Received May 21, 2018, accepted June 14, 2018, date of publication June 18, 2018, date of current version July 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2848586

Lightweight Block Ciphers for IoT: Energy Optimization and Survivability Techniques

BASSAM J. MOHD[1] **AND THAIER HAYAJNEH**[2]
[1]Department of Computer Engineering, The Hashemite University, Zarqa 13133, Jordan
[2]Fordham Center for Cybersecurity, Fordham University, New York, NY 10023, USA

Corresponding author: Bassam J. Mohd (bassam@hu.edu.jo)

This work was supported by the Fordham University, Faculty Research Program.

ABSTRACT With extraordinary growth in the Internet of Things (IoT), the amount of data exchanged between IoT devices is growing at an unprecedented scale. Most of the IoT devices are low-resource devices handling sensitive and confidential data. Conventional encryption methods are inappropriate for low-resource devices. Lightweight block ciphers are used to encrypt data on such devices, as it balances security requirements and energy consumption. The objective of this paper is to explore opportunities to improve performance and optimize energy consumption for cipher designs targeted for low-resource IoT devices. This paper also presents an energy management algorithm to improve IoT survivability against Denial-of-service attacks in the form of battery exhaustion. We developed a simple and effective model for lightweight cipher performance metrics. Model results were compared and validated with published application-specific integrated circuit (ASIC) and field-programmable gate array (FPGA) designs. Using the model, we explored opportunities for performance enhancement in future cipher designs. Our analysis indicates that the optimum energy is achieved when block size is between 48-bit and 96-bit. Also, increasing size of overhead logic from one round to two rounds increases encryption energy-per-bit by 3.4%. Further, the optimum energy is attained when the number of algorithm rounds is 16 or less. Optimum throughput is achieved by implementations with large block sizes and large number of implemented rounds. Next, we present a novel algorithm to manage cipher energy consumption. The algorithm allows low-resource IoT devices to encrypt critical messages during low-energy mode while balancing throughput, energy per bit, and device activity.

INDEX TERMS Security, information security, encryption, cryptography, energy management, energy efficiency, energy harvesting, Internet of Things (IoT), field-programmable gate array (FPGA), application-specific integrated circuit (ASIC).

I. INTRODUCTION

The Internet of Things (IoT) is an intelligent infrastructure of uniquely identifiable heterogeneous computing devices capable of communicating with each other, services, and people through the Internet without human interaction [1], [2]. Alternatively, European Technology Platform on Smart Systems Integration (EPoSS) defines IoT as a worldwide network of interconnected objects uniquely addressable, based on standard communication protocols [3].

Things refers to devices which link the physical and digital words while connected to *Internet* [4]. *Things* includes increasingly embedded systems deployed in various locations such as medical facilities, industrial installations, critical and nomadic environments, private properties and public infrastructures [5], [6]. Example of IoT devices includes digital machines, RFID tags, sensors, actuators and cellphones [2].

Most smart devices are low-resource devices characterized by low computing power, limited battery supply, small area, and/or small memory size [7]. In such devices, data processing and protocols are carefully designed to meet stringent operation requirements [6].

With the significant growth in IoT (expected to be trillions of connected *things* in the near future [4]), data exchanged between IoT devices is growing at unprecedented scale. IoT device designers face several risks and challenges, including energy capacity [8], and data security [2]. Even with network application layer security enhancements [9], such risks and challenges are more critical in particular when low-resource devices exchanging sensitive data. Further, a power attack could potentially drain an IoT device's battery and cause the device to shut down [10].

To ensure confidentiality, data communicated between IoT devices be must encrypted. Conventional encryption algorithms stress the resources of low-resource devices [11], [6]. Lightweight block ciphers require less resources and mitigate encryption overhead by implementing [2], [7]:

- smaller block sizes; 64-bit or less,
- smaller key size; 80-bit or less,
- simple round logic based on simple computations,
- simple key scheduling.

Numerous research works have examined the performance of lightweight ciphers. Unfortunately, many cipher design implementations are dependent on technology and coding style. A fair comparison across published reports is difficult and inaccurate [7]. Accurate comparison requires that compared designs are implemented with the same technology and software tools as well as provide equivalent security level ([1], [7]).

The energy supply in low-resource device is one of the most critical resources [7]. Energy issues (e.g., energy harvesting and low-power chip-sets) are designated as a technology enabler for IoT and are central to the development of the IoT [3]. In fact, the increase of device power requirement surpassed the improvement of battery and energy storage. The aforementioned challenges of energizing the *Things* have been designated as critical to realize IoT [4].

Autonomous IoT devices with embedded computation may be deployed anywhere with limited access to power cord or battery replacement. Such devices present the toughest challenge to provide energy resources [4]. Some IoT devices are equipped with a hybrid power supply techniques, which include energy storage and energy harvesting. Energy harvesting methods extract energy from ambient environment to prolong battery longevity [1]. Hence, energy crunch can be mitigated with multiple power modes and energy harvesting techniques [8]. Operationally, autonomous IoT devices have two basic modes: active and sleep modes, based on required performance and consumed energy. Basic modes could include other (secondary) modes as well. The duty cycle (D) of the device is computed using Eq. 1 [4]:

$$D = \frac{t_{active}}{t_{active} + t_{sleep}} \tag{1}$$

It is desirable to balance maximizing duty cycle and minimizing energy. A malicious power attack keeps the device in active mode causing excessive power consumption and eventually shutting down the device [10]. Smart techniques are important to address such attacks.

Our main contributions in this research work are:

- Examine design options and optimizations for future lightweight ciphers targeting low-resource IoT devices. To do so, we develop simple, effective and efficient performance metric models. By simple we mean it does not involve complex equations. By effective we mean the model is general and applies to as many lightweight block ciphers as possible. By efficient we mean model is used to optimize performance metrics especially power and energy. The models are tested and compared with published research works.
- Examine design options impact on performance. Design options include block size, overhead logic, number of rounds and throughput.
- Propose energy management algorithm for IoT devices. The algorithm optimizes energy consumptions and improve IoT survivability against power attacks in the form of battery exhaustion.

The paper is organized as follows. Section II presents background and related work information. Section III discusses lightweight block cipher design for low-resource device. Section IV develops models for performance metrics. The model results are compared against published reports in section V. Section VI discusses improving future designs. Section VII presents an algorithm to optimize and manage energy. Concluding remarks are discussed section VIII.

II. BACKGROUND AND RELATED WORK

This section presents background information and related work for lightweight block ciphers, modeling and energy for IoT device.

A. LIGHTWEIGHT BLOCK CIPHER FOR IoT

Due to constrained resources, many IoT devices are considered low-resource devices. Such resources include area, memory size, processing power, power consumption and energy. Conventional security mechanisms are not suitable as they require higher computation power and more resources [6], [11]. Designers of low-resource devices have to balance between data security and constrained resources. Achieving said balance was the main motivation for lightweight block ciphers, which are gentler on resources with reasonable and acceptable security level.

Numerous lightweight ciphers were proposed in literature targeting different platforms and optimizing various constraints. To illustrate research progression, lightweight ciphers have gone through three chronological stages [6]. In the initial stage, spanning 80s and 90s, early ideas of lightweight techniques were proposed. Proposed ciphers were mainly compact implementations of conventional ciphers with several new ciphers. Examples of this stage include Noekeon [12], Iceberg [13], Des [14], Tea [15], Camelia [16], Idea [17].

In the second stage, spanning roughly 2005-2012, extensive research was done on lightweight block ciphers to optimize various constraints with emphasis on area. An ISO standard was published on lightweight ciphers [18]. Examples of this stage include mCrypton [19], Present [20], Puffin-2 [21], Klein [22], Led [23], PPRINTcipher [24], Sea [25], Clefia [26], Desl/Desxl [14], MIBS [27], TWIS [28], Lblock [29], Twine [30], Piccolo [31], Hight [32], Katan [33], Ktantan [33], Hummingbird [34] and Hummingbird-2 [35].

Lately, optimization emphasis has shifted from area reduction to security enhancements and latency improvements. Recent examples include Picaro [36], Zorro [37], Prince [38], Rectangle [39], I-Present [40], Pride [41], Simon [42], ITUbee [43], FeW [44], Robin and Fantomas [45], Hisec [46], Speck [42], Lea [47], Halka [48] and Present-GRP [49].

B. MODELING

To rank ciphers, number published articles compared lightweight cipher performance. Unfortunately, fair comparison requires that implementations ([1], [7]):

- are realized with the same technology and process. Different technologies produce different (and sometimes conflicting) results. For example, Hight ASIC implementation results reported in [50] differ than FPGA implementation results in [51].
- are compiled with the same set of design software tools (e.g. synthesis tool) and constraints (e.g. timing, area and power). Synthesis tools vary in optimization capabilities. Even the same synthesis tool produces different results under different constraints.
- employ the same design options e.g. serial or parallel.
- achieve equivalent security level.

Another approach to estimate performance metrics is to develop a technology- and vendor-independent model. Several research works were published on performance metric models, which can be categorized into:

- Curve-fitting models such as the models in [52] and [51]. Such models are cipher-dependent and difficult to generalize.
- Derived mathematical models which are not associated with any specific cipher. However, some models involve error-prone complex mathematical expressions [53]. Such models complicate design optimization.

C. ENERGY FOR IoT DEVICE

Continuous sources of energy for IoT devices are a significant challenge in terms of battery life [54]. Energy harvesting in IoT devices extends service life of the device and facilitate self-sustainability; the process is referred to as Energy Neutrality [4]. Energy harvesting architectures includes ([55]):

- Harvest-Use: energy is harvested and used immediately. Energy production must be greater than energy consumption, otherwise device is disabled. Insufficient energy production would cause device to oscillate between active/sleep modes [55].
- Harvest-Store-Use: energy is harvested and stored for use. The architecture includes a storage to store excess energy to be used later when either harvesting opportunity does not exist or energy has to be increased to improve device capability. This architecture depends on uncontrolled but predictable energy sources, e.g. solar [55]. Due to its benefits, we assume this architecture in our analysis.

Examples of energy harvesting techniques for IoT devices include human-body heat [56], WiFi [57], indoor light [58], electrostatic vibration [59] and magnetic fields [60]. Additionally, researchers have developed algorithms to manage energy by assisting the power management unit, which is part of power supply flow in IoT device [4]. Researchers in [2] proposed a hybrid algorithm of symmetric and asymmetric encryption algorithms for IoT devices. Asymmetric encryption is executed when device has sufficient processor power, energy and memory.

D. IN CONCLUSION

Above discussion shows clearly that existing solutions have shortcomings. Proposed lightweight ciphers emphasize area and latency optimizations. However, the most important metric for low-resource IoT device are security and energy. Optimizing area is not critical as transistors are cheap [61]. In fact, energy optimized cipher will soon take center stage in low-resource. Research work should consider more of energy optimization techniques, as it helps prolong battery life, encrypt data even in-low energy mode and improve survivability. In this work, our main focus is on optimizing energy and energy per bit metrics.

To design future energy optimized ciphers, many design options and parameters should be examined and optimized. Models facilitate the design process, however, existing models are either cipher-specific or mathematically complex. Useful model should be cipher-independent and simple. Our goal is to develop models that are: simple, effective and efficient.

Finally, not enough attention was given in research to monitor, control and prolong battery during high energy consumption periods. Energy in future ciphers should be optimized and monitored during excessive usage and power attacks. Our objective is to develop an algorithm to address this weakness.

III. DESIGN: OVERVIEW, PARAMETERS AND PERFORMANCE METRICS

Fig. 1 illustrates the general structure of a lightweight block cipher algorithm. The design parameters and performance metrics are listed in Table 1. The design (in Fig. 1) consists of the following main units:

- R rounds, each round has two inputs: N_b-bits plaintext from previous round and sub-key; and generates N_b-bits output to next round. The round implements encryption functions of cipher algorithm.

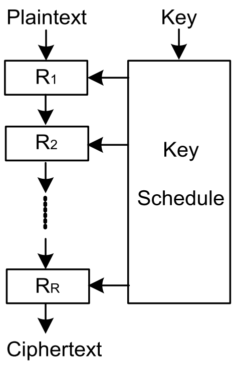

FIGURE 1. Cipher Algorithm.

TABLE 1. **Design parameters and metrics.**

Notations	Description
R	Number of cipher rounds
r	Number of rounds implemented in hardware
N_b	Number of bits per block (i.e. block size)
Th	Throughput (encrypted bits per second)
F	Maximum frequency
T_{cycle}	Cycle time (i.e. clock period)
T_{block}	Time to encrypt one block
C_B	Number of cycles to encrypt one block
A	Design area
E_{block}	Energy to encrypt one block
E_b	Energy to encrypt one bit

- key schedule expands input (master) key into sub-keys used in various rounds. Fixed-key lightweight ciphers (e.g., Ktantan [33]) do not schedule key.

R, complexity of the round function and scheduling vary based on the cipher algorithm. Compared with conventional ciphers, lightweight block ciphers have [7]:

- larger R rounds,
- simpler round function,
- simpler key scheduling.

A Typical lightweight block cipher implementation, illustrated in Fig. 2, consists of the following blocks:

- registers to save initial, intermediate and final texts.
- simple control logic which is mainly finite-state machine. It manages activities and generates sub-keys. We refer to control and key generation logic as overhead logic.
- implementation of r rounds, which complete execution in one cycle.

Maximum frequency is expressed as:

$$F = \frac{1}{T_{cycle}} \tag{2}$$

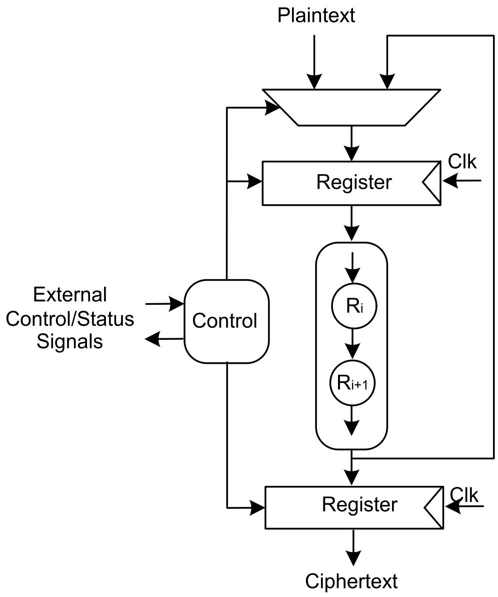

FIGURE 2. **Cipher Implementation.**

Design area A consumes energy of E_{block} to encrypt one block. E_b to encrypt one bit is expressed as:

$$E_b = \frac{E_{block}}{N_b} \tag{3}$$

IV. MODEL

In this section, a simple model for lightweight block cipher metrics is developed. Timing, area, power and energy expressions are presented. To do so, various research works were examined and analyzed, such as those in [50]–[52]. The below discussion refers to parameters and constants defined in Table 1 and Table 2.

TABLE 2. **Constants.**

Notations	Description
T_{reg}	timing delay of registers
T_{comb}	timing delay of combinational logic
C_{idle}	setup cycles to load input plaintext and output ciphertext.
T_{round}	timing delay of one round
τ_1	τ_1=T_{round} when N_b=0
τ_2	timing increase in T_{round} with respect to N_b
A_r	area of implemented r rounds
A_{N_b}	area increase due to N_b
A_{r0}	area of overhead logic, which includes control and key scheduling
A_{r1}	area of single round, assuming minimal (i.e., zero) block size
ρ	A_r growth with respect to r
ρ_1	A_r growth with respect to r, when N_b=0
ρ_2	A_r growth with respect to r, when N_b increases
ν	A_{N_b} growth per bit
F_{AtoC}	capacitance per unit area
α	power per unit area
α_1	r-dependent power per unit area factor
α_2	r-independent power per unit area factor

A. TIMING

The encryption process of one block takes C_B cycles, where the clock cycle is T_{cycle}. C_B is a function of R (from the algorithm) and r (from the implementation). T_{cycle} is reported from implementation and is set by the longest timing paths [61]. The time to encrypt a single block is:

$$T_{block} = C_B \times T_{cycle} \tag{4}$$

The lower bound on T_{cycle} is determined by timing delays of registers (T_{reg}) and combinational logic (T_{comb}), as illustrated in Fig. 3 . Minimum T_{cycle} must satisfy the following expression [61]:

$$T_{cycle} = T_{reg} + T_{comb} \tag{5}$$

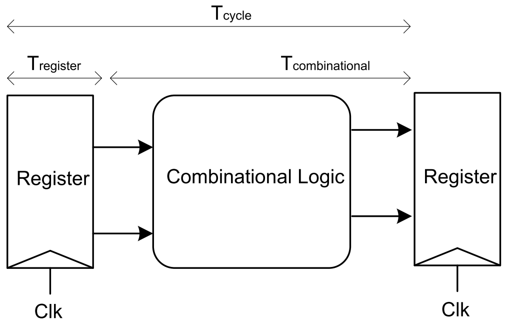

FIGURE 3. T_{cycle}.

Since there are r rounds between two registers (shown in Fig. 2), T_{comb} can be expressed in terms of one round delay (T_{round}):

$$T_{comb} = r \times T_{round} \tag{6}$$

T_{round} consists of two parts: constant τ_1, and N_b-rate τ_2, and is expressed as:

$$T_{round} = \tau_1 + \tau_2 \times N_b \tag{7}$$

Combining Eq. 6 and Eq. 7 results in:

$$T_{comb} = r \times (\tau_1 + \tau_2 \times N_b) \tag{8}$$

Eq. 5 can be re-written as:

$$T_{cycle} = T_{reg} + r \times (\tau_1 + \tau_2 \times N_b) \tag{9}$$

Combining Eq. 4 and Eq. 9 results in:

$$T_{block} = C_B \times (T_{reg} + r \times (\tau_1 + \tau_2 \times N_b)) \tag{10}$$

Also, throughput is defined as [2], [7]:

$$Th = \frac{N_b \times F}{C_B} \tag{11}$$

Since the implementation is capable of executing r rounds per cycle, a single block is encrypted in R/r cycles. Additionally, there are idle cycles (C_{idle}) between blocks to load plaintext and output cipher text. Typically, $C_{idle} = 2$ cycles. Hence, total number of cycles to encrypt a single block is:

$$C_B = \frac{R}{r} + C_{idle} \tag{12}$$

B. AREA

The implementation area depends on r, N_b and overhead logic. Area can be modeled as:

$$A = A_r + A_{N_b} + A_{r0} \tag{13}$$

As r increases, A_r increases non-linearly. Careful examination reveals that A_r is proportional to r^{ρ}, where $\rho < 1$. The A_r growth with respect to r is less than linear because optimization techniques merge some of common logic between rounds. In [51], A_r of four rounds is about 3 × area of single round. Further, it is observed that ρ depends on N_b and can be expressed as:

$$\rho = \rho_2 \times N_b + \rho_1 \tag{14}$$

Hence, A_r is computed as single round multiplied with rounds growth:

$$A_r = r^{\rho} \times A_{r1} = r^{(\rho_2 \times N_b + \rho_1)} \times A_{r1} \tag{15}$$

A_{N_b} increases with N_b linearly as opportunities to minimize logic between bits is minimal. So, A_{N_b} is modeled as:

$$A_{N_b} = \nu \times N_b \tag{16}$$

Combining Eq. 13 -16 results in modeling A as:

$$A = r^{(\rho_2 \times N_b + \rho_1)} \times A_{r1} + \nu \times N_b + A_{r0} \tag{17}$$

C. POWER

Design power consumption is expressed as:

$$P = P_{static} + P_{dynamic} \tag{18}$$

P_{static} represents leakage and static power. In this work, we ignore P_{static} as it contributes to small fraction of the total power [61]. Hence, power is expressed as:

$$P = \alpha \times F \times A \tag{19}$$

where α represents circuit activity factor a, voltage supply V and capacitance per area (F_{AtoC}); and it is expressed as [61]:

$$\alpha = a \times V^2 \times F_{AtoC} \tag{20}$$

Voltage supply and F_{AtoC} are constants. Activity factor a represents how actively design nodes are switching. Because it maximizes data diffusion and confusion, cipher design activates most of the circuit elements and nodes. Increasing r increases levels of logic in the cycle, which in turn results in higher activity factor. Numerous research work attempted to examine impact of logic levels on activity factor [62], [63], [64] and [53]. Examining implementations for cipher designs, α can be roughly estimated as linear relation:

$$\alpha = \alpha_1 \times r + \alpha_2 \tag{21}$$

Eq. 19 can be rewritten as:

$$P = (\alpha_1 \times r + \alpha_2) \times F \times A = \frac{(\alpha_1 \times r + \alpha_2) \times A}{T_{cycle}} \tag{22}$$

D. ENERGY

The energy to encrypt a single block is estimated as:

$$E_{block} = T_{block} \times P \tag{23}$$

Lightweight ciphers have various block sizes (N_b). For fair comparison, energy per bit (E_b) indicates the energy cost to encrypt single plaintext bit for a particular cipher. E_b is expressed as:

$$E_b = \frac{E_{block}}{N_b} \tag{24}$$

Numerous researchers consider E_b as a key performance metric [65]; and one of the most important metrics for low-resource devices as it measures "energy efficiency" of the cipher design [7], [50].

V. COMPARING MODEL WITH PUBLISHED RESULTS

In this section, developed models in section IV are compared with published reports. Hardware design of lightweight block cipher is implemented in either ASIC or FPGA technologies. Because of their in-depth analysis of design options and results, the following research articles were selected to evaluate the models:

- ASIC implementations for Hight cipher [32] presented in [50],
- FPGA implementations for Katan cipher [33] presented in [51].

The goal of the comparison is to illustrate that the models capture the general trends of performance metrics across implementations with reasonable accuracy. Capturing general trends is difficult because area, timing and power must be modeled correctly and accurately across many design flavors.

Initially, model constants are computed for each design, as listed in Table 3. Next, model equations are evaluated to compute various performance metrics. E_b is examined closely since it is the most important metric for low-resource device [7].

TABLE 3. Constants for ASIC and FPGA designs.

Constant	ASIC [50]	FPGA [51]
N_b	64	32/48/64
T_{reg}	11.77	0.0140
τ_1	0.5	0.000143
τ_2	0.00375	8.9375e-07
ρ_1	0.60	0.61
ρ_2	0.0014	0.005
A_{r0}	2445	80
A_{r1}	2042	30
ν	10	2
α_1	0.0002	0.003
α_2	0.001	0.5

A. ASIC DESIGN FOR HIGHT CIPHER

HIGHT, proposed by [32], is a lightweight cipher. It is a variant of generalized Feistel network and targeted for hardware implementations. It has 64-bit block size, 128-bit key size and 32 rounds.

Fig. 4, Fig. 5 and Fig. 6 illustrate power, energy-per-bit and throughput for the design in [50] and presented model. The data in the figures is presented across implementations of various values of r. Examining the results shows that the model tracks the trends for power, energy and throughput with average accuracy of 7.8%, 8.4% and 14.2%, respectively. More importantly, the model agrees with the design that the most efficient energy-per-bit implementations are those with $r = 4$ and $r = 8$.

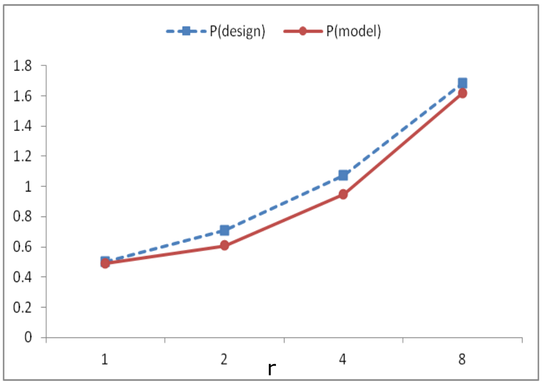

FIGURE 4. Power (mW) versus r.

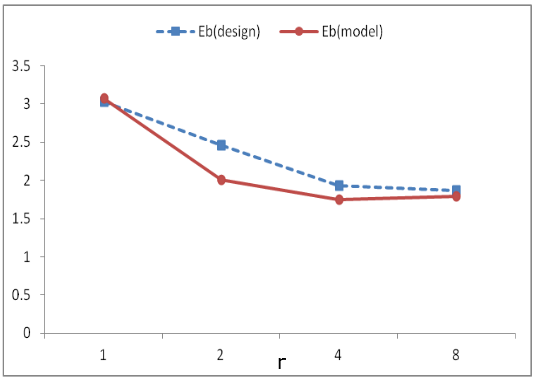

FIGURE 5. Energy per bit (pJ/bit) versus r.

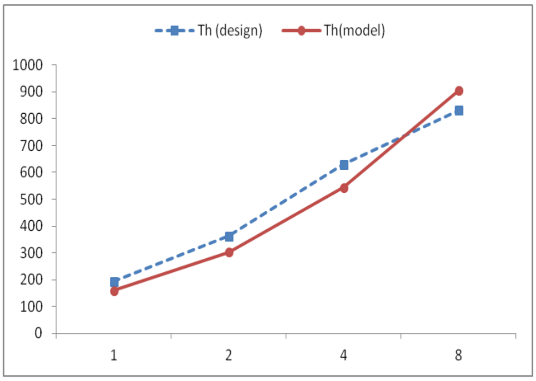

FIGURE 6. Throughput (Mbps) versus r.

B. FPGA DESIGN FOR KATAN CIPHER

Katan is a lightweight cipher; proposed by [33]. It supports 80-bit key size; and 32-, 48- and 64- bit block sizes. Katan includes 254 rounds, each round has a simple encryption function.

[51] presented a detailed analysis of various implementations of Katan cipher including implementations with rounds: 1, 2, 4, 8, 32, 64, 128 and 254. Implementations covered 32-bit, 48-bit and 64-bit block sizes. The objective was to find the implementation with the optimum E_b.

Applying the developed model on Katan cipher was a major challenge due to the many permutations the model must consider and predict. Fig. 7, Fig. 8 and Fig. 9 illustrate E_b reported by the design in [51] and the model. The figures illustrate E_b versus r with different block sizes: 32-bit, 48-bit and 64-bit. The model successfully predicted the implementation with optimum E_b for 32-, 48-, and 64-bit block sizes (N_b), as illustrated in Table 4. In the case of 48-bit, model deviates slightly and predicts 32 rounds

TABLE 4. Minimum E_b for N_b = 32-, 48- and 64-bit.

	32-bit	48-bit	64-bit
Design in ([51])	32	64	32
This work	32	32	32

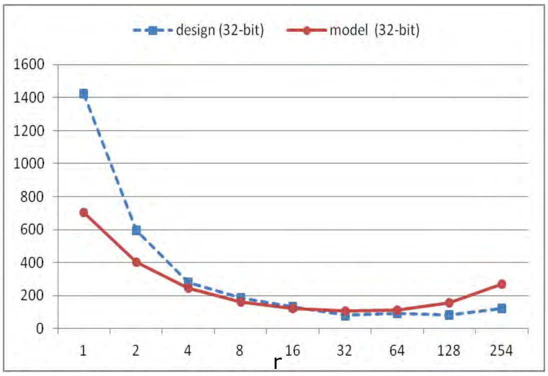

FIGURE 7. E_b (pJ) versus r, N_b = 32-bit.

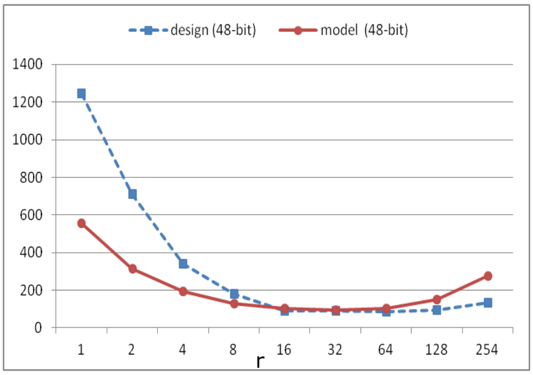

FIGURE 8. E_b (pJ) versus r, N_b = 48-bit.

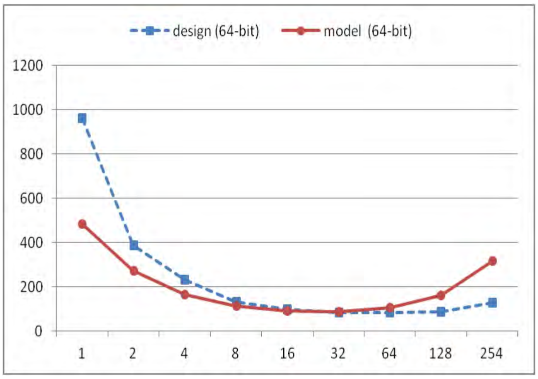

FIGURE 9. E_b (pJ) versus r, N_b = 64-bit.

instead of 64 rounds. More importantly, the model curves trend similarly to the design curves.

VI. OPTIMIZATION FOR FUTURE LIGHTWEIGHT BLOCK CIPHER

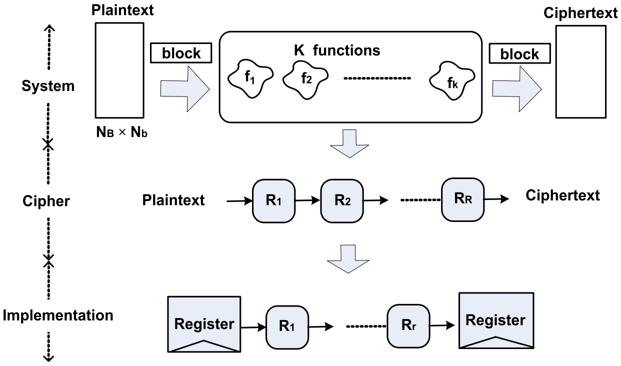

FIGURE 10. Encryption: system, algorithm and implementation views.

In this section, we employ the models to explore opportunities for future performance enhancements. Fig. 10 illustrates encryptions process in different views. The system desires to encrypt a chunk of data with size $= N_B \times N_b$, where N_B is number of blocks. Data is then processed one block at a time. Encryption process is a series of pseudo-random functions: $f_1 \ldots f_K$. The cipher algorithm maps the K pseudo-random functions to R rounds. In low-resource implementations, it is desirable to minimize area and energy per bit. Typically, the implementation realizes r rounds in hardware. In what follows, the impact of design choices on performance metrics is examined using the models. Specifically, we examine:

- block size (N_b),
- round complexity and number of rounds,
- key scheduling (part of the overhead logic),
- throughput.

In generating model results, it is assumed that $N_B = 1000$, Katan cipher and constants in Table 3 (under FPGA column).

A. BLOCK SIZE (N_b)

Fig. 11 illustrates E_b trend versus N_b. To generate the plot, the model is simulated with different values of N_b. For each value of N_b, minimum E_b is then computed by evaluating E_b for different values of $r = 1, 2, 4, 8, 16, 32, 64, 128$ and 254. Examining Fig. 11, it is clear that:

- E_b is high for small block sizes,
- E_b decreases rapidly as block size grows,
- E_b increases slowly for large block sizes,
- optimum E_b occurs at about N_b = [48-bit to 96-bit]. In fact, results of Katan implementations in [50] and [65] show 64-bit implementation has lower E_b compared with 32- and 48-bit implementations.

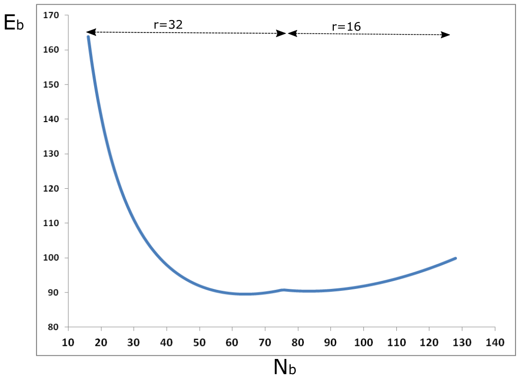

FIGURE 11. E_b versus N_b.

For each value of N_b, the minimum E_b occurs at specific number of implemented rounds, r. For $N_b < 75$, $r = 32$ rounds; for $N_b > 75$, $r = 16$ rounds. This explains the slight blip in the curve around $N_b = 75$ (see Fig. 11). An important point is to determine at what block size minimum E_b occurs for other ciphers. As N_b increases, design area grows; resulting in higher energy. So, an indicator of minimum E_b is to find when area added by N_b becomes dominant in the design. More precisely, it is important to find where area contribution due to N_b overcomes area contribution of single round and overhead logic. To do so, we introduce a ratio in Eq. 25, which divides area parameters that are N_b dependent over area parameters that are N_b independent (see Eq. 17). Fig. 12 plots this ratio across various values of N_b. The optimum-energy N_b value occurs in the ratio range 1-2; it is where the area contribution by N_b becomes significant.

$$ratio = \frac{r^{(\rho_2 \times N_b + \rho_1)} \times \nu \times N_b}{A_{r0} \times A_{r1}} \tag{25}$$

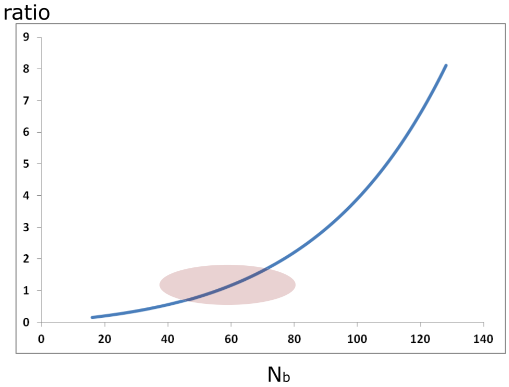

FIGURE 12. Ratio versus N_b.

B. OVERHEAD LOGIC AND KEY SCHEDULING (A_{r0})

In this section, we examine the impact of overhead logic on the cipher energy. Overhead logic includes control logic and key schedule. Typically, control logic is tiny, and cost of overhead logic is dominated by key schedule. We will use the developed model on Katan 64-bit to examine the impact of overhead logic on E_b. We introduce the following ratio to represent the increase in A_{r0} with respect to A_{r1} (which is the area of single round assuming minimal block size).

$$A_{r0/r1} = \frac{A_{r0}}{A_{r1}} \tag{26}$$

Fig. 13 illustrates energy trend versus $A_{r0/r1}$. For each $A_{r0/r1}$, the model was executed with various values of r to compute minimum E_b. Finally, E_b values are normalized to a value of E_b at $A_{r0/r1} = 0.1$. Clearly, increasing $A_{r0/r1}$ results in almost linear increase in E_b. The linearity is broken around $A_{r0/r1} = 1$, as illustrated in Fig. 14, because r (which generates minimum E_b) changes from 16 to 32 rounds. Similarly, linearity broken around $A_{r0/r1} = 23$, because r changes from 32 to 64 rounds. To illustrate the impact of overhead logic on energy, we consider two algorithm alternatives: one with key-scheduling area equivalent to one round area; another with key-scheduling area equivalent to area of two rounds. The second algorithm would increase E_b by 3.4%, resulting in an energy hike (in 64-bit block size) by a factor of 2.2.

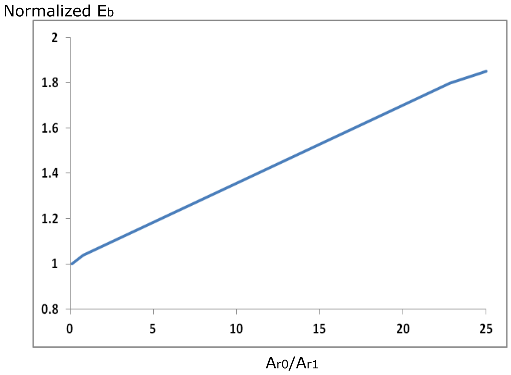

FIGURE 13. Normalized E_b versus $A_{r0/r1}$.

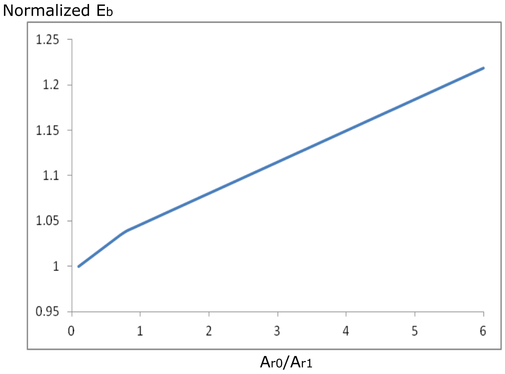

FIGURE 14. Normalized E_b versus $A_{r0/r1}$, smaller range for $A_{r0/r1}$.

C. CIPHER ROUNDS (R)

One of the objectives of our study is to compute the optimum number of rounds in cipher algorithm (R), not to be confused with number of implemented rounds in hardware (r). For low-resource IoT device, optimum R results in minimum E_b. As illustrated in Fig. 10, the algorithm requires K encryption functions to process a single block. Historically, developers of lightweight ciphers opted towards partitioning K functions to small-size groups, with each group mapped to a single round. A tiny cipher round is implemented in a small area, which is a key requirement for low-resource device.

While the intention of the above argument is to reduce implementation area and power, this resulted in cipher algorithms with large R. A large R is not the optimal choice as it might drive up energy of low-resource device. In fact, as transistor size continues shrinking, area is not the dominant metric; rather, energy is the most critical for low-resource devices [7]. So, the key question is how R impacts energy.

To answer the above question, we evaluated the developed model assuming:

- 64-bit Katan,
- total number of rounds is 256,
- several cases were analyzed, in each case the round size was increased to larger size to include ξ of tiny Katan rounds.
- as round size becomes larger, total number of round decreases to R'.
- attempted values of ξ and R' are listed in Table 5.

TABLE 5. Examined cases with various values of ξ and R'.

ξ	Number of Rounds(R')
1	256
2	128
4	64
8	32
16	16
32	8
64	4
128	2
256	1

For each case, as tiny rounds are merged into larger round, the area of a single round (referred to it as A'_{r1}) grows larger. The growth is similar to that of multiple rounds (r) instantiated in hardware. That is, A'_{r1} can be expressed as:

$$A'_{r1} = (\xi \times A_{r1})^{\omega}, \tag{27}$$

where parameter ω depends strongly on the optimization level applied when merging ξ tiny rounds. Strong optimization leads to smaller ω values. Fig. 15 illustrates the E_b trend versus R'. For a clear illustration, axes use a logarithmic scale. Fig. 15 illustrates the following:

- $\omega = 1.0$: represents no-optimization. In this case, large R' decreases E_b,
- $\omega = 0.5$: represents strong optimization. For this case, large R' increases E_b,
- $\omega = 0.8$: represents medium optimization. R' does not impact E_b strongly.

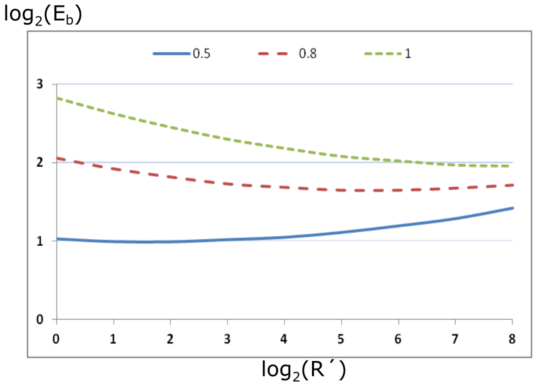

FIGURE 15. $log_2(E_b)$ versus $log_2(R')$ at $\omega = 0.5, 0.8$ and 1.0.

The above analysis shows that optimum number of cipher rounds is heavily dependent on optimization of new large rounds, represented by ω. Since optimization and synthesis algorithms are becoming more powerful and aggressively reduce logic, $\omega = 0.5$ is closer to practical implementations. Consequently, $R' \leq 16$ is a favorable design choice. In fact, researchers in [50] observed the effect of this conclusion and suggested that cipher with tiny-logic rounds is not energy efficient.

Finally, both ω and ρ (which represents the growth of merged r rounds in one hardware stage, shown in Eq. 14 and 15) are dependent on optimization; more optimization results in lower values. ω represents algorithmic (and logic) optimization to merge rounds; while ρ represents logic optimization to merge rounds. Optimization at the algorithmic level (which impacts ω) is stronger than optimization at logic level (which impacts ρ). Hence, it is expected that $\omega < \rho$.

D. THROUGHPUT

Finally, we will examine the impact of design options on throughput of cipher designs. Fig. 16 illustrates that maximum throughput is delivered at higher N_b and r. Maximum throughput is provided at $N_b = 64$ bit and $r = 128$. This implies high throughput is achieved with large block sizes and large number of implemented rounds. However, excessively large r reduces frequency (F) and could potentially reduce throughput, as demonstrated in the case $log_2(r) = 8$, in Fig. 16.

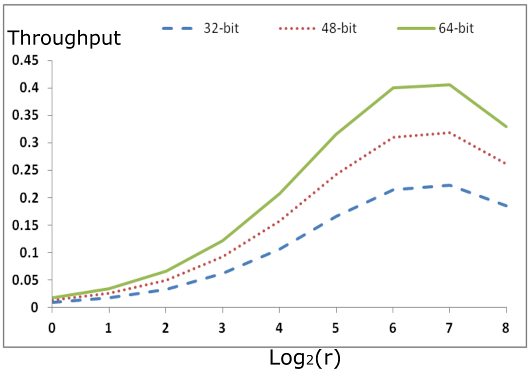

FIGURE 16. Throughput versus $log_2(r)$ at N_b = 32-, 48- and 64-bit.

VII. ENERGY MANAGEMENT ALGORITHM

In this section, we propose an algorithm to manage energy for low-resource devices. The objective is to keep the device operational and maximize duty cycle (D) by lowering energy consumption and prolonging battery life. Such measurement is essential when energy consumption is excessive potentially due to power attack.

Energy levels of IoT devices with energy-harvesting capability exhibits levels depicted in Fig. 17. Similar levels are presented in other researches [55]. Researchers occasionally use energy levels [1], others use power levels [4], [55]. Such energy levels are generated/sampled by a power/energy management unit in the IoT device based on the power source (if any), stored energy, and consumption rates [55].

Fig. 17 displays two important energy levels: E_{normal} and E_{save}. Energy level determines activity level of IoT devices. If it is below E_{save}, the device is placed on sleep mode with limited allowed activity. If the energy level is above E_{normal}, there is no restriction on device activity, and high performance operations are allowed to execute. The IoT device is considered in low-energy mode if energy level is between E_{normal} and E_{save}, only energy optimized operations are allowed to execute.

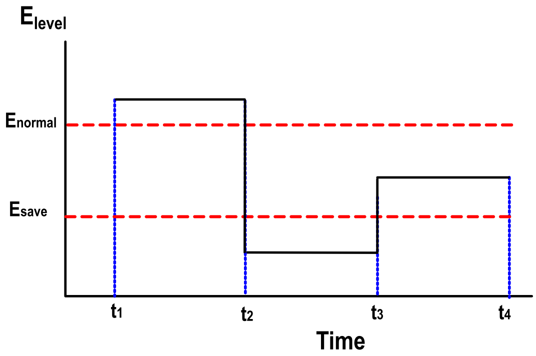

FIGURE 17. Example of Energy Level of Low-Resource Device.

In what follows, we present an algorithm to manage the energy of encryption operation in low-resource IoT device while maximizing device duty cycle. It employs two cipher implementations, each used in a predetermined energy level. We base this solution on earlier stated conclusions: energy is the most important metric for low-resource device and transistors are cheap (in current transistor technology). We assume that cipher design is Katan with a 64-bit block size. We also assume that the two available implementations are:

- performance-optimized (i.e., high-performance) implementation with $r = 32$ [51]. This configuration offers best E_b with optimum throughput.
- low-energy implementation with $r = 1$ [51]. This configuration offers lowest E.

Fig. 18 details the algorithm to manage performance of the IoT. The objectives of the algorithm are to:

- Maximize throughput and minimize E_b whenever possible,
- Maximize cipher active time and duty cycle in particular when energy level is below E_{normal}, as the device might have to process critical data or be under a power attack.

To illustrate proposed algorithm performance, we apply the energy levels in Fig. 17 on three designs. There are three periods in Fig. 17:

- t_1-t_2: energy level is above E_{normal}. During this period, performance-optimized operations are allowed to run. We consider Katan 64-bit with $r = 32$ fits this profile [51].
- t_2-t_3: energy level is below E_{save}, which is a critical energy. In this period, encryption operation is suspended and the IoT device is placed on sleep mode.

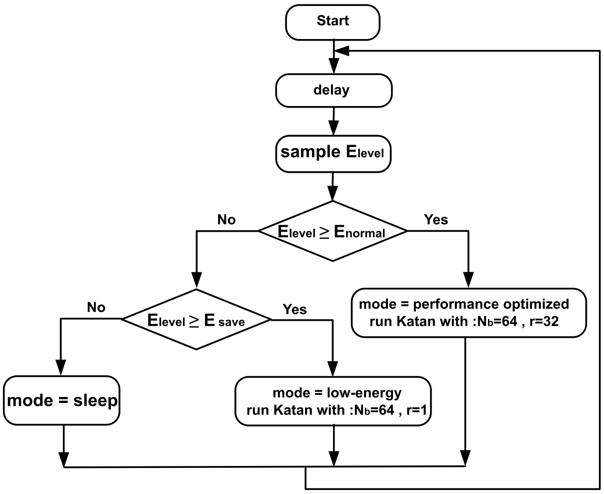

FIGURE 18. Energy-management Algorithm.

- t_3-t_4: energy level is between E_{normal} and E_{save}. In this period, low energy cipher operations can run. In this case, Katan 64-bit with $r = 1$ is appropriate [51].

We compare performance of three designs executing under the same energy levels illustrated in Fig. 17. The designs are:

1) D_1 is a low-energy design and runs when $E_{level} \geq E_{save}$. So, it is active during t_1-t_2 and t_3-t_4. D_1 is Katan implementation with block size of 64-bit and $r = 1$.
2) D_2 is a high-performance design and runs when $E_{level} \geq E_{normal}$. So, the design runs in t_1-t_2. D_2 is Katan implementation with block size of 64-bit and $r = 32$.
3) $D_{algorithm}$ implements the algorithm in Fig. 18 and has two 64-bit block size Katan implementations: $r = 1$ and $r = 32$. The design is active during t_1-t_2 and t_3-t_4.

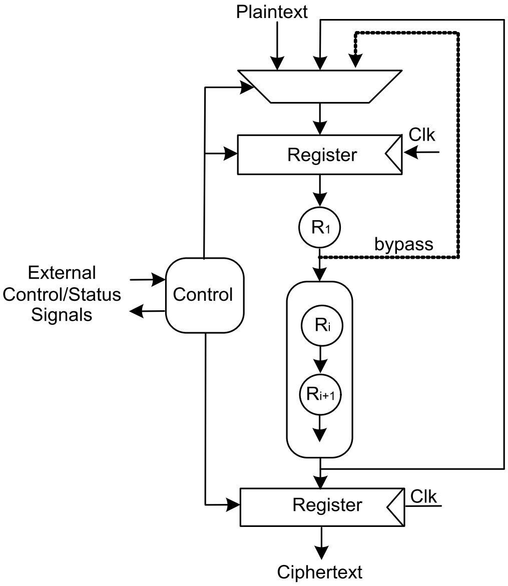

FIGURE 19. $D_{algorithm}$ Implementation.

TABLE 6. Comparing three designs.

Design	#of Enc.	D (%)	Thr (Mbps)	E (mJ)	Thr/E$_b$
D_1	472	0.67	10.1	29.0	0.01
D_2	5170	0.33	110.3	27.9	1.40
D_{alg}	5406	0.67	115.3	42.4	1.0

$D_{algorithm}$ combines the two implementations ($r = 1$ and $r = 32$) by simple bypass technique as shown in Fig 19. For $r = 1$, the bypass is active and the implementation converges to one-round implementation.

We assume that [t_1-t_2] = [t_2-t_3] = [t_3-t_4] = 1 ms. Table 6 illustrates the comparisons of the three designs in terms of number of encryptions, duty cycle (D), throughput, energy (E) and Thr/E_b.

- Clearly, $D_{algorithm}$ computes the most encryption operations because it maximizes throughput, active time and duty cycle. Compared to D_1, $D_{algorithm}$ significantly improves throughput. Also, compared with D_2, $D_{algorithm}$ improves throughput by ∼5%.
- $D_{algorithm}$ has higher energy (as shown in Table 6) since it performs more operations than other designs.
- The last column shows balancing throughput and E_b amongst the designs. Clearly, $D_{algorithm}$ is significantly better than D_1 and slightly lower than D_2, a small cost for maximizing active time.

In conclusion, $D_{algorithm}$ balances performance and energy requirements. It allows encryption operations to continue even during low energy level (i.e. $E_{normal} \geq E_{level} \geq E_{save}$), which could be result of excessive consumption and/or power attack. Above example demonstrates that algorithm improves throughput when compared with low-energy implementation; and increases active time when compared with performance-optimized implementation.

VIII. CONCLUSION

In this paper, we presented an in-depth discussion of lightweight block ciphers for low-resource IoT. In low-resource IoT devices, energy is one of the most challenging resource. The focus of our work was to improve lightweight cipher design performance, and manage energy to prolong battery life in even excessive consumption or power attacks.

Initially, models to capture performance metrics of lightweight cipher were developed. The models allow examination and evaluation of various design choices. The models successfully predicted energy trends in published reports with reasonable accuracy.

Next, the models were employed to guide improving future lightweight ciphers. Several conclusions were derived from the models. For example, the best choice for block size (N_b) for optimum E_b occurs when the area contribution due to N_b overcomes area contributions of single round and overhead logic. Our models show that the best N_b can be found in the range [48-bit to 96-bit]. Also, increasing size of overhead logic from one round to two rounds increases energy per bit by 3.4%. For a 64-bit block size, it results in energy hike by 220%. Additionally, optimal number of rounds in cipher algorithm is heavily dependent on algorithmic and logic optimization techniques. In general, $R \leq 16$ is a favorable design choice as it reduces energy. High throughput is achieved with large block sizes and large number of implemented rounds. However, implementation with very large r degrades both frequency and throughput.

Finally, a novel algorithm to manage cipher energy consumption is presented. It allows low-resource IoT device to encrypt critical messages during low-energy mode. The algorithm balances throughput, energy per bit and cipher active time. Such balancing is helpful to continue encryption operations during intervals of low-energy level caused by excessive energy consumption or power attacks. Results demonstrate that the algorithm enhances throughput when compared with low-energy implementation; and increases active time when compared with performance-optimized implementation.

There are few lessons learned from our research to improve future design of lightweight block ciphers. Since the space of design options and parameters is large, exercising every possible permutation would require significant effort. For successful future design, it is essential to start with near-optimum algorithm, and continue micro-optimizations at the implementation level. To achieve this goal, design options should be simplified and managed by simple and effective model. Further, design optimization should focus on the most important resource: energy. Not only energy guides optimization, but also should be monitored and guarded with smart algorithms from excessive use and power attacks.

Future work should apply the presented model and algorithm to lightweight asymmetric encryption algorithms for low-resource IoT devices. Also, future research could consider monitoring duty cycle to detect suspicious activity in the design. Unjustified high duty cycle would alert the algorithm of potential power attacks.

REFERENCES

[1] P. Kamalinejad, C. Mahapatra, Z. Sheng, S. Mirabbasi, V. C. M. Leung, and Y. L. Guan, "Wireless energy harvesting for the Internet of Things," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 102–108, Jun. 2015.

[2] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions," *J. Ambient Intell. Hum. Comput.*, pp. 1–18, May 2017.

[3] INFSO D.4 Networked Enterprise & RFID. (Sep. 2008). *Internet of Things in 2020—A Roadmap for the Future*. [Online]. Available: https://www.smart-systems-integration.org/public/documents/publications/Internet-of-Things_in_2020_EC-EPoSS_Workshop_Report_2008_v3.pdf

[4] C.-W. Yau, T. T.-O. Kwok, C.-U. Lei, and Y.-K. Kwok, "Energy harvesting in Internet of Things," in *Internet of Everything*. Singapore: Springer, 2018, pp. 35–79.

[5] T. Hayajneh, B. J. Mohd, M. Imran, G. Almashaqbeh, and A. V. Vasilakos, "Secure authentication for remote patient monitoring with wireless medical sensor networks," *Sensors*, vol. 16, no. 4, p. 424, 2016.

[6] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, and C. Manifavas, "A review of lightweight block ciphers," *J. Cryptograph. Eng.*, vol. 8, no. 2, pp. 141–184, 2018.

[7] B. J. Mohd, T. Hayajneh, and A. V. Vasilakos, "A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues," *J. Netw. Comput. Appl.*, vol. 58, pp. 73–93, Dec. 2015.

[8] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. McCann, and K. Leung, "A survey on the IETF protocol suite for the Internet of Things: Standards, challenges, and opportunities," *IEEE Wireless Commun.*, vol. 20, no. 6, pp. 91–98, Dec. 2013.

[9] Z. Shu, J. Wan, D. Li, J. Lin, A. Vasilakos, and M. Imran, "Security in software-defined networking: Threats and countermeasures," *Mobile Netw. Appl.*, vol. 21, no. 5, pp. 764–776, 2016.

[10] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of Things: Security vulnerabilities and challenges," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2015, pp. 180–187.

[11] I. Yaqoob *et al.*, "The rise of ransomware and emerging security challenges in the Internet of Things," *Comput. Netw.*, vol. 129, pp. 444–458, Dec. 2017.

[12] J. Daemen, M. Peeters, G. Assche, and V. Rijmen, "The noekeon block cipher," in *Proc. 1st Open NESSIE Workshop*, 2000, pp. 1–30.

[13] F.-X. Standaert, G. Piret, G. Rouvroy, J.-J. Quisquater, and J.-D. Legat, "ICEBERG: An involutional cipher efficient for block encryption in reconfigurable hardware," in *Proc. Int. Workshop Fast Softw. Encryption*, 2004, pp. 279–298.

[14] G. Leander, C. Paar, A. Poschmann, and K. Schramm, "New lightweight des variants," in *Proc. Int. Workshop Fast Softw. Encryption*, 2007, pp. 196–210.

[15] D. J. Wheeler and R. M. Needham, "TEA, a tiny encryption algorithm," in *Proc. Int. Workshop Fast Softw. Encryption*, 1994, pp. 363–366.

[16] K. Aoki *et al.*, "*Camellia*: A 128-bit block cipher suitable for multiple platforms—Design and analysis," in *Proc. Int. Workshop Sel. Areas Cryptograph.*, 2000, pp. 39–56.

[17] X. Lai and J. L. Massey, "A proposal for a new block encryption standard," in *Proc. Workshop Theory Appl. Cryptograph. Techn.*, 1990, pp. 389–404.

[18] *Information Technology—Security Techniques—Lightweight Cryptography—Part 2: Block Ciphers*, Standard ISO/IEC 29192-2, ISO, Geneva, Switzerland, 2012. [Online]. Available: https://www.iso.org/standard/56552.html

[19] C. H. Lim and T. Korkishko, "mCrypton—A lightweight block cipher for security of low-cost RFID tags and sensors," in *Proc. Int. Workshop Inf. Secur. Appl.*, 2005, pp. 243–258.

[20] A. Bogdanov *et al.*, "PRESENT: An ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems*. Berlin, Germany: Springer, 2007.

[21] C. Wang and H. M. Heys, "An ultra compact block cipher for serialized architecture implementations," in *Proc. Can. Conf. Elect. Comput. Eng. (CCECE)*, May 2009, pp. 1085–1090.

[22] Z. Gong, S. Nikova, and Y. W. Law, "KLEIN: A new family of lightweight block ciphers," in *Proc. Int. Workshop Radio Freq. Identificat., Secur. Privacy Issues*, 2011, pp. 1–18.

[23] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The led block cipher," in *Cryptographic Hardware and Embedded Systems*. Berlin, Germany: Springer, 2011, pp. 326–341.

[24] L. Knudsen, G. Leander, A. Poschmann, and M. J. Robshaw, "PRINTcipher: A block cipher for IC-printing," in *Cryptographic Hardware and Embedded Systems*. Berlin, Germany: Springer, 2010, pp. 16–32.

[25] F.-X. Standaert, G. Piret, N. Gershenfeld, and J.-J. Quisquater, "SEA: A scalable encryption algorithm for small embedded applications," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.*, 2006, pp. 222–236.

[26] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit blockcipher CLEFIA," in *Proc. Int. Workshop Fast Softw. Encryption*, 2007, pp. 181–195.

[27] M. Izadi, B. Sadeghiyan, S. S. Sadeghian, and H. A. Khanooki, "MIBS: A new lightweight block cipher," in *Proc. Int. Conf. Cryptol. Netw. Secur.*, 2009, pp. 334–348.

[28] S. Ojha, N. Kumar, K. Jain, and S. Lal, "TWIS—A lightweight block cipher," in *Proc. Int. Conf. Inf. Syst. Secur.*, 2009, pp. 280–291.

[29] W. Wu and L. Zhang, "LBlock: A lightweight block cipher," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, 2011, pp. 327–344.

[30] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, "Twine: A lightweight, versatile block cipher," in *Proc. ECRYPT Workshop Lightweight Cryptogr.*, 2011, pp. 1–24.

[31] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, "Piccolo: An ultra-lightweight blockcipher," in *Cryptographic Hardware and Embedded Systems*. Berlin, Germany: Springer, 2011, pp. 342–357.

[32] D. Hong *et al.*, "HIGHT: A new block cipher suitable for low-resource device," in *Cryptographic Hardware and Embedded Systems*. Berlin, Germany: Springer, 2006, pp. 46–59.

[33] C. De Canniere, O. Dunkelman, and M. Knezevic, "KATAN and KTANTAN—A family of small and efficient hardware-oriented block ciphers," in *Cryptographic Hardware and Embedded Systems*. Berlin, Germany: Springer, 2009, pp. 272–288.

[34] D. Engels, X. Fan, G. Gong, H. Hu, and E. M. Smith, "Hummingbird: Ultra-lightweight cryptography for resource-constrained devices," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2010, pp. 3–18.

[35] D. Engels, M.-J. O. Saarinen, P. Schweitzer, and E. M. Smith, "The hummingbird-2 lightweight authenticated encryption algorithm," in *Proc. Int. Workshop Radio Freq. Identificat., Secur. Privacy Issues*, 2011, pp. 19–31.

[36] G. Piret, T. Roche, and C. Carlet, "PICARO—A block cipher allowing efficient higher-order side-channel resistance," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, 2012, pp. 311–328.

[37] B. Gerard, V. Grosso, M. Naya-Plasencia, and F.-X. Standaert, "Block ciphers that are easier to mask: How far can we go?" in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2013, pp. 383–399.

[38] J. Borghoff *et al.*, "PRINCE—A low-latency block cipher for pervasive computing applications," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2012, pp. 208–225.

[39] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, "RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms," *Sci. China Inf. Sci.*, vol. 58, no. 12, pp. 1–15, 2015.

[40] M. R. Z'aba, N. Jamil, M. E. Rusli, M. Z. Jamaludin, and A. A. M. Yasir, "I-PRESENTTM: An involutive lightweight block cipher," *J. Inf. Secur.*, vol. 5, no. 3, p. 114, 2014.

[41] M. R. Albrecht, B. Driessen, E. B. Kavun, G. Leander, C. Paar, and T. Yalcin, "Block ciphers—Focus on the linear layer (feat. PRIDE)," in *Proc. Int. Cryptol. Conf.*, 2014, pp. 57–76.

[42] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2015, pp. 1–6.

[43] F. Karakoç, H. Demirci, and A. E. Harmanci, "ITUbee: A software oriented lightweight block cipher," in *Proc. Int. Workshop Lightweight Cryptogr. Secur. Privacy*, 2013, pp. 16–27.

[44] M. Kumar, S. K. Pal, and A. Panigrahi, "Few: A lightweight block cipher," IACR Cryptol. ePrint Arch., Luxembourg, Tech. Rep. 2014/326, 2014.

[45] V. Grosso, G. Leurent, F.-X. Standaert, and K. Varıcı, "LS-designs: Bitslice encryption for efficient masked software implementations," in *Proc. Int. Workshop Fast Softw. Encryption*, 2014, pp. 18–37.

[46] S. S. M. AlDabbagh, A. Shaikhli, I. F. Taha, and M. A. Alahmad, "HISEC: A new lightweight block cipher algorithm," in *Proc. 7th Int. Conf. Secur. Inf. Netw.*, 2014, p. 151.

[47] D. Hong, J.-K. Lee, D.-C. Kim, D. Kwon, K. H. Ryu, and D.-G. Lee, "LEA: A 128-bit block cipher for fast encryption on common processors," in *Proc. Int. Workshop Inf. Secur. Appl.*, 2013, pp. 3–27.

[48] S. Das, "Halka: A lightweight, software friendly block cipher using ultra-lightweight 8-bit s-box," IACR Cryptol. ePrint Arch., Tech. Rep. 2014/110, 2014.

[49] G. Bansod, N. Raval, and N. Pisharoty, "Implementation of a new lightweight encryption design for embedded security," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 142–151, Jan. 2015.

[50] 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*. Berlin, Germany: Springer, 2012, pp. 390–407.

[51] B. J. Mohd, T. Hayajneh, K. M. A. Yousef, Z. A. Khalaf, and M. Z. A. Bhuiyan, "Hardware design and modeling of lightweight block ciphers for secure communications," *Future Gener. Comput. Syst.*, vol. 83, pp. 510–521, Jun. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17304661

[52] B. J. Mohd, T. Hayajneh, Z. A. Khalaf, A. Yousef, and K. Mustafa, "Modeling and optimization of the lightweight HIGHT block cipher design with FPGA implementation," *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 2200–2216, 2016.

[53] B. J. Mohd, T. Hayajneh, M. Z. Shakir, K. A. Qaraqe, and A. V. Vasilakos, "Energy model for light-weight block ciphers for WBAN applications," in *Proc. EAI 4th Int. Conf. Wireless Mobile Commun. Healthcare (Mobihealth)*, Nov. 2014, pp. 1–4.

[54] Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, "Internet-of-Things-based smart cities: Recent advances and challenges," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 16–24, Sep. 2017.

[55] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 443–461, 3rd Quart., 2011.

[56] V. Leonov, "Thermoelectric energy harvesting of human body heat for wearable sensors," *IEEE Sensors J.*, vol. 13, no. 6, pp. 2284–2291, Jun. 2013.

[57] U. Olgun, C.-C. Chen, and J. L. Volakis, "Design of an efficient ambient WiFi energy harvesting system," *IET Microw., Antennas Propag.*, vol. 6, no. 11, pp. 1200–1206, Aug. 2012.

[58] H. Yu and Q. Yue, "Indoor light energy harvesting system for energy-aware wireless sensor node," *Energy Procedia*, vol. 16, pp. 1027–1032, Jan. 2012.

[59] E. O. Torres and G. A. Rincón-Mora, "Electrostatic energy-harvesting and battery-charging CMOS system prototype," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 9, pp. 1938–1948, Sep. 2009.

[60] N. M. Roscoe and M. D. Judd, "Harvesting energy from magnetic fields to power condition monitoring sensors," *IEEE Sensors J.*, vol. 13, no. 6, pp. 2263–2270, Jun. 2013.

[61] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*. New Delhi, India: Pearson, 2015.

[62] E. Boemo, J. P. Oliver, and G. Caffarena, "Tracking the pipelining-power rule along the FPGA technical literature," in *Proc. 10th FPGAworld Conf.*, 2013, pp. 1–9.

[63] S. J. E. Wilton, S.-S. Ang, and W. Luk, "The impact of pipelining on energy per operation in field-programmable gate arrays," in *Proc. FPL*, 2004, pp. 719–728.

[64] N. Rollins and M. J. Wirthlin, "Reducing energy in FPGA multipliers through glitch reduction," Dept. Elect. Comput. Eng., BYU ScholarsArch., Brigham Young Univ., Provo, UT, USA, Tech. Rep. 385, 2005.

[65] L. Batina *et al.*, "Dietary recommendations for lightweight block ciphers: Power, energy and area analysis of recently developed architectures," in *Radio Frequency Identification*. Berlin, Germany: Springer, 2013, pp. 103–112.

BASSAM J. MOHD received the B.S. degree in computer engineering from KFUPM, Dhahran, Saudi Arabia, the M.S. degree in computer engineering from the University of Louisiana at Lafayette, and the Ph.D. degree from The University of Texas at Austin in 2008. He has worked for several semiconductor companies, including Intel, SUN, Synopsys, and Qualcomm. He is currently an Associate Professor with the Computer Engineering Department, The Hashemite University, Jordan. His research interest includes DSP designs, steganographic processors, encryption processors, image processing, speech recognition, and power/energy optimization.

THAIER HAYAJNEH received the M.S. and Ph.D. degrees in information sciences with a specialization in cybersecurity and networking from the University of Pittsburgh, Pittsburgh, PA, USA, in 2009 and 2005, respectively. He was a full-time Faculty Member of computer science with the New York Institute of Technology (NYIT) and the Founding Director of the Center of Excellence in Cyber Security, NYIT, from 2014 to 2016. He is currently the Founder and the Director of the Fordham Center for Cybersecurity, an Associate Professor of computer science, the Program Director of M.S. in cybersecurity and M.S. in data analytics with Fordham University, New York. He published over 65 papers in reputable journals and conferences. His research focuses on cybersecurity and networking, including wireless security, applied cryptography, blockchain, and cryptocurrency. He served on several NSF Cybersecurity review panels and serves as a CAE Reviewer and a Mentor for NSA. He served as the Program Chair on the technical program committee of several leading conferences, including IEEE NSS, GLOBECOM, and ICC. He is also serving as the Editor-in-Chief for EAI *Endorsed Transactions on Pervasive Health and Technology*, an Editor for ACM/ Springer *Wireless Networks*, and a Guest Editor for other prestigious journals. He reviewed several prestigious journals (over 85 reviews) and received Sentinels of Science as a Peer Review Award from Publons.