Exploring Energy Efficiency of Lightweight Block Ciphers

Subhadeep Banik[1(✉)], Andrey Bogdanov[1], and Francesco Regazzoni[2]

[1] DTU Compute, Technical University of Denmark, Kongens Lyngby, Denmark
{subb,anbog}@dtu.dk
[2] ALARI, University of Lugano, Lugano, Switzerland
regazzoni@alari.ch

Abstract. In the last few years, the field of lightweight cryptography has seen an influx in the number of block ciphers and hash functions being proposed. One of the metrics that define a good lightweight design is the energy consumed per unit operation of the algorithm. For block ciphers, this operation is the encryption of one plaintext. By studying the energy consumption model of a CMOS gate, we arrive at the conclusion that the energy consumed per cycle during the encryption operation of an r-round unrolled architecture of any block cipher is a quadratic function in r. We then apply our model to 9 well known lightweight block ciphers, and thereby try to predict the optimal value of r at which an r-round unrolled architecture for a cipher is likely to be most energy efficient. We also try to relate our results to some physical design parameters like the signal delay across a round and algorithmic parameters like the number of rounds taken to achieve full diffusion of a difference in the plaintext/key.

Keywords: AES · Lightweight block cipher · Low power/energy circuits

1 Introduction

In the last few years, we have assisted to the pervasive diffusion of embedded and smart devices, touching every aspect of our lives. These devices, are often used for sensitive applications, such as the ones related to access control, banking and health, and they are often connected to create what is called internet of things. The security needs of these applications lead to the creation of the research area of Lightweight Cryptography, which aims at designing and implementing security primitives fitting the needs of extremely constrained devices. Two main approaches can be followed to achieve this goal: designing new algorithms to be implemented into constrained devices, or trying to implement standards and known algorithms in a lightweight fashion, eventually relaxing the performance constraints. Examples of the first approach are the large number of algorithms proposed in recent years, such as HIGHT [16], KATAN [8], Klein [13], LED [14], Noekeon [10], Present [6], Piccolo [21], Prince [7], Simon/Speck [3]

O. Dunkelman and L. Keliher (Eds.): SAC 2015, LNCS 9566, pp. 178–194, 2016.
DOI: 10.1007/978-3-319-31301-6_10

and TWINE [22]. Possible examples of the second approach are implementations of the Advanced Encryption Standard algorithm (AES) [11], SHA-256 [1], or Keccak [4].

Focusing on block ciphers in particular, it is important to notice that AES still remains the preferred choice for providing security also in constrained devices, even if some lightweight algorithms are now standardized. For this reason, several implementations of AES and its basic transformations (such as S-boxes) targeting low area and low power were proposed in the past, for example, the implementation of Feldhofer *et al.* [12] and the one of Moradi *et al.* [19]. The first design is based on a 8-bit datapath, and occupies approximately 3400 Gate Equivalents (GE). The second design features a mixed data path and requires approximately 2400 GE. The work of Hocquet *et al.* [15] discusses the silicon implementation of low power AES. The authors showed that by exploiting technological advances and algorithmic optimization the AES core, can consume as little as 740 pJ per encryption.

Despite a large number of previous works targeting area and power, only limited efforts were devoted to the optimization of the energy parameter. Energy and power are, for obvious reasons, correlated parameters. Power is the amount of energy consumed per unit time or simply the rate of energy consumption. More specifically, energy consumption is a measure of the total electrical effort expended during the execution of an operation, and the total energy consumed is essentially the time integral of power. However, being directly linked with the battery life or the amount of electrical work to be harvested, energy, rather than power, would become a more relevant parameter for evaluating the suitability of a design. In fact, energy is a much stricter constraint for future cyber-physical systems as well as for the next generation of implantable devices.

Designing for low energy can be significantly different than designing for low power. Furthermore, there is no guarantee that low power architectures would lead to low energy consumption. For instance, block ciphers implemented using smaller datapath and aggressively exploiting serialization to reuse components, result generally in smaller power consumption compared to round based designs having datapath as large as the blocksize of the cipher. However, serial implementations have high latency, which can be significantly larger compared to round based designs. As a result, the energy consumed per encryption for serial designs could be much higher than the corresponding figure for round based designs.

Starting with the AES algorithm, in this work, we carry out a complete exploration of the implementation choice of block ciphers concentrating on their energy consumption, discussing and evaluating the design choice of each round transformation, and the best trade-off between datapath and serialization. From the detailed analysis of this exploration, we extract a model for the energy consumption of a circuit, using as reference, a number of lightweight algorithms recently proposed.

The most significant previous works on this area are the one of Kerckhof *et al.* [17] and the one of Batina *et al.* [2]. The first work, addresses the problem of efficiency for lightweight designs. The authors present 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 stage of the algorithm. The second work explores area, power, and energy consumption of several recently-developed lightweight block ciphers and compares it with the AES algorithm, considering also possible optimization for the non linear transformation. However, no possible optimization was considered for the other transformations, and effects of other design choices, such as serialization were not considered in the work. In another work [18], a comparison of the energy consumptions of fully and partially unrolled circuits was done with respect to the latency in the circuit.

1.1 Contribution and Organization of the Paper

In this paper we complete the analysis started with these works, looking at all the parameters which might affect the energy consumption of a design. We start with the case of AES and investigate how the variation in **(a)** the architectural design of the individual components (S-box, MixColumn), **(b)** frequency of the clock signal and **(c)** serializing or unrolling the design can affect the energy consumption. Furthermore, starting with the detailed analysis of our exploration, we build an energy model for any r-round unrolled architecture of block ciphers. We prove that if all other factors are constant, then the total energy consumed per encryption in an r-round unrolled circuit is quadratic in r. We validate our model by estimating the energy consumed by several lightweight algorithms and comparing it with the figures obtained by simulating their implementations.

The remainder of the paper is organized as follows. Section 2 presents, as a motivating example a detailed study of the AES algorithm from the energy point of view. Section 3 presents our model for estimating the energy consumption of a block cipher, discussing how the contribution of each component is modeled and included into the overall energy consumption equation. Section 4 reports how our model is validated using a number of lightweight algorithms. Section 5, tabulates the final energy figures for all the block ciphers that we have considered, and relates these results to physical parameters like critical path and algorithmic parameters like the minimum number of rounds required to achieve full diffusion of a difference introduced in the plaintext or key. Section 6 concludes the paper.

2 A Case Study of Energy Consumption of AES 128

In this section, we investigate how the choice of architecture can affect the energy performance of implementations of AES 128. In our experiments, we considered three factors that would likely affect the energy metric of the encryption algorithm.

(a) Architecture of S-Box/MixColumn: It is known that the Canright architecture [9] is the one of the most compact representations of the AES S-box in terms of gate area. However it is unlikely to be the most efficient energy-wise. We then experimented with a Lookup table based S-box. However, we found that the Decoder-Switch-Encoder (DSE) architecture [5] is the most energy-efficient. We also considered two different variants of the MixColumn architecture. Considering AES MixColumn to be a linear map from $\{0,1\}^{32} \rightarrow \{0,1\}^{32}$, it can be composed with 152 xor gates by following the mathematical definition. However, as shown by [20], the outputs of several xor gates can be reused and it is possible to get a compact design in 108 gates. The 108 gate variant is likely to be more energy efficient as it provides a balanced datapath and also uses less gates. In Table 1, we present the area and energy per encryption figure for the round based designs for a combination of all the above choices of S-boxes/MixColumns at an operating frequency of 10 MHz (using the standard cell library of the STM 90 nm low leakage process). Clearly the DSE S-box and the 108 gate MixColumn is optimal in terms of energy efficiency.

(b) Clock Frequency: As already pointed out in [17], the energy consumption required to compute an encryption operation should be independent of frequency of operation, as energy is a metric which is a measure of the total switching activity of a circuit during the process. This is true for sufficiently high frequencies, where the total leakage energy consumed by the system is low over the total number of cycles required for encryption. However for the STM 90 nm low leakage process, at frequencies lower than 1 MHz, leakage energy naturally starts to play a significant role, thereby increasing the energy consumption. Furthermore, gates selected by synthesis tools for meeting a high clock frequency can be significantly different from the ones selected for achieving a low clock frequency. The selection of different gates, which is an indirect consequence of the clock frequency, would also affect the energy consumption. In Fig. 1, we present the variation in the energy consumption, for the round based AES architecture (using the DSE S-box and the 108 gate MixColumn) for frequencies ranging from 100 KHz to 100 MHz. We can see that for frequencies higher than 1 MHz, the energy consumption is more or less invariant with respect to frequency.

(c) Width of the Data path/Unrolled Design: We performed our experiments with numerous serialized implementations of AES in which the datapath width varies from 8 to 32 to 64 bits. For our experiments, we used the 8-bit serial architecture described in [19]. For the 32-bit serialized datapath we used three architectures described as follows

1. $\boldsymbol{A_1}$: In this architecture every round is completed in 9 cycles: 4 for the Substitution operation, 4 for the MixColumn operation and 1 for Shift row. This architecture takes 94 cycles to complete one encryption.
2. $\boldsymbol{A_2}$: In this architecture every round is completed in 5 cycles: 4 cycles are used for the combined Substitution operation and MixColumn operation and 1 is used for Shift row. This architecture takes 54 cycles to complete one encryption.

Table 1. Area, energy figures for round based AES 128 using different component architectures

#	S-box	MixColumn	Area (in GE)	Energy (in pJ)	Energy/bit (in pJ)
1	LUT	152 gates	13836.2	797.2	6.23
2	LUT	108 gates	13647.9	755.3	5.90
3	Canright	152 gates	8127.9	753.6	5.89
4	Canright	108 gates	7872.5	708.5	5.53
5	DSE	152 gates	12601.7	377.5	2.95
6	DSE	108 gates	12459.0	**350.7**	2.74

3. $\boldsymbol{A_3}$: In this architecture every round is completed in 4 cycles. Extra multiplexers are used to ensure that each clock cycle performs the Shift Row, Substitution and MixColumn operation on a given chunk of 32 bit data. This architecture takes 44 cycles to complete one encryption.

Similarly, we used three architectures B_1, B_2 and B_3 for the 64-bit serial design that takes 52, 32 and 22 cycles respectively. Thereafter we continue to explore lower latency designs like the round based architecture and the 2, 3, 4, 5, 10 round unrolled architectures.

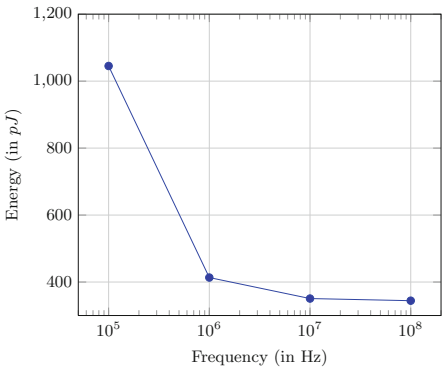


Fig. 1. Energy consumption for round based AES 128 over a range of clock frequencies

We present the area and energy per encryption figure for all the architectures using the DSE S-box, and the 108 gate MixColumn for designs synthesized with the standard cell library based on the STM 90 nm logic process, at a clock frequency of 10 MHz in Table 2. We found that the round based implementation of AES 128 is the most energy efficient. Since the serialized architectures take longer time to complete an encryption operation it was expected that they would consume more energy, but the fact that the round based design was better in

terms of energy than its unrolled counterparts was certainly an interesting result. To understand the reason for this we first need to understand which components of the architecture are consuming the most energy. A breakdown of this energy consumption, by percentage of the total energy, for the various components is shown in Fig. 2.

Table 2. Area and energy figures for different AES 128 architectures

#	Design	Area (in GE)	#Cycles	Energy (pJ)	Energy/bit (pJ)
1	8-bit	2722.0	226	1913.1	14.94
2	32-bit (A_1)	4069.7	94	1123.3	8.77
	32-bit (A_2)	4061.8	54	819.2	6.40
	32-bit (A_3)	5528.4	44	801.7	6.26
3	64-bit (B_1)	6380.9	52	1018.7	7.96
	64-bit (B_2)	6362.6	32	869.8	6.79
	64-bit (B_3)	7747.5	22	616.2	4.81
4	Round based	12459.0	11	**350.7**	2.74
5	2-round	22842.3	6	593.6	4.64
6	3-round	32731.9	5	1043.0	8.15
7	4-round	43641.1	4	1416.5	11.07
8	5-round	53998.7	3	1634.4	12.77
9	10-round	101216.7	1	2129.5	16.64

We can see that the Substitution layer consumes the most part of the energy budget (24 % and 36.3 %) in the round based design and the 2-round unrolled designs respectively. However we also find that in the 2-round unrolled design, the second round functions (Substitution Layer, MixColumn, Add round key and round key logic) consume more energy than the first. To understand the reason for this trend, we need to study the energy consumption model in CMOS gates, and start to analyze the situation from there.

3 CMOS Energy Consumption Model

Currently, static CMOS is the dominant technology used for producing electronic devices. Two main reasons were behind the widespread diffusion of static CMOS: its robustness against noise and its limited static power consumption. With the shrinking of technologies, static power consumption of CMOS is increasing. Nevertheless, static CMOS is likely to continue to be the preferred technology for electronic fabrications in the foreseeable future.

Energy consumption of a static CMOS gate, is defined by the following equation:

$$E_{gate} = E_{load} + E_{sc} + E_{leakage}$$

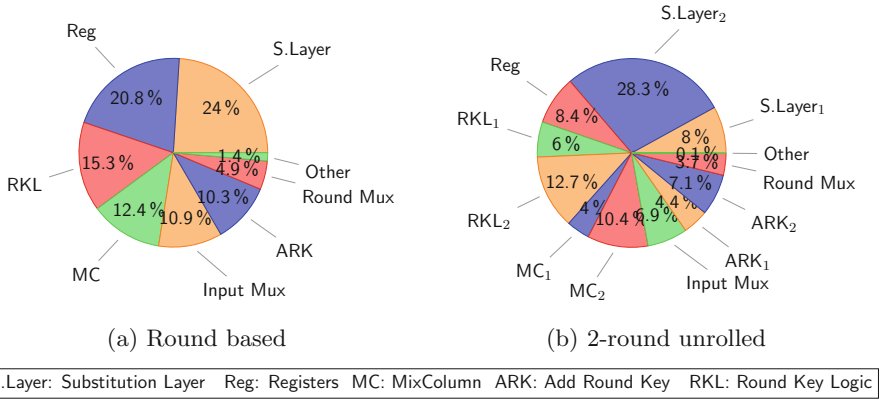


Fig. 2. Energy shares for the round based and 2-round unrolled AES 128

where E_{sc} is the energy due to the short-circuit current. $E_{leakage}$ is the energy consumed due to the sub-threshold leakage current when the transistor is OFF. This component is usually small, but is gaining importance as the technology scaling makes the sub-threshold leakage more significant. E_{load} is the energy dissipated for charging and discharging the capacitive load C_L of a gate when output transitions occur.

Hardware implementations of any cryptographic primitive consist of a number of registers and logic blocks connected together as required by the specifications of the algorithm itself. Block ciphers based on SPN or Feistel designs, consist in particular of a round function and round key generation logic which transform a plaintext and key into a ciphertext by iterating the round function for a specific amount of rounds. Consider an ideal block cipher E operating on a plaintext space $\{0,1\}^{L_p}$ and a key space $\{0,1\}^{L_k}$. Its hardware implementation, illustrated in Fig. 3, would include:

- **A.** A state register (SReg) and a key register (KReg) of L_p and L_k bits respectively, to store the intermediate states produced by the round function and the computed round keys.
- **B.** Two input multiplexers placed before the state register and the key register respectively used to control the updating of the state or the loading of the plaintext or the initial key.
- **C.** Depending on the choices of the designer, one or more instances of the round function (RF_i) and the round key generation logic (RK_i)
- **D.** Additional logic needed to generate control signals, round constants etc.

A designer, depending on the specific requirements of the target application, can implement the algorithm following different strategies. One of the most important decisions in this respect is the number of instances of the round function to be replicated in hardware. The smallest amount of replication happens when a single instance of the round function and the round key are instantiated: implementations following this style are called round based architectures.

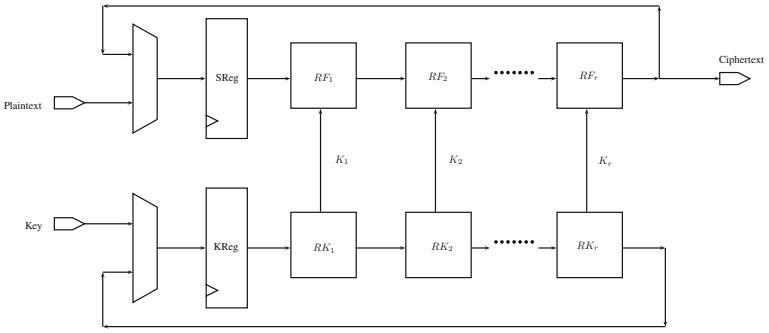


Fig. 3. Block cipher architecture

The round based architecture of a block cipher which has to be executed for R rounds, can be compute the result of an encryption in $R + 1$ clock cycles (1 cycle for the loading of the plaintext/key and the remaining R cycles for executing the R rounds). A designer can instantiate more than one round function, opting for an unrolled architecture. An r-round unrolled architecture consists of r instances of the round function and round key logic. Encryption on such a circuit would take $1 + \lceil \frac{R}{r} \rceil$ clock cycles.

The selection of the number of round functions instantiated depends on the specific optimization parameters. For instance, an r-round unrolled circuit for high values of r would require a smaller number of clock cycles to compute the encrypted ciphertext compared to a round based design. However, its power consumption is usually higher. It is thus interesting to investigate how the value of r affects the energy consumption for a given block cipher. To tackle this problem from a purely analytical point of view, one can make the following observations:

1. Assume that the designer has fixed the logic process and the frequency of operation of the circuit. Consider the input signal seen by the multiplexers i.e., outputs of RF_r and RK_r, respectively in an r-round design. If τ_F, τ_K represent the delays in each of the RF_i, RK_i blocks, then each multiplexer will see a signal that will be switching for around $r\tau_F, r\tau_K$ respectively in every clock cycle before stabilizing. If each of the muxes itself introduce a delay of τ_M, then their outputs will be switching for $r\tau_F + \tau_M, r\tau_K + \tau_M$ in every round. Since in a low leakage environment, the energy consumed is essentially the measure of the total number of logic switches, we can assume that the energy consumed in the each of the multiplexers is proportional to $r\tau_F + \tau_M, r\tau_K + \tau_M$ respectively. Let $E_{Mux,r}$ be used to denote the total energy drawn per cycle by the multiplexers in an r-round unrolled design.

Then we can write (α and β are constants of proportionality)

$$\begin{aligned} E_{Mux,r} - E_{Mux,1} &= \alpha \cdot [(r\tau_F + \tau_M) - (\tau_F + \tau_M)] \\ &\quad + \beta \cdot [(r\tau_K + \tau_M) - (\tau_K + \tau_M)] \\ &= (r-1) \cdot (\alpha\tau_F + \beta\tau_K) = (r-1) \cdot dE_{Mux}, \end{aligned}$$

where dE_{Mux} is therefore the difference between $E_{Mux,r}$ and $E_{Mux,r-1}$, i.e. energy consumed per cycle in the multiplexers in the r and $r-1$ round unrolled architectures. One can see that the $E_{Mux,1}, E_{Mux,2}, \ldots$ forms an arithmetic sequence with difference between successive terms equal to dE_{Mux}.

2. Similarly, we can derive the energy drawn by the registers. However, registers switch only at the positive/negative edge of the clock (assuming a synchronous design). If $E_{Reg,r}$ is the total energy per cycle drawn by the registers in the r-round architecture, we have

$$E_{Reg,r} = E_{Reg,1} + (r-1) \cdot dE_{Reg}.$$

However the value of incremental energy dE_{Reg} when compared to $E_{Reg,1}$ is generally much smaller.

3. By similar arguments, the energy consumed in each successive logic block RF_i and similarly RK_i is likely to constitute two arithmetic sequences. The total energy drawn per cycle by the r round function blocks, given by E_{RF} is

$$\begin{aligned} E_{RF} &= \sum_{i=1}^{r} E_{RF,i} = \sum_{i=1}^{r} E_{RF,1} + (i-1) \cdot dE_{RF} \\ &= rE_{RF,1} + \frac{r(r-1)}{2} \cdot dE_{RF}, \end{aligned}$$

and similarly, the total energy drawn per cycle by the r round key logic blocks is

$$E_{RK} = rE_{RK,1} + \frac{r(r-1)}{2} \cdot dE_{RK}$$

where $E_{RF,i}$ and $E_{RK,i}$ are the energy drawn per cycle by RF_i and RK_i respectively. dE_{RF}, dE_{RK} denote the incremental energy consumption per cycle between successive round function and round key logic blocks respectively. In deriving the above equations we have made the implicit assumption that the capacitive loads driven by the final blocks RF_r, RK_r are the same as the ones driven by the previous blocks RF_i, RK_i (for $i < r$). This, however, is not always true. For example, RF_r drives the multiplexer in front of the state register, and all of the previous RF_i blocks drive the subsequent RF_{i+1}. This may result in small deviation in the actual and the estimated energy consumed in the final block. However the deviation is usually negligible.

4. The energy drawn by the rest of the logic (E_{rem}) may or may not form a sequence with any special property for increasing values of r. This would

naturally depend on the specific algorithm of the block cipher. The value of this figure is usually a small fraction of the total energy drawn by the circuit: in the set of ciphers we have considered in this work, E_{rem} was never exceeding 5 % of the total energy budget.

Summing all the contributions, we can write the total energy E_r consumed per cycle in an r-round unrolled circuit as:

$$\begin{aligned} E_r &= E_{Mux,r} + E_{Reg,r} + E_{RK} + E_{RF} + E_{rem} \\ &= E_{Mux,1} + (r-1) \cdot dE_{Mux} + E_{Reg,1} + (r-1) \cdot dE_{Reg} \\ &+ r \cdot E_{RF,1} + \frac{r(r-1)}{2} \cdot dE_{RF} + r \cdot E_{RK,1} + \frac{r(r-1)}{2} \cdot dE_{RK} + E_{rem} \end{aligned}$$

E_r is a quadratic function in r in the form $Ar^2 + Br + C$, where

$$A = \frac{dE_{RF} + dE_{RK}}{2},\ B = E_{RF,1} + E_{RK,1} + dE_{Reg} + dE_{Mux} - \frac{dE_{RF} + dE_{RK}}{2}$$
$$C = E_{Reg,1} + E_{Mux,1} + E_{rem} - dE_{Reg} - dE_{Mux}.$$

To compute the total energy $\mathbf{E}_r$ which a particular implementation consumes to perform an encryption, the energy required for one round needs to be multiplied for total time required for the computation i.e. $\left(1 + \lceil \frac{R}{r} \rceil\right)$:

$$\mathbf{E}_r = E_r \cdot \left(1 + \left\lceil \frac{R}{r} \right\rceil\right) = (Ar^2 + Br + C) \cdot \left(1 + \left\lceil \frac{R}{r} \right\rceil\right) \tag{1}$$

As before, $\mathbf{E}_r$ is a function in r of the form $\alpha r^2 + \beta r + \gamma + \frac{\delta}{r}$. The analysis for a fully unrolled circuit is slightly different such circuits do not need registers used to store intermediate values. As a result, the total energy consumed by a fully unrolled design does not contain the $E_{Reg,r}$ component. Also, a fully unrolled circuit takes only a single clock cycle to complete an encryption.

4 Application of the Model

In this section we apply our model to determine the most energy efficient configuration for 9 lightweight block ciphers. For each algorithm, we measure **(a)** the parameters E_{Reg}, E_{Mux}, $E_{RF,1}, E_{RK_1}$, E_{rem} and **(b)** the energy differentials $dE_{RF}, dE_{RK}, \ldots$ by simulating the energy consumption of the round based and 2-round unrolled design. Using this data, we predict the energy consumption required for one encryption by changing the number of unrolled rounds. Thereafter, we determine the value of r which achieves the highest energy efficiency. Finally, we compare our predictions with the actual energy consumption estimated using a well recognized gate level power simulator.

Estimation of power consumption (and, as a consequence, energy consumption) can be carried out at different levels. A designer has to trade the desired

precision in the estimation with the time (and the level of circuit details) required for the simulation. A first approximation of power consumption can be achieved by simply counting the amount of switches which each node of the circuit makes during a given time period. This approach is extremely fast. However, the accuracy is very limited, as all the gates are assumed to consume the same amount of power. A better estimation can be obtained by collecting the switching activity of the circuit under test, obtained by simulating a significant and sufficiently large test bench, and annotating it back to the power estimation tool. In this way, the amount of switches is combined with the power fingerprint of each gate indicated in the technological library and produces a more precise estimation. The back-annotation of the switching activity can be carried out in different ways. The first and simpler, consists of annotating only the switching activity at the primary inputs. In this case, the tool estimates the switching of the internal gates. A more precise back annotation involves annotating the exact amount of switching of each gate, as produced by the simulation of a test bench. It is worth mentioning that most precise estimation of power consumption is obtained by simulating a circuit at SPICE level. This simulation, however, requires the availability of technological models (which are often not provided by the foundry) and needs significant amount of time to be carried out. For this reason, SPICE level simulation is not the preferred way to estimate power consumption. Power consumption estimation is also affected by the point in the design flow where it is carried out. A post-synthesized netlist contains all the information for estimating the power consumed by the gates, however it does not have information about the interconnecting wires. To obtain them, it is necessary complete the placement and the routing of the circuit, which is out of the scope of this work.

In this work, we are mainly concerned by the energy consumed by the gates. Hence, we carried out the energy estimation using the following design flow: The design was implemented at RTL level. A functional verification of the VHDL code was done using *Mentorgraphics ModelSim*. *Synopsys Design Compiler* was used to synthesize the RTL design. The switching activity of each gate of the circuit was collected by running post-synthesis simulation. The average power was obtained using *Synopsys Power Compiler*, using the back annotated switching activity. The energy was then computed as the product of the average power and the total time taken for one encryption.

For all the circuits, we set the operating Frequency at 10 MHz and the target library was the standard cell library of the STM 90 nm low leakage process. The operating frequency was fixed at 10 MHz since we have already established that at sufficiently high frequencies, the energy consumption of a circuit is invariant with frequency. We selected a set of 9 lightweight block ciphers of different design flavors. We classified them into two categories:

(a) Iterated ciphers are those all of whose round functions are similar. In this category we have AES 128, Noekeon, Present, Piccolo, TWINE and Simon 64/96. Such ciphers readily fit the model of energy consumption given by Eq. (1).

(b) Non-iterated ciphers are those whose round functions are not all similar. For example, in the cipher LED 128, the most significant bits and the least significant bits of the 128-bit key are alternately added to the state after every 4 rounds. So, in a round based design, to account for the addition of the round key once every four cycles, one needs to place a multiplexer/and gate to filter the key every fourth round. However, in a 2-round unrolled design, this filtering is not needed in the second round function. In a 3-round unrolled design, filtering would be needed in all the rounds, whereas in a 4-round unrolled design, filtering is not needed in any of the rounds. So, this is a cipher in which the structure of the round function varies widely from one architecture to another, we will call this a non-iterative cipher. Another example is Prince, in which 3 different type of round functions are used in the design itself. Finally we have the KATAN64 block cipher which is based on a bitwise Shift register. Since the cipher is based on a Shift register, its functioning is very different from the existing SPN/Feistel designs. Each round consists of the execution of a few simple Boolean Functions over only a limited number of bits of the current state. This is why we do not see a compounding of switching activity across rounds. Such ciphers do not readily fit the model for energy consumption as defined in Eq. (1). But the core logic remains the same, rounds further away from the register would consume more energy that the ones closer to it.

For all the iterated ciphers in our set we measured the values of $E_{Reg,1}, dE_{Reg}$, $E_{Mux,1}, dE_{Mux}$, $E_{RF,1}, dE_{RF}$, $E_{RK,1}$, dE_{RK} E_{rem} and formulate the expression for $\mathbf{E}_r$ as given in Eq. (1). The results are shown in Table 3.

Table 3. Measured parameters for the iterated ciphers (all figures in pJ)

Cipher	Blocksize/ Keysize	$E_{Reg,1}$	dE_{Reg}	$E_{Mux,1}$	dE_{Mux}	$E_{RF,1}$	dE_{RF}	$E_{RK,1}$	dE_{RK}	E_{rem}	$\mathbf{E}_r$
AES 128	128/128	6.75	1.49	3.65	3.20	32.70	8.26	16.10	8.26	0.42	$(6.13 + 6.03r + 20.48r^2) \cdot (1 + \lceil \frac{10}{r} \rceil)$
Noekeon	128/128	3.26	0.86	1.67	1.81	15.53	26.98	0.00	0.00	0.88	$(3.14 + 4.71r + 13.49r^2) \cdot (1 + \lceil \frac{17}{r} \rceil)$
Present	64/80	2.99	0.27	0.58	0.36	1.47	1.59	0.10	0.00	0.20	$(3.15 + 1.40r + 0.795r^2) \cdot (1 + \lceil \frac{32}{r} \rceil)$
Piccolo	64/80	1.56	0.39	0.61	1.00	3.93	7.87	0.74	0.00	0.70	$(1.48 + 2.13r + 3.93r^2) \cdot (1 + \lceil \frac{25}{r} \rceil)$
TWINE	64/80	3.08	0.37	0.76	0.67	1.56	2.16	0.48	0.25	0.42	$(3.23 + 1.82r + 1.25r^2) \cdot (1 + \lceil \frac{36}{r} \rceil)$
Simon 64/96	64/96	3.34	0.30	0.60	0.48	1.19	0.99	0.75	0.42	0.52	$(3.68 + 2.01r + 0.71r^2) \cdot (1 + \lceil \frac{42}{r} \rceil)$

Noekeon, when operated in direct mode, does not use a key schedule operation and hence $E_{RK,1}, dE_{RK}$ parameters are both zero for this cipher. Similarly, in Piccolo, the key schedule consists of selecting different portions of the key depending on the current round number and adding a round constant to it. This functionality can be achieved by a set of multiplexers and xor gates for any r-round unrolled architecture and so $dE_{RK} = 0$ for this cipher. The key schedule of Present is such that extremely slow diffusion occurs in the key path. So, the switching activity of the RK_1 block does not necessarily compound the switching activity in RK_2 and $E_{RK,1} = 0.1, dE_{RK} = 0$ is a reasonable approximation for analyzing less than 5-round unrolled designs.

By analyzing the expressions for $\mathbf{E}_r$ in Table 3, one can conclude that $r = 2$, is the optimal energy configuration for Present, TWINE and Simon 64/96. For AES 128, Piccolo and Noekeon, $r = 1$ is likely to be optimal in terms of energy. In Fig. 4, we compare our estimates for the energy consumption for upto the 4-round unrolled implementation calculated as per the Equation for $\mathbf{E}_r$ in Table 3, with the actual figures. It can be seen that for Present, TWINE and Simon 64/96, our prediction that $r = 2$ is the optimal energy configuration holds good. Similarly our prediction that the round based architecture is the most energy-efficient for AES 128, Noekeon and Piccolo also holds good.

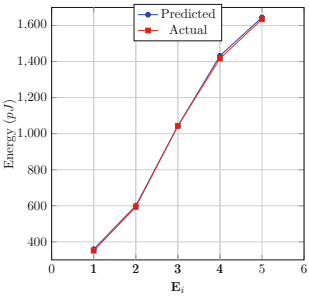
For the non-iterated ciphers, although it is not possible to model the energy consumption of round unrolled designs, the concept holds that successive round functions consume more energy than the previous. The simulation results for all ciphers are given in Table 4. It can be seen that the round based configurations of LED 128 is most energy-efficient. Prince uses three types of round functions: Forward, Middle and Inverse. We implemented 3 architectures for Prince: the round based, Fully unrolled and a Half unrolled design in which Forward/Middle and the Inverse rounds are executed in one cycle each. Again, the round based design was found to be most energy efficient. Finally, we experimented with the round based and 2, 4, 8, 16 and 32 round unrolled versions of KATAN64. As can be seen in Table 4, the 16-round version was found to consume the least energy.

5 Discussion

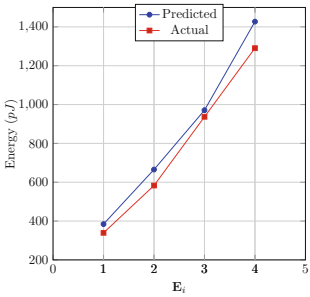
Under a low leakage environment, the energy consumption in an circuit over a period of time is essentially a measure of the electrical work done by the voltage source in order to charge and discharge its gates. We have already seen that in any unrolled architecture, the gates in the later rounds of the design consume more energy, because the switching activity is compounded from one round to the next. Even then, intuitively it makes sense to investigate which degree of unrolling optimizes energy consumption, since an r-round unrolled design will inevitably reduce the total energy required to write updated states onto the state/key registers by a factor of almost r.

We know that the difference in the energy consumptions in any two successive rounds in any unrolled design will depend on the average number of gates that switch in the first round. A physical parameter that is closely related to the average number of gate switchings is the total signal delay in one round. The figures in Table 4 confirm that the ciphers which have low differential energies across successive unrolled architectures are also those in which a signal experiences low delay across a round. These are also the ciphers in which the 2-round unrolled design is more energy efficient. For example in Present, the critical path is composed of 1 S-box and 1 xor gate. In Simon 64/96, the critical path includes 3 xor gates and a single and gate. In Twine, the critical path is made up of 2 xor gates and an S-box. In all other ciphers, the critical path is comprised of atleast one S-box, multiple xor gates and MixColumn layers.

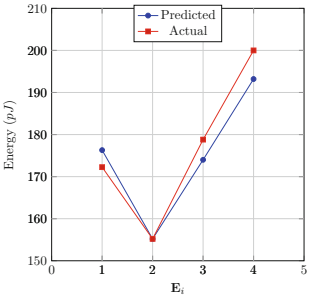
A design parameter that has considerable correlation with the differential energies, is the number of rounds required for full diffusion to take place. This



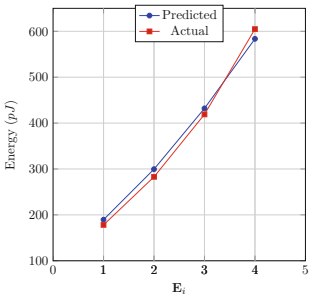
(a) AES 128



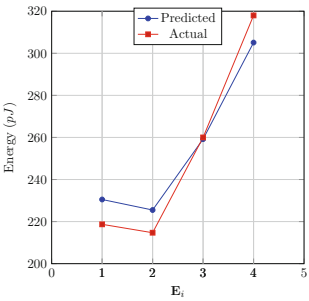
(b) Noekeon



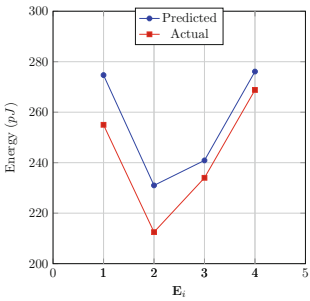
(c) Present



(d) Piccolo



(e) TWINE



(f) Simon 64/96

Fig. 4. Actual and predicted energy consumptions

Table 4. Area, energy and related figures for all the ciphers

#	Cipher	Blocksize/ Keysize	Round Type	Unrolled Rounds	#Cycles	#Rounds for full diffusion	Area(in GE)	Energy (pJ)	Energy/bit (pJ)	Delay per round (ns)
1	AES 128	128/128	SPN	1	11	2	12459.0	**350.7**	2.74	3.32
				2	6		22842.3	593.4	4.64	
				3	5		32731.9	1043.0	8.15	
				4	4		43641.1	1416.5	11.07	
				5	3		53998.7	1634.4	12.77	
2	Noekeon	128/128	SPN	1	18	2	2348.1	**339.2**	2.65	3.41
				2	10		3890.3	583.0	4.55	
				3	7		5434.9	936.7	7.32	
				4	6		6946.6	1290.6	10.08	
3	LED 128	64/128	SPN	1	50	2	1830.8	**656.5**	10.26	5.25
				2	26		2864.7	1216.8	19.01	
				4	14		4780.3	1638.0	25.59	
4	Present	64/80	SPN	1	33	4	1439.9	172.3	2.69	2.09
				2	17		1967.9	**155.2**	2.43	
				3	12		2499.3	178.8	2.79	
				4	9		3000.4	200.0	3.13	
5	Prince	64/128	SPN	1	13	2	2286.5	**149.1**	2.33	4.06
				Half	3		8245.9	358.4	5.60	
				Full	1		7728.6	369.5	5.77	
6	Piccolo	64/80	Feistel	1	26	3	1492.0	**178.1**	2.78	3.28
				2	14		2385.5	282.8	4.42	
				3	10		3268.1	419.0	6.55	
				4	8		4124.7	604.8	9.45	
7	TWINE	64/80	Feistel	1	37	8	1408.2	218.7	3.42	3.10
				2	19		1902.8	**214.7**	3.35	
				3	13		2399.5	260.0	4.06	
				4	10		2850.8	318.0	4.97	
8	Simon 64/96	64/96	Feistel	1	43	4	1480.0	255.0	3.98	2.18
				2	22		1948.7	**212.5**	3.32	
				3	15		2419.0	234.0	3.65	
				4	12		2875.7	268.8	4.20	
9	KATAN64	64/80	Shift register	1	255		983.8	913.6	14.28	2.04
				2	128		1055.4	481.9	7.53	
				4	65		1194.4	269.8	4.22	
				8	33		1459.6	169.1	2.64	
				16	17		1992.4	**140.1**	2.19	
				32	9		3058.1	167.2	2.61	

is defined as the minimum number of rounds that it takes for a difference introduced in any one byte/nibble of the state/key to spread across to all the bytes/nibbles of the current state/key. This figure directly controls the quantum of switching activity across a round, and as Table 4 suggests, the ciphers with low differential energies are also the ones which take more rounds to achieve complete diffusion.

Overall, if we compare the energy consumptions of all the ciphers we find that the 16-round unrolled implementation of KATAN64 consumes least energy. A round in KATAN64 is composed of extremely simple Boolean equations, and hence the trend for KATAN64 is such that unrolling more rounds does not always lead to increase of switching across the rounds. Among the SPN/Feistel architectures, the round based implementation of Prince consumes the least energy, as it takes only 13 cycles to complete an encryption, and the fact that it does not employ any key schedule operation. Close second, is the 2-round unrolled implementation of Present, followed by the round based implementations of Present

and Piccolo. Piccolo benefits from the fact that it does not have a key schedule operation, and hence does not expend any energy on writing values to a key register. Coming in next are the 3 and 4 round unrolled Present and the 2-round unrolled designs of Simon 64/96 and TWINE. It is also interesting to note that if we use a DSE based S-box, then the energy/bit figure of round based AES 128 is quite comparable to Prince and Present.

6 Conclusion

In this paper, we looked at the energy consumption figures of several lightweight ciphers with different degrees of unrolling. By constructing a model of energy consumption, we proved that the total energy consumed in a circuit during an encryption operation has roughly a quadratic relation with the degree of unrolling. In this respect we looked to apply our model to a number of lightweight ciphers and predict the most energy efficient architecture for the design. In the end, we tried to relate the energy consumption in an arbitrarily unrolled architecture of a circuit to physical parameters like critical path in a single round function and algorithmic parameters like number of rounds required to achieve full diffusion.

References

1. Descriptions of SHA-256, SHA-384, and SHA-512. http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256-384-512.pdf
2. Batina, L., Das, A., Ege, B., Kavun, E.B., Mentens, N., Paar, C., Verbauwhede, I., Yalçin, T.: Dietary recommendations for lightweight block ciphers: power, energy and area analysis of recently developed architectures. In: Hutter, M., Schmidt, J.-M. (eds.) RFIDsec 2013. LNCS, vol. 8262, pp. 101–110. Springer, Heidelberg (2013)
3. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The Simon and Speck Families of Lightweight Block Ciphers. IACR eprint archive. https://eprint.iacr.org/2013/404.pdf
4. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: The Keccak Reference. http://keccak.noekeon.org/Keccak-reference-3.0.pdf
5. Bertoni, G., Macchetti, M., Negri, L., Fragneto, P.: Power-efficient ASIC synthesis of cryptographic S-boxes. In: Proceedings of the 14th ACM Great Lakes Symposium on VLSI. ACM, pp. 277–281(2004)
6. Bogdanov, A.A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y., Vikkelsoe, C.: PRESENT: an ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
7. Borghoff, J., et al.: Prince – a low-latency block cipher for pervasive computing applications. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208–225. Springer, Heidelberg (2012)
8. De Cannière, C., Dunkelman, O., Knežević, M.: KATAN and KTANTAN — a family of small and efficient hardware-oriented block ciphers. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer, Heidelberg (2009)

9. Canright, D.: A very compact S-Box for AES. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 441–455. Springer, Heidelberg (2005)
10. Daemen, J., Peeters, M., Assche, G.V., Rijmen, V.: Nessie Proposal: NOEKEON. http://gro.noekeon.org/Noekeon-spec.pdf
11. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer, Heidelberg (2002)
12. Feldhofer, M., Wolkerstorfer, J., Rijmen, V.: AES implementation on a grain of sand. IEEE Proc. Inf. Secur. **152**(1), 13–20 (2005)
13. Gong, Z., Nikova, S., Law, Y.W.: KLEIN: a new family of lightweight block ciphers. In: Juels, A., Paar, C. (eds.) RFIDSec 2011. LNCS, vol. 7055, pp. 1–18. Springer, Heidelberg (2012)
14. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED block cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 326–341. Springer, Heidelberg (2011)
15. Hocquet, C., Kamel, D., Regazzoni, F., Legat, J.-D., Flandre, D., Bol, D., Standaert, F.-X.: Harvesting the potential of nano-CMOS for lightweight cryptography: an ultra-low-voltage 65 nm AES coprocessor for passive RFID tags. J. Cryptograph. Eng. **1**(1), 79–86 (2011)
16. Hong, D., et al.: HIGHT: a new block cipher suitable for low-resource device. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 46–59. Springer, Heidelberg (2006)
17. Kerckhof, S., Durvaux, F., Hocquet, C., Bol, D., Standaert, F.-X.: Towards green cryptography: a comparison of lightweight ciphers from the energy viewpoint. In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 390–407. Springer, Heidelberg (2012)
18. Knežević, M., Nikov, V., Rombouts, P.: Low-latency encryption – is "Lightweight = Light + Wait"? In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 426–446. Springer, Heidelberg (2012)
19. Moradi, A., Poschmann, A., Ling, S., Paar, C., Wang, H.: Pushing the limits: a very compact and a threshold implementation of AES. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 69–88. Springer, Heidelberg (2011)
20. Satoh, A., Morioka, S., Takano, K., Munetoh, S.: A compact Rijndael hardware architecture with S-Box optimization. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 239–254. Springer, Heidelberg (2001)
21. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: *Piccolo*: an ultra-lightweight blockcipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 342–357. Springer, Heidelberg (2011)
22. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: TWINE: a lightweight block cipher for multiple platforms. In: Knudsen, L.R., Wu, H. (eds.) SAC 2012. LNCS, vol. 7707, pp. 339–354. Springer, Heidelberg (2013)