Indonesian Journal of Electrical Engineering and Computer Science
Vol. 20, No. 1, October 2020, pp. 361~369
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v20.i1.pp361-369

Fast lightweight block cipher design with involution substitution permutation network (SPN) structure

Omar A. Dawood
Computer Science Department, University of Anbar, Iraq

Article Info

Article history:

Received Jan 27, 2020
Revised Apr 2, 2020
Accepted Apr 16, 2020

Keywords:

Block cipher
Internet of thing (IoT)
Embedded devices
Involution structure
Lightweight cipher
Symmetric cipher

ABSTRACT

In the present paper, a new cryptographic lightweight algorithm has been developed for the Internet of Things (IoT) applications. The submitted cipher designed with the involution Substitution Permutation Network SPN structure. The involution structure means that the same encryption algorithm is used in the decryption process except the ciphering key algorithm is applied in reverse order. The introduced algorithm encrypts the data with a block size of 128-bit 192-bit or 256-bit, which iterative with 10, 12 and 14-rounds respectively similar to the AES cipher. The design aspect supports an elegant structure with a secure involution round transformation. The main round is built without S-Box stage instead that it uses the on-fly immediate computing stage and the involution of mathematical invertible affine equations. The proposed cipher is adopted to work in a restricted environment and with limited resources pertaining to embedded devices. The proposed cipher introduces an accepted security level and reasonable Gate Equivalent (GE) estimation with fast implementation.

Copyright © 2020 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Omar A. Dawood,
Department of Computer Science,
University of Anbar, Iraq.
Email: Omar-Abdulrahman@uoanbar.edu.iq

1. INTRODUCTION

The Internet of Things (IoT) is a fresh worldview that is quickly making progress in data innovation. The essential mean of this idea is the inescapable gadgets around us, which can collaborate and coordinate with their neighbors to achieve shared objectives through remarkable tending to plans [1]. The widespread of embedded devices that fit for interfacing with the internet has been developed massively. A large number of these pervasive tiny devices will keep running on compelled stages with restricted power and constrained resources [2]. In recent years, IoT interacts considerable attention to depicts the framework of connection to the internet via the smallest devices to be universal and ubiquitous. The pervasive connected devices include sensor nodes, smart cards, FPGA embedded devices, ASIC and Radio-Frequency Identification (RFID). In the era of wireless transmission, there is an urgent need for securing the information from stealing or theft in addition to trust the network protocols [3]. The spread connected devices create new challenges and critical risks regarding security threats. Thus, several innovative lightweight algorithms have been proposed to fit this type of restricted environment in terms of power consumption, area chips, Gate Equivalent and other parameters [4]. Lightweight algorithms are a new direction in cryptographic terminology that deal with implementing cryptographic protocols through a dedicated environment. The lightweight cipher's job is to trust the data with low-cost requirements through the constrained environment in both software and hardware [5]. The main issues addressed by the researchers and designers are how to design a reasonable secure cipher with the least hardware resources and speed of software implementation. The most existing of traditional ciphers are inapplicable to use in embedded devices with low power processing, Latency, and restricted of low consuming energy. Although there are a lot of criticisms about the design of lightweight algorithms

because they do not give a high level of security. This matter generates a general impression about the lightweight models are being less resistance against the malicious attacks [6]. However, the lightweight design issue does not mean that it is a weak or not robust cipher, on the contrary, it has only fewer qualities compared with the traditional symmetric ciphers furthermore it characterized by lightweight and elegant internal structure. Nevertheless, it can have an interesting contribution to apply a trusted modern model. Many designers and developers work to develop new lightweight algorithms for the symmetric cipher fields and they have launched several projects in different trends: lightweight stream ciphers, lightweight block ciphers, lightweight Message Authentication Code (MACs), lightweight Pseudo-Random Number Generator (PRNG) and lightweight hash functions. There are no big differences or precise contradictions between the lightweight and the traditional cipher from the viewpoint of design, since the two models use the same comprehensions and the same strategies, except that the first one with fewer features and less cost and may produce less security margins [7]. There are several modern lightweight algorithms that have emerged recently, which work with various structures and different mathematical and algebraic foundations such as FeW [8], DESL [9], LED [10], KlEIN [11], MIBS [12], KATAN [13], mCrypton [14], SEA [15], HIGH [16], DESXL [17], Piccolo [18], CLEFIA [19] Simon and Speck [20], LEA [21], TEA [22], Puffin [23], XTEA [24], ICEBERG [25], Here, it is an important to refer to the readers for additional information to stay up to date at the edge on developed models.

2. LITERATURE SURVEY

The present study tries to explain some interesting related works about the lightweight symmetric ciphers with SPN structure. A. Bogdanov and et al proposed an-ultra lightweight block cipher algorithm, which is called the PRESENT cipher. The PRESENT structure designed as an SPN structure iterated with 31 rounds. The encryption process includes 64-bit of the data block and 80-bit of a secret key. The PRESENT cipher submitted good features and is considered a benchmark for other lightweight algorithms [26]. J. Borgho and et al introduced a low latency lightweight block cipher that named by PRINCE algorithm. PRINCE cipher encrypts 64-block with 128-bit of ciphering key that separated into two parts each with 64-bits. PRINCE algorithm carries out the instance of encryption process via one clock cycle with low cost and low latency. The encryption process is achieved under the SPN structure of the moderate number of rounds up to 12-rounds [27]. M. R. Albrecht and et al developed a PRIDE lightweight block cipher with SPN structure that looped to 20 rounds. The authors submitted a framework for enhancing the linear layer over the main structure. The algorithm encrypts the electronic data with 128-bit of a secret key that XORed with 64-bit of plaintext. PRIDE cipher uses a bit-slice implementation that gave a significant enhancement for the 8-bit microcontroller implementation. The wide-trail strategy is adopted in PRIDE design to reduce the latency and power consumption in addition to improve hardware efficiency [28]. W. Zhang and et al developed a new fast lightweight symmetric cipher of 25-rounds SPN structure that known by RECTANGLE lightweight algorithm. RECTANGLE cipher also uses a bit-slice technique to increase the implementation speed of software and enhancing hardware flexibility. It encrypts the data block of 64-bit with a ciphering key of 80-bit or 128-bit, and it combines the S-box layer with p-layer to get a good security performance [29].

3. THE ESSENCE OF THE PROPOSED DESIGN STRATEGY

The design of any cipher algorithm should satisfy the standard criteria and the design plan should be subject to the current requirements. The cryptographic algorithms are divided into two basic types Symmetric ciphers and Asymmetric ciphers. The symmetric algorithms use the same cipher-key in enciphering/deciphering processes, While the asymmetric ciphers algorithms work with double ciphering-keys one is public for encryption and the other is private for decryption process [30]. There are two benchmark structures design for the symmetric cipher that determines the design strategy for the building block cipher algorithms. The Feistel Structure (FS) and the Substitution Permutation Network Structure (SPN) are influenced directly to the lightweight design constraints. The proposed cipher is a result of the long development process for some published algorithms that opened the door in front of the design idea. These algorithms inspirited the proposed cipher many good features and the attractive internal structure as explained in [31-33] as a previous works. There are several effective criteria and important conditions that must be satisfied through the design process which can be summarized as follows:

3.1. Memory reservation for look-up table (S-Box)

The amount of memory reservation plays a vital role in design any lightweight ciphers because the lightweight algorithms must work with less memory RAM & ROM. So, the main expensive part of the algorithm that requires a reserved memory is the S-Box. The S-Box design may be built randomly or

mathematically. Most of the lightweight ciphers' design ignores the S-Box part or builds it with no S-Box like the (SIMON) algorithm. Some lightweight ciphers are designed with small S-Boxes of 4*4 or 4*8 similar to the design of (PRESENT) cipher [12, 34].

3.2. The cost of gate equivalent design

The Gate Equivalent (GE) of Integrated Circuits is a necessary factor for design the electronic logic board that must be reduced as little as possible. Most block cipher algorithms designed with two different structures one for encryption and the other for decryption in addition to the ciphering-key algorithm. The design lightweight cipher with two different algorithms is very expensive in terms of the GE and chips area. Thus; some researchers design the algorithm based on the operational mode to cancel the decryption algorithm. The other designers decreasing the cost of design by using the same algorithm in encryption and decryption process with an involutional structure such as KLEIN lightweight cipher. The round transformation with involutions stages means that the main stages are implemented as self-invertible implementation [11].

3.3. The fixed ciphering sub-keys generator

The ciphering key is the backbone for the block cipher algorithm that determines the strength of the algorithm and must be secret. The designers have several methods for constructing the ciphering key algorithm regarding the key-dependent S-Box or random generating method. The ciphering key is built with an intractable manner of one-time pad computational notation. The security concerns about the related key attacks should be taken into account in addition to the key agility scheme [34].

4. THE PROPOSED ENCRYPTION ALGORITHM

The proposed cipher designed to encrypt the data with a block of 128-bit to be compatible with Advance Encryption Standard (AES) cipher but with an elegant structure. The proposed cipher builds to work with involution stages that mean the same mathematical operations in encryption are used in the decryption process. The involution property reduced the number of line-codes, number of GE, implementation time, power-consuming and memory size. Furthermore; the involution property increases the efficiency and the scalability in software implementation. The main structure can be stated in Figure 1.

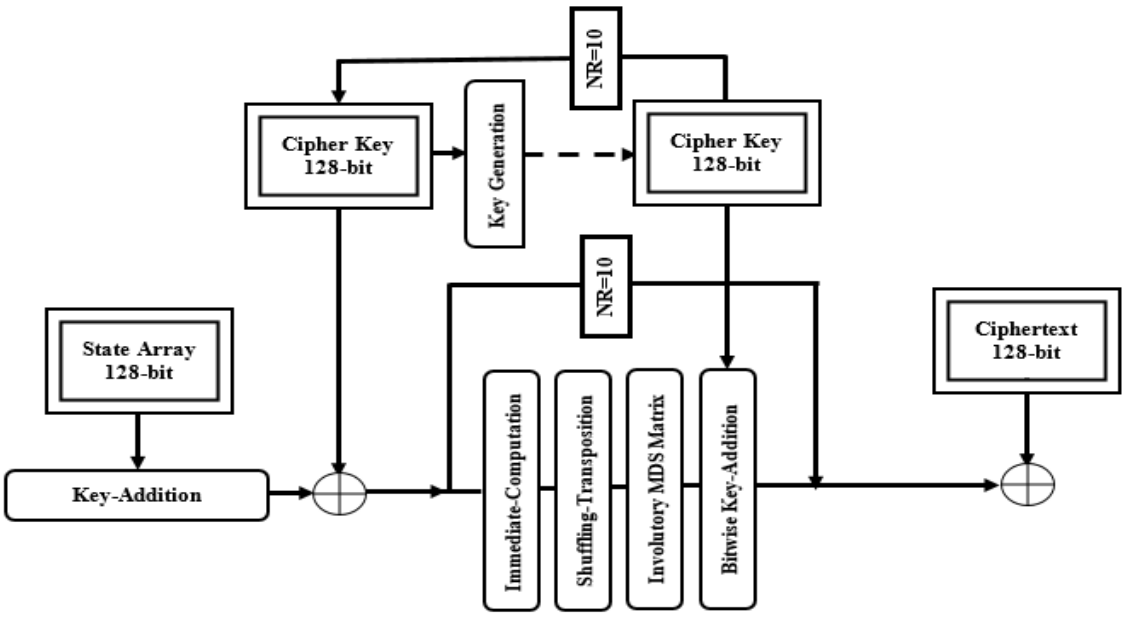

Figure 1. General structure of the proposed lightweight cipher

5. The proposed encryption stages

The round transformation for the proposed cipher can be described as an iterated and product cipher with a compact structure. The encryption round consists of four essential operations which can be explained as follows:

5.1. Involution immediate-computation stage of non-linear layer

This stage is the main responsible for non-linearity and the confusion property which represents the replacement stage for the S-Box design. The main clue behind the immediate computation stage is to ignore the S-box design and memory reservation. The involution immediate computation stage work on-fly computation that encrypts the data instantly. This stage can be summarized in two steps: Compute the multiplicative inverse over the Galois Field GF (28) regarding the irreducible polynomial of modular reduction. Apply self-inverse of non-linear Affine mapping and XORed with a constant vector of self-invertible palindromic value (10100101) or (A5) in Hexa-notation.

$$S(x) = Ax\text{-}1 \oplus b \text{ Over GF(28)} \quad (1)$$

$$S(x)' = AY \oplus b \text{ Over GF(28)} \quad (2)$$

The corresponding Forward/Backward equation in GF (2) is given in (3). and (4).

$$b = bi + b(i+1) \bmod 8 + b(i+2) \bmod 8 + b(i+5) \bmod 8 + b(i+6) \bmod 8 + Ci \quad (3)$$

$$b' = bi + b(i+1) \bmod 8 + b(i+2) \bmod 8 + b(i+5) \bmod 8 + b(i+6) \bmod 8 + Ci \quad (4)$$

Whereas the multiplication of $b*b' =$ Complement (Identity matrix) (5)

The same equation can be represented in the affine matrix form for the multiplication in a forward and backward process to give the Complement-Identity matrix as stated below:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

5.2. The shuffling-transposition stage

This stage is a transposition method responsible for generating the diffusion property by exchanging positions values. It is a new mechanism for rearranging the elements of state array in a symmetric way as shown in Figure 2. and Figure 3. These values are permutated in simple steps to distribute each entry input byte to various output byte. The shuffling stage increases the diffusion properties to the whole structure cipher. Thus, the cryptographic cipher with a high diffusion scheme considers a more secure cipher. The shuffling decryption process is the same encryption process since the shuffling stage is an invertible symmetric transposition of self-inverse operation.

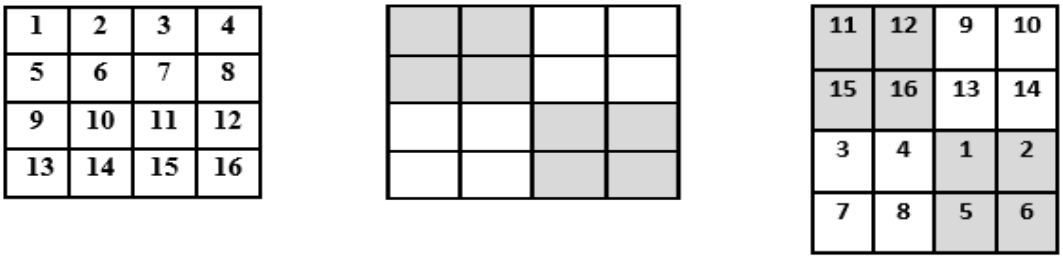

Figure 2. The proposed shuffling diffusion stage

Input										0	1	2	3	4	5	6
Output	1	2		0	5	6	3	4								

Figure 3. The distribution values map

5.3. Involutory MDS matrix of linear-mixing stage

The Maximum Distance Separable (MDS) matrices are remarkable matrices for getting the diffusion functionality in any building blocks. In recent years, many researchers don their best to build the MDS matrices with restricted embedded gate amount to fit the lightweight algorithms. Thus, the proposed algorithm adopts an involutory MDS matrix rather than the ordinary matrix to use the same circuits design in encryption and decryption operations. The mixing stage uses an involutory linear equation of four order linear equation with a self-inverse property that coprime with x4 + 1.

$$a(x) = 03x3 + 05x2 + 03x + 04 \quad (6)$$

$$a'(x) * a(x) = I \quad (7)$$

The MDS matrix implements the addition and multiplication operations column by column that can be regarded as a matrix as follows:

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} \otimes \begin{bmatrix} 04 & 03 & 05 & 03 \\ 03 & 04 & 03 & 05 \\ 05 & 03 & 04 & 03 \\ 03 & 05 & 03 & 04 \end{bmatrix} = \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} \otimes \begin{bmatrix} 04 & 03 & 05 & 03 \\ 03 & 04 & 03 & 05 \\ 05 & 03 & 04 & 03 \\ 03 & 05 & 03 & 04 \end{bmatrix} = \begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix}$$

The proposed MDS matric is more desirable since it requires low-cost hardware in the context of design matter. So the choice of MDS matrix with low coefficient is to reduce the chip area and the involution feature is to decrease the XOR circuits as much as possible.

$$\begin{bmatrix} 04 & 03 & 05 & 03 \\ 03 & 04 & 03 & 05 \\ 05 & 03 & 04 & 03 \\ 03 & 05 & 03 & 04 \end{bmatrix} \times \begin{bmatrix} 04 & 03 & 05 & 03 \\ 03 & 04 & 03 & 05 \\ 05 & 03 & 04 & 03 \\ 03 & 05 & 03 & 04 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5.4. Bitwise key-addition stage

The key addition stage is XORed operation between the plaintext array and ciphering key array in line with the whitening concepts. The reusing bitwise of XORed operation trustworthy the decryption process since it is a self-inverse process.

5.5. Key-scheduling algorithm

The proposed lightweight cipher encrypts the data with a strong ciphering key of the dependent non-linear step with Feistel structure. The key generation process consists of a set of simple operations that collected together give a significant power to encounter the cryptanalysis attacks tightly. The initial key is divided into two parts left and right each with (64-bits) where the right part pass to the complex Function (F) of three operations. These primitives' operations involve the Bitwise-shifting, on-fly computation and the adding of some others constant values or vectors. The key-scheduling algorithm involve a significant Function (F) for generating the ciphering sub-keys for each round. The Function (F) is a key generation process responsible for generating ciphering sub-keys of all rounds. The (F) function map the entry word (32-bit) to the nonlinear affine layer of sub-byte operation, then takes byte-shifting for the resultant word and eventually, XORed with constant word from the pool of table vectors. The right part of a secret key is mixed with the complex function at every stage to eliminate the data correlation. The left part of a secret key is a byte-rotate. The generated ciphering sub-keys is a state array of 128-bit of iterated ciphering key over a several stages with different mathematical operations. The last step includes XORing between the ciphering key and the plaintext (128-bit 128-bit= 128-bit) as stated in Figure 4.

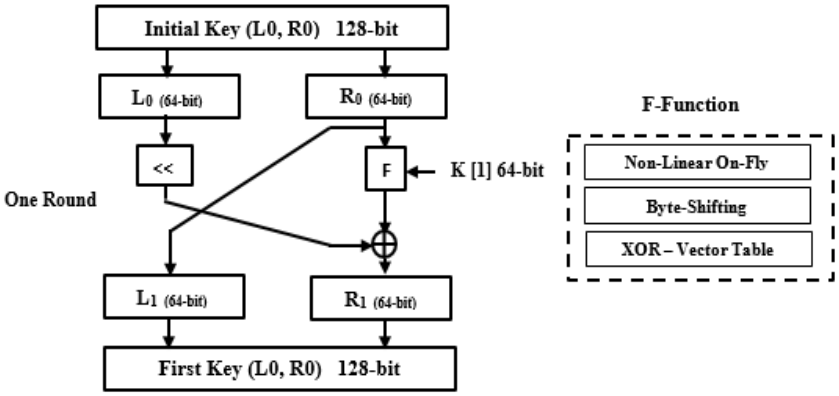

Figure 4. The proposed key scheduling algorithm

6. THE DECRYPTION ALGORITHM

The decryption process in any cipher algorithm designed with SPN structure requires an independent algorithm for reverse operations. This concept is inapposite to the FS that does not need an additional algorithm in the decryption process because it uses the same algorithm in encryption and decryption processes. Both structures decrypt the ciphertext by applying the ciphering keys in reverse order as a standalone algorithm. The main contribution lies in implementing the proposed lightweight algorithm with the involution SPN structure. Therefore, the introduced lightweight cipher encrypts and decrypts the data by the same round transformation or the same structure.

7. ANALYSIS AND IMPLEMENTATION RESULTS

The security of the symmetric algorithm is constantly assessed by designer experts around the world. The main goal is to test the strength of the design frameworks against many types of modern attacks. The analysis aspect for any secure cipher is the attacks that are used to guess the computing requirements for analyzing the algorithm. The linear and differential attacks are the most popular known-plaintext attacks. The mentioned attacks are the most two important issues must be taken into considerations through the design S-Box stage. Although most lightweight ciphers are designed with small precomputation look-up tables of S-Box stage that decreases on-fly-computation time. The present cipher has a balanced architecture with involutional main functions across all encryption modes. The proposed algorithm designed with no S-Box stage, instead of that it has adopted an instant computation strategy. The on-fly computation scheme increases the key dependability on the non-linear function and consequently eliminate the related key attack. The immediate computation technique of non-linear stage designed via self-inverse affine multiplication. The resultant output byte subsequently XORed with a palindromic constant vector. The introduced technique declines the prediction of difference propagation with a differential trail that larger than 21-n. The differential trails probability work with certain differences between the input and output of the n-bit block over the number of rounds. The involutory MDS linear code added to the main round as a resource for diffusion property to prevent the exploiting of linear relations. The proposed MDS equation is obtained with small coefficients that are close to the best solutions that have gotten among several candidates' equations.

It is very difficult to analyses or finds the linear approximations for specific linear expression over a sufficient number of rounds. It is obvious that the differential attacks look for obtaining high probability differentials which depict the correlation between the input and output according to non-linear functionality. From another perspective, the attacker tries to find the affine approximation for the non-linear stage to derive key initialization steps. Therefore, the attacker does the best to get repeated high probability patterns along with the ciphertext parts. These characteristics are iterated number of times and through which the attacker may estimate or deduce specific key-bits. The proposed algorithm maintains 128-bit of a strong ciphering key to overcome the relative short of ciphering key and to keep on an accepted security level. Also, it maintains a sufficient number of rounds since the square attack or reduced key attack is the main reason for breaking some algorithms with reduced rounds. It is good to evaluate and analyze the proposed cipher to give a precise impression of design tradeoffs between the security and the efficiency factors. The key scheduling algorithm mixed the ciphering key with different constant words each with 32-bit to eject the weak and semi-weak generated secret keys. These constant words play an important role in removing the data correlation between each round. The proposed structure encrypts and decrypts the data with the same implementation time to

defeat the timing attacks. It will be possible to change the ciphering key with a larger one in case of encryption sensitive information. The algebraic attacks are more difficult to deduce the algebraic properties from the round transformation. Since the introduced cipher based on well-defined scientific design and coherent mathematical algebraic base. The proposed cipher examined according to the National Institute Standard Technology (NIST) randomness tests and did not give any fluctuate or biases out of random style. The proposed lightweight cipher is compared with other lightweight algorithms regarding to several impact parameters or metrics as shown in Table 1. The aforementioned comparison is based on a number of metrics and measurements described in the following reference [35]. The power of energy consumption per bit for SW& HW implementations is computed as follows:

$$\text{Energy [µJ]} = (\text{Latency [cycles/block]} * \text{Power [µW]}) / \text{block size [bits]} \quad (8)$$

The efficiency for the hardware implementation depicts the trade-off between performance and implementation size that can be computed as (9):

$$\text{HW Efficiency} = \text{Throughput [Kbps]}/\text{Complexity [KGE]} \quad (9)$$

The case of software implementations, the factor or metric is computed by the following (10):

$$\text{SW Efficiency} = \text{Throughput [Kbps]} /\text{Code size [KB]} \quad (10)$$

Where the Throughput refers to the encryption/decryption implementations at a certain frequency as the following formula, where most of the previous studies adopt frequencies of 100 KHz for hardware and 4 MHz for software.

$$\text{Throughput [Kbps]} = Kb/F \quad (11)$$

The required area for representing the hardware implementation of any algorithm is addressed by Gate Equivalent (GE) factor, which is based on the technology applied. Thus, GE metric is computed by dividing the required implementation area by the corresponding area of a NAND2 gate [31]. The GE for the proposed cipher is calculated via a specific simulator tool that is called SPICE (Simulation Program for Integrated Circuits Emphasis).

Table 1. Comparison between the proposed cipher and others lightweight cipher

Algorithm	Type of struc.	Key Size	Block Size	Area (GE)- Gate Equivalent	Throughput (Mbit/s)	Technology [µm]
Proposed Lightweight Cipher 128-bit	SPN	128	128	1190	75	0.18
DESL-56 [9]	Feistel	56	64	1848	44.4	0.18
LED-64 [10]	SPN	64	64	966	5.1	0.18
KlEIN-64 [11]	SPN	64	64	1220	N/A	0.18
PRESENT-80 [12]	SPN	80	64	1570	200.0	0.18
KATAN-64 [13]	Stream	64	64	1054	25.1	0.13
mCrypton-96 [14]	SPN	96	64	2681	492.3	0.13
SEA-96 [15]	Feistel	96	96	3758	103.0	0.13
HIGH-128 [16]	GFN	128	64	3048	188.0	0.25
DESXL-184 [17]	Feistel	184	64	3400	80	0.13
Piccolo-80 [18]	GFN	64	80	1136	237.04	---
CLEFIA-128[19]	GFN	128	128	4950	355.6	0.09

8. CONCLUSIONS

In this paper, a 128-bit lightweight symmetric cipher has been suggested which has SPN involution structure. The proposed cipher designed to be compatible with AES cipher characteristics but with an elegant of lightweight structure. The developed cipher works with byte-oriented operations that have efficient hardware implementation on restricted resources. The round transformation is reduced with lightweight stages to improve the internal mathematical computing for the cipher structure. The main structure uses on-fly computation process as a replacement for S-box design in a non-linear layer. Thus, the submitted cipher concludes several important issues in pertaining to design a reliable lightweight cipher of accepted security margin that has the applicability in constrained environments. The developed lightweight cipher has dedicated to work efficiently on pervasive embedded devices and for the IoT applications, especially for the extremely restricted environment.

REFERENCES

[1] Y. NarasimhaRao, P. Surya Chandra, V. Revathi3, N. Suresh Kumar, "Providing enhanced security in IoT based smart weather system", *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, Vol. 18, No. 1, pp. 9-15, April 2020.

[2] Jessica Velasco, et al, "Internet of things-based (IoT) inventory monitoring refrigerator using arduino sensor network", *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, Vol. 18, No. 1, pp. 508-515, April 2020.

[3] O. A. Dawood, O. I. Hammadi, and T. K. Asman, "Developing a New Secret Symmetric Algorithm for Securing Wireless Applications," in *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, pp. 152-158, 2018.

[4] F. Gianni, S. Mora, and M. Divitini, "RapIoT toolkit: Rapid prototyping of collaborative Internet of Things applications," *Futur. Gener. Comput. Syst.*, vol. 95, pp. 867-879, Jun. 2019.

[5] Uma Narayanan, Varghese Paul and Shelbi Joseph, "A light weight encryption over big data in information stockpiling on cloud", *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 17, No. 1, pp. 389-397 January 2020.

[6] O. A. Dawood, A. M. Sagheer, and S. S. Al-Rawi, "Design Large Symmetric Algorithm for Securing Big Data," in *2018 11th International Conference on Developments in eSystems Engineering (DeSE)*, pp. 123-128, 2018.

[7] F. Armknecht, V. Mikhalev, On lightweight stream ciphers with shorter internal states, in *FSE*. LNCS, vol. 9054, pp. 451-470, (Springer, 2015).

[8] M. Kumar, S. K. Pal, and A. Panigrahi,"FeW: a lightweight block cipher", IACR Cryptology ePrint Archive, 2014.

[9] G. Leander, C. Paar, A. Poschmann, and K. Schramm, "New Lightweight DES Variants," *Fast Softw. Encryption*, vol. 4593, pp. 196-210, 2007.

[10] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED block cipher," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6917 LNCS, pp. 326-341, 2011.

[11] Z. Gong, S. Nikova, and Y. W. Law, "KLEIN: A new family of lightweight block ciphers," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7055 LNCS, pp. 1-18, 2012.

[12] M. Izadi, B. Sadeghiyan, S. S. Sadeghian, and H. A. Khanooki, "MIBS: A new lightweight block cipher BT - 8th International Conference on Cryptology and Network Security, CANS 2009, December 12, 2009 - December 14, 2009," vol. 5888 LNCS, pp. 334-348, 2009.

[13] C. De Canniere, O. Dunkelman, and M. Knezevic, "KATAN & KTANTAN - A Family of Small and Efficient Hardware-Oriented Block Ciphers," *Cryptogr. Hardw. Embed. Syst. CHES 2009, Springer, LNCS*, vol. 5747, pp. 272-288, 2009.

[14] T. Lim, C.H., Korkishko, "mCrypton - A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors," vol. 3786, pp. 243-258, 2006.

[15] F. X. Standaert, G. Piret, N. Gershenfeld, and J. J. Quisquater, "SEA: A scalable encryption algorithm for small embedded applications," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3928 LNCS, pp. 222-236, 2006.

[16] D. Hong *et al.*, "HIGHT: A New Block Cipher Suitable for Low-Resource Device," vol. 4249, pp. 46-59, 2006. 10.1007/11894063_4

[17] V. Nachef, J. Patarin, and E. Volte, *Feistel Ciphers*. Cham: Springer International Publishing, 2017.

[18] T. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, "Piccolo: An ultra-lightweight blockcipher," 2011.

[19] M. Katagi and S. Moriai, "The 128-Bit Blockcipher CLEFIA," Springer, Heidelberg, vol. 4593, pp. 181-195, 2011.

[20] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," *Proc. 52nd Annu. Des. Autom. Conf. - DAC '15*, pp. 1-6, 2015.

[21] D. C. Kim *et al.*, "LEA: A 128-bit block cipher for fast encryption on common processors," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8267 LNCS, pp. 286-313, 2014.

[22] D. J. Wheeler and R. M. Needham, "TEA, a tiny encryption algorithm," Lecture Notes in Computer Science. 1995.

[23] H. Cheng, H. M. Heys, and C. Wang, "PUFFIN: A novel compact block cipher targeted to embedded digital systems," *Proc. - 11th EUROMICRO Conf. Digit. Syst. Des. Archit. Methods Tools, DSD 2008*, pp. 383-390, 2008.

[24] D. J. Wheeler and R. M. Needham, "Correction of XTEA," Technical Report, Computer Laborarory, Universoty of Cambridge, October 1998.

[25] 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," Lecture Note in Computer Science (LCNS), Vol. 3017. pp. 279-298, 2011.

[26] Bogdanov, L.R. Knudsen, G. Leander, C. Paar, and A. Poschmann, "PRESENT: An Ultra-Lightweight Block Cipher", *Cryptographic Hardware and Embedded Systems, CHES 2007*, Springer, LNCS, 4727, pp.450-466, 2007.

[27] J. Borghoff *et al.*, "PRINCE - A low-latency block cipher for pervasive computing applications," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7658 LNCS, pp. 208-225, 2012.

[28] S. Matsuda *et al.*, "RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms," *Sci. China Inf. Sci.*, vol. 58, no. 15, pp. 408-425, 2015.

[29] M. R. Albrecht, B. Driessen, E. B. Kavun, and I. Ag, "PRIDE Block Ciphers – Focus On The Linear Layer," pp.57-76, 2012.

[30] M. Sagheer, S. S. Al-Rawi, and O. A. Dawood, "Proposing of developed advance encryption standard", In *Developments in E-systems Engineering (DeSE)*, 6-8 Dec. 2011 Dubai, United Arab Emirates, pp, 197-202. IEEE, 2011. DOI: 10.1109/DeSE.2011.74.
[31] O. A. Dawood, A. M. S. Rahma, and A. M. J. Abdul Hossen, "The New Block Cipher Design (Tigris Cipher)," *Int. J. Comput. Netw. Inf. Secur.*, vol. 7, no. 12, pp. 10-18, 2015.
[32] O. A. Dawood, A. M. S. Rahma, and A. M. J. Abdul Hossen, "New Symmetric Cipher Fast Algorithm of Revertible Operations' Queen (FAROQ) Cipher," *Int. J. Comput. Netw. Inf. Secur.*, vol. 9, no. 4, pp. 29-36, Apr. 2017.
[33] O. A. Dawood, A. M. S. Rahma, A. Mohssen, and J. A. Hossen, "The Euphrates Cipher," *IJCSI Int. J. Comput. Sci. Issues*, vol. 12, no. 2, pp. 154-160, 2015.
[34] O. A. Dawood, Ali M. Sagheer, and Salah Sleibi Al-Rawi. "Design large symmetric algorithm for securing big data" In 2018, *11th IEEE International Conference on Developments in eSystems Engineering (DeSE)*, pp. 123-128. 2018.
[35] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, and C. Manifavas, "A review of lightweight block ciphers," *J. Cryptogr. Eng.*, vol. 8, no. 2, pp. 141-184, 2018.

BIOGRAPHY OF AUTHOR

Omar A. Dawood was born in Habanyah, Anbar, Iraq (1986). He got B.Sc. (2008), M.Sc. (2011) in Computer Science from the College of Computer Science and Information Technology, University of Anbar, Iraq. He was ranking the first during his B.Sc. and M.Sc. studies. He got the Ph.D from the Computer Science Department, University of Technology-Baghdad, Iraq (2015). He is a teaching staff member in University of Anbar, His research interests are: Data and Network Security, Coding, Number Theory, Cryptography, Algorithms' Design.