International Journal of Engineering and Technology Innovation, vol. 12, no. 2, 2022, pp. 167-181

# Lightweight Compressive Sensing for Joint Compression and Encryption of Sensor Data

Anil Kumar Chatamoni*, Rajendra Naik Bhukya

Department of Electronics and Communication Engineering, University College of Engineering, Osmania University, Hyderabad, Telangana, India

Received 01 October 2021; received in revised form 18 December 2021; accepted 19 December 2021

DOI: https://doi.org/10.46604/ijeti.2022.8599

## Abstract

The security and energy efficiency of resource-constrained distributed sensors are the major concerns in the Internet of Things (IoT) network. A novel lightweight compressive sensing (CS) method is proposed in this study for simultaneous compression and encryption of sensor data in IoT scenarios. The proposed method reduces the storage space and transmission cost and increases the IoT security, with joint compression and encryption of data by image sensors. In this proposed method, the cryptographic advantage of CS with a structurally random matrix (SRM) is considered. Block compressive sensing (BCS) with an SRM-based measurement matrix is performed to generate the compressed and primary encrypted data. To enhance security, a stream cipher-based pseudo-error vector is added to corrupt the compressed data, preventing the leakage of statistical information. The experimental results and comparative analyses show that the proposed scheme outperforms the conventional and state-of-art schemes in terms of reconstruction performance and encryption efficiency.

**Keywords:** sensor data, block compressive sensing, stream cipher, structurally random matrix, pseudo error vector

## 1. Introduction

In recent years, Internet of Things (IoT) has been developing at an exponential speed, impacting people's lives. IoT is a gigantic network of connected devices with a combination of different technologies and systems. Particularly, it is a network of "things" or "devices" used to transmit data among each other in the form of signal, text, audio, image, and video. The IoT network is mixed with distinct components like sensors, mechanical and digital machines, computing devices, Internet, etc. The total number of connected devices in IoT is increasing exponentially and will reach 75.44 billion by the end of 2025 [1].

The wireless sensor network (WSN) is among the imperative elements of the IoT network. A general WSN consists of a data processing center or fusion center and many resource-constrained distributed sensors. The sensor nodes transmit the collected observation data over the wireless channel to the fusion center. They have a low-powered limited transmission range. Since the data is transmitted wirelessly over insecure channels, it can be very easily intercepted by an attacker. Therefore, security and energy efficiency become the major concerns in IoT-based WSNs. The WSN technology is used to perform healthcare monitoring, surveillance, environmental or atmosphere monitoring, process control, performance monitoring, and emergency management in e-healthcare, military, environmental, industrial, home, and other commercial applications [2].

This research proposes a joint compression and encryption scheme with the use of block compressive sensing (BCS), structurally random matrix (SRM), and stream cipher [3]. SRM is used to generate the random measurement matrix. Then, the BCS technique is applied to perform the initial encryption and compression. Further, the pseudo-random error is added to the

* Corresponding author. E-mail address: anilkumarou2016@gmail.com

Tel.: +040-27098254

compressed measurement to ensure the final encryption. The encryption provides two-layer security by adding random errors and generating the measurement matrix. With the correct keys, the original image is reconstructed after performing the decryption and compressive sensing (CS) recovery processes.

The remaining sections of this study are organized as follows. Section 2 presents the literature review. BCS, SRM, and stream cipher are described in section 3. In section 4, the proposed lightweight CS scheme is explained with a detailed description. The experimental results and analyses of the proposed new scheme are discussed in section 5. In the end, section 6 provides the concluding remarks.

## 2. Literature Review

In WSN, multi-hop routing is used to deliver data from sensors to sink nodes. This increases the number of transmissions and consequently the energy consumption in the network. To overcome this issue, the data observed from different sensors should be compressed. The CS scheme is the best match to compress the gathered and transmitted data, wherein sampling and compression are done in one step [4-5].

Currently, to improve the industrial environment in Industry 4.0, IoT provides system security using cyber-physic systems (CPS). IoT has played a key role in the growth and development of Industry 4.0, especially in real-time monitoring and cybersecurity. A new IoT architecture is proposed for online status monitoring of gas-insulated switchgear (GIS) instead of the conventional observation methods [6]. Also, a novel IoT architecture is provided for secure and reliable online monitoring of induction motor status [7]. These architectures are utilized in modern machine learning techniques to detect cyber-attacks. New and improved security mechanisms based on machine and deep learning techniques need to be further developed.

Security is another concern because the data is wirelessly transmitted over insecure channels. Hence, energy efficiency and security become the main challenges in IoT-based WSNs [8]. CS can also be considered an encryption scheme when the measurement matrix is used as a key [9]. Consequently, joint encryption and compression of data can be achieved at the same time to increase energy efficiency.

Reducing the amount of stored and transmitted data and conserving the devices are the essential challenges in the growth of IoT systems. The majority of IoT applications had an independent implementation of data encryption and data compression. CS is the best choice for IoT devices due to its capability of joint compression and encryption. Recently, some lightweight CS schemes have been proposed for joint compression and encryption [10-11]. Bellasi and Benini [12] analyzed the energy efficiency of CS in wireless sensors, developed the power estimation models, and derived a framework for the total consumption of different CS architectures.

Recently, Kaur et al. [13] proposed a secure and energy-efficient model for e-health IoT networks. The secure transmission of biomedical images is achieved with CS and a hyper-chaotic map. CS is applied to the biomedical images to generate the compressed images. To encrypt the biomedical images, the compressed images are initially diffused and permutated row-wise to generate the scrambled images. Finally, the scrambled images are diffused and permutated column-wise to generate the encrypted images. Zhang et al. [14] proposed an enhanced CS-based data collection method in WSN by using asymmetric semi-homomorphic encryption to improve security. The sparse compressive matrix is used to reduce the computation cost.

The security and energy efficiency of lightweight CS-based schemes are highly dependent on the implementation of CS and the generation of measurement matrices. In recent studies, CS is replaced with BCS for making the scheme much more energy-efficient [15-16]. In BCS, an image is partitioned into non-overlapping blocks, and CS is applied independently on each block with a different or same measurement matrix.

Measurement matrices are categorized into two types: random and deterministic. Independent identically distributed (i.i.d.) Gaussian random matrices, partial Fourier matrices, and Bernoulli matrices are widely used as random measurement matrices. Circulant matrices, Toeplitz matrices, and binary matrices are used as deterministic measurement matrices. There are other methods to generate measurement matrices, such as chaos [17] and linear feedback shift register (LFSR) [18]. SRM is used for fast and efficient real-time CS applications which have merits like block-based processing and fast computation [19].

In CS, a measurement matrix is used for the compression of an original image. The measurement matrix can be used as a key in performing simultaneous compression and encryption of images. However, the simultaneous encryption and compression method using a measurement matrix as a key does not guarantee security. With the observation of information leakage and histogram analysis, the information about the original image is revealed, showing that CS compression schemes require additional encryption to accomplish confidentiality.

## 3. Preliminaries

### *3.1. Block compressive sensing (BCS)*

A signal acquisition and processing scheme which aims to sample and reconstruct a sparse signal below the Nyquist rate of sensor data is called CS. Using a measurement matrix ($\phi$) of size $m \times n$ ($m << n$), a high-dimensional sparse signal $x$ of size $n \times 1$ is converted into a lower-dimensional signal $y$ of size $m \times 1$. This dimensionality reduction procedure shown in Eq. (1) is called CS encoding or CS compression.

$$y_{m\times 1} = \phi_{m\times n} x_{n\times 1} \tag{1}$$

The accurate recovery of $x$ from the measurement vector $y$ is accomplished through the solution of the $l_1$-optimization problem, which is shown in Eq. (2).

$$x = \arg\min \|x\|_1 \text{ s.t. } y = \phi x \tag{2}$$

Most sensor signals are not sparse. To make them sparse, the signal of interest into another transform domain uses a sparsifying basis Ψ. Then, the signal recovery problem becomes:

$$x = \arg\min \|x\|_1 \text{ s.t. } y = \phi x = \phi \Psi^{-1} x \tag{3}$$

The two-dimensional signal recovery using Eq. (3) is a computational burden process. In BCS, the image acquisition is accomplished through the same operator in the block-by-block method. The image is divided into $c$ non-overlapping blocks of size $b \times b$, and CS is applied independently on each block with the following measurement matrix:

$$\phi_{BCS} = \begin{bmatrix} \phi_1 & 0 & 0 & 0 \\ 0 & \phi_2 & 0 & 0 \\ \cdots & \cdots & \ddots & \vdots \\ 0 & 0 & 0 & \phi_c \end{bmatrix} \tag{4}$$

The information carried out by the non-overlapping BCS empowers parallel sampling and reconstruction. Finally, image reconstruction is conducted through the simultaneous recovery of each block using Eq. (3).

The fundamental definition of CS is converting high-dimensional data into lower-dimensional data. This dimensionality reduction procedure generates the compressed data smaller than the original ones. These compressed data reduce the storage space when stored in a hard disk or memory card, and minimize the transmission cost when transmitted over the channel. Thus, by using the CS method, the storage space is reduced and the transmission cost is minimized.

### 3.2. CS with structural random matrix (SRM)

SRM is a novel method for fast and efficient CS. The measurement matrix using SRM is denoted as ($\phi_{SRM}$), and is a product of random permutation matrix ($R$), fast computable transform ($F$), and random subsampling matrix ($D$). The mathematical representation is as follows.

$$\phi_{SRM} = DFR \tag{5}$$

SRM pre-randomizes the signal of interest by multiplying it with the matrix $R$, and then applies a fast computable transform on a randomized signal to spread the information to all measurement coefficients. Finally, the operator $D$ is multiplied by randomly selected transform coefficients. The structure of SRM is depicted as:

$$\phi_{SRM} = \begin{bmatrix} D_1 & 0 & 0 & 0 \\ 0 & D_2 & 0 & 0 \\ \cdots & \cdots & \ddots & \vdots \\ 0 & 0 & 0 & D_c \end{bmatrix} \begin{bmatrix} F_1 & 0 & 0 & 0 \\ 0 & F_2 & 0 & 0 \\ \cdots & \cdots & \ddots & \vdots \\ 0 & 0 & 0 & F_c \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_c \end{bmatrix} \tag{6}$$

The advantages of SRM are block-based processing support, high computational speed, less complexity, and ease in hardware implementation.

### 3.3. Stream cipher

Stream cipher is a cryptographic method used to encrypt and decrypt the data based on a symmetric key. Stream cipher can encrypt plaintext messages of variable length into a ciphertext. The scheme of stream cipher is shown in Fig. 1. The keystream generator produces a pseudo-random sequence (keystream) using a secret key. The generated keystream bit is XORing with plaintext bit to create a ciphertext bit. At the end of this one-bit-at-a-time process, the ciphertext is generated, which is called stream cipher encryption. In the decryption process, the same original plaintext is regenerated from the ciphertext using the same key. Stream cipher is most suitable for high-speed and low-complexity operations.

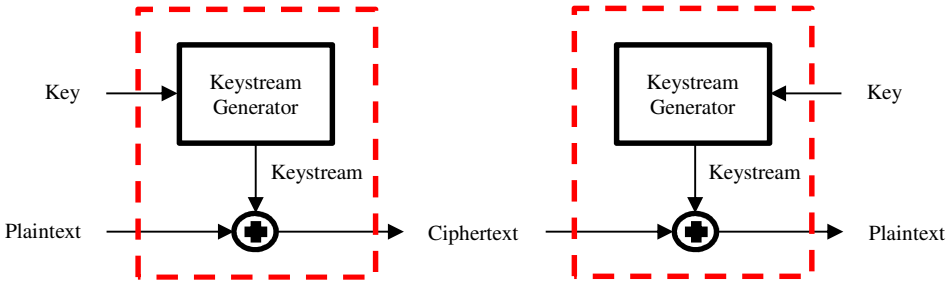



Fig. 1 Stream cipher scheme

## 4. Proposed Lightweight Scheme

In this section, the SRM-based BCS scheme is discussed, and a novel lightweight CS scheme is proposed. In the SRM-based BCS scheme, as shown in Fig. 2, a 32 × 32 block diagonal Walsh-Hadamard transform (WHT) is considered for the generation of the measurement matrix. The compressed bitstream is generated with CS projection and SRM. Due to the inherent property of CS, it yields the initial encryption. There is a possibility of information leakage because the linear combination of input samples urges the CS encoder to preserve some original image characteristics. To restrict the information leakage, the SRM-based BCS scheme requires additional encryption of compressed samples.

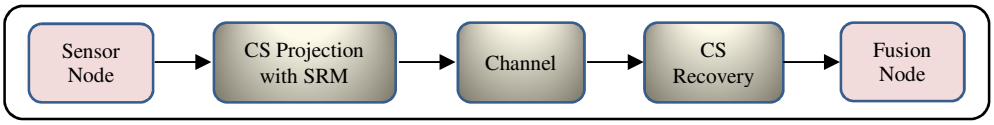



Fig. 2 SRM-based CS scheme

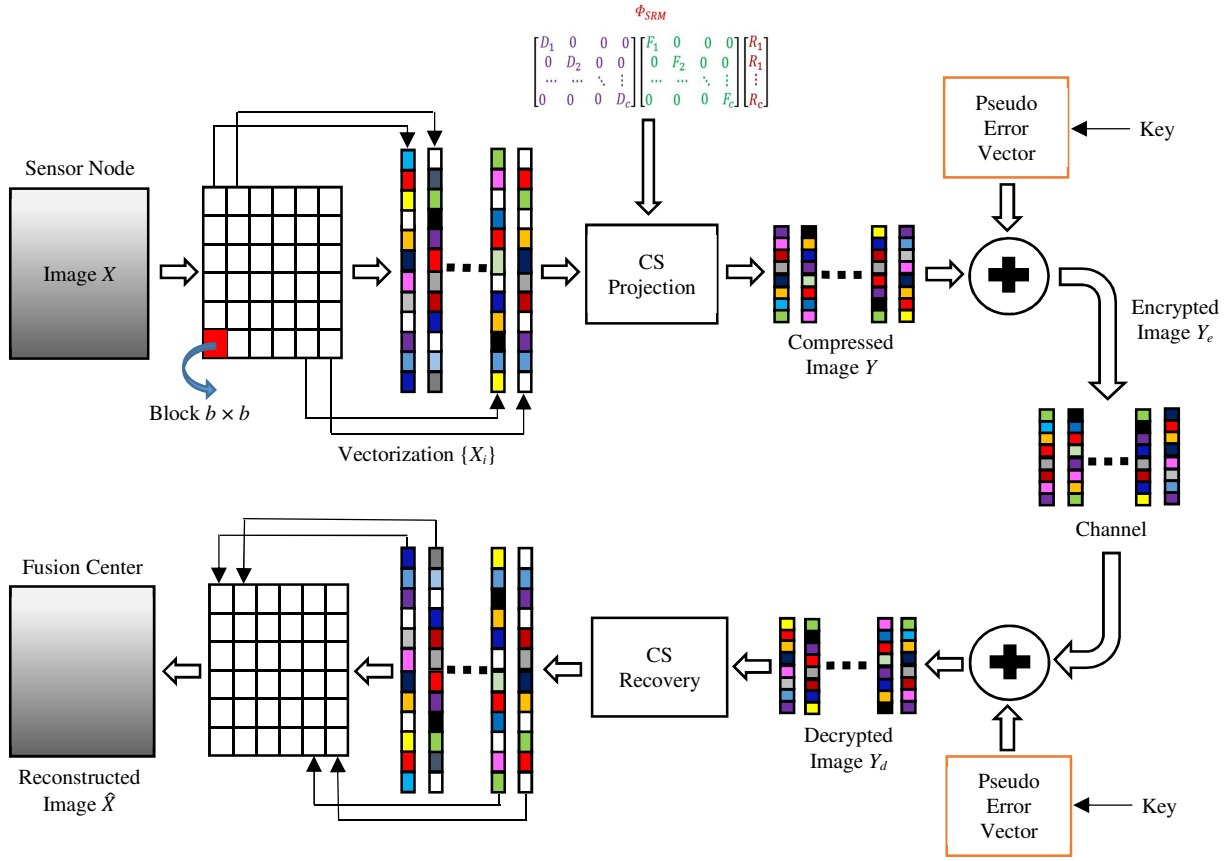



Fig. 3 Proposed lightweight CS scheme

The proposed lightweight CS scheme is shown in Fig. 3. The image captured from the sensor node is applied to BCS projection to generate the measured coefficients with the help of an SRM-based measurement matrix, which results in the initial encryption and compression. Further encryption is obtained by encrypting the measured coefficients with symmetric key-controlled stream cipher. The step-wise procedure of the proposed scheme is as follows.

***Step 1***: *Pre-process the captured image*

The captured image from the sensor node is considered the original image ($X$) of size $N \times N$, which is divided into equal and non-overlapping blocks of size $b \times b$. Then, the total number of blocks is $B = N^2 / b^2$. Each block is vectorized into a column vector of size ($b^2 \times 1$) of a preprocessed matrix of size ($b^2 \times B$).

$$X = \{x_i\} \text{ s.t. } i = 1, 2, \ldots, B \tag{7}$$

where $x_i$ is the coefficient vector of the $i^{\text{th}}$ block.

***Step 2***: *Generate the measurement matrix*

The random permutation matrix ($R$) and random sub-sampling diagonal matrix ($D$) can be generated using a secret key $K_1$. Each row and column of $R$ and $D$ have only one position of value 1. A $32 \times 32$ block diagonal WHT is used as a fast transform ($F$) to generate the measurement matrix. The size of $D$ determines the size of the $\phi$ matrix. SRM requires storing only the diagonals of $D$, $R$, and $F$ instead of the entire matrices, which greatly reduces the storage space and computational complexity. In this SRM method, each block is equally important, because of the random selection of sub-sampled signals from the original signal. The size of the measurement matrix ($\phi_{SRM}$) is $M \times b^2$, where $M = CR \times b^2$. $CR$ refers to the compression ratio (CR), which is the number of measurements over the total number of coefficients.

$$\phi_{SRM} = D_{M \times b^2} F_{b^2 \times b^2} R_{b^2 \times b^2} \tag{8}$$

***Step 3***: *Obtain the measured coefficient matrix through BCS*

BCS is performed by superimposing the coefficient vector on the measurement matrix, which results in the corresponding compressed measured coefficient vector of size $M \times 1$.

$$y_i = \phi_{SRM} x_i \tag{9}$$

where $y_i$ is the measured coefficient vector of the $i^{th}$ block. With the sequential procedure, at the end of all the blocks, a compressed image ($Y$) of size $M \times b$ is generated. The measured coefficient matrix is formed as:

$$Y = \{y_i\} \text{ s.t. } i = 1, 2, ....., B \tag{10}$$

This results in the compression and initial encryption of the original image. The histogram analysis in section 5.2.1 shows that the histogram of the compressed image is not uniformly distributed. In other words, CS measurements reveal some of the characteristics of the original image, which is why further encryption is required.

***Step 4***: *Encrypt the measured coefficients*

The compressed coefficients are encoded to generate a bitstream. The resultant bitstream is randomly corrupted through the addition of key-controlled pseudo-random error to get the uniform distribution of CS measurements. If $K_2$ is considered the key for pseudo error vector ($\eta$) generation, the individual pseudo-random error bits are XORed with the individual bits of the encoded measured coefficients, and the resultant encrypted vector stream is as follows.

$$\hat{y}_i = y_i \oplus \eta \tag{11}$$

Stream cipher is used to generate one row of the matrix at each clock cycle. After that, the encrypted image $Y_e$ is generated.

$$Y_e = \{\hat{y}_i\} \text{ s.t. } i = 1, 2, ....., B \tag{12}$$

This lightweight encryption process restricts the leakage of statistical information.

***Step 5***: *Decrypt the measured coefficients*

The same and opposite of the encryption process is produced by the decryption process. With the correct key $K_2$, the pseudo error vector is generated and XORed with the encrypted measured coefficients, resulting in the decrypted measured coefficients.

$$\tilde{y}_i = \hat{y}_i \oplus \eta \tag{13}$$

Then, the decrypted image $Y_d$ is generated as:

$$Y_d = \{\tilde{y}_i\} \text{ s.t. } i = 1, 2, ....., B \tag{14}$$

***Step 6***: *Recover the measurement coefficients*

The optimization problem defined in Eq. (2) is applied with the correct key $K_1$ on the decoded-decrypted measured coefficients to recover the measurement coefficient vectors of size ($b^2 \times 1$). Then, the size of the resultant recovered coefficient matrix is ($b^2 \times B$).

***Step 7***: *Reconstruct the original image*

The reconstructed image $\hat{X}$ of size $N \times N$ is generated by converting the recovered vector data into the corresponding blocks of $b \times b$ size.

The above step-wise procedure of the proposed scheme is represented with pseudo-code. The pseudo-code of the encoding process for estimating the encrypted image is summarized in Algorithm 1, and the pseudo-code of the decoding process for reconstructing the original image is summarized in Algorithm 2.

**Algorithm 1:** Encoding process

**Inputs:** The original gray image $X$ of size $N \times N$, keys $K_1$ and $K_2$, and block size $b$
**Initialize** $Y = [\ ]$, $Y_e = [\ ]$;
1: $B = N^2 / b^2$ Compute the total number of blocks
2: $X = im2col\ (X, [b\ b], 'distinct')$ Rearrange the image blocks into columns
3: $X = \{\ x_i\ \}$ $x_i$ is the coefficient column vector of the $i^{th}$ block
4: $Phi = DFR$ Calculate the measurement matrix using $K_1$
5: Calculate the compressed data segment:
**for** $i = 1: B$ **do**
$y_i = Phi \times x_i$ $y_i$ is the compressed column vector of the $i^{th}$ block
$Y(:, i) = y_i$ Compressed image
**end for**
6: Generate the pseudo error vector ($Eta$) using $K_2$
7: Estimate the encrypted data segment:
**for** $i = 1: B$ **do**
$y_{enc\_i} = xor(Y(:, i), Eta)$
$Y_e\ (:, i) = y_{enc\_i}$
**end for**
**Output:** The encrypted image $Y_e$

**Algorithm 2:** Decoding process

**Inputs:** The encrypted image $Y_e$, keys $K_1$ and $K_2$, and block sizes $b$ and $N$
**Initialize** $Y_r = [\ ]$, $Y_d = [\ ]$;
1: $B = size\ (Y_e, 2)$ Compute the column size
2: Generate the pseudo error vector ($Eta$) using $K_2$
3: Estimate the decrypted data segment:
**for** $i = 1: B$ **do**
$y_{dec\_i} = xor(Y_e(:, i), Eta)$
$Y_d\ (:, i) = y_{dec\_i}$
**end for**
4: $Phi = DFR$ Calculate the measurement matrix using $K_1$
5: $Y_r = l_1\ minimization\ (Y_d, Phi)$ Recovered image
6: $N = sqrt((B \times b^2))$
7: $X_r = col2im(Y_r, [b\ b], [N\ N], 'distinct')$ Rearrange the columns into image blocks
**Output:** The reconstructed image $X_r$

## 5. Experimental Results and Analyses

This section provides experimental results to assess the performance of the proposed scheme. The standard "Lena" image and some other images from the dataset Miscellaneous [20], with 8 bits/pixel grayscale of size 512 × 512, are used. The original image is partitioned into 32 × 32 size blocks. The total number of blocks becomes 256. Each block is converted into a 1024 × 1 vector and then formed into a pre-processed matrix of size 1024 × 256. A 128- bit secret key ($K_1$) is used to generate the diagonal elements of $D$ and $R$. The size of the SRM-based measurement matrix is 358 × 1024 with $CR = 0.35$ resulting in the compressed image of size 358 × 256.

The stream cipher that produces the bitstream is a good candidate for the generation of pseudo-error vectors. Among various stream ciphers, Bivium with a 128- bit secret key ($K_2$) is considered because of its high throughput and medium hardware complexity [21]. The iterative $l_1$ norm minimization algorithm is considered to restore the coefficient matrix of size

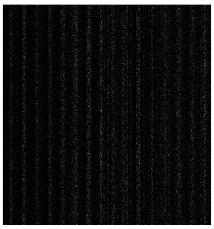
1024 × 256, and a 512 × 512 image is reconstructed with vector to block conversion. A set of simulations is carried out through MATLAB R2019a in Intel(R)Core(TM) i7-6700CPU, 64-bit windows 10 with 8 GB RAM desktop to validate the effectiveness of the proposed scheme.

A 512 × 512 pixels "Boat" image, as shown in Fig. 4(a), is considered the captured original image. After successful compression and encryption, the generated encrypted image of size 358 × 256 is shown in Fig. 4(b). The peak-to-signal noise ratio (PSNR) value is 31.6244 dB with $CR$ = 0.35. The encrypted image is corrupted and there is no correct information leakage of the original image. With the correct key value, decryption and decompression are performed to get the recovered image, as shown in Fig. 4(c). The proposed scheme has $2^{256}$ keyspace, which is adequate to resist brute attacks. The recovered image, which is not correlated with the original image, is shown in Fig. 4(d) and is obtained by a single-bit change in the key value of $K_2$. Hence, the proposed scheme is key-sensitive and provides better security.

(a) Original image

(b) Encrypted image with CR = 0.35 and PSNR = 31.6244 dB

(c) Reconstructed image with correct keys



(d) Reconstructed image with one-bit change in K2

Fig. 4 Experimental results on the "Boat" image

### *5.1. Performance analyses*

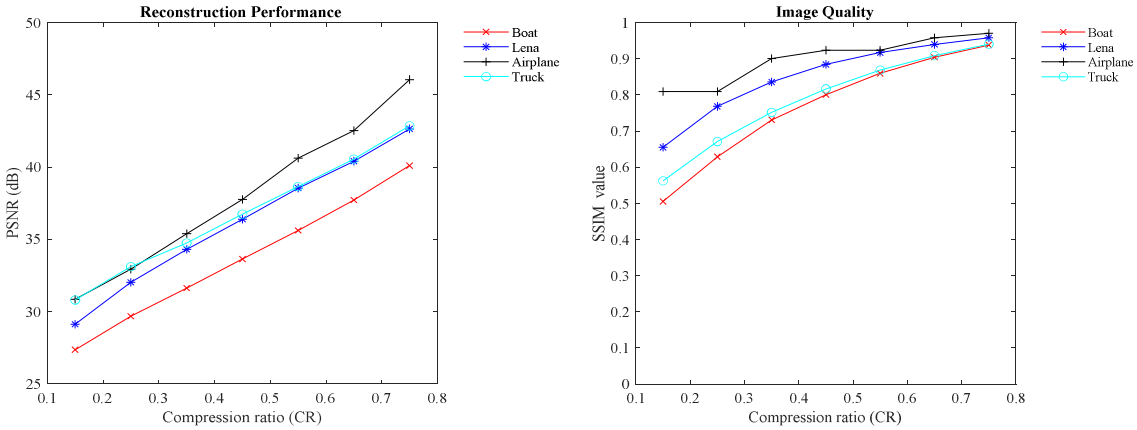
The performance analyses of the proposed scheme are measured in two metric values. PSNR is used to evaluate the reconstruction performance, and the structural similarity index measure (SSIM) is used to measure the image quality. The PSNR value is defined as:

$$PSNR = 20\log_{10}(255/\sqrt{MSE}) \tag{15}$$

where MSE is the mean squared error, which is estimated as:

$$MSE = \frac{1}{N \times N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left\|X(i,\ j) - \hat{X}(i,\ j)\right\|^2 \tag{16}$$

The graph shown in Fig. 5(a) represents the PSNR curve as a function of CR. The x-axis value represents CR and the y-axis value represents PSNR. The PSNR curves depicted in Fig. 5(a) represent the image recovery of different images with different CRs. It is observed that, with the least CR, the PSNR value is more than 25 dB, which is a competent performance value in image recovery. Hence, the proposed scheme has good image recovery performance with fewer measurements. SSIM is a quality measurement used to find the similarity between two images by comparing the resultant image with the reference image. The SSIM graph shown in Fig. 5(b) represents the reconstructed image quality of different images with different CRs. The value of SSIM is more than 0.5 in all cases, meaning that the proposed scheme has stable image reconstruction.




(a) PSNR (b) SSIM

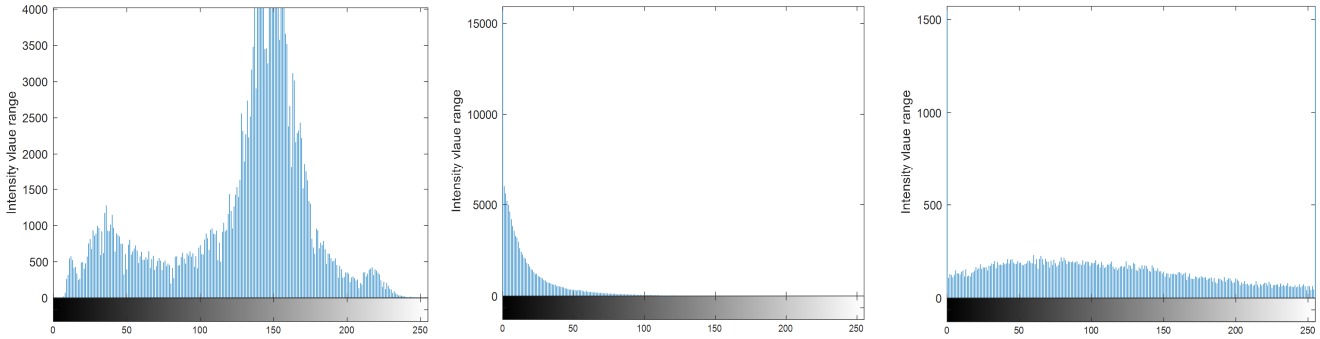
Fig. 5 Performance evaluation of the proposed scheme with PSNR and SSIM values

### 5.2. *Security analyses*

The reliability of the proposed scheme is verified with security analyses. The security analyses are examined in terms of the histogram analysis, correlation coefficient analysis between two adjacent pixels, information entropy analysis, diffusion analysis, and timing analysis.

#### 5.2.1. *Histogram analysis*

An image histogram represents the pictorial distribution of pixel intensities. The distribution pattern of pixels determines the quality of encryption. The peaks in the non-uniform histogram reveal the most information. The histogram of the plain image, the compressed image, and the encrypted image are shown in Figs. 6(a)-(c) respectively. The histogram of the compressed image contains peaks and it is easy for an eavesdropper to get any information, whereas the encrypted image histogram is uniformly distributed and does not contain any peaks, which restricts the statistical attacks.




(a) Original image (b) Compressed image (c) Encrypted image

Fig. 6 Histogram analysis of the "Boat" image

#### 5.2.2. *Correlation coefficient analysis*

Correlation coefficients (CC) are analyzed to measure the correlation between two adjacent pixels in horizontal, vertical, and diagonal directions. Theoretically, the CC value is very low (near 0) in an encrypted image and high (near 1) in a plain mage, which means that pixels are not close to each other in the encrypted image and are very close to each other in the plane image. The mathematical expression for the CC of two adjacent pixels is defined as:

$$CC = \frac{\frac{1}{S}\sum_{i=1}^{S}[p_i - E(p)][q_i - E(q)]}{\sqrt{\frac{1}{S}\sum_{i=1}^{S}[p_i - E(p)]^2}\sqrt{\frac{1}{S}\sum_{i=1}^{S}[q_i - E(q)]^2}} \tag{17}$$

$$E(p) = \sum_{i=1}^{S} p_i / s \tag{18}$$

$$E(q) = \sum_{i=1}^{S} q_i / s \tag{19}$$

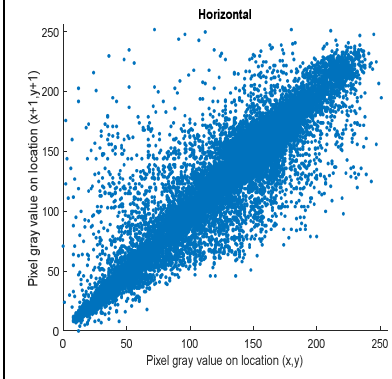
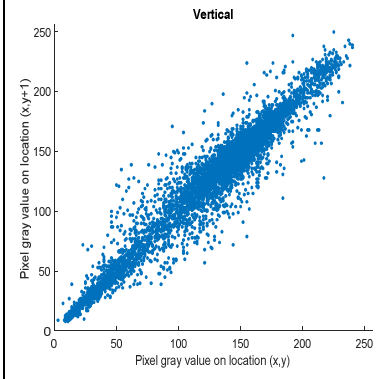
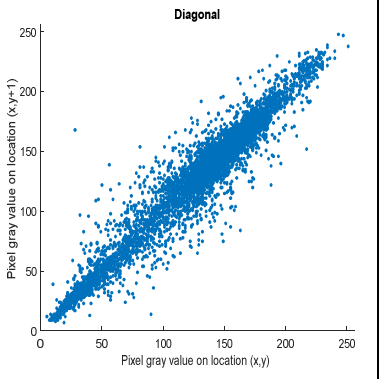
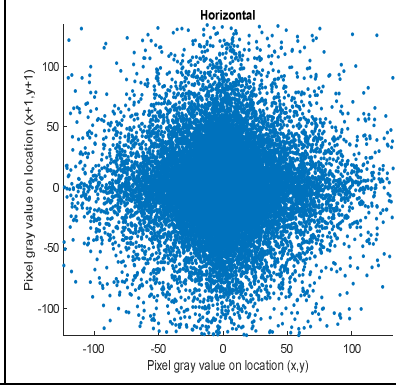
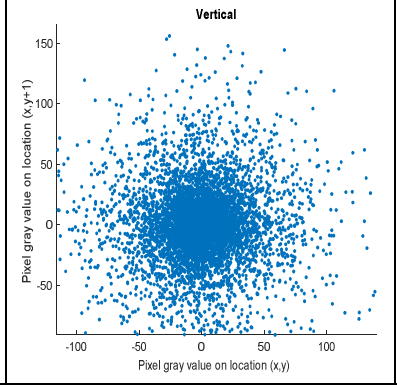
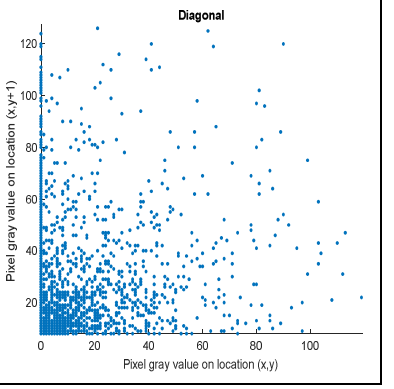
where $p$ and $q$ are grayscale values of two adjacent pixels. $S$ is the total number of selected adjacent pixel pairs. $E(p)$ and $E(q)$ are the mean values of $p_i$ and $q_i$ respectively.

8,000 pairs of neighboring pixels in each direction are randomly selected to calculate the CC values in both the encrypted image and its corresponding original image. The CC values of different original images and their corresponding encrypted images are tabulated in Table 1. From Table 1, it can be seen that the original images' CC values are very close to 1 and the encrypted images' CC values are very close to 0, which indicates that the proposed scheme can resist statistical attacks. Table 2 shows the pixel distribution of the original "Boat" image and encrypted "Boat" image in all three directions. The pixel distribution of the original image has a unique pattern, whereas the pixel distribution of the encrypted image has a uniform pattern in all three directions.

Table 1 Cross-correlation values of original and encrypted images in horizontal, vertical, and diagonal directions

| Image | Horizontal | | Vertical | | Diagonal | |
|---|---|---|---|---|---|---|
| | Original | Encrypted | Original | Encrypted | Original | Encrypted |
| Boat | 0.93711 | 0.00054 | 0.97182 | -0.00416 | 0.97827 | 0.00065 |
| Lena | 0.96627 | -0.00135 | 0.96896 | 0.00173 | 0.93571 | -0.00142 |
| Airplane | 0.95596 | 0.00171 | 0.95547 | -0.00261 | 0.93143 | 0.00065 |
| Truck | 0.98652 | 0.00035 | 0.95287 | 0.00072 | 0.91589 | -0.01258 |

Table 2 Pixel distribution of encrypted and original "Boat" images in horizontal, vertical, and diagonal directions

| Image | Horizontal | Vertical | Diagonal |
|---|---|---|---|
| Encrypted |  |  |  |
| Original |  |  |  |



*5.2.3. Information entropy analysis*

The information entropy is a statistical quantity of the pixel distribution used to evaluate the performance of the encryption scheme and it is expressed as:

$$E = -\sum_{j=0}^{2^k-1} p(A_j)\log_2 p(A_j) \tag{20}$$

where $p(A_j)$ is the probability of $(A_j)$. The standard value of entropy is 8 because $2^8$ possible values are presented in the image. The entropy values before and after encryption are presented in Table 3. From Table 3, the entropy of the encrypted image is close to 8, i.e., there is more randomness in the encrypted image as compared to the original image. Thus, the proposed scheme is secure against entropy attacks.

Table 3 Information entropy values of different original images and their corresponding encrypted images

| Image | Original image | Encrypted image |
|---|---|---|
| Boat | 7.2189 | 7.9915 |
| Lena | 7.8055 | 7.9997 |
| Airplane | 7.3298 | 7.9995 |
| Truck | 7.1532 | 7.9938 |

*5.2.4. Diffusion analysis*

The strength of an encryption scheme is measured with diffusion performance. The differential attack is used to determine the diffusion performance. The impact of a small change in the plain image on the corresponding encrypted image determines the ability of the proposed scheme to resist differential attacks. The difference between the original image $I_1(i, j)$ and encrypted image $I_2(i, j)$ is measured by the number of pixel change rate (NPCR) and unified average changing intensity (UACI). These values are defined as:

$$NPCR = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} D(i, j)\times 100\% \tag{21}$$

$$D(i, j) = \begin{cases} 0 & I_1(i, j) \neq I_2(i, j) \\ 1 & I_1(i, j) = I_2(i, j) \end{cases} \tag{22}$$

$$UACI = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} \frac{I_1(i, j) - I_2(i, j)}{255}\times 100\% \tag{23}$$

Theoretically, the NPCR and UACI values are 99.6094% and 33.4635% respectively. Table 4 shows that the NPCR and UACI values are near the theoretical values. This shows that the proposed scheme has better resistance to the differential attacks.

Table 4 NPCR and UACI values for plain image sensitivity test

| Image | NPCR | UACI |
|---|---|---|
| Boat | 99.6018 | 33.5523 |
| Lena | 99.6041 | 33.4898 |
| Airplane | 99.6095 | 33.4912 |
| Truck | 99.6063 | 33.5524 |

*5.2.5. Timing analysis*

In real-time, for processing images, the encryption scheme should be implemented as quickly as possible. The simulation is carried out twenty times, and the average execution time for the encryption process, decryption process, and total time is displayed in Table 5.

Table 5 Total execution time (in sec.)

| Image | Encryption | Decryption | Total |
|---|---|---|---|
| Boat | 2.1643 | 0.9276 | 3.0919 |
| Lena | 1.6351 | 1.0901 | 2.7252 |
| Airplane | 2.8074 | 2.2970 | 5.1044 |
| Truck | 2.8022 | 1.6296 | 4.4318 |

### 5.3. Average evaluation results across whole datasets

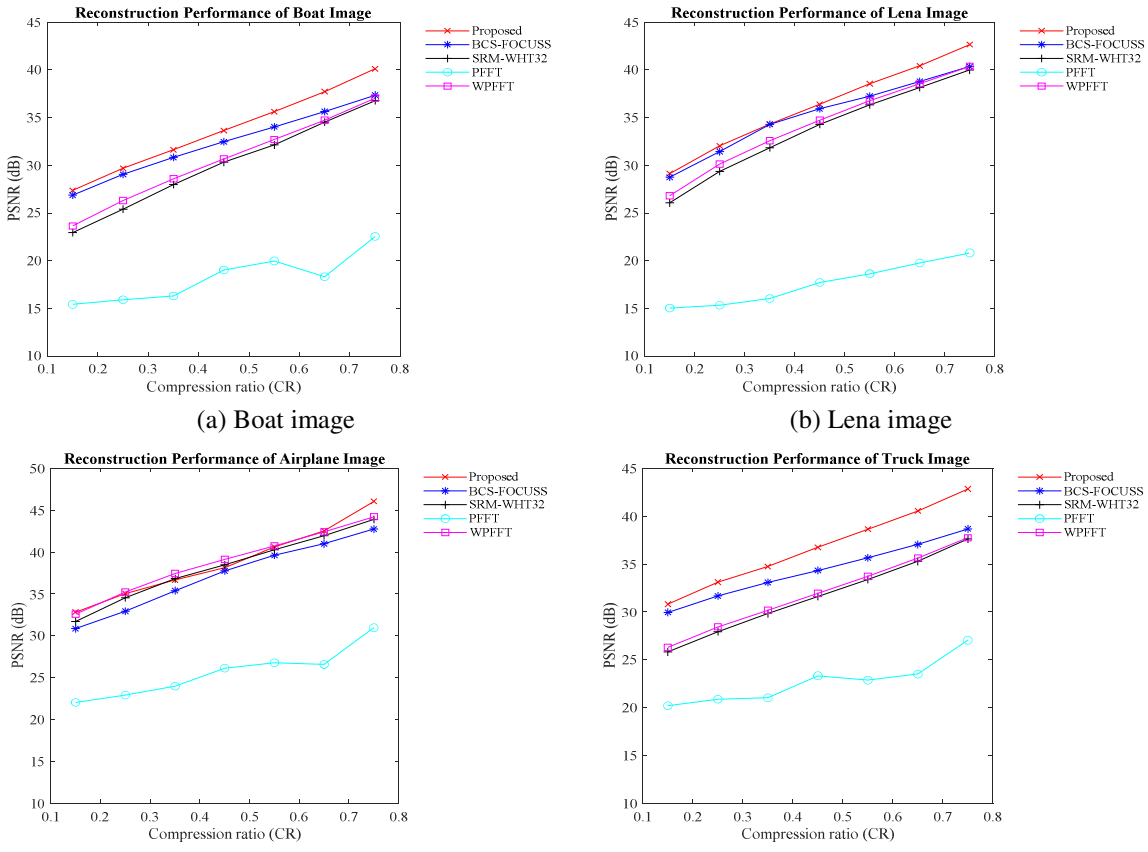
To verify the superiority of the proposed scheme, the evaluation metrics are measured across datasets [20]. Different datasets, namely Textures, Aerials, Sequences, and Miscellaneous, are considered. The simulation is carried out over 20 times to remove the randomness, and all the evaluation metrics as shown in Table 6 are averaged over all the images in the datasets.

Table 6 Evaluation metrics averaged over all images for different datasets

| Image dataset | PSNR | SSIM | Correlation coefficients | | | Information entropy | NPCR | UACI | Total execution time |
|---|---|---|---|---|---|---|---|---|---|
| | | | H | V | D | | | | |
| Textures | 28.4885 | 0.9380 | 0.00008 | -0.00015 | -0.00013 | 7.9968 | 99.6032 | 33.15 | 3.140483 |
| Aerials | 31.7218 | 0.9751 | 0.000645 | 0.00925 | -0.00410 | 7.9935 | 99.6089 | 31.48 | 3.502147 |
| Sequences | 30.7502 | 0.9535 | 0.000042 | -0.000001 | 0.00017 | 7.9918 | 99.6015 | 33.49 | 2.933631 |
| Miscellaneous | 30.3295 | 0.9894 | -0.00051 | 0.00172 | 0.00853 | 7.9925 | 99.6046 | 31.55 | 3.664448 |

### 5.4. Comparative analyses

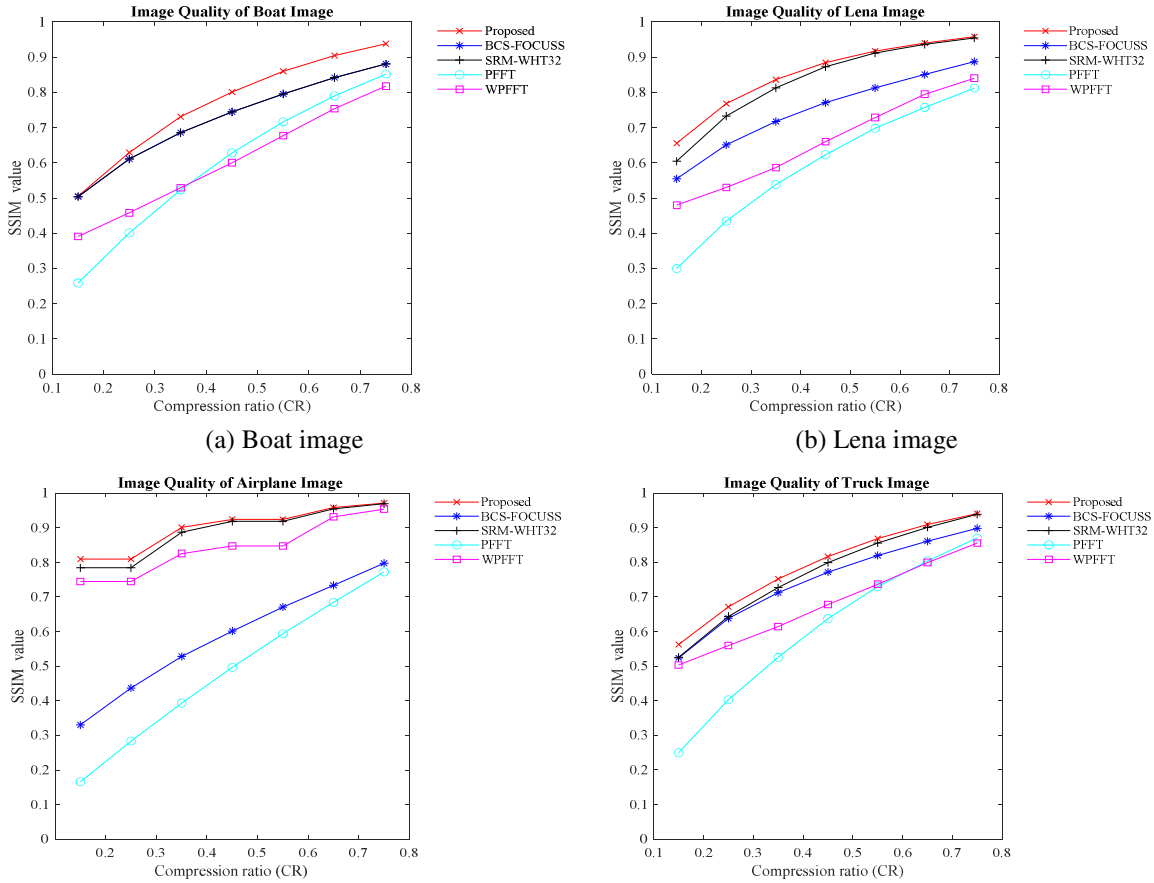
A comparison is made in terms of two aspects: reconstruction performance and encryption efficiency. In this comparative analysis, various approaches are compared with the help of an image database that has 512 × 512 size images like ‘‘Boat”, ‘‘Lena”, ‘‘Airplane”, and ‘‘Truck”. BCS is considered with non-overlapping and independently-sampled 32 × 32 size blocks, and different CR values are compared. In recent years, the most popular iterative image recovery algorithm, i.e., the BCS based on the smoothed projected Landweber (BCS-SPL) algorithm, had Wiener filtering for smoothness and hard thresholding operations for sparsity in every iteration. Information loss is possible due to hard thresholding, which results in a reduction in image recovery. The suggested BCS-focal underdetermined system solver (BCS-FOCUSS) algorithm is compared with the BCS-SPL algorithm with discrete wavelet transform (DWT), discrete cosine transform (DCT), and dual tree discrete wavelet transform (DDWT), resulting in significant performance gain and better reconstruction quality [16].




(a) Boat image (b) Lena image

(c) Airplane image (d) Truck image

Fig. 7 Comparison between the reconstruction performance of the proposed scheme and other schemes by using PSNR vs CR for images




(a) Boat image (b) Lena image

(c) Airplane image (d) Truck image

Fig. 8 Comparison between the image quality of the proposed scheme and other schemes by using SSIM vs CR for images

The proposed scheme is compared with other schemes wherein a random sensing matrix is generated with different implementations, such as partial fast Fourier transform (PFFT) in the time domain, PFFT in the wavelet domain (WPFFT), and SRM with 32 × 32 block diagonal WHT [18, 22].

The proposed scheme is compared with BCS-FOCUSS and other sensing matrix approaches in terms of PSNR and SSIM. The PSNR and SSIM comparison graphs using “Boat”, “Lena”, “Airplane”, and “Truck” images are shown in Fig. 7 (a)-(d) and Fig. 8 (a)-(d) respectively. With this comparison, the proposed scheme has better reconstruction performance. The proposed scheme is compared with other state-of-art encryption techniques using 512 × 512 “Lena” image to verify the level of encryption efficiency. A lightweight cryptosystem, based on a new chaotic S-box and advanced encryption standard, is explained in the highly constrained IoT devices scenario [23]. Khan and Munir [24] proposed a new technique for image encryption, wherein advanced encryption standard (AES) was extended to the Galois field of any characteristic. An encryption scheme with Lorenz system chaos-based logarithmic key generation as described by Tariq et al. [25] had reasonable digital multimedia security over the existing benchmark techniques.

Table 7 Encryption efficiency for different encryption schemes

| Algorithm | Correlation coefficients | | | Information entropy | Recovered image PSNR (dB) | NPCR | UACI |
|---|---|---|---|---|---|---|---|
| | Horizontal | Vertical | Diagonal | | | | |
| Proposed | -0.0013 | 0.0017 | -0.0014 | 7.9997 | 31.62 | 99.60 | 33.48 |
| [23] | -0.0636 | 0.0465 | 0.0669 | 7.9401 | - | - | - |
| [24] | -0.0147 | -0.1297 | 0.0027 | 7.9522 | - | 99.52 | 33.51 |
| [25] | −0.0026 | 0.0031 | −0.0043 | 7.9974 | - | 99.62 | 31.03 |
| [26] | 0.0053 | 0.0078 | 0.0042 | 7.9895 | 28.95 | 99.52 | 32.71 |
| [27] | -0.0022 | 0.0023 | 0.0034 | - | 31.15 | 99.56 | 33.45 |

An efficient, lightweight, and robust technique for image encryption using 2-D von-Neumann cellular automata (IEVCA) is suggested for IoT applications [26]. The meaningful image encryption scheme is implemented based on BCS and singular value decomposition (SVD) embedding [27]. The comparison made in terms of the CC in all three directions, information entropy, recovered image performance, NCPR, and UACI are provided in Table 7. Experimental results show that the proposed scheme has improved reconstruction performance and better encryption efficiency.

## 6. Conclusions

In this study, a novel lightweight CS scheme is proposed for simultaneous compression and encryption of sensor data in IoT. Firstly, BCS with SRM-based measurement matrix is applied to the captured image to generate the compressed and initial encrypted image. Then, a stream cipher-based pseudo-error vector is intentionally added to corrupt the compressed data, which yields further encryption. Finally, with the correct key values, the reconstructed image is obtained after successful decryption and CS recovery. The experimental results give the PSNR value in the range of 25-40 dB depending on the compression ratio. With the experimental results and analyses, the proposed two-layered lightweight encryption scheme shows improved reconstruction performance and better encryption efficiency compared to the conventional and other state-of-art methods. There is a possibility of reduced storage space and minimized transmission cost due to the usage of BCS. The proposed study is limited to the encryption of grayscale images. The proposed method can be applied to the encryption of color images with different sizes and further extended to video encryption. The proposed lightweight CS method provides a promising solution to be implemented on other security applications in the future, especially in IoT-based Industry 4.0 using advanced machine learning algorithms.

## Conflicts of Interest

The authors declare no conflict of interest.

## Acknowledgments

The authors wish to express their gratitude to Visvesvaraya Ph.D. Scheme, Ministry of Electronics and Information Technology (MeitY), Government of India, grant no. MEITY-PHD-1589 for funding this research work. The authors appreciate the comments from editors and anonymous reviewers for improving this study.

## References

[1] T. Alam, "A Reliable Communication Framework and Its Use in Internet of Things (IoT)," International Journal of Scientific Research in Computer Science, Engineering, and Information Technology, vol. 3, no. 5, pp. 450-456, May 2018.

[2] H. M. A. Fahmy, Concepts, Applications, Experimentation and Analysis of Wireless Sensor Networks, 2nd ed., Cham: Springer International Publishing, 2021.

[3] S. A. Jassim and A. K. Farhan, "A Survey on Stream Ciphers for Constrained Environments," 1st Babylon International Conference on Information Technology and Science, pp. 228-233, August 2021.

[4] D. L. Donoho, "Compressed Sensing," IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289-1306, April 2006.

[5] S. P. Tirani, A. Avokh, and S. Azar, "WDAT-OMS: A Two-Level Scheme for Efficient Data Gathering in Mobile-Sink Wireless Sensor Networks Using Compressive Sensing Theory," IET Communications, vol. 14, no. 11, pp. 1826-1837, July 2020.

[6] M. Elsisi, M. Q. Tran, K. Mahmoud, D. E. A. Mansour, M. Lehtonen, and M. M. F. Darwish, "Towards Secured Online Monitoring for Digitalized GIS Against Cyber-Attacks Based on IoT and Machine Learning," IEEE Access, vol. 9, pp. 78415-78427, March 2021.

[7] M. Q. Tran, M. Elsisi, K. Mahmoud, M. K. Liu, M. Lehtonen, and M. M. F. Darwish, "Experimental Setup for Online Fault Diagnosis of Induction Machines via Promising IoT and Machine Learning: Towards Industry 4.0 Empowerment," IEEE Access, vol. 9, pp. 115429-115441, August 2021.

[8] A. Salim, W. Osamy, A. M. Khedr, A. Aziz, and M. A. Mageed, "A Secure Data Gathering Scheme Based on Properties of Primes and Compressive Sensing for IoT-Based WSNs," IEEE Sensors Journal, vol. 21, no. 4, pp. 5553-5571, February 2021.
[9] M. Zhang, X. J. Tong, J. Liu, Z. Wang, J. Liu, B. Liu, et al., "Image Compression and Encryption Scheme Based on Compressive Sensing and Fourier Transform," IEEE Access, vol. 8, pp. 40838-40849, March 2020.
[10] G. Kuldeep and Q. Zhang, "A Novel Efficient Secure and Error-Robust Scheme for Internet of Things Using Compressive Sensing," IEEE Access, vol. 9, pp. 40903-40914, March 2021.
[11] Y. Zhang, Y. Xiang, L. Y. Zhang, Y. Rong, and S. Guo, "Secure Wireless Communications Based on Compressive Sensing: A Survey," IEEE Communications Surveys and Tutorials, vol. 21, no. 2, pp. 1093-1111, May 2019.
[12] D. E. Bellasi and L. Benini, "Energy-Efficiency Analysis of Analog and Digital Compressive Sensing in Wireless Sensors," IEEE Transactions on Circuits and Systems—I: Regular Papers, vol. 62, no. 11, pp. 2718-2729, November 2015.
[13] M. Kaur, D. Singh, V. Kumar, B. B. Gupta, and A. A. A. E. Latif, "Secure and Energy Efficient-Based E-Health Care Framework for Green Internet of Things," IEEE Transactions on Green Communications and Networking, vol. 5, no. 3, pp. 1223-1231, September 2021.
[14] P. Zhang, S. Wang, K. Guo, and J. Wang, "A Secure Data Collection Scheme Based on Compressive Sensing in Wireless Sensor Networks," Ad Hoc Networks, vol. 70, pp. 73-84, March 2018.
[15] S. Zhou, Y. He, Y. Liu, C. Li, and J. Zhang, "Multi-Channel Deep Networks for Block-Based Image Compressive Sensing," IEEE Transactions on Multimedia, vol. 23, pp. 2627-2640, August 2021.
[16] A. S. Unde and P. P. Deepthi, "Block Compressive Sensing: Individual and Joint Reconstruction of Correlated Images," Journal of Visual Communication and Image Representation, vol. 44, pp. 187-197, April 2017.
[17] A. K. Chatamoni, R. N. Bhukya, and P. R. Jeripotula, "A Novel Approach Based on Compressive Sensing and Fractional Wavelet Transform for Secure Image Transmission," International Journal of Intelligent Engineering and Systems, vol. 14, no. 4, pp. 11-21, August 2021.
[18] R. Dautov and G. R. Tsouri, "Establishing Secure Measurement Matrix for Compressed Sensing Using Wireless Physical Layer Security," International Conference on Computing, Networking, and Communications, pp. 354-358, May 2013.
[19] T. T. Do, L. Gan, N. H. Nguyen, and T. D. Tran, "Fast and Efficient Compressive Sensing Using Structurally Random Matrices," IEEE Transactions on Signal Processing, vol. 60, no. 1, pp. 139-154, January 2012.
[20] "The USC-SIPI Image Database," http://sipi.usc.edu/database/database.php?volume=misc, 1981.
[21] M. Soos, K. Nohl, and C. Castelluccia, "Extending SAT Solvers to Cryptographic Problems," International Conference on Theory and Applications of Satisfiability Testing, pp. 244-257, June 2009.
[22] E. J. Candès, J. Romberg, and T. Tao, "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information," IEEE Transactions on Information Theory, vol. 52, no. 2, pp. 489-509, February 2006.
[23] B. M. Alshammari, R. Guesmi, T. Guesmi, H. Alsaif, and A. Alzamil, "Implementing a Symmetric Lightweight Cryptosystem in Highly Constrained IoT Devices by Using a Chaotic S-Box," Symmetry, vol. 13, no. 1, 129, January 2021.
[24] M. Khan and N. Munir, "A Novel Image Encryption Technique Based on Generalized Advanced Encryption Standard Based on Field of Any Characteristic," Wireless Personal Communications, vol. 109, no. 2, pp. 849-867, May 2019.
[25] S. Tariq, M. Khan, A. Alghafis, and M. Amin, "A Novel Hybrid Encryption Scheme Based on Chaotic Lorenz System and Logarithmic Key Generation," Multimedia Tools and Applications, vol. 79, no. 31, pp. 23507-23529, August 2020.
[26] S. Roy, M. Shrivastava, U. Rawat, C. V. Pandey, and S. K. Nayak, "IESCA: An Efficient Image Encryption Scheme Using 2-D Cellular Automata," Journal of Information Security and Applications, vol. 61, 102919, September 2021.
[27] L. Zhu, H. Song, X. Zhang, M. Yan, T. Zhang, X. Wang, et al., "A Robust Meaningful Image Encryption Scheme Based on Block Compressive Sensing and SVD Embedding," Signal Processing, vol. 175, 107629, October 2020.