İSTANBUL UNIVERSITY CERRAHPAŞA

RESEARCH ARTICLE

Electrica 2020; 20(2): 168-176

A Lightweight Cryptography Algorithm for Secure Smart Cities and IOT

Ahmed Mohsin Abed[1], Ali Boyacı[2]

[1]Institute of Science and Technology, Istanbul Commerce University, İstanbul, Turkey
[2]Department of Computer Engineering, Istanbul Commerce University, İstanbul, Turkey

Cite this article as: A.A. Abed, A. Boyacı. "A Lightweight Cryptography Algorithm for Secure Smart Cities and IOT". *Electrica*, 2020; 20(2): 168-176.

ABSTRACT

Lightweight cryptography has a major role in cybersecurity for smart cities and the Internet of Things, and it can be further secured by the development of lightweight encryption algorithms that work in restricted devices and with limited specifications such as wireless network sensors. The restricted devices have a small memory and a simple processor and limited power, and to secure them need lightweight cryptography algorithms and take into account the limited specifications at the same time. In addition to the importance of decreasing memory and power consumption in lightweight encryption algorithms are important, increasing security is more important. When the lightweight cryptography algorithm more secure and less consumption of memory and energy that means it is better and more efficient. In this article, we have developed a lightweight cryptographic algorithm, and through studies, analysis and comparison with other lightweight cryptography algorithms show that the proposed algorithm is more efficient.
Keywords: Lightweight cryptography, internet of things, smart cities

Introduction

A report of the United Nations on 16 May 2018 Indicates that 55% of the world lives in the cities and will attend to 68% in 2050, and it emphasizes that governments should focus on integrated policies to improve the lives of both cities and rural dwellers [1]. In addition to the application of some Internet of things services in the cities, some can also be applied in rural areas, for example in irrigation and agriculture. There are many benefits in the trend toward urbanization, and many challenges. Consolidating a growing population in a smaller space can help conserve resources and other benefits, and will have some Challenges, as cybersecurity threats especially in smart cities and Internet of things applications. There are many difficulties in the development of Smart Cities and Internet of Things applications that need to world keen endeavors in scholarly and industry research. To reach better ideas, innovations and solutions to improve and develop departments and services and overcome difficulties, challenges, risks and threats including cybersecurity.

Cybersecurity is protecting the confidentiality, integrity and availability of information-whether it is personally identifiable information, email or other of communication, credit card numbers, intellectual property or government secrets and other data, information and the services that connect through the internet [2].

Smart cities are radically dependent on the Internet of Things (IoT) and that will be connected to sensors that connect to the Internet through specific protocols for communications, information exchange and data. The sensors have very limited specifications such as memory, speed, processing and power. To secure them, they need lightweight and highly efficient cryptography algorithms.

The lightweight Cryptography algorithms designed to work in many highly restrictive environments (eg. sensor networks, healthcare, distributed control systems, Internet of things, and electronic physical systems) where they use low power and will use circuits much more limited than those on the simplest mobile phones. These devices are usually connected to each other wirelessly to work in coordination to accomplish some tasks. Because cryptographic algorithms designed for desktop environments and servers, most of them are not compatible with restricted devices [3].

There are many lightweight algorithms (AES, RC5, PRESENT, Simon, Speck, HIGHT, LEA etc.), But they vary in terms of memory consumption and energy, in addition to the factor most im-

Corresponding Author:
Ali Boyacı

E-mail:
aboyaci@ticaret.edu.tr

Received: 23.03.2020

Accepted: 27.04.2020

DOI: 10.5152/electrica.2020.20002

Content of this journal is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

portant that is security. Despite the abundance of algorithms and the efficiency of some of them, it still needs to develop, improve and search for better algorithms and solutions because of the difficulty of obtaining the required level of security in the lightweight cryptography algorithms to take into account the specifications of the restricted devices at the same time.

Review of the Literature

Can increase of the security level for the cryptography algorithms by increasing the size of the block, the size of the key or the number of rounds [4], but sometimes negatively affecting its speed and consumption of energy and memory. In addition to increasing the block size, key size and number of rounds that lead to increase the security level, the design of the algorithm's structure, function and generator of sub keys and operators play a very large role in the strength level of the algorithm's security.

Can improve the strength of certain algorithms by making some changes and improvements, for example, Triple DES or 3DES algorithm derived from the DES algorithm by adding some changes and improvements such as increasing the size of the key or other changes that would lead to increase the strength of the algorithm. As example of lightweight algorithms is the case for the TEA (Tiny Encryption Algorithm) and XTEA (Extended Tiny Encryption Algorithm), XXTEA. TEA and XTEA is a Feistel cipher with 64-bit block, 128-bit key and 64 rounds, but there are some differences, including a more complex key-schedule, XORs, other additions and rearrangement of the shifts' Also, and XXTEA algorithm is a Feistel cipher with 64-bit block, 128-bit key and also contains some changes.

In block cipher, some algorithms use Feistel structure where the Block is split into two halves and their size is equal in the balanced algorithms like DES, Simon and others, and in two different sizes in the unbalanced algorithms as in the Skipjack algorithm, XXTEA and others. One of the major advantage of using Feistel structure is that the encryption and decryption operations are almost same. Some other algorithms use Substitution-Permutation Network (SPN) structure, such as AES and present and prince and others, Several alternating rounds of substitution and permutation leads to the confusion and spread of Shannon properties that require changing the encryption text in a pseudo-random way [5].

There is an algorithm (SIT algorithm with key size 64 bit, block size 64 bit and five rounds), it is a mixture based on Feistel and SP networks. Thus making use of the advantages of both approaches to develop a lightweight algorithm that presents more security in the IoT environment while keeping the computational complexity at a moderate level [5].

Although the SIT algorithm it is designed to reduce power and memory consumption and increase speed, it needs to improve some processes add operations that are more complex, in the structure and sub-key generator and increase of key size, to increase of security level with taking into account the consumption of energy and memory on the same time.

To be the cryptography algorithms more secure, the key size must be at least 128 bits as studies suggest in recent years from the National Institute of Standards and Technology, NIST [6]. In addition to when the key size larger, the key space will be larger and more resistant against brute force attack.

The proposed algorithm in this article like SIT algorithm but with the key size 128 bit and block size 64 bit and nine rounds, and added many operations for algorithm structure and sub-key generator and add other functions.

And from the operations that added, operations that used in TEA XTEA XXTEA algorithms like shift operation and Delta number, Delta is the golden number or 9E3779B9 [7].

We made the improvements above to make it more complex and efficiency and more secure, and according to the next analyzes, studies and comparisons about the proposed algorithm, that it is more secure and more efficient compared with other algorithms.

Proposed Algorithm

Proposed Algorithm and Encryption

The proposed algorithm is a lightweight encryption algorithm with a block size equal 64-bit, key size equal 128-bit and number of rounds is nine rounds. The proposed algorithm is a mixture of Feistel structure and Substitution-Permutation network SPN structure.

Some algorithms have the size of sub-keys equal to the size of the main keys and some algorithms have the main key size different from the sub-keys sizes, in the proposed algorithm have sub-keys size is different from the master key size where the size of the sub-keys is 16 bits while the key size The main is 128-bit.

The main key divided to eight segments, then each segment equal 16 bit and each one divided to eight sub-segments, each sub-segment equal to two bits, this means we have 64 sub-segments of 2 bits, and through the substitution of segments, we will get on new eight segments, as shown following equations:

$$S_a = s_1 s_2 \ddagger s_{17} \ddagger s_{18} \ddagger s_{33} \ddagger s_{34} \ddagger s_{49} \ddagger s_{50}$$

$$S_b = s_9 \ddagger s_{10} \ddagger s_{25} \ddagger s_{26} \ddagger s_{41} \ddagger s_{42} \ddagger s_{57} \ddagger s_{58}$$

$$S_c = \|_{j=1}^{8} S_{8(j-1)+3}$$

$$S_d = \|_{j=1}^{8} S_{8(j-1)+4}$$

$$S_e = s_5 \ddagger s_6 \ddagger s_{21} \ddagger s_{22} \ddagger s_{37} \ddagger s_{38} \ddagger s_{53} \ddagger s_{54}$$

$$S_f = s_{13} \ddagger s_{14} \ddagger s_{29} \ddagger s_{30} \ddagger s_{45} \ddagger s_{46} \ddagger s_{61} \ddagger s_{62}$$

$$S_g = \|_{j=1}^{8} S_{8(j-1)+7}$$

$$S_h = \|_{j=1}^{8} S_{8(j-1)+8}$$

where is a segment and is a sub-segment. After producing eight segments the Function 1 used for the segments, the functions has special design and depends on P and Q tables show in Figure 1, the transformations made by P and Q are shown in the Tables 1 and 2.

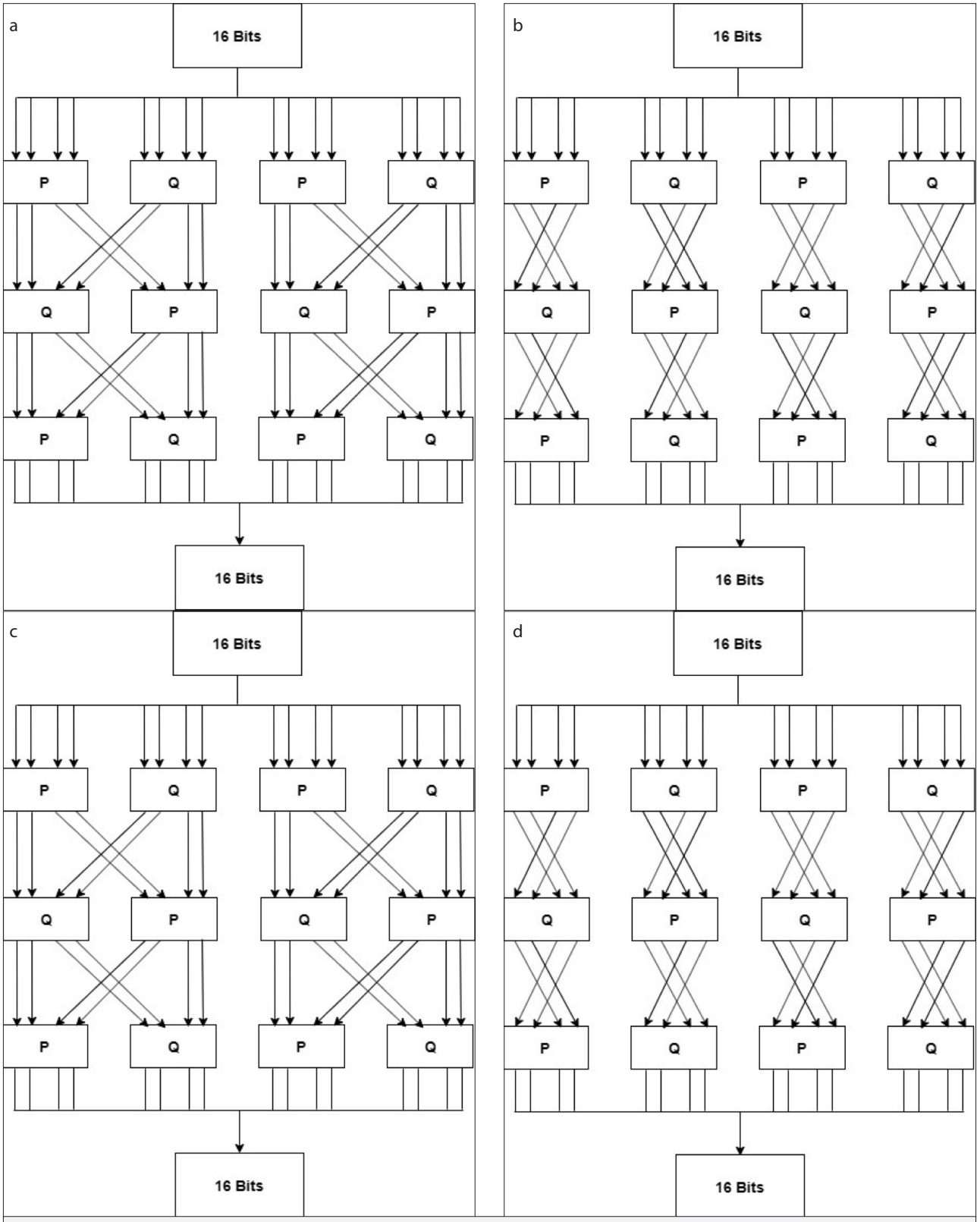


Figure 1. a-d. Function 1(a), Function 2 (b), Function 3 (c), Function 4 (d)

Table 1. p table

S_"i "	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
p(S_"i ")	3	F	E	0	5	4	B	C	D	A	9	6	7	8	2	1

Table 2. Q table

S_"i "	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Q(S_"i ")	9	E	5	6	A	2	3	C	F	0	4	D	7	B	1	8

$$Ka_i f = f(S_i) \quad (1)$$

where $i = (a, b, c, d, e, f, g, h)$. The next step is to get $Ka_i f$ by passing the 16-bits of S_i to the Function 1 as shown in Equation (1). The Function 1 applied in sub-keys generator and in algorithm structure (Encryption structure), while the Functions 2, 3 and 4 applied in Encryption structure only.

The tables and function perform to linear and non-linear transformations for the bits to get confusion and diffusion that can be seen in Tables 1 and 2.

The outputs (a, b, c, d, e, f, g, h) of each function is arranged in a 4×4 matrix named $\boldsymbol{M}$ shown next, but some operations are applied to some functions before arranging them in the matrixes as following equations:

$$\mathrm{bs} = b \ll 4$$

$$\mathrm{ds} = d \gg 8$$

$$\mathrm{fs} = f \ll 4$$

$$\mathrm{hs} = h \gg 8$$

$$M_a = \begin{bmatrix} Ka_1f_1 & Ka_1f_2 & Ka_1f_3 & Ka_1f_4 \\ Ka_1f_5 & Ka_1f_6 & Ka_1f_7 & Ka_1f_8 \\ Ka_1f_9 & Ka_1f_{10} & Ka_1f_{11} & Ka_1f_{12} \\ Ka_1f_{13} & Ka_1f_{14} & Ka_1f_{15} & Ka_1f_{16} \end{bmatrix}$$

$$M_b = \begin{bmatrix} Ka_2f_1 & Ka_2f_2 & Ka_2f_3 & Ka_2f_4 \\ Ka_2f_5 & Ka_2f_6 & Ka_2f_7 & Ka_2f_8 \\ Ka_2f_9 & Ka_2f_{10} & Ka_2f_{11} & Ka_2f_{12} \\ Ka_2f_{13} & Ka_2f_{14} & Ka_2f_{15} & Ka_2f_{16} \end{bmatrix}$$

$$M_c = \begin{bmatrix} Ka_3f_1 & Ka_3f_2 & Ka_3f_3 & Ka_3f_4 \\ Ka_3f_5 & Ka_3f_6 & Ka_3f_7 & Ka_3f_8 \\ Ka_3f_9 & Ka_3f_{10} & Ka_3f_{11} & Ka_3f_{12} \\ Ka_3f_{13} & Ka_3f_{14} & Ka_3f_{15} & Ka_3f_{16} \end{bmatrix}$$

$$M_d = \begin{bmatrix} Ka_4f_1 & Ka_4f_2 & Ka_4f_3 & Ka_4f_4 \\ Ka_4f_5 & Ka_4f_6 & Ka_4f_7 & Ka_4f_8 \\ Ka_4f_9 & Ka_4f_{10} & Ka_4f_{11} & Ka_4f_{12} \\ Ka_4f_{13} & Ka_4f_{14} & Ka_4f_{15} & Ka_4f_{16} \end{bmatrix}$$

$$M_e = \begin{bmatrix} Ka_5f_1 & Ka_5f_2 & Ka_5f_3 & Ka_5f_4 \\ Ka_5f_5 & Ka_5f_6 & Ka_5f_7 & Ka_5f_8 \\ Ka_5f_9 & Ka_5f_{10} & Ka_5f_{11} & Ka_5f_{12} \\ Ka_5f_{13} & Ka_5f_{14} & Ka_5f_{15} & Ka_5f_{16} \end{bmatrix}$$

$$M_f = \begin{bmatrix} Ka_6f_1 & Ka_6f_2 & Ka_6f_3 & Ka_6f_4 \\ Ka_6f_5 & Ka_6f_6 & Ka_6f_7 & Ka_6f_8 \\ Ka_6f_9 & Ka_6f_{10} & Ka_6f_{11} & Ka_6f_{12} \\ Ka_6f_{13} & Ka_6f_{14} & Ka_6f_{15} & Ka_6f_{16} \end{bmatrix}$$

$$M_g = \begin{bmatrix} Ka_7f_1 & Ka_7f_2 & Ka_7f_3 & Ka_7f_4 \\ Ka_7f_5 & Ka_7f_6 & Ka_7f_7 & Ka_7f_8 \\ Ka_7f_9 & Ka_7f_{10} & Ka_7f_{11} & Ka_7f_{12} \\ Ka_7f_{13} & Ka_7f_{14} & Ka_7f_{15} & Ka_7f_{16} \end{bmatrix}$$

$$M_h = \begin{bmatrix} Ka_8f_1 & Ka_8f_2 & Ka_8f_3 & Ka_8f_4 \\ Ka_8f_5 & Ka_8f_6 & Ka_8f_7 & Ka_8f_8 \\ Ka_8f_9 & Ka_8f_{10} & Ka_8f_{11} & Ka_8f_{12} \\ Ka_8f_{13} & Ka_8f_{14} & Ka_8f_{15} & Ka_8f_{16} \end{bmatrix}$$

Then apply on XNOR on matrix $\boldsymbol{M}_{\mathbf{a}}$ and opposite it, and apply XOR on matrix $\boldsymbol{M}_{\mathbf{h}}$ and opposite it as follow equations. That to make the sub-keys more complexity and more independent and from this sub-keys *K9* because those sub-keys shared by creating sub-key nine.

$$M_{a2} = \left(\overline{M_a \oplus \overline{M_a}}\right) \quad (2)$$

$$M_{h2} = M_h \oplus \overline{M_h} \quad (3)$$

To obtain the matrices are transformed into eight arrays of 16 bits, the arrangement of these bits are shown in following equations:

$$A = a_4 + a_3 + a_2 + a_1 + a_5 + a_6 + a_7 + a_8 + a_{12} + a_{11} + a_{10} + a_9 + a_{13} + a_{14} + a_{15} + a_{16}$$

$$B = b_1 + b_5 + b_9 + b_{13} + b_{14} + b_{10} + b_6 + b_2 + b_3 + b_7 + b_{11} + b_{15} + b_{16} + b_{12} + b_8 + b_4$$

$$C = c_1 + c_2 + c_3 + c_4 + c_8 + c_7 + c_6 + c_5 + c_9 + c_{10} + c_{11} + c_{12} + c_{16} + c_{15} + c_{14} + c_{13}$$

$$D = d_{13} + d_9 + d_5 + d_1 + d_2 + d_6 + d_{10} + d_{14} + d_{15} + d_{11} + d_7 + d_3 + d_4 + d_8 + d_{12} + d_{16}$$

$$E = e_4 + e_3 + e_2 + e_1 + e_5 + e_6 + e_7 + e_8 + e_{12} + e_{11} + e_{10} + e_9 + e_{13} + e_{14} + e_{15} + e_{16}$$

$$F = f_1 + f_5 + f_9 + f_{13} + f_{14} + f_{10} + f_6 + f_2 + f_3 + f_7 + f_{11} + f_{15} + f_{16} + f_{12} + f_8 + f_4$$

$$G = g_1 + g_2 + g_3 + g_4 + g_8 + g_7 + g_6 + g_5 + g_9 + g_{10} + g_{11} + g_{12} + g_{16} + g_{15} + g_{14} + g_{13}$$

$$H = h_{13} + h_9 + h_5 + h_1 + h_2 + h_6 + h_{10} + h_{14} + h_{15} + h_{11} + h_7 + h_3 + h_4 + h_8 + h_{12} + h_{16}$$

From next equations, we will get on sub-keys (K1, K3, K5, K7), through shift five bits to the right for previous arrays (A, C, E, G), we will get on sub-keys (K2, K4, K6, K8) through shift nine bits to the left for previous arrays (B, D, F, H). Then we will get on sub key nine (K9), by use XOR of shift five bits to the left for T1 (Equation 5) and shift nine bits to the right for T2 (Equation 6), where T1 equal XOR of K1 opposite (NOT) and D, T2 equal XOR of K5 and NOT of H.

$$\mathrm{K1} = A \gg 5$$

$$\mathrm{K2} = B \ll 9$$

$$\mathrm{K3} = C \gg 5$$

$$\mathrm{K4} = D \ll 9$$

$$\mathrm{K5} = E \gg 5$$

$$\mathrm{K6} = F \ll 9$$

$$\mathrm{K7} = G \gg 5$$

$$\mathrm{K8} = H \ll 9$$

$$\mathrm{T1} = \overline{K1} \oplus D \quad (4)$$

$$\mathrm{T2} = K5 \oplus \overline{H} \quad (5)$$

$$\mathrm{K9} = (T1 \ll 5) \oplus (T2 \gg 9) \quad (6)$$

In Figure 2, we saw how creating nine sub-keys from the main key. In the next equations, we will explain the algorithm structure and encryption. In the begin, the main block divided to four blocks (each block equal 16 bits) and each block divided to eight subblocks (each block equal two bits), this means we have 32 subblocks of two bits, then produce six blocks through next equations:

$$Ba_1 = \mathrm{b}_{26} \# \mathrm{b}_{18} \# \mathrm{b}_{10} \# \mathrm{b}_2 \# \mathrm{b}_{32} \# \mathrm{b}_{24} \# \mathrm{b}_{16} \# \mathrm{b}_8$$

$$Bb_1 = \mathrm{b}_{30} \# \mathrm{b}_{22} \# \mathrm{b}_{14} \# \mathrm{b}_6 \# \overline{\mathrm{b}_{30}} \# \overline{\mathrm{b}_{22}} \# \overline{\mathrm{b}_{14}} \# \overline{\mathrm{b}_6}$$

$$Bc_1 = \mathrm{b}_{29} \# \mathrm{b}_{21} \# \mathrm{b}_{13} \# \mathrm{b}_5 \# \overline{\mathrm{b}_{29}} \# \overline{\mathrm{b}_{21}} \# \overline{\mathrm{b}_{13}} \# \overline{\mathrm{b}_5}$$

$$Bd_1 = \mathrm{b}_{28} \# \mathrm{b}_{20} \# \mathrm{b}_{12} \# \mathrm{b}_4 \# \overline{\mathrm{b}_{28}} \# \overline{\mathrm{b}_{20}} \# \overline{\mathrm{b}_{12}} \# \overline{\mathrm{b}_4}$$

$$Be_1 = \mathrm{b}_{27} \# \mathrm{b}_{19} \# \mathrm{b}_{11} \# \mathrm{b}_3 \# \overline{\mathrm{b}_{27}} \# \overline{\mathrm{b}_{19}} \# \overline{\mathrm{b}_{11}} \# \overline{\mathrm{b}_3}$$

$$Bf_1 = \mathrm{b}_{25} \# \mathrm{b}_{17} \# \mathrm{b}_9 \# \mathrm{b}_1 \# \mathrm{b}_{31} \# \mathrm{b}_{23} \# \mathrm{b}_{15} \# \mathrm{b}_7$$

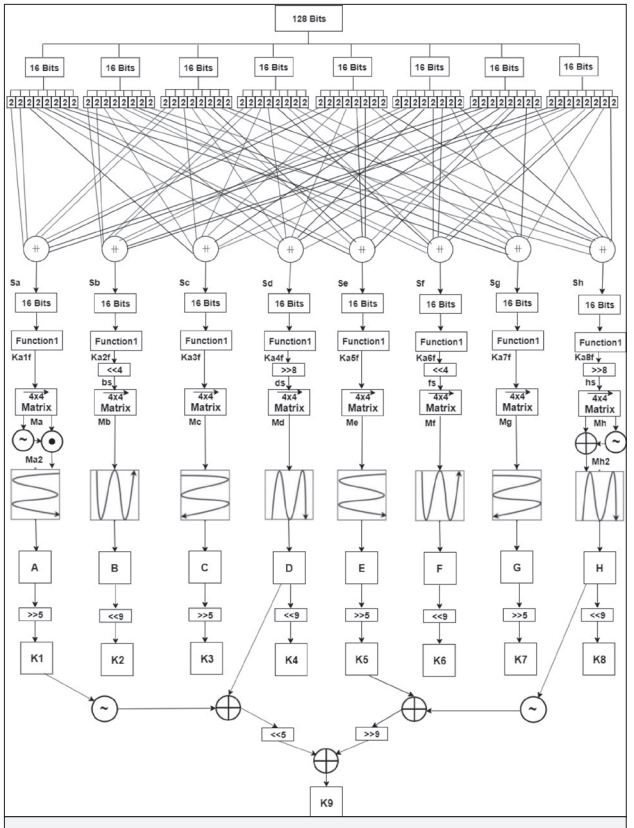


Figure 2. Sub-keys generator

where B means Block, b means sub-block. Then in every round, will be obtained new blocks through the following equations:

$$Ba_2 = (Ba_1 \gg 5) \oplus K1 \quad (7)$$

$$Bb_2 = \left(\overline{Bb_1 \oplus K1}\right) \oplus Delta \quad (8)$$

$$Bc_2 = F_1\left(\overline{Bb_1 \oplus K1}\right) \oplus Bd_1 \quad (9)$$

$$Bd_2 = \left(F_1\left(\overline{Be_1 \oplus K1}\right) \oplus Bc_2\right) \oplus Delta \quad (10)$$

$$Be_2 = \left(\overline{Be_1 \oplus K1}\right) \quad (11)$$

$$Bf_2 = (Bf_1 \gg 5) \oplus (\mathrm{K1} \oplus Delta) \quad (12)$$

where $F_1 = Function\ 1$. After nine rounds, there are some operations to get on the block cipher from this operations, three functions (Function 1, Function 2, Function 3, Function 4), shown following equations:

$$B_1 = F_1(Bf_{10}) \quad (13)$$

$$B_2 = F_4(Bb_{10}) \oplus F_2(Bc_{10}) \quad (14)$$

$$B_3 = F_1(Bd_{10}) \oplus F_3(Be_{10}) \quad (15)$$

$$B_4 = F_1(Ba_{10}) \quad (16)$$

$$BC = B_2 \# B_3 \# B_1 \# B_4 \quad (17)$$

where $BC = Block\ Cipher$, $F_1 = Function\ 1$, $F_2 = Function\ 2$, $F_3 = Function\ 3$, $F_4 = Function\ 4$.

Analysis of Proposed Algorithm and Encryption

Linear and Differential Cryptanalysis

In the proposed algorithm, shown in Figure 3, and the functions 1, 2, 3 and 4 are inspired by [8] whose cryptanalysis shows in the complete cipher that differential and linear attacks does not have the success. where the input and output correlation is so large if the linear approximation is done for two rounds, Also, the round transformation is preserved uniform, that treats every bit in a similar way and provides resistance against differential attacks.

Correlation Coefficient Analysis

A correlation is a statistical measure of the relationship between two variables, and it ranges between +1 and –1. positive one which that means that variables move in the one direction along which means an ideal correlation, whereas zero implies that there's no relationship between the variables and if negative one refers to an ideal indirect correlation, which means that one variable goes up, the other goes down.

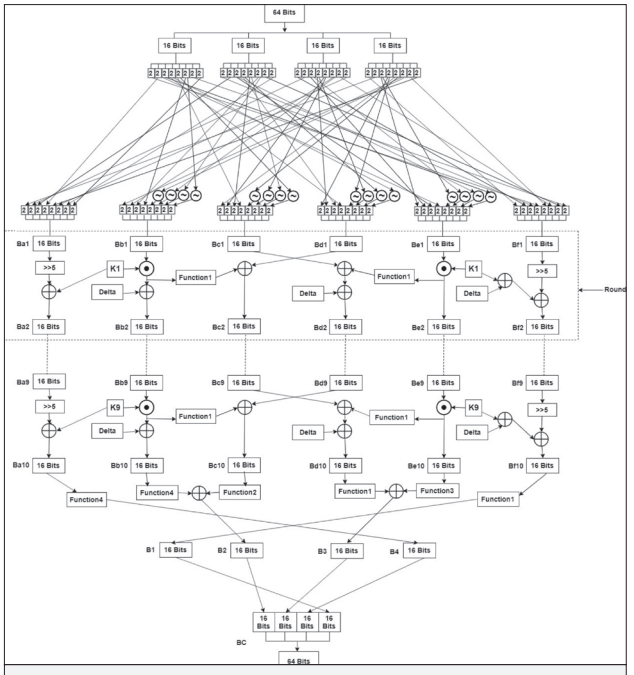


Figure 3. Structure of the Proposed Algorithm (encryption)

In the encryption of images means the result of the correlation that the closer to zero (if positive or Negative), that relationship between the encrypted image and the original image is weaker and this means that the level of encryption stronger. If that the result of the correlation farther away from zero, whether negative or positive means that the relationship between the original image and encrypted image stronger this means that the encryption level is weaker. If that result of correlation equal zero this means there is no relationship between the original and encrypted image and refers to the best level of encryption [8, 9].

$$\mathrm{CR} = \frac{\mathrm{cov}(\mathrm{X},\mathrm{Y})}{\sqrt{D(X)}\,\sqrt{D(Y)}} \tag{18}$$

$$\mathrm{cov}(\mathrm{X},\mathrm{Y}) = \frac{1}{256}\sum_{1}^{256}(X - E(X))(Yi - E(Y)) \tag{19}$$

where, X and Y are the pixels and neighboring pixels of the original and encrypted image, $cov(X,Y)$ is the covariance between X and Y, D(X) is variance of X, and E(X) is the expected value of X.

Information Entropy Analysis

The entropy of information estimates the uncertainty of a random variable. When the entropy applied to evaluate image encryption, the larger value of entropy refers to a greater level of security, and it is secure from a brute force attack when an entropy value that is very close to a perfect value of eight.

$$\mathrm{E} = \sum_{i=0}^{255} \mathrm{P(i)}\log\left(\frac{1}{\mathrm{P(i)}}\right) \tag{20}$$

where E is Entropy, $P(i)$ is the probability of the presence of pixel i.

Histogram Analysis

The histogram of the image is a graphical and statistical representation for the distribution of pixel values information. The histogram of the perfect encrypted image must be a uniformly distributed and fully different in comparison with the original image, to prevent extract any information from the histogram for the encrypted image [10].

In the histogram, the image is highly random and highly resistant against the statistical attacks if the intensity of the pixels is uniformly distributed. [11].

Key Space Analysis

Key space means number of bits that used to encrypt image, and for good encryption, that key space should be as large as possible to repulse brute force attack. Key space size means the total number of different keys of same number of bits use for encryption [12]. For high security, the key space should be greater than [13]. In proposed algorithm key size is 128 bit and according to equation (62) the key space is this mean key space size is good to repulse brute force attack.

$$\mathrm{Ks} = 2^{Kz} \tag{21}$$

where Ks is key space and Kz is key size.

Related Keys

Through performing cipher operations and by using unknown or partially known keys can be made an attack. The related key attack predominately depends on having symmetry in key expansion block or upon either slow diffusion In the proposed algorithm, the sub-keys process is designed for fast and non-linear diffusion to the difference of the main key and sub-keys.

Interpolation Attacks

Dependent these attacks on the simple structures for the cipher components that may yield a rationalistic expression with a handy intricacy. In the proposed algorithm, the S-box expression for along with the diffusion layer makes this type of attack impractical.

Other Analysis

There are important things that must be taken into account in the lightweight cryptography algorithms are the speed or processing time, the amount of memory and energy consumption of the algorithm during data encryption operations, as the proposed algorithm was designed to suit Internet of things environments. The memory and energy consumption are directly proportional but inversely proportional to time and speed.

Results

Through some analysis for the proposed algorithm, where the simulations were performed on a desktop computer with Intel(R) Core(TM) i7 CPU L 620 @2.00 GHz., 4GB RAM., and Win-

Table 3. Correlation and entropy analysis

		Correlation		Entropy	
Image	**Size**	**Original image**	**Encrypted image**	**Original image**	**Encrypted image**
Lena	256 x 256	0.9744	0.0001	7.4509	7.9976
Panda	256 x 256	0.9811	-0.0005	7.4938	7.9971

Table 4. Comparison between the proposed algorithm and other algorithms

The Algorithm	**Block Size**	**Key Size**	**Rounds**	**Code Size**	**RAM**
PRESENT	64	80	32	1738	274
Simon	64	96	42	1370	188
Speck	64	96	26	2552	124
SIT	64	64	5	826	22
AES	128	128	10	23090	720
LEA	128	128	24	3700	432
RC5	64	128	20	20044	360
HIGHT	64	128	32	13476	288
Proposed	64	128	9	823	144

dows 10 Professional operating system, and by using MATLAB R2015a. In addition to some other algorithms on FELICS. We gets the results in Table 3 as shown Figures 4 and 5.

From the results show the proposed algorithm high efficient in comparison with another algorithm especially with those al-

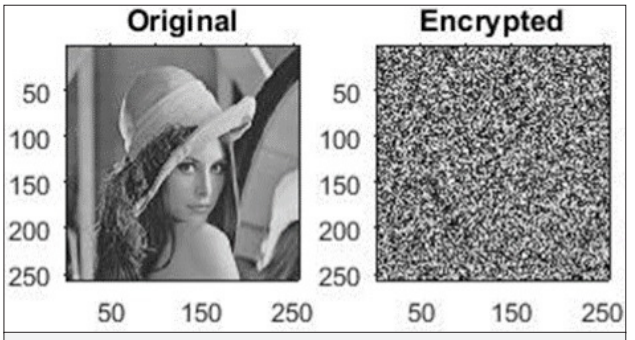


Figure 4. Encryption of Lena image

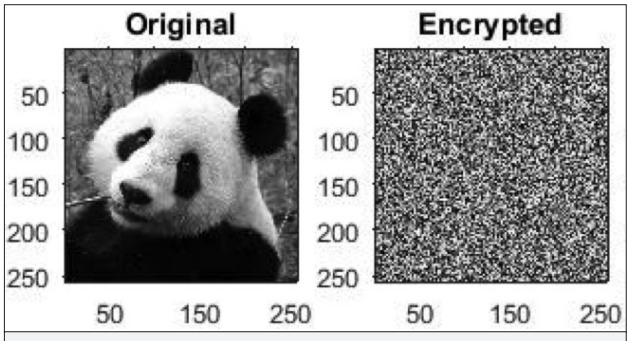


Figure 5. Encryption of Panda image

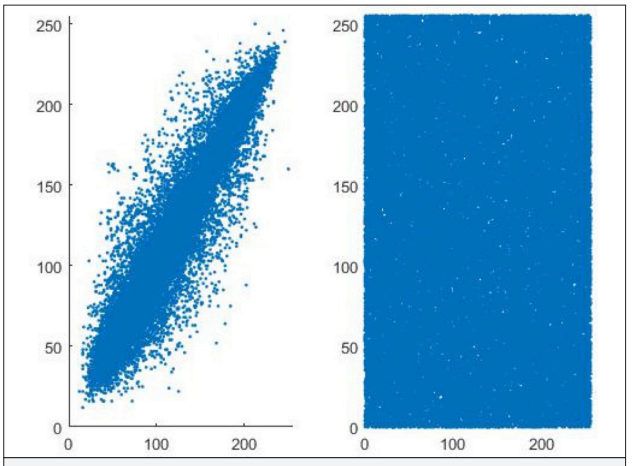


Figure 6. Correlation of Lena image

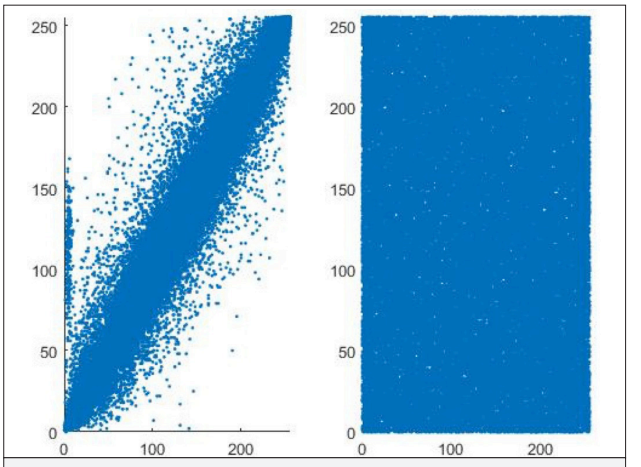


Figure 7. Correlation of Panda image

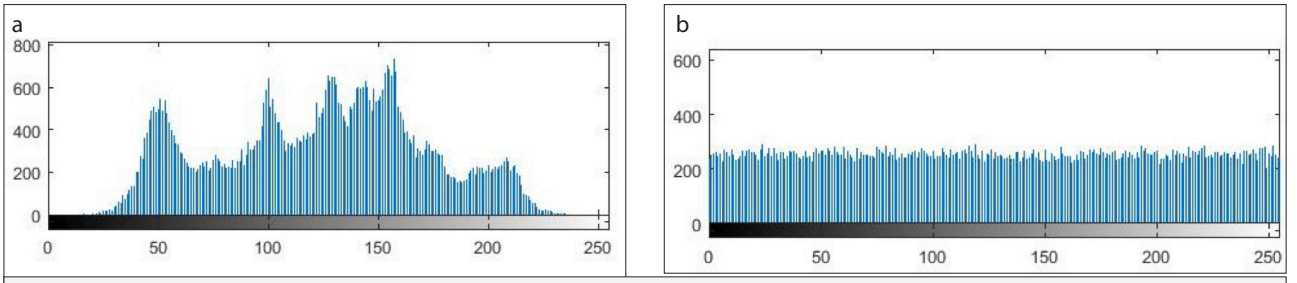


Figure 8. a, b. Histogram Graph of Original Lena (a), Histogram Graph of Encrypted Lena (b)

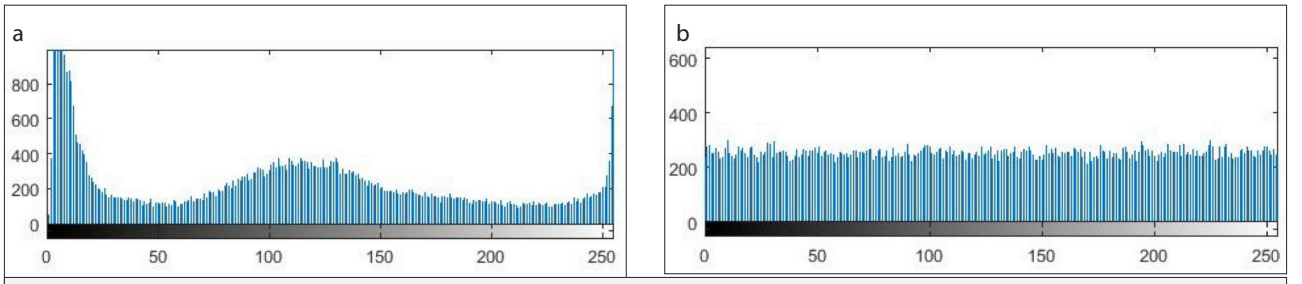


Figure 9. a, b. Histogram Graph of Original Lena (a), Histogram Graph of Encrypted Lena (b)

gorithms of 128 bit. In Table 4, the Block size and key size are in bits, while the code size and RAM are in bytes.

As can be seen clearly on Figures 6 and 7 that the relationship between the encoded and original images is very weak, this is inversely proportional to the strength of the encryption. In the histogram comparison as shown in Figures 8 and 9, we notice a large difference in the distribution of data between the original and encrypted images as, and this indicates the strength of the encryption for the proposed algorithm.

Conclusion

The improvement and development of lightweight cryptography algorithms are necessary to improve the security of the Internet of things and smart cities and take into account its specification. Through the results in this article, the proposed algorithm is high-security with low memory and energy consumption, which are directly proportional, and it Convenient to the environment of the internet of things.

The Security can be improved by increase of key size and block size and rounds number, in addition, to change the function and operation in the algorithm but maybe that will make it not convenient to internet of things environment if high consumption for memory and power.

Peer-review: Externally peer-reviewed.

Conflict of Interest: The authors have no conflicts of interest to declare.

Financial Disclosure: The authors declared that this study has received no financial support.

References

1. United Nations, "2018 Revision of World Urbanization Prospects," United Nations: Department of Economic and Social Affairs, Available from URL: https://www.un.org/development/desa/publications/2018-revision-of-world-urbanization-prospects.html.
2. P. W. Singer, A. Friedman, "Cybersecurity and Cyberwar," Oxford University Press, New York, p. 320, 2014.
3. Information Technology Laboratory: Computer Security Resource Center, "Lightweight Cryptography," The National Institute of Standards and Technology (NIST), Available from URL: https://csrc.nist.gov/projects/lightweight-cryptography.
4. W. Stallings, "Cryptography and Network Security: Principles and Practice," 5th Ed. Pearson, New York, 2010, p. 744.
5. M. Usman, I. Ahmed, M. I. Aslam, S. Khan and U. A. Shah, "SIT: A Lightweight Encryption Algorithm for Secure Internet of Things," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 8, no. 1, pp. 402-411, 2017.
6. E. Barker, "National Institute of Standards and Technology: Special Publication," part 1, revision 4, 800-57, 2016.
7. V. R. Andem, "A Cryptanalysis of the Tiny Encryption Algorithm," M.Sc. Thesis, The Graduate School, University of Alabama, Alabama, 2003.
8. P. S. L. M. Barreto and V. Rijmen, "The KHAZAD Legacy-Level Block Cipher," vol. 97, pp. 1-20, 2000.
9. P. Ramasamy, V. Ranganathan, S. Kadry and et.al, "An Image Encryption Scheme Based on Block Scrambling," Modified Zigzag Transformation and Key Generation Using Enhanced Logistic-Tent Map. Entropy, vol. 21, no. 7, pp. 1-7, 2019.
10. H. N. Abdullah, "Image Encryption Using Hybrid Chaotic Map," International Conference on Current Research in Computer Science and Information Technology (ICCIT), Slemani, Iraq, pp. 121-125, 2017
11. G. Maddodi, A. Awad, D. Awad, et.al, "A New Image Encryption Algorithm Based on Heterogeneous Chaotic Neural Network Generator and DNA Encoding," Springer, vol. 77, no. 19, pp. 24701-24725, 2018.
12. T. Kaur, R. Sharma, "Security Definitive Parameters for Image Encryption Techniques," International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 5, pp. 109-112, 2013.



Ahmed Mohsin Abed received B.Sc. degree in Software Engineering from Baghdad College for Economic Sciences University in 2008. He is currently, M.Sc. Student at Department of Computer Engineering in Istanbul Commerce University. His research interests include Cyber Security for Smart Cities and Internet of Things, Cryptography and Lightweight Cryptography.



Ali Boyacı received the B.S. and M.Sc. degrees in computer science from İstanbul University, İstanbul, Turkey, in 2007 and 2010, respectively, and the Ph.D. degree from the Yıldız Technical University, İstanbul, Turkey, in 2015. He worked as a software engineer at Nortel Networks and project leader at Huawei from 2007 to 2012. Currently, he is an Assistant Professor with the Department of Computer Engineering, İstanbul Commerce University, İstanbul. Ali Boyaci's current research interests include computer networks and embedded systems.