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, Istanbul, Turkey
[2]Department of Computer Engineering, Istanbul Commerce University, Istanbul, 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. It can be further secure 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, a simple processor, and limited power. To secure them, need lightweight cryptography algorithms and take into account the limited specifications at the same time. In addition, decreasing memory and power consumption in lightweight encryption algorithms, increasing security is also essential. When the lightweight cryptography algorithm is more secure and less consumes memory and energy, 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 efficient.

Keywords: Internet of things, lightweight cryptography, smart cities

Introduction

A report of the United Nations on 16 May 2018 indicates that 55% of the world lives in cities will attend to 68% in 2050. It emphasizes that governments should focus on integrated policies to improve the lives of both cities and rural dwellers [1]. In addition, the application of Internet of Things (IoT) services in cities can be applied in rural areas, for example, irrigation, and agriculture. There are many benefits to the trend toward urbanization and many challenges. Consolidating a growing population in a smaller space can help conserve resources and other benefits. It will have some challenges, as cybersecurity threats, especially in smart cities and the IoT applications. There are many difficulties in the development of Smart Cities and IoT 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 as well as 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 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 IoT, and connected to sensors that connect to the Internet through specific protocols for communications, information exchange, and data. The sensors have minimal 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 extremely restrictive environments (e.g., sensor networks, healthcare, distributed control systems, IoT, and electronic and physical systems) where they use low power. They will use circuits much more limited than those on the simplest mobile phones. These devices are usually connected 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.). However, they vary in terms of memory consumption and energy in addition to security— the most important factor. 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.

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.

Review of the Literature

It can increase 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 energy consumption and memory. In addition to increasing the block size, key size, and the number of rounds that lead to an in increase the security level, the design of the algorithm's structure, function, and generator of sub-keys and operators play a huge role in the strength level of the algorithm's security.

It can improve the strength of specific algorithms by making some changes and improvements, such as 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 an example of lightweight algorithms is the case for the Tiny Encryption Algorithm (TEA) and Extended Tiny Encryption Algorithm (XTEA), XXTEA. TEA and XTEA is a Feistel cipher with a 64-bit block, 128-bit key, and 64 rounds. However, there are some differences, including a more complex key-schedule, XORs. It includes other additions and rearrangement of the shifts'. XXTEA algorithm is a Feistel cipher with a 64-bit block, 128-bit key, and also contains some changes.

In block cipher, some algorithms use a Feistel structure where the block split into two halves. Their size is equal in the balanced algorithms such as DES and Simon, and in two different sizes in the unbalanced algorithms as in the Skipjack algorithm, XXTEA, and others. One of the significant advantages of using the Feistel structure is that the encryption and decryption operations are almost the same. Some other algorithms use Substitution-Permutation Network (SPN) structure, such as AES and present and prince. Several alternating rounds of substitution and permutation lead 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, both approaches are used 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 is designed to reduce power and memory consumption and increase speed, it needs to improve some processes by adding more complicated operations in the structure, sub-key generator, and increase of key size, to increase the security level by taking into account the consumption of energy and memory simultaneously.

For the secured cryptography algorithms, 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 the brute force attack.

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

Moreover, 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 mentioned above to make it more complicated and efficient and more secure. According to the next analyzes, studies and comparisons about the proposed algorithm are more secure and efficient than 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 nine rounds. The proposed algorithm is a mixture of Feistel structure and 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. The proposed algorithm has sub-keys size different from the master key size where the size of the sub-keys is 16 bits, while the key size of the main is 128-bit.

The main key divided into eight segments. 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:

$$S_{\mathrm{a}} = \mathrm{s}_1\mathrm{s}_2 \mathbin{+\!\!\!+} \mathrm{s}_{17} \mathbin{+\!\!\!+} \mathrm{s}_{18} \mathbin{+\!\!\!+} \mathrm{s}_{33} \mathbin{+\!\!\!+} \mathrm{s}_{34} \mathbin{+\!\!\!+} \mathrm{s}_{49} \mathbin{+\!\!\!+} \mathrm{s}_{50}$$

$$S_{\mathrm{b}} = \mathrm{s}_9 \mathbin{+\!\!\!+} \mathrm{s}_{10} \mathbin{+\!\!\!+} \mathrm{s}_{25} \mathbin{+\!\!\!+} \mathrm{s}_{26} \mathbin{+\!\!\!+} \mathrm{s}_{41} \mathbin{+\!\!\!+} \mathrm{s}_{42} \mathbin{+\!\!\!+} \mathrm{s}_{57} \mathbin{+\!\!\!+} \mathrm{s}_{58}$$

$$S_{\mathrm{c}} = \Vert_{j=1}^{8}\ S_{8(\mathrm{j}\text{-}1)+3}$$

$$S_{\mathrm{d}} = \Vert_{j=1}^{8}\ S_{8(\mathrm{j}\text{-}1)+4}$$

$$S_{\mathrm{e}} = \mathrm{s}_5 \mathbin{+\!\!\!+} \mathrm{s}_6 \mathbin{+\!\!\!+} \mathrm{s}_{21} \mathbin{+\!\!\!+} \mathrm{s}_{22} \mathbin{+\!\!\!+} \mathrm{s}_{37} \mathbin{+\!\!\!+} \mathrm{s}_{38} \mathbin{+\!\!\!+} \mathrm{s}_{53} \mathbin{+\!\!\!+} \mathrm{s}_{54}$$

$$S_{\mathrm{f}} = \mathrm{s}_{13} \mathbin{+\!\!\!+} \mathrm{s}_{14} \mathbin{+\!\!\!+} \mathrm{s}_{29} \mathbin{+\!\!\!+} \mathrm{s}_{30} \mathbin{+\!\!\!+} \mathrm{s}_{45} \mathbin{+\!\!\!+} \mathrm{s}_{46} \mathbin{+\!\!\!+} \mathrm{s}_{61} \mathbin{+\!\!\!+} \mathrm{s}_{62}$$

$$S_{\mathrm{g}} = \Vert_{j=1}^{8}\ S_{8(\mathrm{j}\text{-}1)+7}$$

$$S_{\mathrm{h}} = \Vert_{j=1}^{8}\ S_{8(\mathrm{j}\text{-}1)+8}$$

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

$$K\mathrm{a}_{\mathrm{i}}\, f = \mathrm{f}(S_{\mathrm{i}}) \tag{1}$$

where *i = (a, b, c, d, e, f, g, h)*. The next step is to $K\mathrm{a}_{\mathrm{i}} f$ get by passing the 16-bits of S_i to Function 1, as shown in Equation (1). Function 1 applied in the sub-keys generator and algorithm

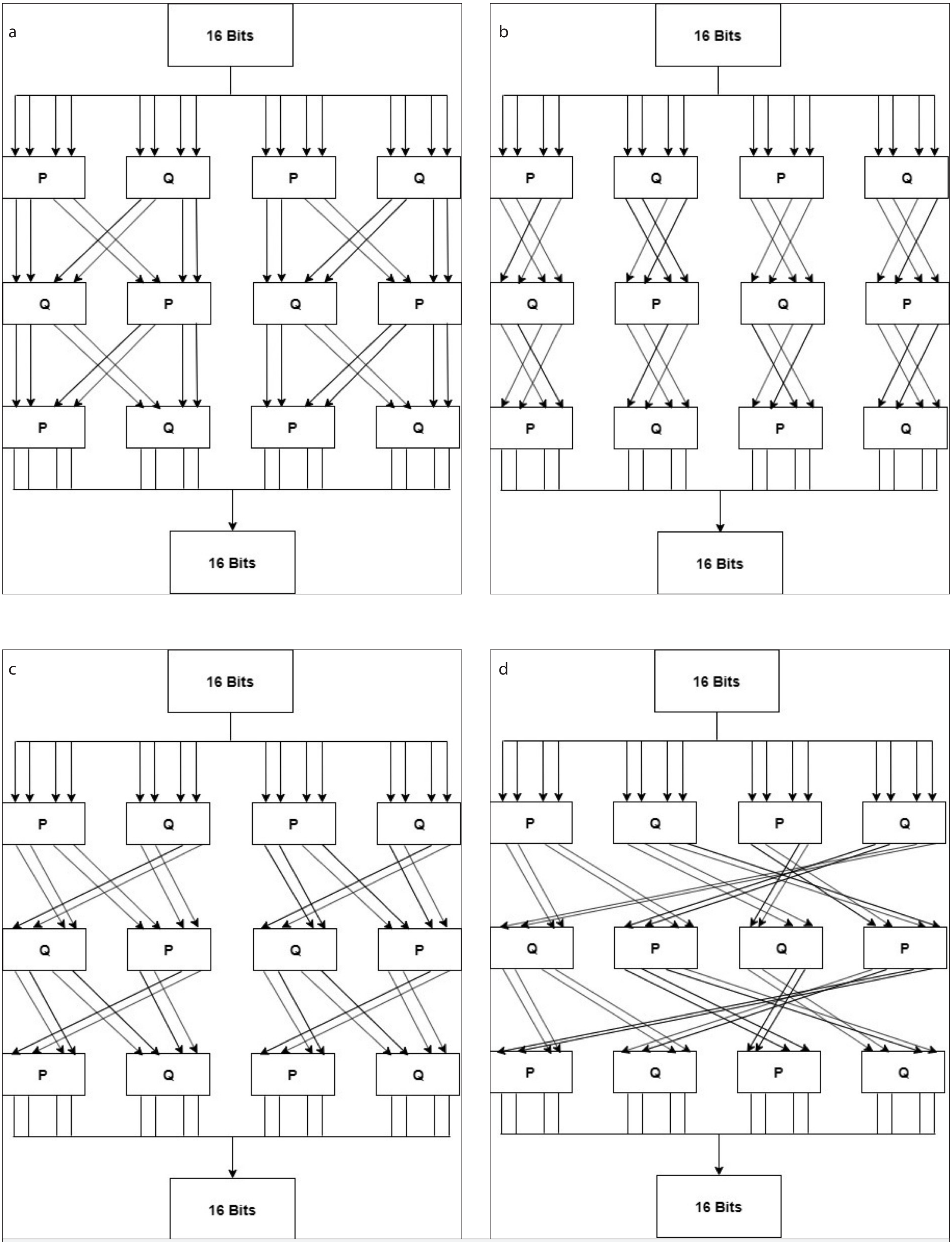

Figure 1. a-d. Function 1(a), Function 2 (b), Function 3 (c), Function 4 (d)

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

structure (Encryption structure), while Functions 2, 3, and 4 applied in Encryption structure only.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\boldsymbol{M}_{\mathbf{a2}} = \left(\overline{\boldsymbol{M}_\mathbf{a} \oplus \overline{\boldsymbol{M}_\mathbf{a}}}\right) \tag{2}$$

$$\boldsymbol{M}_{\mathbf{h2}} = \boldsymbol{M}_\mathbf{h} \oplus \overline{\boldsymbol{M}_\mathbf{h}} \tag{3}$$

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

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

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

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

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

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

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

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

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

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), using 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.

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

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

$K3 = C >> 5$

$K4 = D << 9$

$K5 = E >> 5$

$K6 = F << 9$

$K7 = G >> 5$

$K8 = H << 9$

$$T1 = \overline{K1} \oplus D \tag{4}$$

$$T2 = K5 \oplus H \tag{5}$$

$$K9 = (T1 << 5) \oplus (T2 >> 9 \tag{6}$$

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

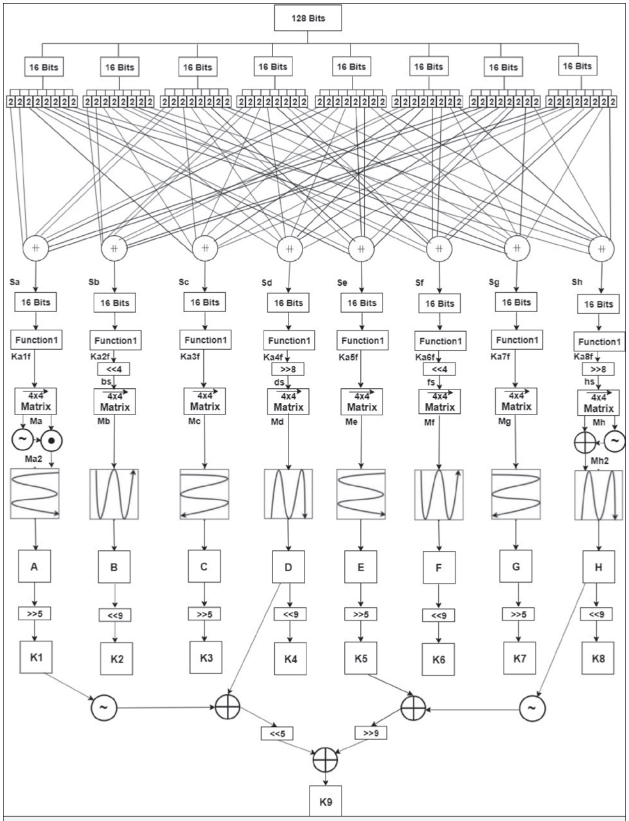

Figure 2. Sub-keys generator

$$Ba_1 = b_{26} + b_{18} + b_{10} + b_2 + b_{32} + b_{24} + b_{16} + b_8$$
$$Bb_1 = b_{30} + b_{22} + b_{14} + b_6 + \overline{b_{30}} + \overline{b_{22}} + \overline{b_{14}} + \overline{b_6}$$
$$Bc_1 = b_{29} + b_{21} + b_{13} + b_5 + \overline{b_{29}} + \overline{b_{21}} + \overline{b_{13}} + \overline{b_5}$$
$$Bd_1 = b_{28} + b_{20} + b_{12} + b_4 + \overline{b_{28}} + \overline{b_{20}} + \overline{b_{12}} + \overline{b_4}$$
$$Be_1 = b_{27} + b_{19} + b_{11} + b_3 + \overline{b_{27}} + \overline{b_{19}} + \overline{b_{11}} + \overline{b_3}$$
$$Bf_1 = b_{25} + b_{17} + b_9 + b_1 + b_{31} + b_{23} + b_{15} + b_7$$

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

$$Ba_2 = (Ba_1 >> 5) \oplus K1 \tag{7}$$

$$Bb_2 = (Bb_1 \oplus K1) \oplus Delta \tag{8}$$

$$Bc_2 = F_1\,(Bb_1 \oplus K1) \oplus Bd_1 \tag{9}$$

$$Bd_2 = (F_1\,(Be_1 \oplus K1)\ Bc_2) \oplus Delta \tag{10}$$

$$Be_2 = (Be_1 \oplus K1) \tag{11}$$

$$Bf_2 = (Bf_1 >> 5) \oplus (\mathrm{K1} \oplus Delta) \tag{12}$$

where $F1$ = *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), showing the following equations:

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

$$B_2 = F_4\,(Bd_{10}) \oplus F_2\,(Bc_{10}) \tag{14}$$

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

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

$$BC = B_2 + B_3 + B_1 + B_4 \tag{17}$$

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

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 do not have the success. The input and output correlation is so large if the linear approximation made for two rounds. Moreover, the round transformation is preserved uniform, which similarly treats every bit 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 that means that variables move in the one direction along, which means an ideal correlation. In contrast, zero implies that there is no relationship between the variables. If negative one

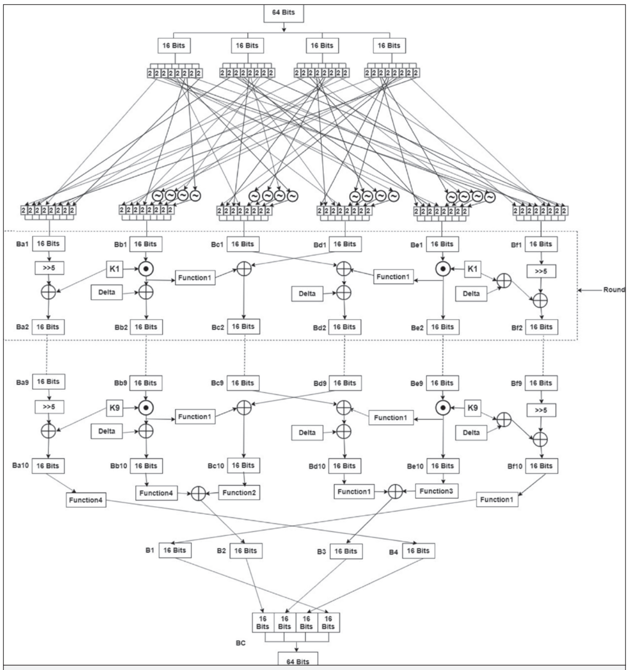

Figure 3. Structure of the Proposed Algorithm (encryption)

refers to an ideal indirect correlation, which means that one variable goes up, the other goes down.

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 is 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 equals zero, this means there is no relationship between the original and encrypted image and refers to the best level of encryption [8, 9].

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

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

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 the 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 is applied to evaluate image encryption, the larger value of entropy refers to a greater security level. It is secure from a brute force attack when an entropy value very close to an absolute 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 uniformly distributed and entirely different from the original image to prevent extracting 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 uniformly distributed. [11].

Key Space Analysis

Key space means the number of bits used to encrypt images, and for good encryption, that key space should be as large as possible to repulse the brute force attack. Key space size means the total number of different keys of the same number of bits used for encryption [12]. For high security, the key space should be greater than 2^{100}. In the proposed algorithm key size is 128 bit, and according to equation (62), the key space is 2^{128} this means key space size is suitable to repulse the 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 designed for fast and non-linear diffusion to the difference of the main key and sub-keys.

Interpolation Attacks

These attacks depend 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

Important things taken into account in lightweight cryptography algorithms are the speed of processing time, the amount of memory and energy consumption of the algorithm during data encryption operations, as the proposed algorithm designed to suit the IoT environments. The memory and energy consumption are directly proportional but inversely proportional to time and speed.

Table 3. Correlation and entropy analysis

		Correlation		Entropy	
Image	**Size**	**Original image**	**Encrypted image**	**Original image**	**Encrypted image**
Lena	256×256	0.9744	0.0001	7.4509	7.9976
Panda	256×256	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

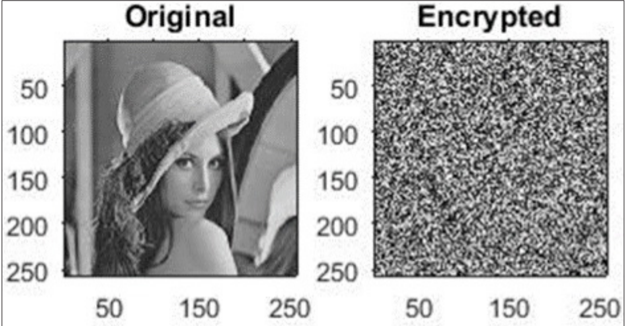

Figure 4. Encryption of Lena image

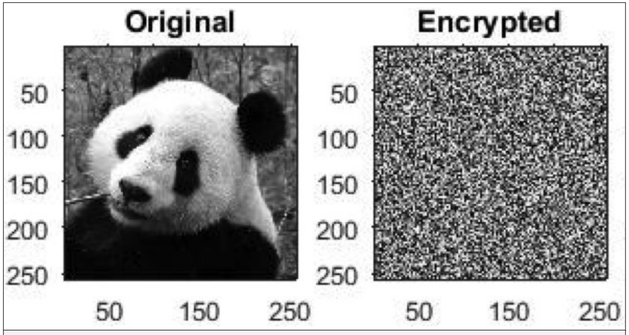

Figure 5. Encryption of Panda image

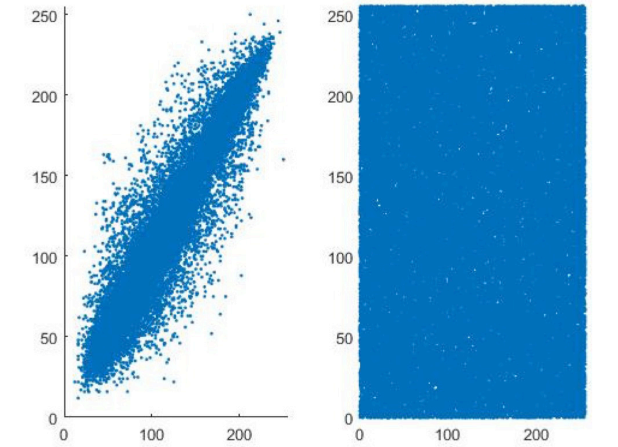

Figure 6. Correlation of Lena image

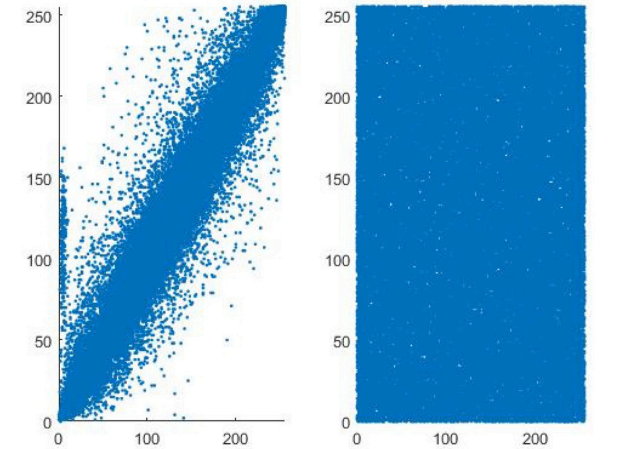

Figure 7. Correlation of Panda image

Results

Through some analysis for the proposed algorithm, where the simulations performed on a desktop computer with Intel(R) Core(TM) i7 CPU L 620 @2.00 GHz., 4GB RAM, and Windows 10 Professional operating system, and by using MATLAB R2015a.

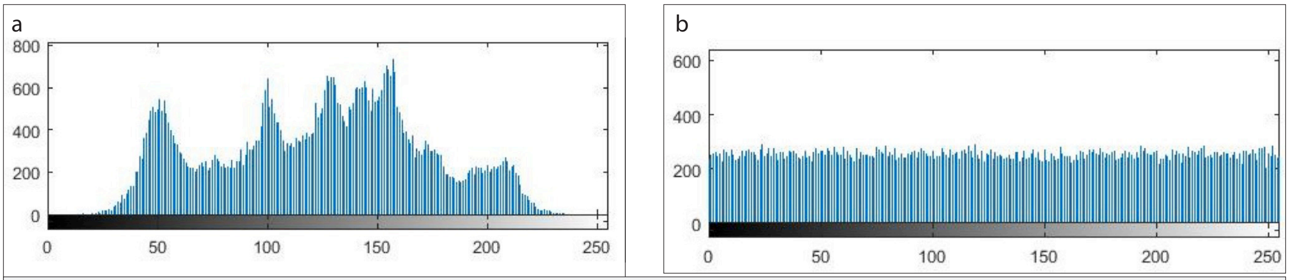

Figure 8. a, b. Histogram Graph of Original Lena (a), Histogram Graph of Encrypted Lena (b)

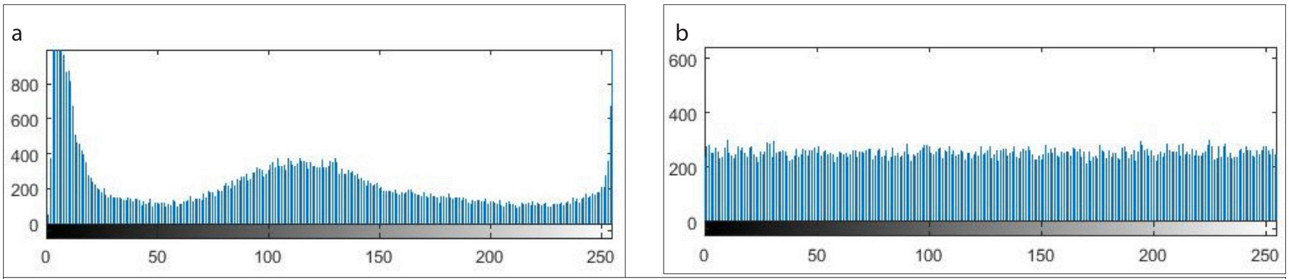

Figure 9. a, b. Histogram Graph of Original Panda (a), Histogram Graph of Encrypted Panda (b)

In addition to some other algorithms on FELICS. We get the results in Table 3, as shown in Figures 4 and 5.

The results show the proposed algorithm high-efficiency compared with another algorithm, especially with algorithms of 128 bit. In Table 4, the block and key sizes are in bits, while the code size and RAM are in bytes.

As shown in Figures 6 and 7, the relationship between the encoded and original images is 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, which 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 IoT and smart cities and take into account its specification. The results in this article show that the proposed algorithm is directly proportional to high security and low memory and energy consumption. It is convenient for the environment of the IoT.

The security can be improved by increasing key and block sizes and rounds number in addition to changing the function and operation in the algorithm. However, maybe that will make it not convenient for the IoT 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. [Crossref]
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. [Crossref]
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. [Crossref]
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 [Crossref]
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. [Crossref]
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.