rounds and the energy consumed.

These results are in line with the ones reported in literature for ASIC [14]. The only algorithm which shows a different trend

#	Cipher	Blocksize/ Keysize	Round Type	# Unrolled rounds (r)	# Cycles	Total Power mW	Total Energy nJ	# Instances	Device Utilization # Regs	#LUTs	#Slices	Energy per Instance (nJ)
1	AES 128	128/128	SPN	1	11	216.51	238.20	12	4752	17301	4997	**19.85**
				2	6	486.04	291.60	8	3138	19713	5341	36.45
				3	5	637.54	318.78	6	2352	18959	5202	53.13
				4	4	787.91	315.16	4	1572	16393	4536	78.79
2	LED 128	128/128	SPN	1	50	210.82	1054.20	60	11880	20711	5975	**17.57**
				2	26	373.79	971.95	35	6989	18931	5612	27.77
				4	14	514.19	719.85	15	2955	14373	4240	47.99
3	Prince	64 /128	SPN	1	13	188.28	244.80	40	8299	21644	6197	**6.12**
4	Present	64/80	SPN	1	33	130.13	429.60	80	18320	18778	5827	5.37
				2	17	140.60	238.80	60	13740	20209	5939	**3.98**
				3	12	167.16	200.80	40	9160	14760	4382	5.02
				4	9	181.78	163.80	30	6960	13840	4444	5.46
5	Piccolo	64/80	Feistel	1	26	156.98	408.00	60	8940	17800	5219	**6.80**
				2	14	204.14	285.95	35	5355	15549	4950	8.17
				3	10	328.58	328.50	25	3918	15356	4709	13.14
				4	8	395.07	316.00	20	3120	14569	4348	15.80
6	Twine	64/80	Feistel	1	37	96.51	357.00	60	13800	16349	5424	**5.95**
				2	19	153.30	291.20	40	9200	15852	5015	7.28
				3	13	195.91	254.70	30	6900	16042	5111	8.49
				4	10	233.41	233.50	25	5725	15627	4961	9.34
7	Simon 64/96	64/96	Feistel	1	43	94.43	406.20	60	15960	16066	5443	6.77
				2	22	120.63	265.20	40	10680	14608	4918	**6.63**
				3	15	137.85	206.70	30	7860	13933	4612	6.89
				4	12	226.17	271.25	25	6829	15904	4933	10.85

TABLE I: A comparison of Energy consumptions of 7 Lightweight Block Ciphers. Bold indicates the minimum energy consumption of each algorithm

compared to its ASIC counterpart is Twine. In ASIC, Twine consumed the minimum amount of energy when two rounds were unrolled. In FPGAs, the energy consumption is directly related with the amount of instantiated rounds.

A graphical representation of the results is reported in Figure 1, where the energy consumed as a function of the number of unrolled rounds is depicted. From the figure, it is clearly visible that the most energy efficient algorithm is Present and that the design where two rounds are unrolled is the best one. Also, it can be seen that, except for AES 128, Piccolo and LED 128, the difference in energy consumption when unrolling multiple rounds is minimal. This fact is due to the high leakage which FPGAs have.

The separation of the dynamic and the total energy for each algorithm is depicted in Figure 2. It can be noticed that despite the fact that the static energy consumption is significant, it is not sufficiently large to overshadow the difference in energy consumption due to unrolling multiple rounds.

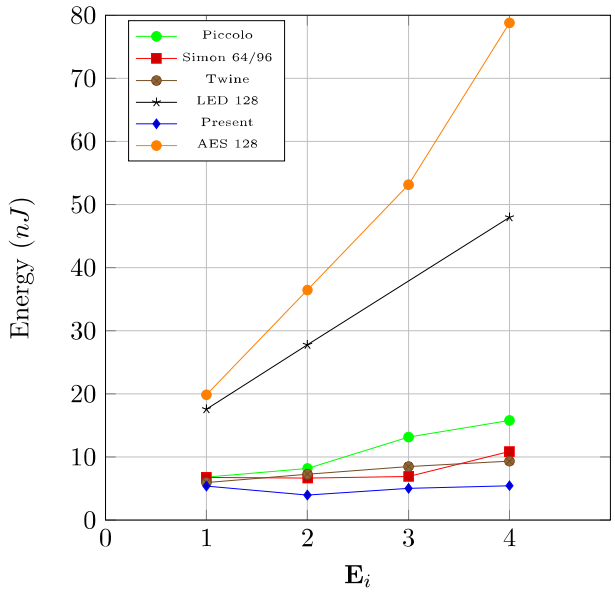


Fig. 1: Energy consumption vs Number of Unrolled rounds (all algorithms)

VI. Conclusions

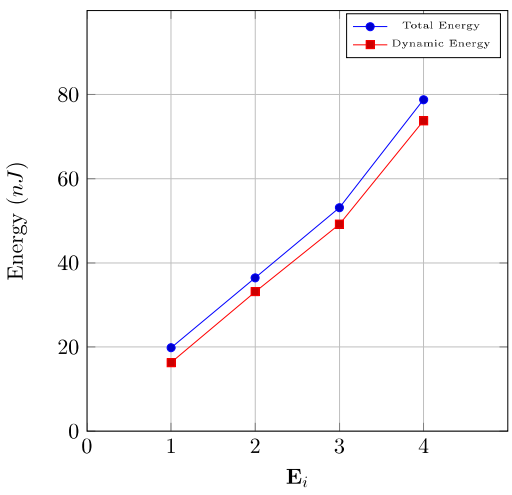
In this paper, for the first time, we tackled the problem of energy efficiency of lightweight block ciphers implemented on reconfigurable devices and we explored the effects that round unrolling might have on the energy consumption. We considered, as case study, a central node in a network connecting several devices of the internet of things and we instantiated multiple instances of the same algorithm on the target FPGA.

Our results showed that Present is the most energy efficient algorithm and that, even if the energy consumption is significantly affected by the high leakage of FPGAs, the relation between the amount of unrolled rounds and the energy consumption is in line with results reported in literature for dedicated hardware.

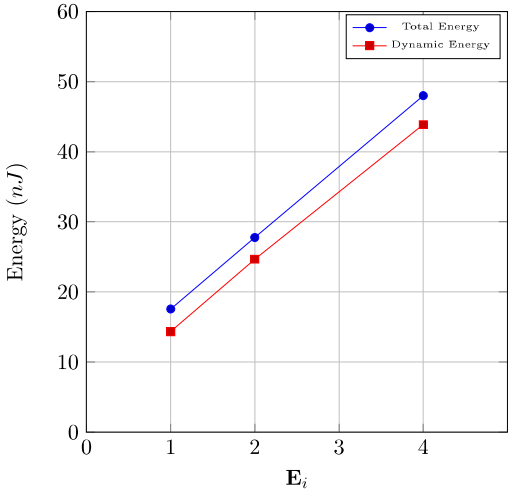
Our work represents a first step towards energy efficient implementations of cryptographic algorithms on reconfigurable hardware, a research area which is of crucial importance for the enabling reliable deployment of an efficient internet of things and for a pervasive diffusion of cyber-physical systems.

References

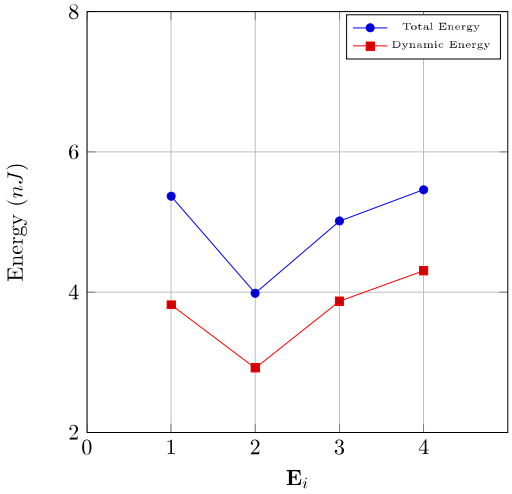
[1] P. Chodowiec and K. Gaj, "Very compact fpga implementation of the aes algorithm," in *Cryptographic Hardware and Embedded Systems-CHES 2003*. Springer, 2003, pp. 319–333.



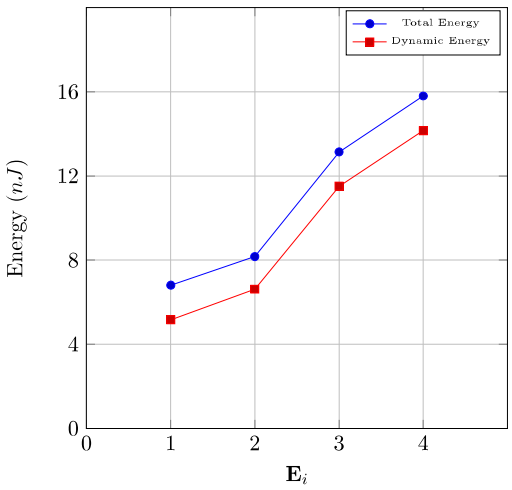
(a) AES 128



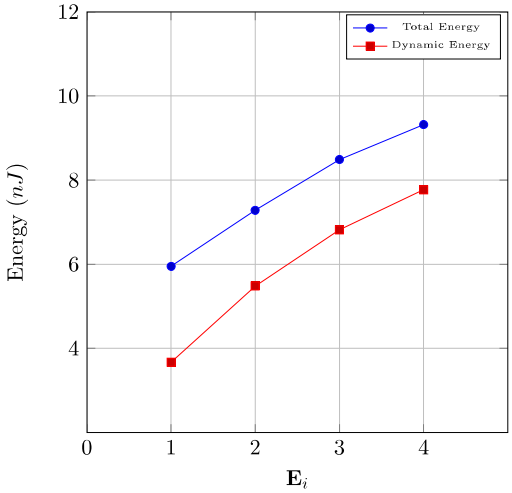
(b) LED 128



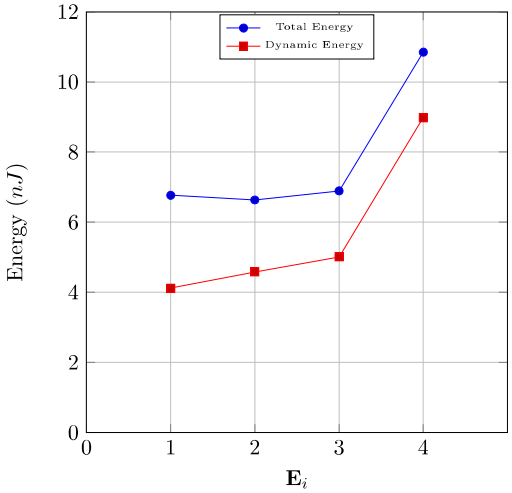
(c) Present



(d) Piccolo



(e) Twine



(f) Simon

Fig. 2: Energy consumption vs Number of Unrolled rounds

[2] G. Rouvroy, F. Standaert, J. Quisquater, and J. Legat, "Compact and efficient encryption/decryption module for FPGA implementation of the AES rijndael very well suited for small embedded applications," in *International Conference on Information Technology: Coding and Computing (ITCC'04), Volume 2, April 5-7, 2004, Las Vegas, Nevada, USA*. IEEE Computer Society, 2004, pp. 583–587. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/ITCC.2004.1286716

[3] S. Kumar, C. Paar, J. Pelzl, G. Pfeiffer, and M. Schimmler, "Breaking ciphers with copacobana–a cost-optimized parallel code breaker," in *Proceedings of the 8th international conference on Cryptographic Hardware and Embedded Systems*. Springer-Verlag, 2006, pp. 101–118.

[4] T. Güneysu and A. Moradi, "Generic side-channel countermeasures for reconfigurable devices," in *Cryptographic Hardware and Embedded Systems–CHES 2011*. Springer, 2011, pp. 33–48.

[5] F. Regazzoni, Y. Wang, F.-X. Standaert *et al.*, "Fpga implementations of the aes masked against power analysis attacks," *Proceedings of COSADE*, vol. 2011, pp. 56–66, 2011.

[6] P. Sasdrich, O. Mischke, A. Moradi, and T. Güneysu, "Side-channel protection by randomizing look-up tables on reconfigurable hardware," *proceedings of COSADE*, 2015.

[7] H. Guntur, J. Ishii, and A. Satoh, "Side-channel AttacK User Reference Architecture board SAKURA-G," in *Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference on*, Oct 2014, pp. 271–274.

[8] E. Kusse and J. Rabaey, "Low-energy embedded fpga structures," in *Low Power Electronics and Design, 1998. Proceedings. 1998 International Symposium on*. IEEE, 1998, pp. 155–160.

[9] A. Gayasen, Y. Tsai, N. Vijaykrishnan, M. Kandemir, M. J. Irwin, and T. Tuan, "Reducing leakage energy in fpgas using region-constrained placement," in *Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays*. ACM, 2004, pp. 51–58.

[10] J. Goodman and A. P. Chandrakasan, "An energy-efficient reconfigurable public-key cryptography processor," *Solid-State Circuits, IEEE Journal of*, vol. 36, no. 11, pp. 1808–1820, 2001.

[11] S. Kerckhof, F. Durvaux, C. Hocquet, D. Bol, and F.-X. Standaert, "Towards green cryptography: a comparison of lightweight ciphers from the energy viewpoint," in *Cryptographic Hardware and Embedded Systems–CHES 2012*. Springer, 2012, pp. 390–407.

[12] L. Batina, A. Das, B. Ege, E. B. Kavun, N. Mentens, C. Paar, I. Verbauwhede, and T. Yalçın, "Dietary recommendations for lightweight block ciphers: Power, energy and area analysis of recently developed architectures," in *Radio Frequency Identification*. Springer, 2013, pp. 103–112.

[13] M. Knežević, V. Nikov, and P. Rombouts, "Low-latency encryption–is lightweight= light+ wait?" in *Cryptographic Hardware and Embedded Systems–CHES 2012*. Springer, 2012, pp. 426–446.

[14] S. Banik, A. Bogdanov, and F. Regazzoni, "Exploring energy efficiency of lightweight block ciphers," in *Selected Areas in Cryptography–SAC 2015*. Springer, 2015.

[15] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*. Berlin, Heidelberg, New York: Springer Verlag, 2002.

[16] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher," in *CHES*, ser. Lecture Notes in Computer Science, P. Paillier and I. Verbauwhede, Eds., vol. 4727. Springer, 2007, pp. 450–466. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74735-2_31

[17] J. Guo, T. Peyrin, A. Poschmann, and M. J. B. Robshaw, "The LED Block Cipher." in *CHES*, ser. Lecture Notes in Computer Science, B. Preneel and T. Takagi, Eds., vol. 6917. Springer, 2011, pp. 326–341. [Online]. Available: http://dblp.uni-trier.de/db/conf/ches/ches2011.html#GuoPPR11;http://dx.doi.org/10.1007/978-3-642-23951-9_22;http://www.bibsonomy.org/bibtex/211cbbf7a6a56388196f9775b6da9ed9a/dblp

[18] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knezevic, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. S. Thomsen, and T. Yalçin, "PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract." in *ASIACRYPT*, ser. Lecture Notes in Computer Science, X. Wang and K. Sako, Eds., vol. 7658. Springer, 2012, pp. 208–225. [Online]. Available: http://dblp.uni-trier.de/db/conf/asiacrypt/asiacrypt2012.html#BorghoffCGKKKLNPRRTY12

[19] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, "TWINE : A Lightweight Block Cipher for Multiple Platforms." in *Selected Areas in Cryptography*, ser. Lecture Notes in Computer Science, L. R. Knudsen and H. Wu, Eds., vol. 7707. Springer, 2012, pp. 339–354. [Online]. Available: http://dblp.uni-trier.de/db/conf/sacrypt/sacrypt2012.html#SuzakiMMK12

[20] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, "Piccolo: An Ultra-Lightweight Blockcipher." in *CHES*, ser. Lecture Notes in Computer Science, B. Preneel and T. Takagi, Eds., vol. 6917. Springer, 2011, pp. 342–357. [Online]. Available: http://dblp.uni-trier.de/db/conf/ches/ches2011.html#ShibutaniIHMAS11

[21] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "SIMON and SPECK: Block Ciphers for the Internet of Things." *IACR Cryptology ePrint Archive*, vol. 2015, p. 585, 2015. [Online]. Available: http://dblp.uni-trier.de/db/journals/iacr/iacr2015.html#BeaulieuSSTWW15;http://eprint.iacr.org/2015/585;http://www.bibsonomy.org/bibtex/210c431e11e383320d2d861920990dc12/dblp

[22] I. Kuon and J. Rose, "Measuring the Gap Between FPGAs and ASICs," *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, vol. 26, no. 2, pp. 203–215, Feb. 2007. [Online]. Available: http://dx.doi.org/10.1109/TCAD.2006.884574

[23] B. Preneel and T. Takagi, Eds., *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, ser. Lecture Notes in Computer Science, vol. 6917. Springer, 2011. [Online]. Available: http://dblp.uni-trier.de/db/conf/ches/ches2011.html