2023 International Conference on Modeling, Simulation & Intelligent Computing (MoSICom)
December, 7-9, 2023, BITS Pilani Dubai Campus, Dubai, UAE

Performance Analysis of Lightweight Ciphers for securing images

Keerthan Nekkanti
Department of Computer Science and Engineering
Manipal Institute of Technology,
Manipal Academy of Higher Education,
Manipal, Karnataka, India
keerthannekkanti@gmail.com

Lade Sai Sumanth
Department of Computer Science and Engineering
Manipal Institute of Technology,
Manipal Academy of Higher Education,
Manipal, Karnataka, India
ladesaisumanth2000@gmail.com

Renuka A
Department of Computer Science and Engineering
Manipal Institute of Technology,
Manipal Academy of Higher Education,
Manipal, Karnataka, India
renuka.prabhu@manipal.edu

Musica Supriya
Department of Computer Science and Engineering
Manipal Institute of Technology,
Manipal Academy of Higher Education,
Manipal, Karnataka, India
musica.supriya@manipal.edu

Arun Kumar[1,2]
[1]Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India
[2]Department of Computer and Software Engineering, YooBee College of Creative Innovation, Auckland,
NewZealand
Arun.kumar@yoobeecolleges.com

2023 International Conference on Modeling, Simulation &amp; Intelligent Computing (MoSICom) | 979-8-3503-9341-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/MoSICom59118.2023.10458747

***Abstract*—In the digital age, the security of sensitive information and data has become a paramount concern. The modern conventional encryption algorithms are very efficient, but they are very complex and need a lot of resources. This has led to the development and implementation of lightweight ciphers designed specifically for image encryption. Four algorithms namely Present, Lightweight encryption Algorithm (LEA), Twine and Secure IoT(SIT) are implemented to encrypt and decrypt images and their performance is analyzed. The correlation values of the images before encryption is found to be around 0.9 showing strong correlation whereas after encryption it is found to be very less in the range oof 0.0005 which clearly shows the randomization of the pixels and hence good security. The execution times was less than one second for LEA algorithm and hence efficient for multimedia applications.**

Keywords*— light weight ciphers, image encryption, Performance Analysis

I. Introduction

Cryptography has been evolving and is becoming more efficient in both security and performance and at the same time minimizing the consumption of resources. However, with the increase in usage of mobile and handheld devices the IoT industry needs more efficient algorithms that provides security. The memory is a major parameter that identifies the possibility of execution in a device. Power is another important parameter to be considered in energy constrained devices, whereas power dissipation measurement is found to be important in wireless devices. High speed critical component for devices such as cameras or sensors with large amount of data transmissions,. At the same time, smooth functioning is necessary for the time sensitive AI enabled devices such as self-driving systems. This has led to the invention of new lightweight ciphers Although lightweight cryptography entered in early 2000's, the industry still has need for more efficient algorithms and determining the specific applications of the invented algorithms. Lightweight ciphers are a class of encryption algorithms that prioritize efficiency and low computational overhead while maintaining a high level of security. These ciphers are suitable for resource-constrained devices and real-time applications, making them an excellent choice for image encryption, where large volumes of data need to be processed quickly and securely. Image encryption is the process of transforming the pixel values of an image to render it unreadable without the proper decryption key. The aim is not only to prevent unauthorized access but also to protect the integrity of the image during transmission or storage. Lightweight ciphers offer a practical and efficient solution for this task, as they balance the trade-off between security and performance. Image encryption involves unique security requirements, such as preserving the visual quality of the image while ensuring its confidentiality and integrity. Lightweight ciphers provide efficient encryption for resource-constrained devices and applications, making them suitable for image encryption due to the data-intensive nature of images. Image encryption using lightweight ciphers finds applications in various fields, including medical imaging, secure communication, and data storage. The increase in usage of IoT devices in limited resources areas such as Radio Frequency Identification(RFID) tags, home voice controllers, agriculture etc., has increased the need of identifying efficient algorithms for security. The lightweight algorithms named, Present, LEA, SIT and Twine are implemented to encrypt the images and are analysed based on few metrics. These ciphers are symmetric key lightweight cryptographic algorithms which take a small space and only use very limited resources. They are also time efficient algorithms. The metrics used for evaluating these algorithms are entropy of images, statistical or correlation analysis, execution times of both encryption and decryption process, Unified Average Changing Intensity (UACI) and Number of Pixel Change Rate (NPCR).

II. Literature Review

In [1] the authors have evaluated and analysed the lightweight cipher based on few parameters. A cipher comparison is done based on an overview of lightweight cryptographic algorithms which would give an idea about this research field. The authors in [2] provide a performance comparison of most common symmetric ciphers like Data Encryption Standard and Advanced Encryption Standard along with, Blowfish, Rivest Cipher 4, High Security and

979-8-3503-9341-5/23/$31.00 ©2023 IEEE

Light weight (HIGH) and Secure IoT(SIT), that are used to provide cloud security services and have proposed NLCA-128. They studied memory usage for encryption, and decryption, key-space analysis, avalanche analysis, and differential cryptanalysis and claim that NLCA-128 performs better. A comparison of light weight ciphers is brought in [3] by carrying out hardware and software implementations. Authors in [4] propose a data compression algorithm that is lightweight for image encryption which is more suitable with IoT devices that have a low data rate. This makes use of combination of compression and encryption technique to encrypt the image data in a single round, thus reducing the computational complexity and also reducing its data volume. The compression technique is applied in blocks and is scan-based and the encryption method is a selective pixel encryption approach. In [5] authors have done a thorough survey on the different image encryption techniques that uses chaotic maps in spatiotemporal domains, spatial domain, and transform domain and then have classified them into relevant categories. They have concluded that image encryption domain needs to be explored more and needs in depth understanding of the issues concerning security, efficiency of computing and parameter adjustments. They conclude after analyzing the various methods, that the majority of the works do not use proper validation techniques to validate the performance of the algorithms and therefore a standard benchmark should be devised to measure the performance.

In this paper four lightweight ciphers which are constructed in different ways have been chosen for analysis. Ref [6] proposed a block cipher which used the Substitution Permutation Network(SPN) and named it as Present which is suitable for hardware implementations. Present uses a block size of 64 bits, a key length which can be 80 or 128 bits and make use of 31 rounds. Exclusive OR operation is performed with round key for each round. Four-bit Substitution-Box(S-Box) is applied for each round and performed in parallel that increases confusion and diffusion. Bit permutation is used to introduce diffusion. Attacks like weak key attack, and linear attack can be applied to Present due to its weak diffusion property. A new lightweight block cipher Lightweight encryption Algorithm(LEA) that uses 32-bit words having simple operations such as addition, rotation and exclusive-OR called as ARX structure and does not use S-box structure was proposed in [7]. LEA uses a standard block size made up of 128 bits with different key sizes which can be of 128 or 192 or 256-bits having different round sizes ranging as 24 or 28 or 32 rounds respectively. LEA algorithm makes use of two XOR operations, a modulo 2^{32} addition operation and bit-wise rotation operation.. Modulo 2^{32} addition is a non-linear function with two inputs each of 32-bits and one output. and Bitwise rotations and word wise swapping introduces diffusion. Decryption is similar to encryption procedure. An array of 32-bit words is used to store the 192-bit round keys sequence that is generated using the of LEA and key schedule algorithm. The authors in [8] present a cipher Twine which is lightweight cipher and uses a 64-bit block and a a key size of 80/ 128 bits. The necessary round keys are computed before hand with a single loop of two rounds by eliminating the shuffling between blocks in the rounds. The compact implementation of Twine is possible due to the usage of the Feistel structure. A lightweight 64 bit symmetric key block cipher SIT (Secure IoT), is proposed in [9], which is with uses a key size of 64-bits comprising of five rounds. SIT uses an hybrid approach that uses a combination of Feistel and SPN structure. The proposed approach uses logical operations in addition to the swapping and substitution operations. Energy efficiency is improved because only five rounds of encryption are used, with each round using a different key. A 64-bit key is fed to the key expansion block that uses a F-function for key generation.

III. Methodology

In this section, an overview of the various algorithms and their implementation details are discussed.

A. Overview of the Algorithms

A brief description of the algorithms is presented in this section.

a) Present: Present is a symmetric key block cipher which makes use of 64 bits as the size of each of the blocks. It can be implemented either by using 80bit key or with 128bit key size. Present uses bit permutations which is more suitable for implementing in hardware not very software friendly. Present is a hardware friendly and a power efficient algorithm. In resource-constrained environments, Present has proven to be efficient than AES. Present is ultra-low lightweight cipher suited for domas like agriculture where the scarcity of power is a major concern. This algorithm is a 64bit size block size consisting of 32 rounds. Although it can be done by either using 80bit key or 128bit keys size, we are using 128bit key to increase the key space of the algorithm, which will increase the security of the algorithm. Each round has three same consecutive steps, and they are adding round key, substitution layer or the S-layer, and permutation Layer or the P-layer. However, the 32nd round has only addRoundKey step. The decryption has the same methods as encryption and uses the same key but only executed in inverted order. The Substitution -Box used in the Sustitution layer and the Permutation-Box in Permutation-layer is inverted by inverting the mapping. The process starts with the last round key i.e., 32nd and ends with first round. The key-scheduling process produces thirty-two round keys each of length 64 by taking one 128bits as inputs.

b) Lightweight encryption Algorithm(LEA) : The LEA Algorithm was developed to provide efficient protection for big data and cloud computing devices with high-speed environment and also for IoT devices and mobile devices with lightweight environment. LEA can be done using three different length keys i.e., 128bit, 192bit, 256bit. LEA is almost twice faster than AES in encryption. Many software environments make use of the LEA cipher. The encryption process produces 128-bit ciphers text by taking 128bit plain as input and output. It uses modular addition, bitwise shift, bitwise exclusive or and concatenation. This has twenty-four rounds, each having the same set of four operations. Decryption is the exact opposite of encryption. While the encryption used modular addition, decryption uses modular subtraction to counter it. The direction of bitwise shift at every point is reversed in decryption. The key length can be 128bit, 192bit, and 256bit. 128 bit key length is used in this implementation. The key scheduling takes 128 bits as input and generates 24 round keys each of length 192 bit. The input gets sliced into 4 equal parts of length 32 bits and then each part will be processed separately. The formed result will then be concatenated making the length 192bit.

c) Twine: Twine cipher uses a block size of 64 bits and makes use of key length of 80 bits. Round-keys are generated from the cipher key, using the key schedule. Four bit S-boxes in Twine's round function introduce non-linearity and diffusion is introduced through permutation by permuting the the 16 blocks. The decryption of Twine is the same as encryption with inverse block shuffle being used. The decryption is similar to encryption because it follows the Feistel Structure but uses non-linear functions sparsely.

d) Secure IoT(SIT): SIT operates on plaintext comprising of 64-bits and a key which is again 64 bits. The encryption process makes use of five rounds of encryption. Each round consists of various modules to introduce diffusion and confusion. Energy consumption increases if the number of rounds are increased. Therefore, many ciphers are designed to have more number of rounds to enhance security. Each round manipulates 64 bits of data that improves the security. The Feistel structure creates confusion and diffusion of data that makes it immune to attacks. The encryption key is entered into the key generation block which creates five different keys that are unique and not repeating.

B. Implementation

The implementation of the above mentioned algorithms was done Open CV library of Python. Python Imaging Library(PIL) was used convert the image to greyscale and for converting the image into bytes and vice versa. To plot the histograms and for few others, matplot library was used. NumPy, pandas modules were also used whenever needed in evaluating the algorithms and storing the results. Image encryption algorithm consists of the following steps parts

- Image Pre-processing
- Image encryption
- Image decryption

In Image Pre-processing, the original-colored image is first converted into grayscale and then the image is resized to the desired value. A standard size of 256 * 256 was used. To test the effectiveness of the various lightweight encryption algorithms in encrypting the image, images of different sizes were used. The hex function is used to convert the image data to hexadecimal value. The hexadecimal string so obtained is fed as plaintext to the above algorithms Present, LEA, Twine, and SIT. This operation gives the encrypted image. Again the encrypted image is fed back to the decryption algorithm to get hexadecimal string which is then converted back into binary string. This binary data is used to create a new image and resized for displaying.

IV. Result Analysis

In this section the entropy, correlation analysis, NCPR and UACI is computed for the various images and compared and analyzed. The encrypted image and the decrypted images along with the original image, for the four algorithms are shown in Figure 1. The decrypted image is compared to the original image and all the images are matched for both the algorithms with no loss in data.

A. Metrics Used for the Evaluation of the Results

The various metrics such as image entropy, correlation analysis, NPCR and UACI have been used, the definition of these metrics and the result obtained is given below.

1) Image Entropy : is used to indicate the randomness of the pixels in an image. Usually, the higher value of image entropy means that the image is clearer. This is used to understand the strength of the encryption algorithm and hence can be used as a quality check parameter. The entropy is denoted by H and for an image consisting of pixels denoted by P is given by the formula

$$H(I) = -\sum_{i=1}^{256} P(I_i)log_2 P(I_i) \qquad (1)$$

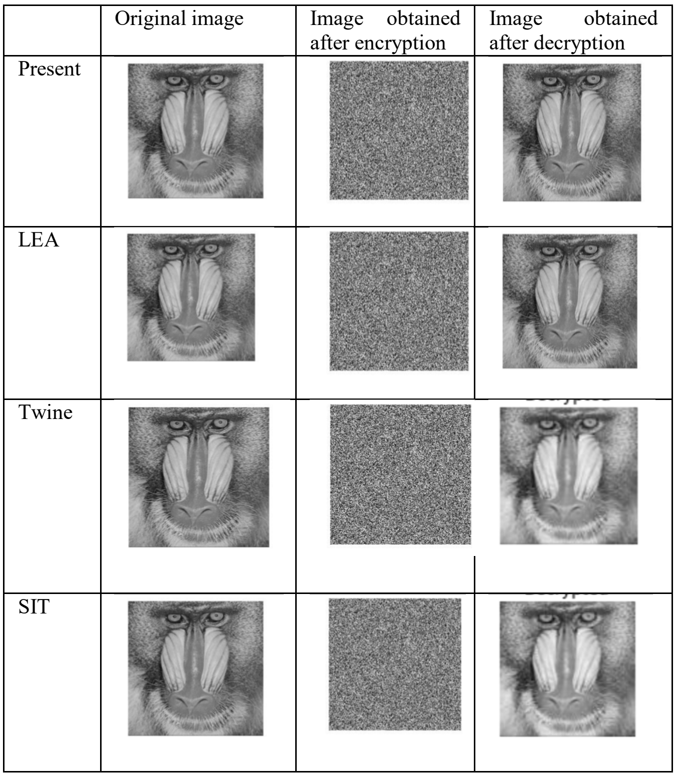
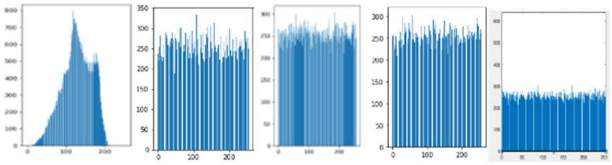

Original image
Image obtained after encryption
Image obtained after decryption
Present
LEA
Twine
SIT

Fig. 1. Original, encrypted and decrypted images for various algorithms

TABLE I. Entropy values of various algorithms

Names of images	Entropy of original image	Entropy of encrypted image			
		Present	LEA	Twine	SIT
Baboon	7.232	7.997	7.997	7.996	7.997
Onion	7.341	7.993	7.993	7.993	7.997
Lena	7.461	7.996	7.996	7.996	7.997
Football	6.686	7.997	7.998	7.996	7.997
Panda	7.509	7.990	7.997	7.981	7.997

The entropy values for the various algorithms are shown in Table I. It is observed from the results that all the encrypted images have entropy almost equal to 8 which is the max value of an entropy indicating maximum randomness. The histogram can also be used to check whether the encrypting algorithm is efficient or not since a flat histogram indicates a random distribution. The histograms of the encrypted and decrypted images along with the original are given in Figure 2. It is observed that the histogram of the encrypted images are almost flattened which indicate that the security provided by algorithms is good.



(a) (b) (c) (d) (e)

Fig. 2. Entropy for various algorithms (a) Original image (b) Present (c) LEA (d) Twine (e) SIT

2) Correlation Analysis: is used to find whether one variable is dependent on the other or in other words to obtain the relationship between two variables. It is used to evaluate the performance of the cryptographic algorithms. In image cryptography, adjacent pixels are used to find the dependency between them.. The formula for the correlation between two variables x and y is given by the formula shown in (2), which ranges from -1 to 1. The higher the absolute value of correlation the stronger the correlation.

$$Y_{x,y} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \qquad (2)$$

where the variance D(x) and the covariance cov(x,y) can be calculated using (3) and (4) respectively.

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)^2 \qquad (3)$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \qquad (4)$$

The mean E(x) can be evaluated using the formula given in (5)

$$E(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i) \qquad (5)$$

TABLE II. CORRELATION VALUES OF ORIGINAL AND ENCRYPTED IMAGES FOR VARIOUS CIPHERS

Name of image	**Correlation of original image**	**Correlation of encrypted image**			
		Present	***LEA***	***Twine***	***SIT***
Baboon	0.867725	-0.00052	-0.00271	0.0032	0.0041
Onion	0.983094	0.006974	-0.00779	0.0054	0.0020
Lena	0.920925	0.000173	0.002057	0.0051	0.0041
Football	0.927558	0.000308	0.002816	0.0001	0.0023
Panda	0.983168	0.007558	0.002049	0.0118	0.0030

The correlation values for the original image and the encrypted images are shown in Table II. From the results given in Table II, it is observed that the original images have higher correlation and encrypted images have lower correlation. So, encrypted images pixels are almost not related to their adjacent ones where decrypted are. The same can also be inferred from the scatter plots shown in Fig. 3, which suggests that there is practically no relation between the pixels in the encrypted image.

3) Encryption Time: Execution time for encryption is considered as a factor of evaluation for almost every algorithm these days. In addition, when dealing with low resource applications like IoT devices, execution time plays a crucial part. The lesser the execution time the better the algorithm. Table III shows the execution time of all the algorithms for different images.

4) Number of Pixel Change Rate (NPCR): is used to measure the amount by which the encrypted image is deviated from the original image and the formula is given by (6).

$$NPCR = \left(\frac{\sum_{i=0}^{H}\sum_{j=0}^{W} C(i,j)}{(H*W)}\right) * 100 \qquad (6)$$

where C(i) is given by the formula shown in (7)

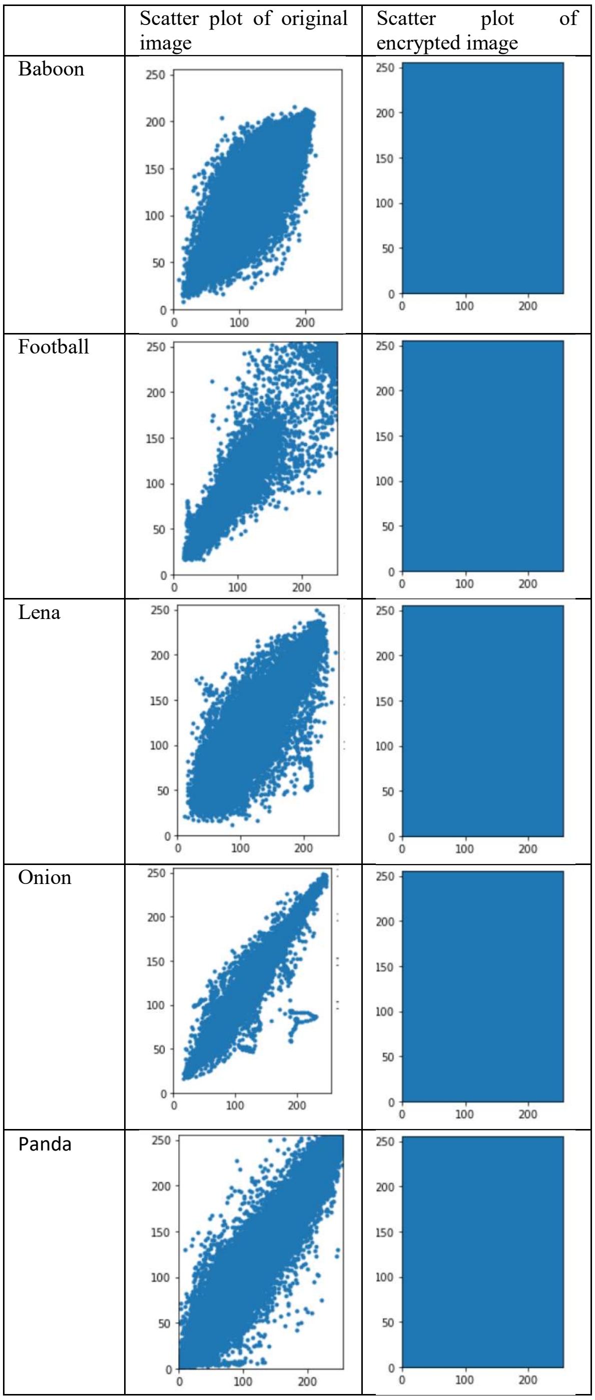


Fig. 3. Scatter plots for original image and the encrypted image using LEA encryption for various images.

$$C(i,J) = \begin{cases} 1, & image(i,j) not\ equal\ to\ enc(i,j) \\ 0, & image(i,j)\ \ equal\ to\ enc(i,j) \end{cases} \quad (7)$$

where the W and H are the heights and width of the image respectively. Here image is the original image and enc is the encrypted image. A good encryption algorithm should have a value greater than or equal to 99. The Table IV shows the NPCR values of different images for the various algorithms.

TABLE III. EXECUTION TIMES IN SECONDS FOR VARIOUS ALGORITHMS

Image	Present	LEA	Twine	SIT
Baboon	4.762415	0.411396	6.48	9.84
Onion	1.931566	0.157263	6.43	9.71
Lena	3.474593	0.298245	6.33	9.80
Football	7.512887	0.549255	6.40	9.80
Panda	16.039922	1.428447	6.43	10.21

TABLE IV. NPCR VALUES FOR VARIOUS ALGORITHMS

Image	Present	LEA	Twine	SIT
Baboon	99.6109	99.6154	99.6216	99.6262
Onion	99.6633	99.5847	99.6216	99.5941
Lena	99.5847	99.6012	99.5605	99.5972
Football	99.6044	99.6154	99.6323	99.6109
Panda	99.5941	99.5486	99.6307	99.6246

5) Unified Average Changing Intensity (UACI): UACI is the measure of the degree of the change of encrypted image when the original image is changed by one bit. This is helpful in showcasing the strength of algorithms towards the differential attacks. It can be calculated between any two images of the same size. It indicates the change in intensity between two images. UACI is given by the formula in (8)

$$UACI = \left(\frac{1}{H*W} \sum_{i=0}^{H} \sum_{j=0}^{W} \frac{|C(i,j)-C\prime(i,j)|}{255} \right) * 100 \quad (8)$$

where C and C' are encrypted images before and after changing 1-bit. W and H are width and height of the images respectively. A good algorithm should produce a value close to 33%. Table V shows the results of UACI for various algorithms on different images.

TABLE V. UACI VALUES FOR VARIOUS ALGORITHMS

Name of image	Present	LEA	Twine	SIT
Baboon	49.9206	50.0651	49.8761	26.0561
Onion	50.2018	49.8401	49.9986	14.8720
Lena	49.8429	49.9955	49.9455	13.3172
Football	50.1904	50.0135	49.9865	22.6245
Panda	50.3362	50.2738	49.9865	20.3556

V. CONCLUSION

After conducting all the evaluations on the algorithms, the algorithms perform equally good and are secure. In case of encryption time and UACI, LEA algorithm performed well. Almost all algorithms performed equally good in image entropy proving to be efficient. At the end, LEA algorithm efficiency surpassed and would better suit for resource constrained IOT devices as far as software implementation is considered. Future work would be implementing these algorithms in hardware.

REFERENCES

[1] Mathews, R., & Jose, D. V. (2020). Analysis of lightweight cryptographic algorithms for internet of things. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3734786

[2] Thabit, F., Alhomdy, S., & Jagtap, S. (2021). Security analysis and performance evaluation of a new lightweight cryptographic algorithm for cloud computing. Global Transitions Proceedings, 2(1), 100–110. https://doi.org/10.1016/j.gltp.2021.01.014

[3] A review of lightweight block ciphers, George Hatzivasilis1 · Konstantinos Fysarakis1 · Ioannis Papaefstathiou1Charalampos Manifavas2

[4] Hedayati, R., Mostafavi, S. A Lightweight Image Encryption Algorithm for Secure Communications in Multimedia Internet of Things. Wireless Pers Commun 123, 1121–1143 (2022).

[5] Zia, U., McCartney, M., Scotney, B. et al. Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains. Int. J. Inf. Secur. 21, 917–935 (2022).

[6] Bogdanov, Andrey & Knudsen, Lars & Leander, Gregor & Paar, Christof & Poschmann, Axel & Robshaw, Matthew & Seurin, Yannick & Vikkelsoe, C(2007). PRESENT: an ultra-lightweight block cipher. Lect Note. Comput. Sci.. 4727. 450-466.

[7] Hong, D., Lee, J.K., Kim, D.C., Kwon, D., Ryu, K.H. and Lee, D.G., 2013, August. LEA: A 128-bit block cipher for fast encryption on common processors. In International Workshop on Information Security Applications (pp. 3-27). Springer

[8] Suzaki, T., Minematsu, K., Morioka, S. and Kobayashi, E., 2011, November. Twine: A lightweight, versatile block cipher. In ECRYPT Workshop on Lightweight Cryptography (Vol. 2011)

[9] Usman, M., Ahmed, I., Aslam, M.I., Khan, S. and Shah, U.A., 2017. SIT: A Lightweight Encryption Algorithm for Secure Internet of Things International Journal of Advanced Computer Science and Applications, 2017.