International Journal of Cryptology Research 9(1): 76 - 89 (2019)

LED and SIMECK FPGA Implementation

Jun-Hoe Phoon*[1], **Denis C.-K. Wong**[1], and **Wai-Kong Lee**[2]

[1]*Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman*
[2]*Faculty of Information and Technology, Universiti Tunku Abdul Rahman*

E-mail: dezphoon@1utar.my, wklee@utar.edu.my, deniswong@utar.edu.my*

ABSTRACT

Lightweight cryptography has been an essential research area in the cryptographic community in recent years, due to the booming of Internet of Things and Ubiquitous Computing technology. Implementation of cryptographic algorithms on embedded devices (e.g. microcontroller and FPGA) is important to protect these systems. However, it is not straightforward to implement cryptographic algorithms in embedded devices, due to the limited computational capability and resources. In this paper, we present the lightweight implementation of LED and SIMECK in Xilinx Artix-7 FPGA.

Keywords: AES, FPGA, SIMECK, lightweight block cipher, LED

1 INTRODUCTION

Lightweight cryptography has been an essential research area for cryptographic community in recent years. Due to the advancement of Internet of Things

(IoT), lightweight cryptography is used in various applications to safeguard the cyber-attack. Different lightweight block ciphers with various design strategies have been presented Hatzivasilis et al. (2017), Hayajneh et al. (2015), Kitsos et al. (2012). The advanced encryption standard (AES) is one of the most matured and well-known block ciphers in hardware and software Farashahi et al. (2014). However, the resource consumption for AES block cipher is relatively high; which makes low-resource devices implementation impossible. Therefore, many lightweight block ciphers have been proposed as an alternative to AES for hardware implementation. Lightweight block cipher algorithms play an important role in the security for resource-constrained devices, such as radio frequency identification (RFID) tags, smart cards, and wireless sensor network (WSN) nodes. In 2011, Guo et al. (2011) proposed the LED block cipher that claims to have the smallest area footprint compared to comparable block ciphers. In 2015, Yang et al. (2015) proposed Simeck, a combination of Simon and Speck by Beaulieu et al. (2013), both lightweight block cipher families with 32 to 128 bits block size and 64 to 256 bits key length; both proposed by National Security Agency (NSA). They are suitable for lightweight hardware implementation such as embedded CPUs that are used in the low-area cryptographic application systems.

2 PREVIOUS WORK

In the past, there had been several attempts to implement LED in embedded platform. In 2014, N.Nalla et al. (2014) implemented the LED and PHOTON in Spartan-3 and Artix-7 FPGA with a study of tradeoff between speed and area with different diffusion matrix. Their work presents the least amount of area implementation to date with the introduction on SRL16 dedicated in Xilinx FPGA devices as shift registers. In 2017, a comparison of lightweight block ciphers (AES, SIMON, SPECK, PRESENT, LED and TWINE) in hardware and software were performed by Diehl et al. (2017). They compared their own implementation in Kintex-7 against existing literatures and benchmarked the findings based on throughput, area and efficiency (TP/A). In the same year, Marchand et al. (2017) published similar work to Diehl et al. (2017) by implementing full-fledge KLEIN, LED, Liliput and Ktantan with encryption and decryption in Spartan 3 and 6 FPGA. They presented serial (least area) and full-width (best speed) results for all the implementations. In 2018, Abbas et al. (2018) proposed a novel comparator based implementation for LED and PHOTON. Comparators were used to replace multipliers, XORs and irreducible polynomial operation to perform matrix multiplication in the mix column step.

In 2018, Bhoyar et al. (2018) integrated Simeck-32/64 cipher into the IEEE 802.15.4 transceiver using Kintex-7 which is the only Simeck FPGA implementation to date besides the implementation in the original proposal by Yang et al. (2015).

2.1 LED

LED is a 64-bit block cipher based on a substitution-permutation network (SPN). It supports any key lengths from 64 to 128 bits. In this article, we will focus on a few main versions: 64-bit key LED (named LED-64) and 128-bit key LED (named LED-128). The number of rounds N depends on the key size, LED-64 has N = 32 rounds while LED-128 has N = 48 rounds.

One can view the 64-bit internal state as a 4×4 matrix of 4-bit nibbles and the round function as an AES-like permutation composed of the following four operations:

1. AddConstants (AC): the internal state is bitwise XORed with a round-dependent constant.
2. SubCells (SC): the PRESENT Bogdanov et al. (2007) S-box is applied to each 4-bit nibble of the internal state.
3. ShiftRows (SR): nibble row i of the internal state is cyclically shifted by i positions to the left.
4. MixColumnsSerial (MCS): each nibble column of the internal state is transformed by multiplying it once with Maximum Distance Separable (MDS) matrix χ^4 (or two times with matrix χ^2, or four times with matrix χ).

$$\chi = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & 1 & 2 & 2 \end{pmatrix}; \chi^2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & 1 & 2 & 2 \\ 8 & 6 & 5 & 6 \end{pmatrix};$$

$$(\chi)^4 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & 1 & 2 & 2 \end{pmatrix}^4 = \begin{pmatrix} 4 & 1 & 2 & 2 \\ 8 & 6 & 5 & 6 \\ B & E & A & 9 \\ 2 & 2 & F & B \end{pmatrix}$$

The key schedule of LED is very simple. In the case of LED-64, the key K is repeatedly XORed to the internal state every 4 rounds (with whitening key operation). In the case of LED-128, the key K is divided into two 64-bit subparts $K = K_1||K_2$, each XORed alternatively to the internal state every 4 rounds. The 4-round operation between two key addition is called a step.

2.2 SIMECK

Simeck is block cipher family, which is denoted as Simeck $2n/mn$, where $2n$ is the block size and mn, key size. Word size (n) has to be 16, 24 or 32 while m is in the range of 2^4. Simeck block ciphers are designed for extremely low power devices. Similar to the Simon family, it has two main blocks, round function and key scheduling algorithm. The number of rounds varies from 32 to 72 based on the value of m and n as shown in Table 1. First of all, plaintext is divided into two words, which are processed by round function. Let m_0 and l_0 represent the two words of plaintext. The round function is defined as:

$$R(m_i, l_i) = (l_i \oplus f(m_i) \oplus k_i, m_i) \tag{1}$$

where k_i is the round key and function f is defined as

$$f(m_i) = (m_i \wedge (m_i << 5) \oplus (m_i << 1)) \tag{2}$$

where '$\wedge$' is AND operator and '$<<$' is left shift operator. For each round, a distinct Round Key k_i is generated with the help of key expansion algorithm. The main key (K) is divided into four words and is initially stored as (t_2, t_1, t_0, k_0). The lowest word of K is loaded into k_0 while the highest word is stored in t_2. Round function is used for updating key values.

$$\begin{aligned} k_{i+1} &= t_i \\ t_{i+3} &= k_i \oplus f(t_i) \oplus C \oplus (Z_j)_i \end{aligned} \tag{3}$$

where Z_j is simeck constant and $(Z_j)_i$ represents the i^{th} bit of Z_j while $C = 2^n - 4$, is a binary constant.

Table 1: Simeck parameters

Word size (n)	**Key words** (m)	**Key size** (mn)	**Number of rounds**
16	4	64	32
24	3, 4	72, 96	36, 36
32	3, 4	96, 128	42, 44
48	2, 3	96, 144	52, 54
64	2, 3, 4	128, 192, 256	68, 69, 72

3 PROPOSED ARCHITECTURE OF LED AND SIMECK

3.1 Architecture of LED

This section describes the design of our proposed LED block cipher architecture. One step is performed in one cycle similar to the architecture of N.Nalla et al. (2014) where the total clock cycles for the round function is 48 clock cycles for 128-bits key size.

The implementation of AC, SC and SR is the as the implementation by N.Nalla et al. (2014) as the implementation is straight-forward which includes mapping, shifting and XOR operation. The proposed implementation of MCS is through the use of a look-up table for the matrix multiplication to prevent usage of multipliers. The look-up table consists of precomputed values for the matrix multiplication. The equation for MCS can be shown in Eq (4).

$$sum+ = field\_mult[MDS[i][k] \times 16 + state[k][j]]; \tag{4}$$

where:

1. field_mult is the proposed look-up table.
2. MDS is the maximum distance separable matrix (χ^4).
3. state is the internal state of the LED.
4. i, k and j denotes the i^{th} column of MDS and state; k^{th} bit of MDS; j^{th} bit of state.

As a result, one step of the round function is completed within one clock cycle as shown in Figure 1.

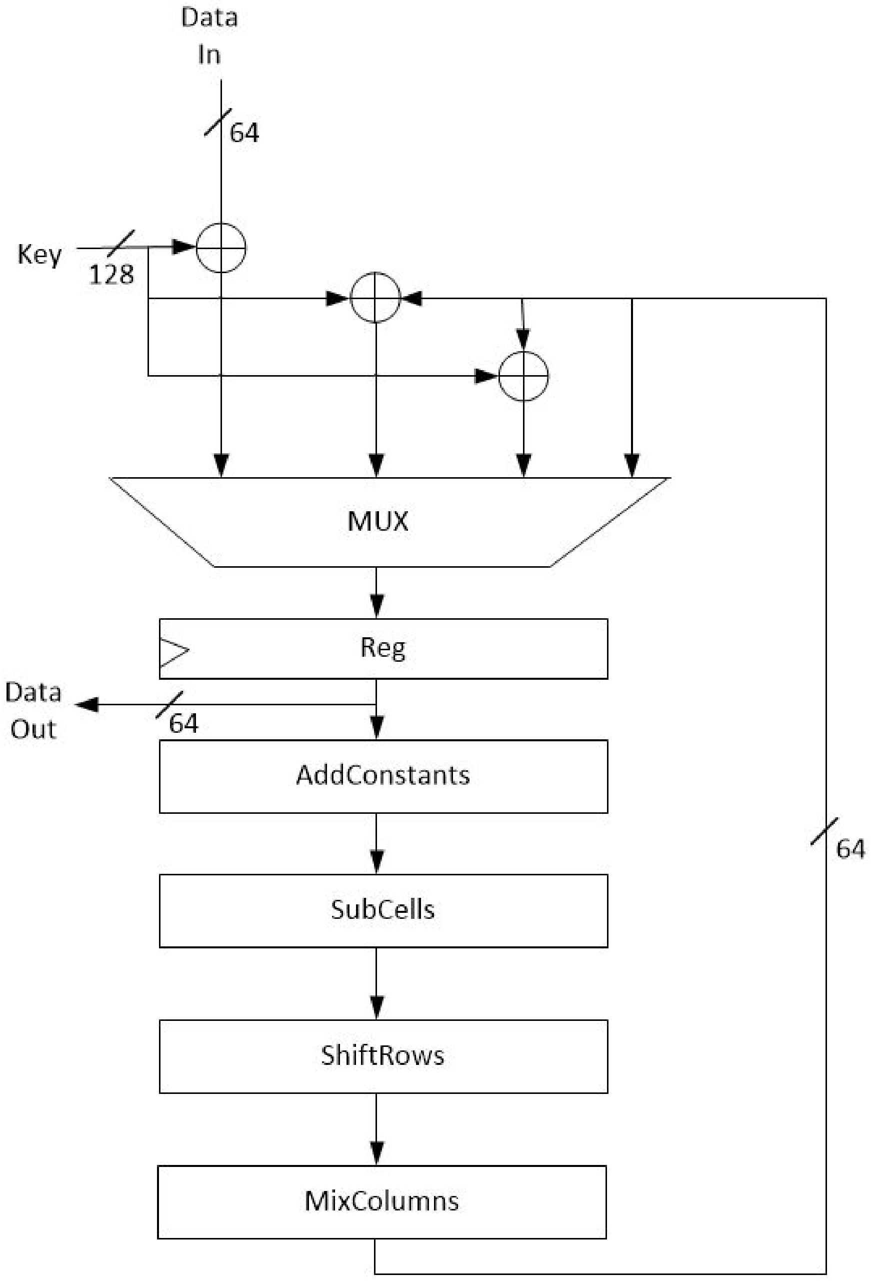

Figure 1: Architecture of LED encryption module.

3.2 Architecture of Simeck

This section describes the parallel architecture of Simeck. In the original proposal by Yang et al. (2015), two different hardware architectures are proposed for Simeck cipher namely, parallel and partially serial. Though partially serial architecture reduces power and area consumption, it suffers from the delay whereas the parallel architecture is 4 $\times$ faster than the serial architecture based on our findings. Therefore, we aimed for better speed performance as Artix-7 has sufficient resource to fully implement the parallel architecture and at the same time provide better speed compared to the serial architecture.

In the parallel architecture, one round of data and key schedule are processed in a clock cycle. Such architecture reduces delay and provides high throughput. Figure 2 shows the round function and it has two 32-bit registers, one 32-bit multiplexer and a combinational logic circuit. Mode input enables the multiplexer to choose either from plaintext data (Data In) or feedback data. The combinational logic circuit has three 32-bit XOR gates, one 32-bit AND gate and two cyclic shifters. The round function has two 32-bit inputs, data-in and k_i. It generates 32-bit data-out after 44 rounds. Data-out along with the previous 32-bit data out forms the 64-bit encrypted word. The only difference compared to the key schedule function shown in Figure 3 as compared to the round function is the number of shift registers which are four in our case. The rest of the circuit remains the same.

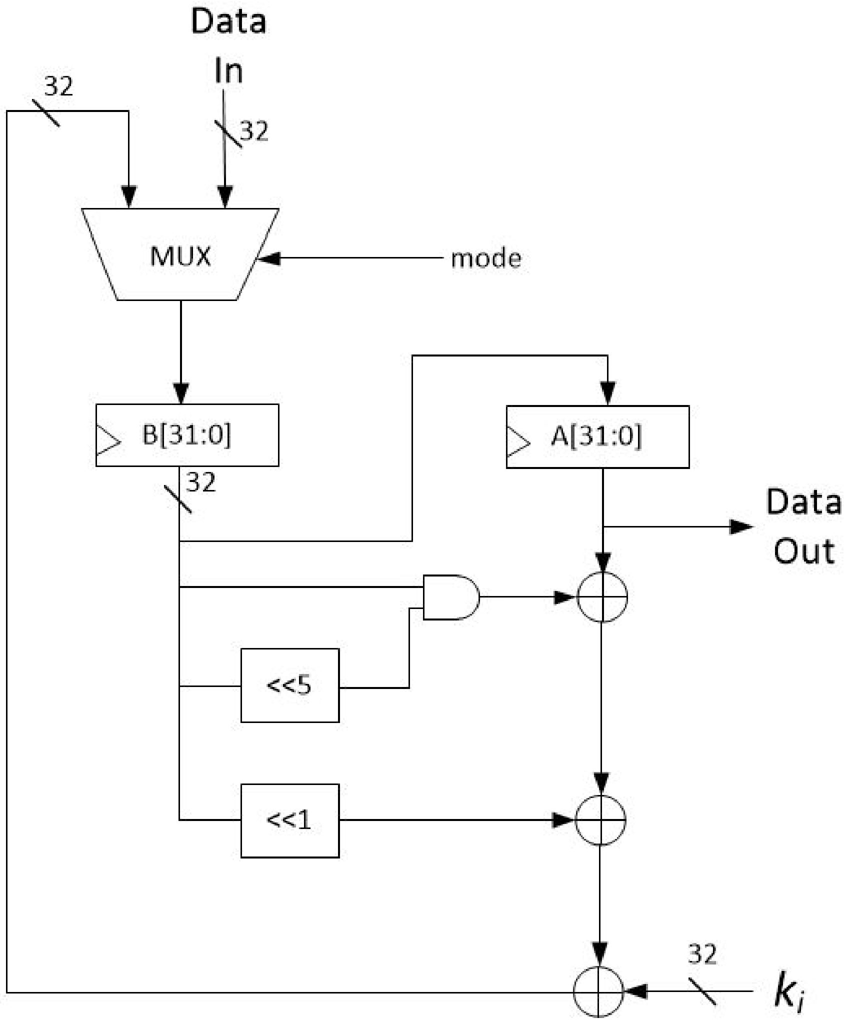

Figure 2: Parallel Architecture of Simeck Round Function.

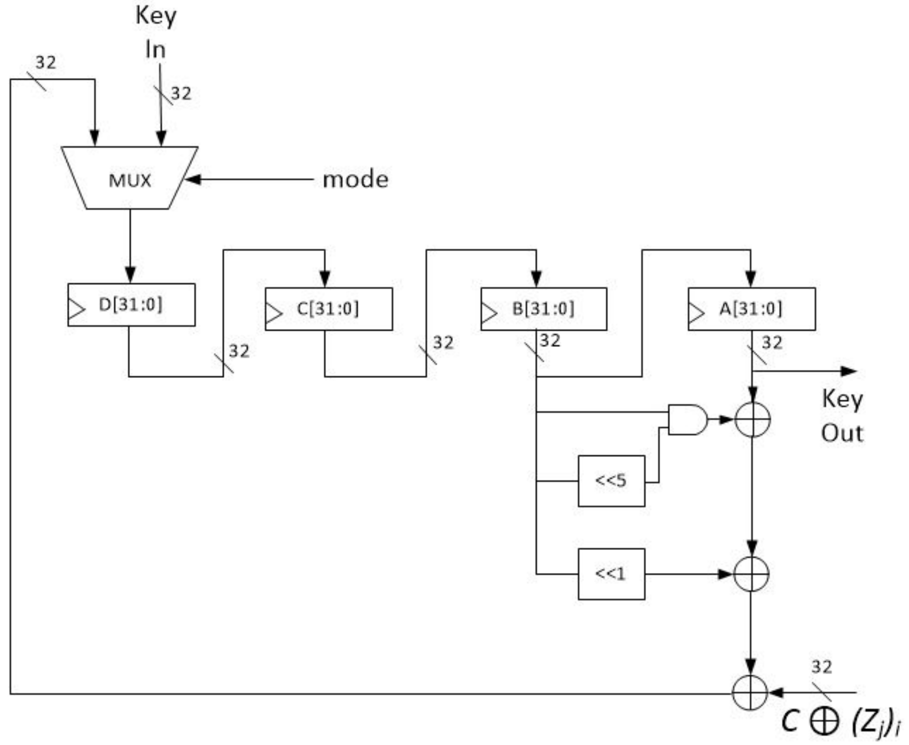

Figure 3: Parallel Architecture of Simeck Key Schedule.

3.3 Parameter Selection

We chose the parameter LED-64/128 and Simeck-64/128 as they provide the equivalent security of AES-128. The key size is 128-bit while each block is of 64-bit wide. This is typically suitable for IoT applications that process small amount of data frequently.

4 RESULTS

For all the techniques proposed in this paper, the implementations and experiments were carried out on Xilinx Artix-7 FPGA family. The results were obtained post place-and-route (PAR) on Xilinx Artix-7 XC7A100T FPGA using Xilinx Vivado 2017.2 and is written in Verilog.

The throughput is calculated through the equation below:

$$Throughput = Frequency * \frac{block\_size}{TotalClockCycles} \tag{5}$$

The results for our implementations is recorded in Table 2. Comparison with other existing work is difficult as works from Banik et al. (2015) and Marchand et al. (2017) did not include the maximum frequency of their implementation and the implementation platform is different. Work by Abbas et al. (2018) only implemented LED-64/64 in serial whereas our implementation is LED-64/128 in full-width. Besides, they have unclear calculation of obtaining the throughput. Despite of that, our LED implementation has lower slice utilization compared to N.Nalla et al. (2014) (44%), Banik et al. (2015)(159%) and Marchand et al. (2017)(597%). However, our LED implementation is still less efficient compared to N.Nalla et al. (2014). As for Simeck implementation, comparison is unfair as the parameters and platform are different but our implementation still promises 13.8 × better throughput despite of the difference in parameters and platform.

Table 2: Performance comparison of similar block ciphers

Scheme	**Plat.**	**Freq (MHz)**	**LUT**	**FF**	**Slices**	**Throughput (Mb/s)**	**Efficiency (Mbps /Slice)**
LED-64/128 (this work)	Artix7	181.81	258	142	109	242.41	2.22
SIMECK-64/128 (this work)	Artix7	172.41	139	334	92	220.68	2.40
LED-64/128 [a]	Artix7	282.43	261	76	158	376.57	2.39
LED-64/128 [b]	Spar6	-	-	-	283	46	0.16
LED-64/128 [c]	Spar6	-	-	-	760	-	-
LED-64/64 [d] **(Serial)**	Artix7	532	-	-	40	34060	851
SIMECK-32/64 [e]	Kin7	16	174	192	-	16	-

[a] N.Nalla et al. (2014)
[b] Banik et al. (2015)
[c] Marchand et al. (2017)
[d] Abbas et al. (2018)
[e] Bhoyar et al. (2018)

5 CONCLUSION

In this paper, we present a lightweight implementation of LED and SIMECK in Xilinx Artix-7 with reasonably high throughput (242.41 Mb/s and 220.68 Mb/s respectively) and efficiency (2.22 and 2.40 respectively). We proposed a look-up table for mix columns operation in the LED block cipher to prevent usage of multipliers and to save resources. In addition, we present the lightest parallel architecture of Simeck-64/128 to date which is crucial for IoT applications that has limited area and power constraint.

REFERENCES

Abbas, Y. A., Jidin, R., Jamil, N., Z'aba, M. R., and Al-Azawi, S. (2018). Small Footprint Mix-Column Serial for PHOTON and LED Lightweight Cryptography. In *2018 International Conference on Advanced Science and Engineering (ICOASE)*, pages 70–74.

Banik, S., Bogdanov, A., and Regazzoni, F. (2015). Exploring the energy consumption of lightweight blockciphers in FPGA. In *2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, pages 1–6.

Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L. (2013). The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404. `https://eprint.iacr.org/2013/404`.

Bhoyar, P., Dhok, S., and Deshmukh, R. (2018). Hardware Implementation of Secure and Lightweight Simeck32/64 Cipher for IEEE 802.15.4 Transceiver. *AEU - International Journal of Electronics and Communications*, 90.

Bogdanov, A., Knudsen, L., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y., and Vikkelsoe, C. (2007). PRESENT: an ultra-lightweight block cipher. volume 4727, pages 450–466.

Diehl, W., Farahmand, F., Yalla, P., Kaps, J., and Gaj, K. (2017). Comparison of hardware and software implementations of selected lightweight block ciphers. In *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, pages 1–4.

Farashahi, R. R., Rashidi, B., and Sayedi, S. M. (2014). FPGA Based Fast and High-throughput 2-slow Retiming 128-bit AES Encryption Algorithm. *Microelectron. J.*, 45(8):1014–1025.

Guo, J., Peyrin, T., Poschmann, A., and Robshaw, M. (2011). The LED Block Cipher. pages 326–341.

Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I., and Manifavas, C. (2017). A review of lightweight block ciphers. *Journal of Cryptographic Engineering*, 8:141–184.

Hayajneh, T., Mohd, B., and Vasilakos, A. (2015). A Survey on Lightweight Block Ciphers for Low-Resource Devices: Comparative Study and Open Issues. *Journal of Network and Computer Applications*, 58.

Kitsos, P., Sklavos, N., Parousi, M., and Skodras, A. (2012). A comparative study of hardware architectures for lightweight block ciphers. *Computers & Electrical Engineering*, 38:148–160.

Marchand, C., Bossuet, L., and Gaj, K. (2017). *Ultra-Lightweight Implementation in Area of Block Ciphers*, pages 177–203.

N.Nalla, Peyrin, Anandakumar, T., and Poschmann, A. (2014). A Very Compact FPGA Implementation of LED and PHOTON. volume 8885, pages 304–321.

Yang, G., Zhu, B., Suder, V., Aagaard, M., and Gong, G. (2015). The Simeck Family of Lightweight Block Ciphers. pages 307–329.