Journal of Theoretical and Applied Information Technology
15th February 2023. Vol.101. No 3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A LIGHTWEIGHT SPATIAL DOMAIN IMAGE ENCRYPTION ALGORITHMS: A REVIEW PAPER

ISSA JACAMAN[1] , MOUSA FARAJALLAH [2]
[1,2] College of Information Technology and Computer Engineering (CITCE)
Palestine Polytechnic University, Palestine
E-mail: [1]176006@ppu.edu.ps, [2] mousa_math@ppu.edu

ABSTRACT

In our modern era, multimedia images have increased exponentially. Such data is being stored, transmitted through the public internet, and being used or produced by limited resource devices such as smartphones, Internet of Things devices, and healthcare devices. The conventional algorithms fail to protect such data while being processed by limited resource devices as it requires the most computational cost and increases communication overhead. In this short review, several lightweight encryption algorithms that overcome the conventional algorithm are considered. Moreover, we highlight some of the existing algorithms that are involved in encrypting images in the spatial domain for resource-limited devices and provided a comparison among them for their performance and robustness. The presented algorithms' techniques were categorized for better understanding of the five categories in-compression approach, coefficient correlation, edge detection, most significant bits, and byte stream with a specific encryption ratio.

Keywords: *Chaotic Encryption, Selective Encryption, Cellular Automata, Lightweight Encryption, Stenography.*

1. INTRODUCTION

Users share images and videos very frequently using their own resource-limited devices. This has brought the challenge of protecting private data while storing and transmitting data through the public internet with limited resource devices. Selective encryption [1] is considered the best solution for lightweight encryption [2] in real-time resource-limited applications. Many researchers have adopted selective encryption to encrypt the important data of an image. Various methods were presented to enhance selective encryption, such as using the least significant bits, and encryption within the MQ coding system (a context-based adaptive arithmetic coder). Regarding the previous review works, in the image encryption area, a few review works were done such as [3] [4], they describe and evaluate (cryptoanalysis) the encryption algorithms with respect to the image being in spatial, transform, spatiotemporal, optical, or compressive sensing domain. They consider encryption algorithms in a general overview being in different domains. Meanwhile, in this brief review, we focus on the algorithms in the spatial domain only and being lightweight. Furthermore, we categorized them in accordance with their encryption algorithm techniques and compared these techniques in the evaluation section, and as future work to investigate more into these techniques to be able to adopt a suggested hybrid encryption algorithm.

There are two image domains, spatial and frequency domains. The presented approaches in this literature review are both selective and full cellular automata image encryption in the spatial domain. The difference is that in the frequency domain, encryption is applied to selected frequency coefficients of image data based on predefined criteria. Meanwhile, in the spatial domain, encryption is applied on the bit-level or pixel-level using confusion and diffusion to change their actual values [3].

1.1. Lightweight Encryption Algorithms Categories

Encryption algorithms in [4] [5] [6] [7] [8] [9] can be categorized based on the selectivity method used in encrypting digital images. Figure 1 illustrates these categories; [7] used the In-compression approach, [4] calculated the coefficient correlation of pixel blocks, [8] considered encrypting only the most significant bits of an Image, [9] [5] encrypted only the edges using Edge detection, and [7] encrypted an image through its byte-stream with encryption ratio.

Journal of Theoretical and Applied Information Technology
15th February 2023. Vol.101. No 3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

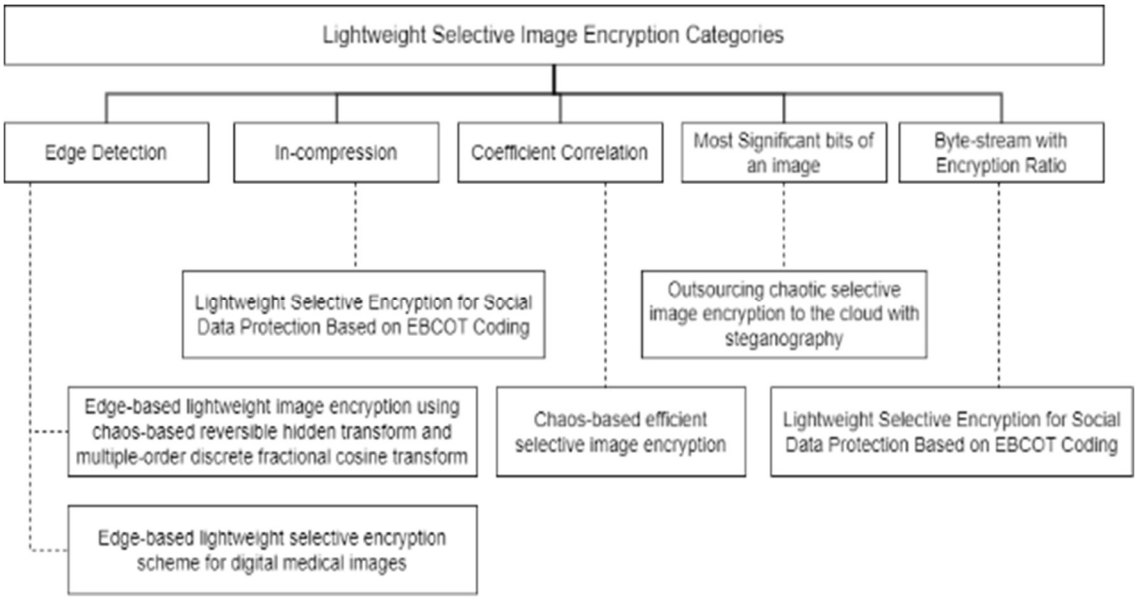

Figure 1 Lightweight Spatial Domain Image Encryption Categories

1.1.1. In-Compression approach

This approach requires a modification of the encoder and decoder system during an image byte stream (during compression) [10]. It results in an encrypted image. The compression process reduces and eliminates the duplicated blocks making the encryption more secure. In other words, constant and similar blocks are reduced and then encrypted.

1.1.2. Coefficient correlation

Calculating the correlation of an image block helps define whether this block of an image contains important data. This important data is critical for concealing the whole image if all these blocks are encrypted. Depending on a threshold value, the calculated Coefficient Correlation of a block is either encrypted or left as it is. Encrypting these blocks conceals the important data of an image, making it visually indistinguishable.

1.1.3. Edge detection

Edge detection reveals the important visual information corresponding to discontinuities in the physical properties of an image [11]. In other words, sharp changes in colour or intensity of an image block help identify the important data of an image that makes an image distinguishable. Encrypting these areas tends to hide such information to make this important data concealed visually. This technique reduced the computational cost and time instead of encrypting all the image blocks.

1.1.4. Most Significant bits of an Image

The most significant bits (MSB) of the image pixels can be selected as the important data of an image. Since the MSB holds the most important data of a pixel, for example, the 8^{th} most significant bit of a pixel holds half the information ($\frac{2^7}{256}\times100\%$) and the 7^{th} bit holds 25% of the information ($\frac{2^{67}}{256}\times100\%$) and so on. Encrypting this part will lose the pixel colour (visually).

1.1.5. Byte-stream with Encryption Ratio

Another approach is to selectively encrypt an image by ratio. A byte-stream of an image can be encrypted [12] within a specified ratio. Authors need to experimentally test their algorithm in order to reach the ultimate ratio percentage for encrypting the bit-steam to guarantee the robustness of image encryption.

2. RELATED WORKS

In this section, some of the recent and interesting research algorithms that address lightweight selective image encryption are described. As an example of chaotic encryption algorithms [4] [5] [6] [8] [9] are discussed, whereas in [7] Pseudo Random Number Generator (PRNG) is used during its presented encryption process.

2.1 Chaos Based Efficient Selective Image Encryption

In [4], the authors presented a lightweight, secure encryption scheme for digital images. The presented scheme starts by dividing plaintext images into a number of blocks, and correlation coefficient values are calculated for each block. Then, the blocks with the maximum values of correlation

coefficients (C.C) are encrypted by XORing pixel-wise with random numbers generated from a skew tent map (based on a predefined threshold value). Finally, using two random sequences generated from TD-ERCS chaotic map, the whole image is permuted. For confusion, the final encrypted image is shuffled row-wise and column-wise, respectively.

Last decade, the use of social networks has significantly increased the demand for sharing multimedia data. Consequently, many algorithms have been developed to increase its security and difficulty against eavesdropping attacks. However, this has increased the computational cost and communication overhead and does not yet provide security against new zero-day attacks. This has motivated researchers to propose algorithms against these issues for better security and performance (cost).

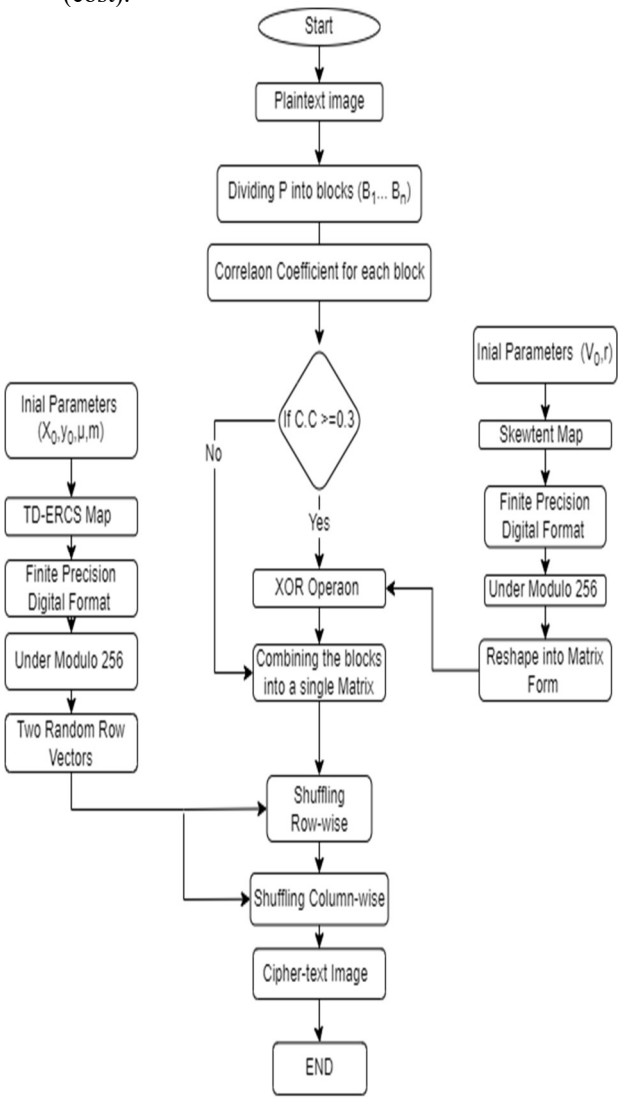

Figure 2 Flow Chart Of The Design Scheme

The presented scheme can be summarized with the following steps (illustrated in figure 2 [4]):

1- First, divide the plain text image into blocks $B=[B_1,B_2,\ldots,B_{256}]$ with total blocks of 256.
2- Then define a threshold value and calculate the correlation coefficient of each block.
3- Each block of the plain-text image having a correlation coefficient (C.C) greater than the predefined threshold value (T=0.3), is bit-wised XOR-ed with a random number matrix Ψ. Matrix Ψ is obtained by arranging vector ζ in matrix Ψ, where ζ=Module(Y,256) by which is defined by $Y=V\times10^{14}$ and V is a random vector generated by iterating V_{n+1} 65,536 times such that t=0.1000 and V_0=0.5000 are the initial conditions for Skew Tent Map:

$$V_{n+1}=f(V_n,r) \quad (1)$$

$$V_{n+1}=\begin{cases}\frac{V_n}{r} & \text{if } V_n\in[0,r]\\ \frac{(1-V_n)}{(1-r)} & \text{if } V_n\in(r,1]\end{cases} \quad (2)$$

4- Diffused blocks $Diff_{Block_n}$ are now generated $Diff_{Block_n}=bitxor(B_n,\Psi)$ where n is the block (B_n) number undergoing the XOR operation.
5- All blocks are combined to get the diffused image $Diff_{image}$.
6- The row-wise permutation and then column-wise permutation is done on the diffused image using two random numbers generated X" and Y". These two numbers are generated using the mathematical representation of the Tangent Delay Ellipse Reflecting Cavity Map System (TD-ERCS)

$$X=X^1,X^2,\ldots,X^{256} \quad (3)$$
$$Y=Y^1,Y^2,\ldots,Y^{256} \quad (4)$$

7- Finally, the ciphertext image is obtained.
8- To decrypt the cipher image, all previous steps are applied reversely.

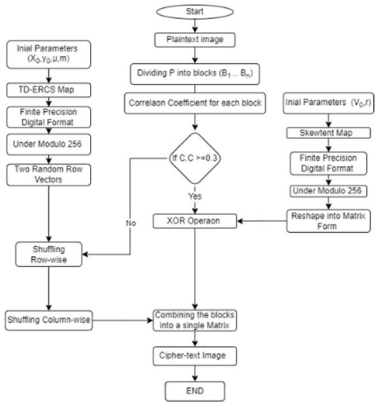

Figure 3 Flow Chart Of The Suggested Scheme

2.2 Edge-based Lightweight Image Encryption Using Chaos-based Reversible Hidden Transform and Multiple-order Discrete Fractional Cosine Transform

In [5], the authors proposed an encryption scheme consisting of edge detection (based on advanced cellular neural network structure "CNN"), chaos-based reversible hidden transform and multiple-order discrete fractional cosine transform. This scheme encrypts the image regions with semantic information, whereas the other smooth regions will not be encrypted. About fifty percent of image blocks were fully encrypted; thus, the computation cost is decreased.

Most image encryption algorithms encrypt all the image blocks regardless of contour features or other semantic information in an image; they only consider pixels and bits. This requires a high computational cost. Motivated by this, the authors propose in this paper a lightweight scheme that has a low computational cost that encrypts only contour features and other semantic information of an image. Moreover, transmitting a fully encrypted image after compressing it can result in a loss of compression ratio (to some extent). This loss can be remitted to some extent with the authors' presented encryption scheme.

Encryption can be categorized into two categories, full and selective (partial) encryption. Full encryption encrypts whole information, whereas selective encrypts a particular bit-stream. The major difference between selective and full is the computational cost, as it is higher in full encryption. Even though selective encryption has a trade-off between security and complexity, it has wider practical applications.

In short, in [5] the authors presented a new scheme for lightweight image encryption that can be summarized with the following: The image in pre-processing step undergoes an edge detection step; it is an essential step to recognize significant contour features. Authors used edge detection based on Cellular Neural Network (CNN) with low computational cost. Then the identified significant blocks are encrypted using Cross Chaotic Map-based Reversible Hidden Transform (CCM-based RHT) and Multiple-Order Discrete Cosine Transform (MODFrCT).

RHT transforms (maps) a pair to another one at a lower computational complexity such that:

$$y=\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad (5)$$

$$y=\begin{bmatrix} \alpha x_1+\beta x_2 \\ \beta x_1+\alpha x_2 \end{bmatrix} \quad (6)$$

where $y=\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ is the transformed pair of pixels. And its inverse transform:

$$\tilde{x}=\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} \quad (7)$$

$$\tilde{x}=\begin{bmatrix} \frac{\alpha y_1-\beta y_2}{\alpha^2-\beta^2} \\ \frac{\beta y_1+\alpha y_1}{\beta^2-\alpha^2} \end{bmatrix} \quad (8)$$

such that α and β are secret parameters and it changes for each image where α+β=1 and 0≤α, β≥1.

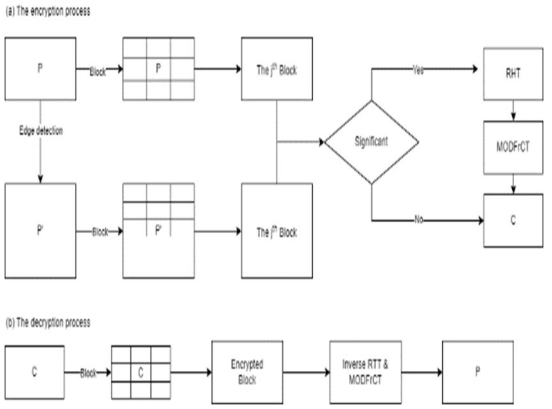

Figure 4 Block Diagram (a) The Encryption Process And (b) Decryption Process

The authors illustrated the encryption/decryption process as shown in figure 4 [5]:
To encrypt an image, they needed to:

1. Input a grey-scale image *P* with size N×N:
   a. Use an Edge detector to recognize the significance of each block.
   b. From the original image P, generate the output of the detected edge as a binary image P'. In binary image P', the detected pixel is referred to as "1" and "0" for the otherwise.
   c. Divide P and P' into non-overlapping m×m pixels blocks. The number of detected blocks in an image is $n=(N/m)^2$.

d. Calculate the significant degree of each detected pixel $\gamma_i=d_i/m^2$ of each block such that $d_i=1,2,\ldots,n$ and d_i is the detected pixels of a block i.
e. Set threshold level (T, $(0 < T \leq 1)$ and obtain Binary Significant Vector (BSV) for each block. The BSV value "1" indicates a significant block while "0" for the contrary. This BSV is required to decrypt the image on the receiver side.

2. Generate the keystream used in the Reversible Hidden Transform (RHT) and Multiple-Order Discrete Fractional Cosine Transform (MODFrCT) based on the Cross Chaotic Map (CCM).
   a. Get the number of the significant blocks Φ, and iterate CCM Φ times with initial conditions a_0 and b_0 to obtain two key vectors *a* and b of length *Φ*,
   b. Generate keystream α_j and β_j such that j=1,2,..., Φ with the equation $a_j = (|a_j| + |b_j|)/2$.
   c. Set $(|a_j|+|b_j|)\in(0,2)$ as the orders of the MODFrCT, namely $P_j=|a_j|+|b_j|$.
3. Encrypt the significant blocks in sequence:
   a. The j^{th} significant block is then transformed by the RHT with the corresponding parameters α_j and b_j.
   b. Perform MODFrCT of the j^{th}sequence with the corresponding order P_j. Then each sequence is replaced with the original block in the same position.
4. The final encrypted image is produced.

The decryption process is much simpler, as the receiver needs to have BSV to perform the inverse of MODFrCT and RHT.

The authors experimented using several typical images such as Lena, Aerial, Boat, Goldhill, Baboon, Peppers, and woman. They found significant contour features in these images have been largely hidden.

2.3 Ievca An Efficient Image Encryption Technique For Iot Applications Using 2-D Von-Neumann Cellular Automata

Authors in [6] presented a lightweight encryption algorithm using 2-D Von-Neumann Cellular Automata (2VCA) with five neighbours, called IEVCA. It has all the properties of a good image cypher, including correlation immunity and lossless. It has passed all the randomness tests of DIEHARD and NIST statistical test suites.

It achieves a high level of diffusion and confusion by using pixel substitution of colour images. IEVCA is robust and achieves high security as it has successfully undergone the security and performance analysis that conventional ones apply for.

Limited resource devices such as the Internet of Things (IoT) work as sensors sending images through the internet to cloud storage for further processing. Critical applications such as defence, and healthcare, require images to be encrypted before transmitting them to the public network to gateway fog nodes. Since conventional encryption algorithms cannot be deployed due to resource-limited devices. Motivated by this, authors in [6] presented encryption based on Cellular Automata (CA) for image encryption due to its simplicity in implementation, efficiency, and resistance to security attacks.

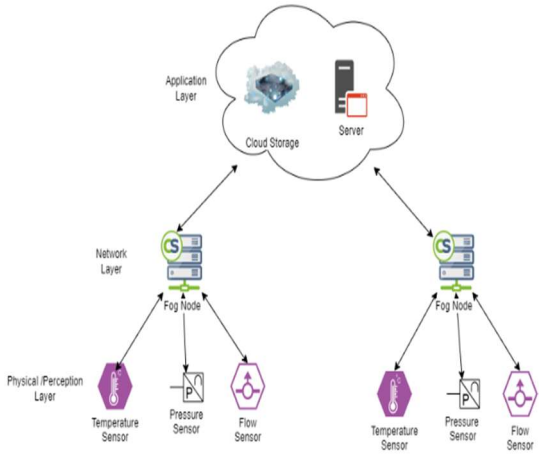

Figure 5 Three-Layer IoT Architecture

The presented scheme is implemented in the physical layer of a three-layer IoT deployment scenario; see figure 5. The three layers are the physical layer where sensors reside and send the capture data to the above layer (network layer), the network layer where fog nodes exist, and they restrict massive network traffic towards the upper application layer (eliminating unnecessary traffic), and application layer where high-end computers and server cloud storage receive the sensors' data or analytical results. End users interact with the application layer.

Cellular Automata is a mathematical model consisting of simple components that act together under transformation rules to construct a complex system. CA can be achieved within many dimensions, such as 1-D CA and 2-D CA. In the presented encryption algorithm, 2-D von Neumann CA (2VCA) was considered. Periodic-boundary and

null boundary considerations are used in the presented 2VCA IEVCA work. The extreme boundary cells in the periodic boundary are considered adjacent to each other while finding neighbours. Meanwhile, the extreme boundary cells in the null boundary are connected to the logic "Zero".

2.3.1. 2-D CA rule generation

A cell in CA is transformed from 0 to 1 or vice versa according to specific CA rules. These rules are affected by the nine neighbours of a cell (including the cell itself). Most of the CA rules are constructed through primary rules. As an example of a primary rule, if a cell value is 0 and has at least three alive neighbours, then that cell's value is changed to 1. This rule is a primary rule, and other rules are generated using these primary rules. For example, let Rule 1, Rule 2, Rule 4, Rule 8, and Rule 16 be primitive rules such that:

$$\text{Rule1}::[S_t]=[S_t] \quad (9)$$

$$\text{Rule2}::[S_{(t+1)}]=[S_t][M_2] \quad (10)$$

$$\text{Rule4}::[S_{(t+1)}]=[M_2] \quad (11)$$

$$\text{Rule8}::[S_{(t+1)}]=[S_t][M_1] \quad (12)$$

$$\text{Rule16}::[S_{(t+1)}]=[M_1][S_t] \quad (13)$$

And M_1, M_2 be any matrices for instance:

$M_1=\begin{pmatrix}0&1&0\\0&0&1\\0&0&0\end{pmatrix}$, $M_2=\begin{pmatrix}0&0&0\\1&0&0\\0&1&0\end{pmatrix}$, and $S_{(t+1)}$ be the start of a cell at time t+1 and S_t be the start of a cell at time t. More complex rules can be generated from the above primary rules, such as:

Rule 11P = Rule 1P + Rule 2P + Rule 8P (14)

$\therefore [S_{t+1}]=[S_t] + [S_t][M_2] + [S_t][M_1]$

Rule 24P = Rule 8P + Rule 16P (15)

$\therefore [S_{t+1}]=[S_t]\,[M_1] + [M_2][S_t]$

Rule 25P = Rule 1P + Rule 8P+Rule 16P (16)

$\therefore [S_{t+1}]=[S_t]+[S_t]\,[M_1] + [M_2][S_t]$

Rule 30P = Rule 2P + Rule 4P+Rule 16P (17)

$\therefore [S_{t+1}]=[S_t]\,[M_2]+ [M_1][S_t]+[S_t]+[S_t]\,[M_1]$

Note: P indicates the cells are under periodic boundary conditions, whereas N is for null-boundary conditions.

A rule vector (CARV) is a set containing a set of rules. And as an example, it can be symbolized such as:

$$\text{CARV}=\begin{pmatrix}31N&11N&22N\\26N&2N&26N\\7N&15N&26N\end{pmatrix} \quad (18)$$

In the case of a group of cells having k as the initial configuration and when it undergoes a certain number of transitions and ends up with the initial state k, then this CA is called Group CA (GCA). In this authors' presented work, Group CA (GCA) is obtained from different rule vectors. Rule vectors belong to different classes to ensure a high degree of confusion and diffusion property in the cipher images. Authors generated Von Neumann GCA rule vectors under both null-boundary and periodic-boundary conditions.

2.3.2. Image Encryption Algorithm

The encryption algorithm generates the encrypted image "I_{enc}" and the secret "symmetric" key K (to be used in the decryption algorithm). The encryption algorithm encrypts an input colour image (figure 6) [6] "I" of size m×n using 2D CA rule vectors (CARVs). First, red R(MXN), green G(MXN) and blue B(MXN) channels are extracted from the plain colour image I. Then each channel "R(mxn), B(mxn), G(mxn)" is converted into binary format; each pixel is converted into 8 bits "Bin(R)(mx(8n), Bin(B)(mx(8n), Bin(G) (mx(8n)".

Binary image blocks are substituted using the CARVs rule list. From this CARV list three rule vectors (k_1,k_2,k_3) are selected randomly to encrypt these three channels (red, green, and blue). The encryption is done with a random number of round iterations (r_{itr}). In each iteration, the binary image goes through random CARVs taken from CARVList through the Rule scheduler. Then, these binary images are converted into red, green, and blue channels to be combined as one encrypted image.

The decryption process uses the same secret key "K" for decrypting "I_{enc}". It follows the same steps as in the encryption process, except it decrypts the encrypted binary images.

2.4 Lightweight Selective Encryption for Social Data Protection Based on EBCOT Coding

To effectively protect social media while its data is being generated, sent, transmitted, and shared through online social media platforms, the authors presented a novel design based on an agnostic selective encryption concept based on the embedded block coding with optimized truncation (EBOCT) system. Inspired by SE, the authors presented an effective agnostic selective encryption that encrypts a small subset of the byte-stream 8% of the stream) based on arithmetic techniques.

The trending development of social sensing systems has brought the urge to protect data while being generated, stored, and transmitted. Meanwhile, most traditional encryption algorithms are unsuitable for data protection in social sensing and data-sharing systems. Traditional encryption methods are designed based on the assumption of having one sender and one receiver during the communication process. However, this is not

efficient when many users are involved as receivers. Besides that, existing Selective Encryption (SE) methods are not suitable for today's social sensing data since they are strictly format reliance and implementing them on such data is very costly. Motivated by this, authors in [7] presented a selective encryption scheme based on an in-compression approach.

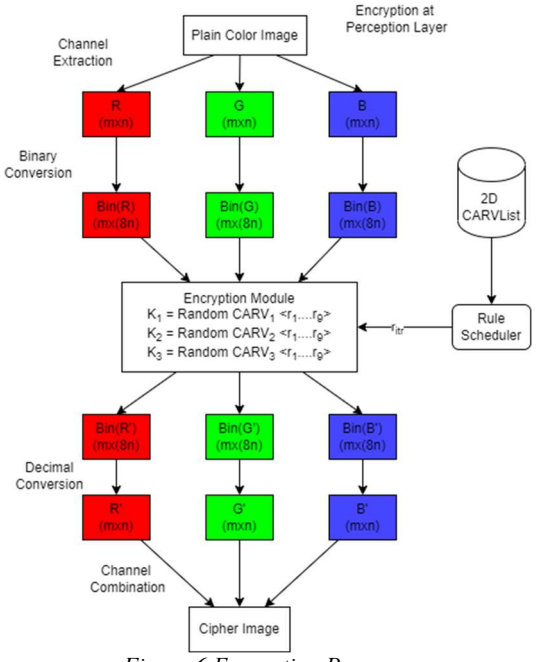

Figure 6 Encryption Process

To protect social data generated, sent, and transmitted through social sensing systems authors presented a novel design in [7]; they adopted an agnostic selective encryption concept based on the embedded block coding with an optimized truncation system (EBCOT).

During the EBCOT coding process, redundant data content is removed; thus, data will be very sensitive to the tiny changes since there is propagation for the decoding process such that a small ratio of SE could lead to very different output results, which can resist recovery from the attackers.

Authors presented this basic design to selectively encrypt in a lightweight manner some bitstream in the middle of the coding system process such that the output data files are protected.

The authors used the arithmetic coding system in JPEG2000 to propose SE encryption. The MQ coder in this arithmetic coding system is a context-based adaptive binary arithmetic coding system (AC). JPEG200 standard is mainly formed by two tiers; tier-1 is the entropy coding combined with the MQ coding, and tier-2 is the packetization process which generates the code packets from code streams. The authors adopted a tier-1 coding system from JPEG2000 as a compression process of the bitstream, and within this tier, they performed the Selective Encryption (SE) process with a percentage ratio of the bitstream.

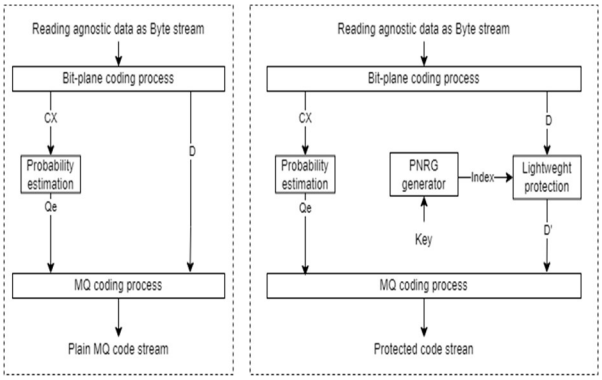

Figure 7 Architecture Of How The Coding-based SE Is Performed. (a) On Left Is The Normal MQ Coding Process. (b) On Right, Presented SE With The MQ Coding Process

As illustrated in figure 7 [7], data are read as byte streams agnostically, disregarding their original format. Then, the byte stream is encoded into the context information (CX) using a bit-plane encoding process. MQ coding process uses the context information (CX) to control the adaptivity of AC by generating the probability estimation (Q_e) from the CX. In bitstream D, authors adopted SM2LSB-plane encoder to selectively encrypt and protect D using a lightweight selective encryption algorithm [7] symbolized by SE_C function:

$$SE_C(F, K, R)$$

Where F is the data content representing the data input, K is a secret key, and R is the selection ratio.

Authors fragmented data context "F" bitstream into N fragmentations such that $F = F_1, F_2, F_3, \ldots, F_3$ for parallel processing encryption as displayed in figure 8 [7]. Each fragment is selectively encrypted within the compression system in tier-1 by an algorithm summarized with $Enc_{(F)}=SE_C(F,K,R)$, where K: is the secret key used for pseudorandom number generator (PRNG), R: is the ratio of selective encryption of bit steam, and F: is a fragment of data content.

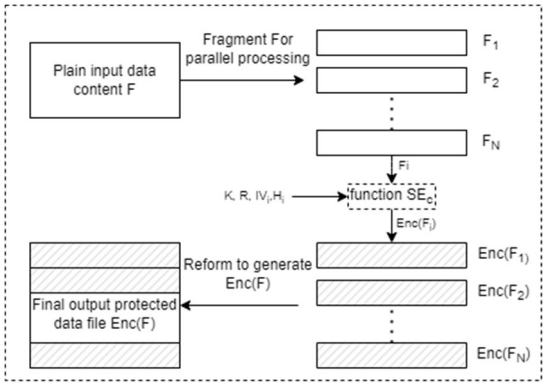

Figure 8 System architecture of how the fragmentation of F is processed

The authors obtained the proper ratio R through the protection analysis test, in which most of the tests concluded a ratio of 8% with a high level of protection. The authors used the secret key "K" in the pseudorandom number generator to selectively protect the bit stream. However, if this secret key is reused, the generated random number will be repeated (same), and it is exposed to a chosen/known-plaintext attack. Thus, the authors increased the inputs of PRNG to include three other parameters: the secret key "K", the hash value of the input plaintext fragment "H" and the initialization vector "IV".

2.5 Outsourcing Chaotic Selective Image Encryption to The Cloud with Steganography

With the help of steganography, authors in [8] proposed a scheme for outsourcing chaotic selective image encryption to the cloud; to protect image data from being exposed to a third-party cloud service.

Devices such as smartphones and real-time communication devices face a challenge in encrypting images because of their limited resources (energy and computational power) thus traditional encryption paradigm (entire bit stream image encryption) is no longer suitable. Besides that, image encryption outsourcing also encounters other challenges on how to protect data images and not be revealed to third-party cloud outsourced encryption services. Many researchers have been seeking chaotic encryption for image encryption, even though many chaotic image ciphers are computationally extensive (cannot be managed with resource-limited devices). Motivated by this, the authors considered the problem of selectively encrypting a plain image by a chaotic map and distributing the encrypted image to other user with no sufficient computational power or energy supply to be outsourced on the cloud.

In the authors' scheme, a resource-limited client sends a stego image (contains selective secret information image) to a cloud service to do the chaotic encryption. The Cloud service sends the encrypted stego image back to the user. The user shares encrypted image with other users.

In [8], the presented scheme is shown in following figure 9 [8].

A client chooses the important part of data in an image M of size m×n to be selectively encrypted. The image M can be constructed as the 4 most significant bits (MSBs) H and the 4 least significant bits L (LSBs), i.e., M=H || L. Important information (I) of the plain image is masked by doing the XORs with L such that I=H⊕L.

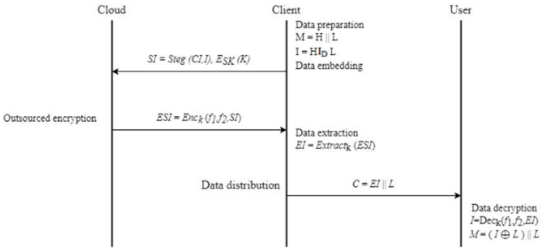

Figure 9 The Framework Of The Presented Scheme

Then I is embedded into a cover image (CI) using Single Match 2 Least Significant Bit (SM2LSB) to produce a stego image (SI) so that (SI=Steg(CI, I). Then, the client sends a stego image (SI) to the cloud for chaotic encryption along with the encrypted key (K) using a fast encryption cipher with the shared key (SK). Cloud has no knowledge of the existence of hidden data.

In the cloud, the stego image is encrypted using two predetermined chaotic maps f_1,f_2 i.e. ESI=Enc_k(f_1,f_2,SI); f_1 to permutate the pixel position of SI and f_2 to perform XOR masking with each pixel value of the permutated (SI). The cloud encrypts the whole stego image SI and sends it back to the client. The client extracts the encrypted embedded important data (EI), such that EI=$Extract_K$(ESI)

Now, the client can transfer the selectively encrypted image (C) to other users securely through public channels. The client needs to get the encrypted image (C) by concatenating (EI) and the least significant bit L such that C=EI || L.

The selectively encrypted image (C) can be decrypted by obtaining the secret key (K) and running the decryption algorithm to get the important information I such i.e. $I = Dec_K(f_1, f_2, EI)$. The client can know which data (EI) is selectively encrypted by splitting C into EI and L such that C=EI || L. After obtaining the important information I, the user can easily get the decrypted image M by M=(I⊕L)||L.

Authors adopted SM2LSB presented in [13] since it offers lower probability detection of hidden important data by making fewer changes to the cover image compared to 2LSB replacement. The main idea behind Single Match 2LSB is to embed 2-bit information into 2 LSB of the cover image, and a third LSB is used as a flag indicating the mismatch position.

SM2LSB maximizes embedding capability while keeping high security in the spatial domain by reorganizing and embedding important data to a cover image. In data reorganization, important data (I) containing 4 bitplanes is reorganized into 2 bitplanes of I', and by doing so, the vulnerability of having consecutive 0s and 1s in I' is avoided. The size of I' has the same size as the used cover image (to be able to support chaotic encryption in a later step). It reorganizes 4 bitplanes of I into 2 bitplanes by extracting each bit of 4 bitplanes in a repeated raster scan order for all the coordination of I and buffering the extracted bits onto a sequence. The authors used this sequence to construct the first and second bitplanes of I'. In data embedding, SM2LSB ensures the minimal changes of 2LSB in the cover image for much lower detection compared to 2LSB.

The receiver decrypts using key $Dec_{SK}(K)$ the 2LSB of ESI and partially decrypts the third LSB (flag to indicate the position of mismatch bit). This finally results in obtaining the hidden decrypted data.

2.6 Edge-Based Lightweight Selective Encryption Scheme for Digital Medical Images

To overcome the computation complexity and high processing time, the authors in [9] adopted an edge-based lightweight selective encryption in their work. They used a combination of One Time Pad (OTP), edge detection (Prewitt edge) and a Chaotic map approach. Authors used edge detection only to encrypt the significant image blocks and thus reducing the computational time using the OTP algorithm. To resist well know attacks, the authors used the chaotic map in [9] to produce a highly sensitive key with an appropriate large key space and at the same time having a relatively high image quality.

The rapid growth and the storing of transferred medical images through the public network have a high demand to secure such traffic, considering the special structure of these medical images. Traditional algorithms are designed for textual data and not for images which have their own complexity. Medical images are in large data volumes and have a strong correlation between pixels and high redundancy.

Chaos-based encryption has also been considered in the authors' presented work in [9]. Since chaotic encryption can efficiently and securely encrypt images due to the randomness of its output. However, chaotic encryption still has computation complexity and high processing time, especially in real-time applications. Motivated by this, the authors have proposed selective encryption of the medical images in [9] to overcome the mentioned issues.

In [9], the authors presented a scheme (illustrated in figure 10 [9]) that starts by decomposing a medical image into non-overlapping blocks of pixels of a specific size. Then, using Prewitt edge detection, significant image blocks were identified according to a specific threshold value. Then, using the chaotic map, a matrix of random keys is generated that equals the total number of significant blocks in an image. Finally, each identified significant block is encrypted in sequence using the one-time pad algorithm.

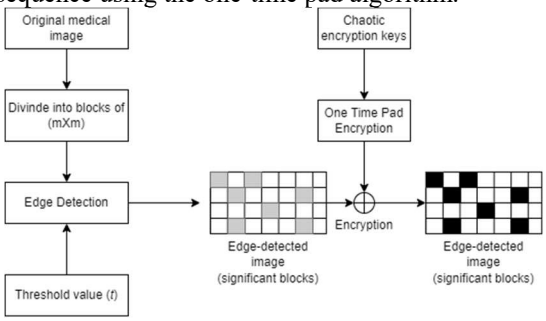

Figure 10 Block Diagram For Edge-Based Medical Image Encryption

3. EVALUATION CRITERIA

This section consists of comparison tables for all the presented encryption schemes related to selective and chaotic map encryption algorithms. The Cryptoanalysis tests in this survey include Histogram Analysis, Correlation Coefficient Analysis, Key Space Test, Key Sensitivity Test, Number of Pixels Change Rate (NPCR), Uniform Average Change Intensity (UACI), Uniform Histogram Deviation (UHD), Irregular Deviation (I_D), Plaint text and ciphertext attack (Cryptoanalysis attack), Time Analysis, Information Entropy Analysis, Selective Encryption Domain, Relative Mean Square Error function (RMSE), Mean Square Error function (MSE), DIEHARD randomness tests, National Institute of Standards and Technology (NIST) randomness, Visual Effects for images, Peak signal-to-noise ratio (PSNR), Structural Similarity (SSIM), Difference Test, Normalized Mutual-Information (NMI), Plain Text Sensitivity, Compression Performance, Speed

Performance, Chi-square, Difference Image Histogram (DIH), 2LSB steganalysis, Computational cost, Communication Cost, Security dependency, Performance, Normalize Cross Correlations (NCC), and Robustness test against different levels of noise.

Table 1: Cryptoanalysis Tests

Paper	*Number of Tests*
[4]	12
[5]	2
[6]	10
[7]	9
[8]	4
[9]	8

Table 1 demonstrates the number of cryptoanalysis tests authors performed on their presented work. We can observe that authors in [4] [6] [7] and [9] have done the most test within the selected encrypted algorithms.

Distinctive cryptoanalysis tests of each algorithm are summarized in tables 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 and 19. These tables exemplify the ideal values of these cryptoanalysis tests (contrast analysis, time analysis, I_D, RMSE, DIEHARD randomness tests, NIST randomness test, robustness test against different levels of noise test, MSE, visual effects for images test, MNI, difference test, speed performance, compression performance, plain text sensitivity, chi-square, DIH, 2LSB steganalysis as in [14] and NCC) and the corresponding obtained values in authors' evaluation tests.

Meanwhile, tables 20, 21, 22, 23, 24, 25, 26, 27, 28 and 29 illustrate the ideal values of the tests (Histogram Analysis (HA), Correlation Coefficient Analysis (CC), Key Space Test (KST), key sensitivity tests, NPCR, UACI, Plaint text and ciphertext attack (PT&CC) and Information Entropy Analysis (IE), PSNR and SSIM) and a comparison among the corresponding test results obtained in [4] [6] [9] [5] [7] [8].

Table 2: Contrast Analysis

	Contrast Analysis
Ideal Value	To prove the robustness and security of the presented image encryption image scheme, the difference value between the intensity of a pixel and its neighbour pixel must be large enough [4].
Authors Values [4]	Contrast values for a colour image, grey image and binary image are as follows respectively 10.1145, 10.1874, and 10.0166. A high contrast value indicates a good image encryption scheme.

Table 3: Time Analysis

	Time Analysis
Ideal Value	Much lower encryption A comparison with [15] and [16].
Authors' Values [4]	

Table 4: I_D

	Irregular Deviation "I_D"
Ideal Value	It measures the closeness of the statistical distribution of the deviation between the input and encrypted image to the uniform statistical deviation [17]. A good enough encryption algorithm if the I_D value is close to uniform distribution [18]. This measurement is good for comparison with other encryption algorithms and is not to be used alone.
Authors' Values [4]	I_D values of a colour image, grey image and binary image are as follows respectively 32820, 33541, and 32561.

Table 5: RMSE

	RMSE
Ideal Value	A value of 0 indicates a totally different image. A value closer to 1, indicates no difference between the two images. [5]
Authors' Values [5]	Between 0.6 and 0.7

Table 6: DIEHARD Randomness Tests

	DIEHARD randomness tests
Ideal Value	p-values of range [0,1) are accepted [6]
Periodic VCA [6]	Authors got p-values within the range of (0.162652 to 0.985227) for DIEHARD tests.
Null VCA [6]	Authors got p-values within the range of (0.1527293to 0.999153) for DIEHARD tests.

Table 7: NIST Randomness

	NIST randomness
Ideal Value	p-values larger than 0.001 are considered acceptable and indicate a high degree of randomness in ciphertext images [6]
Periodic VCA [6]	The authors got p-values within the range of (0.1242 to 0.9856) for NIST tests.
Null VCA [6]	The authors got p-values within the range of (0.1365 to 0.9944) for NIST tests.

Table 8: Robustness Test Against Different Levels Of Noise

	Robustness test against different levels of noise
Ideal Value	"Salt and Pepper" noise was applied. Decrypted images with noise levels (1%, 10%) are easily recognized
Periodic VCA [6]	
Null VCA [6]	

Table 9: MSE

	MSE
Ideal Value	A high value indicates an unrecognizable plain image from a cipher image
Periodic VCA [6]	The authors obtained the following range of values of MSE: 82.38 and 93.37
Null VCA [6]	The authors obtained the following range of values of MSE: 82.48 and 93.37

Table 10: Visual Effects for images

	Visual Effects for images
Ideal Value	Obvious hard visual degradation of the encrypted image
Authors' Values [7]	Obvious visual degradation is reached when 5% to 8% of the encoded bitstream is encrypted for the four encrypted file types (image, video, GPS log and ASCII).

Table 11: NMI

	NMI
Ideal Value	A value of 0 implies mutual information is found between a plaintext and its ciphertext thus recovery is impossible. However, a value of 1 means two are identical. Nevertheless, a close value of 1 indicates a very possible recovery of the plain text.
Authors' Values [7]	As illustrated to authors through their experiments, an encryption ratio larger than 8% achieved a value of NMI very close to zero except for the MP4 file by which the NMI value remained with a higher value of 0.1.

Table 12: Difference Test

	Difference Test
Ideal Value	Optimal difference ratio is 50% [7].
Authors' Values [7]	The Optimal ratio of 50% is reached when the encryption ratio is set to 8% for the four encrypted file types (image, video, GPS log and ASCII).

Table 13: PTS, CP, And SP

	Speed Performance
Ideal Value	Not measured, but the authors suggested boosting the speed of EBCOT encoding by using GPGU instead of CPU-based implementation.
Authors' Values [7]	

Table 14: Compression Performance

	Compression Performance
Ideal Value	With the video-type files, more than 20% of the bitstream was generated because of the selective encryption method. Meanwhile, a 10% more bitstream is generated with the other file types (ASCII, Image, and GPS)
Authors' Values [7]	

Table 15: Plain Text Sensitivity

	Plain Text Sensitivity
Ideal Value	A 1-bit difference in a plaintext should produce a totally different ciphertext with a different percentage of 50% (in a bit level) [7].
Authors' Values [7]	Authors got a range of values from 49.22% to 50.41% difference.

Table 16: Chi-Square

	Chi-square
Ideal Value	An ideal probability value of zero implies a cover image does not have any embedded data [14] [8].
Authors' Values [8]	The presented steganographic system gave a result of 0.

Table 17: DIH

	DIH
Ideal Value	The correlation between the LSB bit plane and the remained bit planes is weak, and it becomes weaker and weaker when more secret messages are embedded in LSB. Eventually, the LSB plane will be independent of the remained bit planes [19].
Authors' Values [8]	Lower detection by 34.6% compared to 2LSB replacement

Table 18: 2LSB Steganalysis

	2LSB steganalysis as in [20]
Ideal Value	A method in [20] was used to find the detection probability for the hidden message "length by 2LSB".
Authors' Values [8]	Authors could reduce the detection probability of the hidden message by 43.4% compared to the standard 2LSB replacement.

Table 19: NCC

	NCC
Ideal Value	A value close to 1 implies a high similarity between the two compared images. Otherwise, a value close to 0 indicates completely different images [9].
Authors' Values [9]	Comparing the original and cipher images, a value larger than 0.82

Table 20: Key Sensitivity Test

	Key Sensitivity Test
Ideal Value	A slight change (1-bit difference) in the key produces a completely different cipher image for the same plain text (a 50% difference between ciphertext and plaintext) [4], and a 100% difference between the two produced ciphertexts [21] In other words, a tiny change of the private key produces a non-intelligible recovered image [22]
Authors' Values [4]	With a one-bit change in a key, a totally different ciphertext image is generated. Authors could obtain a 98.2212% to 99.5991% difference between two ciphertext images by changing one bit of the key
Periodic VCA [6]	Generates a completely different decrypted unrecognized image on a slightly modified private key. Key sensitivity test results ranged from 99.65% to 99.76%.
Null VCA [6]	With a one-bit change in a key, a totally different ciphertext image is generated. Authors could obtain a 99.63% to 99.74% difference between two ciphertext images by changing one bit of the key.
Authors' Values [5]	A totally different cyphertext image is produced with only 10^{-16} slight change of the initial key values
Authors' Values [7]	Authors got a range of values from 49.11% to 50.90% difference between plaintext image and its ciphertext.

Table 21: HA

	Histogram Analysis
Ideal Value	An obtained uniform and flat histogram is an indication of the existence of a high level of randomness in pixel values of cipher image [6] [23]. Thus the ideal histogram of an encrypted image must be uniform (flat)
Authors' Values [4]	The authors got a uniform pixel distribution (histogram) of the ciphertext thus no important information is disclosed about the ciphertext statistics.
Periodic VCA [6]	Authors could acquire a uniform and flat histogram
Null VCA [6]	Authors could acquire a uniform and flat histogram
Authors' Values [9]	Authors got histogram graphs of the encrypted images approaches Gaussian distribution which indicates the frequency distribution is hidden within the encrypted histogram.

Table 22: CC

	Correlation Coefficient Analysis
Ideal Value	An ideal value must be very close to zero. Otherwise, if the value lays near 1.0 means a highly correlated ciphertext and fails to resist statistical attacks [4].
Authors' Values [4]	The authors obtained a range of results close to zero. Horizontal: from -0.0163 to 0.1007, vertical: from -0.033 to 0.1081, and diagonal: from -0.0474 to 0.1081
Periodic VCA [6]	The authors obtained a range of values very close to zero for Horizontal: from -0.0021 to 0.0010, Vertical: from -0.0030 to 0.0007, and Diagonal: from -0.0057 To 0.0011)
Null VCA [6]	The authors obtained a range of values very close to zero for Horizontal: from -0.0024 to 0.0053, Vertical: from 0.0018 to 0.0078, and Diagonal: from 0.0023 To 0.0051)
Authors' Values [9]	Close to zero (-0.0763 to -0.6463)

Table 23: KST

	Key Space Test
Ideal Value	An ideal value of key space is 2^{128} to be brute force attack resistance [24].
Authors' Values [4]	The authors obtained a key space value of 2^{299}
Periodic VCA [6]	Authors obtained $2^{81\times p}$ where p is a non-zero integer of random blocks.
Null VCA [6]	
Authors' Values [9]	The key constraint for a block size of 3×3 is 160,000

Table 24: NPCR

	NPCR
Ideal Value	Ideal value of NPCR is 99.6094% [25] [26].
Authors Values [4]	The authors got NPCR values 99.2304%, 99.1547%, and 99.3421% for colour, grey, and binary images respectively.
Authors' Values [9]	Mean value of 99.017%
Periodic VCA [6]	Presented scheme in [6] obtained NPRC values within a range between 99.6123% to 99.7412%
Null VCA [6]	The authors got NPCR values 99.2304%, 99.1547%, and 99.3421% for colour, grey, and binary images respectively.

Table 25: UACI

	UACI
Ideal Value	Ideal value of UACI is 33.4635% [25] [26]
Authors Values [4]	The authors got UACI values of 33.0341%, 33.2072%, and 33.0021% for colour, grey, and binary images respectively.
Authors' Values [9]	Mean value of 32.63
Periodic VCA [6]	Presented scheme [6] obtained UACI values within a range between 33.3564% to 33.4822%
Null VCA [6]	The authors got UACI values of 33.0341%, 33.2072%, and 33.0021% for colour, grey, and binary images respectively.

Table 26: Plain text and ciphertext attack

	Plain text and ciphertext attack
Ideal Value	Resistance against chosen plaintext attack is consequently resistant against the ciphertext-only attack and chosen ciphertext attacks [6].
Authors' Values [4]	Not feasible since the key is very sensitive to slight changes (having a slight change of 1 bit to the key, produces a totally different ciphertext image).
{Periodic VCA [6]	Authors achieved a high level of confusion and diffusion by obtaining a one-to-many mapping between a plain image and a cipherimage. Thus, resistance against chosen plaintext attack. Hence, also resistant to other attacks (ciphertext-only attacks and chosen ciphertext attacks).
Null VCA [6]	

Table 27: Information Entropy Analysis

	Information Entropy Analysis
Ideal Value	Ideal entropy value for random image with 256 values is 8 [27] [21].
Authors' Values [4]	The authors got entropy values of 7.9921, 7.9969 and 7.9925. Thus, resistant to entropy attacks.
Periodic VCA [6]	Information entropy values varied from 7.9973 to 7.9997.
Null VCA [6]	Information entropy values varied from 7.9884 to 7.9938.

Table 28: SSIM

	SSIM
Ideal Value	A value close to 1 implies a high similarity between the two compared images. Otherwise, a value close to 0 indicates completely different images [9] [7].
Authors' Values [7]	The SSIM Optimal Value is reached when the selection ratio to the encrypted stream is 8% for the four encrypted file types (image, video, GPS log and ASCII). SSIM values were less than 0.01 except for ASCI was 0.086.
Authors' Values [9]	Comparing the original and cipher images, a value larger than 0.82 is obtained. This indicates the original plaintext and its ciphertext images are approximately the same.

Table 29: PSNR

	PSNR
Ideal Value	A large difference between the protected image and its original one is indicated by the low PSNR value [7]. In [6] considers 30 as an ideal value, whereas in [9], a value of 34 dB is considered acceptable. Meanwhile, in [28] a value greater than 38dB is acceptable.
Authors' Values [7]	The low value of PSNR of 8db is acquired when encrypting 8% of the encoded bitstream for the four encrypted file types (image, video, GPS log and ASCII).
Authors Values [8]	Authors got values above 40db.
Periodic VCA [6]	The authors obtained the following range of values of PSNR: 28.42 and 28.95
Null VCA [6]	The authors obtained the following range of values of PSNR: 28.41 and 28.96
Authors' Values [9]	The authors had a mean value of 39.84 dB

We can observe from the previous evaluation tables, all the presented works have cryptanalysis test results near the ideal values, thus making them efficient for lightweight image encryption algorithms in limited-resource devices.

During presenting these works, the following notes are considered:

1- The authors in [4] encrypt the selected blocks having larger value than a specific threshold, and then they diffuse and permutate all the image's blocks (plain and cipher blocks). We suggest performing the permutation only on the plain blocks, this would decrease the overall computational cost.

2- In [5], the authors do not mention if coloured images were involved in the experiment or not. It is unknown if images involving texts are also considered, especially if the text in the image is big enough and is surrounded by a smooth region; this might cause their encryption scheme not to completely conceal the text as it only encrypts the edges of the text but not the text itself as a complete word. Moreover, the authors' work in [5] is suitable for some special commercial multimedia applications.

3- In the cryptography scheme [6], many open questions could arise upon reading this, such as: Is the system prone to bit errors? Estimation probability Q_e is generated using the context

information of the binary data D but is it a better approach to consider D' instead of D for Q_e !

4- Meanwhile, the presented work in [8] uses grey-scale images since they adopted SM2LSB [13] for grey-scale images, thus making this scheme unfeasible for colourful images. The authors were not clear about this limitation.
5- The work presented in [9] deals with grey scale medical images and their work gives promising results and more realistic real time image encryption needs to be done for better evaluation and improve the security and performance aspects.
6- Meanwhile, [6] present an encryption technique which is not a selective one, but its cryptoanalysis results outperform the other selective encryption techniques. This brings to the field a new future work to have a hybrid technique that combine cellular automata and selective encryption.
7- Presented algorithms in [5] [7] [8] [9] deals with plain text images as a grey ones meanwhile in [4] [6] colour images are the input of their presented algorithms. As a demonstrated in the comparison tables among the presented algorithms, their tests results were all near the ideal values. However, algorithms presented in [6] holds the test results with the best values near the ideal ones. Work presented in [6] does not selectively encrypt images but it is designed to be a lightweight encryption algorithm for IoT devices; this creates a future work.

4. PROBLEMS AND OPEN RESEARCH

Based on the evaluations in the above tables we can do an overall evaluation for the five encryption categories beside the cellular automata algorithm as in Table 30.

Table 30: Categories evaluated beside the cellular automata

Category	Work	Overall Evaluation
In-compression approach	[7]	Good
Coefficient correlation	[4]	Good
Edge detection	[9] [5]	Good
Most significant bits of an image	[8]	Good
Byte stream with encryption ration	[7]	Good
Cellular Automata	[6]	Best cryptoanalysis results among the others

As noted in Table 30, the different encryption algorithm techniques met the overall evaluation, and all of them were near the optimal value; this encourages a new hybrid approach that combines one of these techniques and cellular automata.

We suggest combining one of the selective encryption algorithm techniques +such as "coefficient correlation" along with cellular automata (CA); this will bring CA encryption technique into new level by performing selective encryption of an image blocks instead of full encryption, and it is done by selecting and encrypting specific blocks using 2D-CA. This brings new approach of research in more depth of the mentioned categorized encryption techniques along with the cellular automata.

5. DISCUSSION AND CONCLUSION

The main contribution to this review is to summarize some of the existing lightweight image encryption in the spatial domain. This helps researchers in directing their work by obtaining the most effective and efficient technique for performing a lightweight image encryption for limited-resource devices. More papers shall be reviewed for each presented category technique (in-compression, coefficient correlation, edge detection, most significant bits, and encryption during byte stream), and expand our taxonomy with more algorithm techniques along their evaluation criterion (cryptoanalysis tests results comparison) based on the technique. A future work is suggested to investigate more algorithm techniques and combine more than two together to produce a better hybrid image encryption regarding performance and robustness.

In this review, some of the lightweight encryption algorithms in spatial domain were highlighted. With the resource limited devices, the conventional algorithms fail to protect multimedia images that are exponentially growing. This review started by introducing the difference between the spatial domain and the frequency domain. Then categorized the presented algorithms into five categories. Then we presented each algorithm in a separate section. In the last section, all the cryptoanalysis tests performed for each work are presented with comparisons among them. All these tests were passed successfully and near the ideal values for the presented works in [4] [6] [7] [8] [9] [5].

REFERENCES:

[1] M. a. G. G. a. H. W. a. D. O. a. E. A. S. arajallah, "Selective Encryption of the Versatile Video Coding Standard," *IEEE Access,* vol. 10, pp. 21821-21835, 2022.

[2] Z. a. E. A. S. a. F. M. a. K. A. a. L. R. a. D. O. Fawaz, "Lightweight chaos-based cryptosystem for secure images," in *8th International Conference for Internet Technology and Secured Transactions (ICITST-2013)*, 2013.

[3] A. a. X. D. a. A. S. A. a. o. Kulsoom, "An efficient and noise resistive selective image encryption scheme for gray images based on chaotic maps and DNA complementary rules," *Multimedia Tools and Applications,* vol. 75, no. 1, 2016, pp. 1-23.

[4] J. S. a. A. J. Khan, "Chaos based efficient selective image encryption," *Multidimensional Systems and Signal Processing,* vol. 30, no. 2, 2019, pp. 943-961.

[5] Y. X. D. W. W. &. T. Y. Zhang, "Edge-based lightweight image encryption using chaos-based reversible hidden transform and multiple-order discrete fractional cosine transform," *Optics & Laser Technology,* vol. 54, 2013, pp. 1-6.

[6] S. S. M. P. C. V. N. S. K. &. R. U. Roy, "IEVCA: An efficient image encryption technique for IoT applications using 2-D Von-Neumann cellular automata," *Multimedia Tools and Applications,* vol. 80, no. 21, 2021, pp. 31529-31567.

[7] H. Q. M. L. M. &. M. Z. Qiu, "Lightweight selective encryption for social data protection based on EBCOT coding," *IEEE Transactions on Computational Social Systems,* vol. 7, no. 1, 2019, pp. 205-214.

[8] T. H. J. &. S. J. Xiang, "Outsourcing chaotic selective image encryption to the cloud with steganography," *Digital Signal Processing,* vol. 43, 2015, pp. 28-37.

[9] O. A. &. A. M. Khashan, "Edge-based lightweight selective encryption scheme for digital medical images," *Multimedia Tools and Applications,* vol. 79, no. 35, 2020, pp. 26369-26388.

[10] I. a. F. M. a. A. N. a. H. W. Hraini, "Joint crypto-compression based on selective encryption for WMSNs," *IEEE Access,* vol. 9, 2021, pp. 161269-161282.

[11] D. a. T. S. a. o. Ziou, "Edge detection techniques-an overview," *Pattern Recognition and Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii,* vol. 8, 1998, pp. 537-559.

[12] R. a. F. M. a. H. R. Qumsieh, "Joint block and stream cipher based on a modified skew tent map," *Multimedia Tools and Applications,* vol. 78, no. 23, 2019, pp. 33527-33547.

[13] O. &. A. B. Khalind, "Single-mismatch 2LSB embedding steganograph," in *IEEE International Symposium on Signal Processing and Information Technology*, 2013.

[14] A. &. P. A. Westfeld, "Attacks on steganographic systems," *International workshop on information hiding*, 1999, pp. 61-76.

[15] A. M. H. A. H. &. A. M. A. Ayoup, "Efficient selective image encryption," *Multimedia tools and applications,* vol. 75, no. 24, 2016, pp. 17171-1718.

[16] I. I. W. &. M. A. Ullah, "Selective region based images encryption," in *2013 2nd National conference on information assurance (NCIA)*, 2013, December.

[17] H. M. &. M. M. A. Elkamchouchi, "Measuring encryption quality for bitmap images encrypted with rijndael and kamkar block ciphers," in *Proceedings of the Twenty-Second National Radio Science Conference*, 2005.

[18] J. S. u. R. A. A. J. &. H. Z. Khan, "A new chaos-based secure image encryption scheme using multiple substitution boxes," in *2015 Conference on information assurance and cyber security (CIACS)*, 2015.

[19] T. &. P. X. Zhang, "Reliable detection of LSB steganography based on the difference image histogram," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*, 2003.

[20] C. S. X. Q. J. &. X. Z. Niu, "Steganalysis of two least significant bits embedding based on least square method," in *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, 2009.

[21] C. Zhu, "A novel image encryption scheme based on improved hyperchaotic sequences," *Optics communications,* vol. 285, no. 1, 2012, pp. 29-37.

[22] X. &. L. D. Wang, "A novel image encryption algorithm using chaos and reversible cellular automata," *Communications in Nonlinear Science and Numerical Simulation,* vol. 18, no. 11, 2013, pp. 3075-3085.

[23] Y.-Q. &. W. X.-Y. Zhang, "A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice," *Information Sciences*, 2014, pp. 273, 329-351.

[24] I. Ecrypt, "European network of excellence in cryptology ii," *Yearly Report on Algorithms and Keysizes (2009-2010)*, 2010, pp. 539-556.

[25] Y. N. J. P. &. A. S. Wu, "NPCR and UACI randomness tests for image encryption," *Cyber journals: multidisciplinary journals in science and technology,* vol. 1, no. 2, 2011, pp. 31-38.

[26] Z. &. Z. Y. Hua, "Image encryption using 2D Logistic-adjusted-Sine map," *Information Sciences,* vol. 339, 2016, pp. 237-253.

[27] X. &. Z. Z. Zhang, "Chaos-based image encryption with total shuffling and bidirectional diffusion," *Nonlinear Dynamics,* vol. 75, no. 1, 2014, pp. 319-330.

[28] F. A. &. A. R. J. Petitcolas, "Evaluation of copyright marking systems," *Proceedings IEEE International Conference on Multimedia Computing and Systems,* vol. 1, 1999, pp. 574-579.

[29] J. &. A. F. Ahmad, "Efficiency analysis and security evaluation of image encryption schemes," *computing,* vol. 23, 2010, p. 25.

[30] M. a. K. V. Kaur, "A comprehensive review on image encryption techniques," *Archives of Computational Methods in Engineering,* vol. 27, no. 1, 2020, pp. 15-43.

[31] U. a. M. M. a. S. B. a. M. J. a. A. M. a. M. J. a. S. A. Zia, "Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains," *International Journal of Information Security*, 2022, pp. 1-19.