Compressed sensing based fingerprint imaging system using a chaotic model-based deterministic sensing matrix

Workneh Wolde Hailemariam · Pallavi Gupta

Received: 9 July 2021 / Revised: 25 March 2022 / Accepted: 2 July 2022 / Published online: 11 August 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
A secured compressed sensing (CS) systems design approach uses a novel deterministic sensing matrix to sense and transmit fingerprint images. The performance of the CS system was studied in detail by varying CS and security parameters. The sampling and sparse coefficient are the parameters considered from compressed sensing, whereas the encryption key is from the security scheme. The simultaneous compression and encryption has been achieved by multiplying the sparse modeled data with the proposed deterministic partial bounded orthogonal sensing matrix. A chaotic model-based permutation is applied to scramble the DCT matrix rows to build the sensing matrix. Recovering and decryption of the compressed image are accomplished with the help of the $L_1$ optimization method. The experimental test shows that a sparse vector of 121 widths has been recovered by taking about 25 samples. This indicates that up to 1 : 5 compression ratio is supported without damaging the fingerprint minutiae. If only compression is required without encryption, up to a 1 : 16 ratio can be achieved. The peak signal-to-noise ratio (PSNR) is 27.65 dB for both compression ratios under fulfilsments of all necessary security requirements. The 7.20 value of the entropy, histogram analysis, and the correlation analysis show the proposed scheme possesses adequate randomness. Furthermore, the ability of the system resistance against attacks is proved by 100% NPCR (Net Pixel Change Rate) and 0.92% UACI (Unified Average Changing Intensity) values.

Keywords Compressed sensing · Fingerprint image · Sparse representation · Optimization · Encryption · Chaotic model
1 Introduction

Nowadays, several facilities that serving the community are essentially becoming smart and IoT (Internet of Things)-based. As a result, those facilities need a tight authentication process to identify who accesses them. One of the well-known authentications methods is fingerprint sensing-based identity verification. In the past, several technological advancements have been made, and most of them were a sensors array-based that includes capacitive [26, 31, 38, 56], ultrasonic [30], and optical [35] MEMS (Micro-Electro-Mechanical System) sensors array with a whole sampling method to read fingerprint data. This sampling approach is implemented by reading and recording the outputs of all sensors’ element of the imaging device. Complete sampling is not efficient in terms of the optimal design approach. Such approach will lead to a high amount of energy-wasting during transmission.

Once any traditional or electronic technique records the fingerprint image, there are several well-known standards to process further or store them. For better visibilities, an image resolution of 1,000 PPI (pixel per inch) has been proposed [57]. This 1,000 PPI standard consumes more memory resources than the other standards, such as 500 PPI. Fingerprint database center like the national Automatic Fingerprint Identification Systems (AFIS) contains several millions of fingerprint records that consume nearly 100TB of memory resource [33]. Therefore, the rapid growth in storage requirement arose from storing those uncompressed fingerprint images.

Among fingerprint compression protocols that are currently implemented widely, JPEG (Joint Photograph Editor Group), JPEG2000, and WSQ (Wavelet Scalar Quantization) [24] are a few techniques to mention. However, in JPEG compression which is lossy, visibilities problems are reported in Ref. [25] due to its formulation that involves dropping out some components. Unlike the former standard, JPEG2000 [63] can be configured as either lossless or lossy compression standards. During image processing, both WSQ and JPEG2000 are use the Daubechies filter [7] to transform the signal. The observed loss is due to the transformation from time to frequency domain, plus some neglected transform coefficients during image processing. Hence lossy behavior regardless of the fingerprint minutiae number is one of the problems that limit their application.

Recently, a compression technique that works primarily for fingerprints by employing Multispectral technology is reported in Ref. [60]. This multispectral technology captures the images at different illumination conditions, wavelength, and orientation. On the other hand, sparse data representation-based compression technique is reported in Ref. [59]. The proposed method shows superiority in some databases and worsens over JPEG and JPEG2000. Sparse representation of data is the central idea applied in this work and will be discussed later. However, the shortcoming of this concept of compression is that it works for recorded images only, not during sensing process. In the multispectral-based technique, an optical fingerprint sensor is required. However, this technique has some security drawbacks as it may process a molded finger surface in case of illegitimate accesses.

Transmission of an image with a biometric feature should be secured and reliable. As a result of that, a cryptosystem is employed to store them for reproduction, authentication, or recognition. In the past, Pseudo-random number-based encoding methods were used for data encryption purposes. Some of them are phase encoding schemes by using joint transform [29], exclusive-OR encryption [28], and fractional Fourier transformation methods [72]. Additionally, encrypting fingerprint images using a fusion-based encryption scheme is also studied in Ref. [23]. Double stage chaotic biometric image encryption scheme by using two maps named Arnold (permutation), and Henon (substitution) for pixel shuffling is studied in
Ref. [46]. Like the previously studied schemes, this encryption again works independently with respect to sensing. Therefore, the lack of simultaneous recording, compression, and encryption is a significant drawback of those technologies to ensure optimality.

Most recently, an alternative method called secured compressed sensing (CS) systems has been used intensively to perform simultaneous compression and encryption on data to be sensed [61, 65, 70, 73, 74] for fingerprint and other images. In most CS-based image compression and encryption, sparse representation of image data has been used. Furthermore, either random or chaotic-based sensing matrix has been used for simultaneous encryption and compression to ensure security. Hence, all the success that has been seen in the above encryption and compression ability of the CS makes it a promising scheme in future secure image sensing and transmission. The main problem in the reported CS-based approach is that each and every element of the sensing matrix requires an independent chaotic computation that makes it computationally expensive.

In previous studies the concept of CS has been examined in quite a lot domain. The implementations and applications of CS includes compressive imaging, biomedical, communications, pattern recognition [52] and gait recognition [62]. However, the last application recently has been implemented using concept of IoT and deep learning neural network as reported in Ref. [1, 2]. The presented work focused on the CS application that involves the combination of image processing, communication system, and novel sensing matrix to sense, compress, and transmit fingerprint image.

To summarize, in fingerprint imaging, there are several steps in between sensing and storing or transmission. All the previously reported methods are focused on few aspects of fingerprint image processing. To mention some of their drawback, lack of compression in Ref. [57], lack of sensing method in Ref. [7, 24, 25, 63], in Ref. [23, 28, 29, 46, 72] only encryption of already recorded image, and in Ref. [61, 65, 70, 73, 74] the cost of computational work problems are clearly observed. All those problems initiated us to work more on optimized imaging scheme and that involves a simultaneous sensing, compression, and encryption in one side or before transmission. On the other or during receiving, a simultaneous decompression and decryption.

This paper proposes a novel scheme for sensing, secured transmission and the reception of fingerprint images. The proposed method uses a novel, chaotically structured, deterministic, and partial bounded orthogonal sensing matrix to successfully transfer and receive fingerprint images. The proposed system has been computationally implemented using an algorithm that governs the construction of the proposed sensing matrix via MATLAB script. We have developed a CS-based fingerprint imaging scheme by following the above-explained approach. The significant contributions that the proposed schemes can provide are stated below.

1. A novel chaotic model-based, partial bounded orthogonal, and deterministic sensing matrix is proposed, and
2. A new kind of fingerprint sensing and secured transmission scheme is proposed by utilizing the novel deterministic sensing matrix.

Hence the remainder of this paper is organized into seven sections, including this introduction part. Section two will discuss the theoretical background of compressed sensing. Following section two, in section three, the detailed methodology of designing the whole CS and its components like dictionary and sensing matrix with their algorithm is discussed. The experimental protocol for fingerprint data set organization, training, and evaluation also
explained in different parts of section three. The performance of the sensing matrix is studied by varying compressed sensing, and security threat model parameters in sections four and five. Additionally, the performance analysis based on comparison with existing methods in related work is summarized in section six. Finally, the whole work is summarized by pointing out the main contribution in the last section of the paper.

2 Background studies

The CS-based methodology works based on the concept developed by Candes et al. [12], and Donoho [17]. This formulation allows sampling from a few acquired data either from their native or transformed domain rather than from the whole elements in the sensing system. Following the transformation, signal compression and encryption can be possible by changing it from higher to lower-dimensional version by multiplying it with a measurement matrix [22].

The CS system needs some requirements for the input signal to fulfill before processed it. The signal must be sparse in its original or transform version. The signal’s sparsity in its transform version might come in two types. The first one, which is a widely used one, is obtained over some dictionary formed analytically by DCT and wavelet transformation of itself [5, 11]. The other one, which is less lossy, is coming over some learned dictionary designed based on prior knowledge about several correlated signals [4, 14, 34, 40, 41, 43, 47].

Furthermore, the Concept of CS provides an alternative approach to recover sparse signals from fewer measurements at a rate less than the Nyquist rate [12, 17]. A signal, which is sparse over a particular, orthogonal, and over completed dictionary matrix \( \Phi \), is mathematically related with its corresponding measurement signal, \( y \), according to (1).

\[
y = \Phi x
\]

In (1), \( x \in \mathbb{R}^{N \times 1} \), \( y \in \mathbb{R}^{M \times 1} \), \( \Phi \in \mathbb{R}^{M \times N} \), and \( M < N \).

According to Ref. [68], for a given integers \( k \) and \( N \) with \( k < N \), the sets of \( k \)-sparse, \( S_k \) vectors in \( \mathbb{R}^N \) define as

\[
S_k := \left\{ x \in \mathbb{R}^N : |\text{Supp}(x) \leq k \right\}
\]

Because of a non-inequality of the two integers \( M \) and \( N \), (1) yields an under-completed system of linear equations and lead to a non-unique solution. To get a unique solution, one has to define a constraint and apply it on (1) like the the following optimization problems \( L_p \)

\[
L_p : \quad \min \| x \|_p \quad \text{s.t.} \quad \hat{y} = \Phi \hat{x}
\]

\[
\| x \|_p = \left( \sum_{i=1}^{N} x_i^p \right)^{1/p}
\]

In both, (3) and (4), the range of \( p \) can be any value between 0 and 1, and the other acceptable value is 2. Equation (3) must be solved to get sparse vector \( \hat{x} \) from \( \hat{y} \) by solving \( p = 0 \) and \( p = 1 \) optimization problems. In this project, a well-known greedy type method known as Orthogonal Matching Pursuit (OMP) [49, 51] which is belongs to \( p = 0 \) optimization method is used.
In case of $L_0$ type optimization method, the solving process starts from $y$ followed by looking for a column of $\Phi$, which is most correlated with it. Once the correlated column is known, the resulting scalar product value of that column of $\Phi$ and $y$ itself will be taken as a reference to predict the next correlated column without repetition. According to [32], this algorithm is non-invariant type.

As can be observed, to achieve the goal of the $L_0$ optimization task, the dictionary matrix $\Phi$ must be known. The widely used method to build the dictionary matrix is the $K-SVD$ algorithm [4].

Now, a compressed sensing that is used to dimensionally compresses the measurement vector $y$ to $Y$ is mathematically defined as

$$Y = \Psi y$$ (5)

Where, $Y \in \mathbb{R}^{N' \times 1}$ is the compressed version of $y$ and $\Psi \in \mathbb{R}^{N' \times M}$ is a sensing matrix. Furthermore, to enable compression effectively, the sensing matrix $\Psi$ must be rectangular or $N' < M < N$.

If $y$ is not a sparse vector, the sparse representation of $y$ or $x$ by solving (1) must be used instead. Therefore, by plugging the sparse representation of $y$ in (5), the final equation for compressed sensing becomes,

$$Y = \Psi x$$ (6)

The formulation based on (5) directly works for communication, and natural signals compressible under Fourier basis [42]. However, non-compressible signal under Fourier basis, further signal processing is required to transform them to sparse signals so that (6) can be applied.

The sensing matrix denoted by $\Psi$ in (5) is a matrix that governs the selection of sparsely modeled signals using a predetermined order of the sampling process. A sensing matrix can be a random [16] or deterministic type [53]. Like the dictionary matrix, the sensing matrix is also under-completed. Hence, (5) needs another optimization approach to recover the sparse vector $y$ from $Y$ after compression. The $L_0$ type optimization problem, in this case, or recovery, is a non-deterministic polynomial hard or $NP-Hard$ problem. To leverage this hardness, transforming the problem to the $L_1$ type, which is given by (7), is the best option.

In addition to the sparsity of $x$, the sensing matrix $\Psi$ must satisfy the Restricted Isometric Property ($RIP$) (8) with the restricted isometric constant $\delta_S$ [10] for effective recovery of $x$ from $Y$.

$$\min \| x \|_1 \quad s.t. \quad \hat{Y} = \Psi \hat{x}$$ (7)

$$(1 - \delta_S) \| x \|_2^2 \leq \| \Psi x \|_2^2 \leq (1 + \delta_S) \| x \|_2^2$$ (8)

Among several already available approaches to solve (7), the prime dual interior point method is the most widely used. According to the prime dual method, the solution can be achieved by narrowing the dual gap between the feasible solution of the prime and dual problem. It starts from an arbitrary chosen initial point and goes in the specific direction...
to find the optimal solution by applying the classical Newton method \[13\]. This approach is computationally possible as free software package \[9\] are already available to solve the problem iteratively. Once the recovery of \(\hat{x}\) from \(\hat{Y}\) is completed, by multiplying \(\hat{x}\) with the dictionary matrix, the original measurement vector \(\hat{y}\) can be obtained.

### 3 The proposed method

The proposed method to derive the sensing matrix is mainly focused on the basic Discrete Cosine Transform (DCT) matrix which is given by \(9\) and \(10\) with a chaotic model-based row scrambling feature added to it via pseudo-random permutation. A chaotic system is a dynamic system that oscillates forever without even repeating itself or shows no tendency towards steady-state value \[45, 55\]. The proposed sensing matrix performance is studied after the derivation is fully completed.

#### 3.1 Proposed sensing matrix design for the compressed sensing system

For sensors array composed of \(M\) by \(N\) elements, the locations at which measurements have to be taken are stored in the proposed sensing matrix derived from a DCT matrix defined by \(9\) and \(10\).

\[
\Psi = \begin{pmatrix}
\Psi_{1,1} & \Psi_{1,2} & \cdots & \Psi_{1,N} \\
\Psi_{2,1} & \Psi_{2,2} & \cdots & \Psi_{2,N} \\
\vdots & \vdots & \ddots & \vdots \\
\Psi_{N,1} & \Psi_{N,2} & \cdots & \Psi_{N,N}
\end{pmatrix}
\]  

\[(9)\]

\[
\Psi_{k,n} = \sum_{n=1}^{N} \psi_n \cos \left( \frac{\pi}{N} \left( n + \frac{1}{2} \right) k \right) \quad k = 1, 2, 3, \ldots, N
\]

\[(10)\]

Equation 10 helps us to generate the required sensing matrix whose row width is not less than \(k \times \log(N)\) \[12\] by arranging selected rows according to the recommended sequence governed by our proposed algorithm. Then, a partial bounded orthogonal sensing matrix is selected at a high probability. A partial bound, orthogonal sensing matrix is a sub-matrix of the DCT matrix with the same number of columns with \(k \times \log(N)\) number of rows. The resulting partial bounded orthogonal matrix was found to obey RIP. Unlike \[36\], and \[27\], the scrambling feature was added before the derivation of the sensing matrix by swapping the rows of the DCT matrix. Compared to those schemes, the advantage is that this approach does not require building a new matrix to compress image patches securely.

The actual sequence of the DCT matrix row can be generated from the index of the Logistics map recursive equation. Logistics map is mathematically given by \(11\) with the population \(z_{t+1}\) at the index \(t\), the ratio of the existing population to the entire population \(z_t\), and growth rate \(r\).

\[
z_{t+1} = rz_t(1 - z_t)
\]

\[(11)\]
The value of $r$ is any value between zero and four and the value of $t$ varies from 1 to the maximum row size ($N$). The values of $z_{t+1}$ and $z_t$ are always between zero and one. As depicted in Figs. 1 and 2, the logistic map is highly chaotic when the value of $r$ is 3.57 or more. The detailed step that we followed to re-arrange the matrix rows sequence is summarized in Algorithm 1. According to the principle of probability, the resulting row combination has $1/N!$ probability of existence in the set of random permutations.

Algorithm 1: Chaotic model-based row index re-arrangement (scrambling) of the DCT matrix.

Input: Chaotic control parameters such as initial state $z_0$, growth rate $r \in (3.57, 4]$ and the $N$ by $N$ DCT matrix.

Output: Equivalent form of the DCT matrix.

1. Using the recursive logistic map (11) expression, generate a one-dimensional vector $[z_t]$ of size $N$.
2. Let $[N]$ be the index set $\{1, 2, 3, ..., N\}$ and their corresponding values of $[z_t]$ are $\{z[1], z[2], z[3], ..., z[N]\}$
3. Performs an ascending order operation upon the $[z_t]$ sequence which yields another new vector $[\hat{z}_t]$ with a new index set $[\hat{N}]$.
4. Iterate through the set $[\hat{z}_t]$ to obtain their corresponding index subset in $[N]$ and denote it by $[\mathcal{R}]$.
5. Perform a direct mapping from set $[N]$ to $[\mathcal{R}]$ or $([N] \leftrightarrow [\mathcal{R}])$ and elementary rows switching upon DCT accordingly.
6. Finally express the orthogonal equivalent matrix of DCT, by $\hat{\Psi}$.

The diagrammatical representation of the above algorithm is given in Fig. 2 for $r = 3.7$, $t = 121$ and initial state value 0.45.

Once the equivalent matrix is obtained, the parameters used, such as $N$, $r$, $t$, and $z_0$, would be taken as part of the encryption keys for the sensing as well as the recovery system. The sensing matrix obeys the RIP, if the smallest set $q (M, N)$ which formed by row selection from $\hat{\Psi}$ with high probability is equal to $O (M \times \log (N))$ [54]. A Multiplicative Linear Congruential Generator (MLCG) [8] based row sequence selection method with slight modification can be employed to extract the sensing matrix from $\hat{\Psi}$ at a high probability.

The modified MLCG sequence generator has been re-written including a shifting properties to it and mathematically expressed as.

$$k_{i+1} = S + (ak_i + b) \mod 2^{B_t} \quad 0 \leq a, b \leq 2^{B_t-1}, (M - N - 1) \leq S \leq N$$

(12)

Where $a$ is the multiplier, $b$ is the increment, $B_t$ is the number of bits, and $S$ is an integer used to extend the row index selection by shifting. Hence, by appropriately choosing the values $S$, we can generate several sensing matrices from $\hat{\Psi}$ without repetition.
Algorithm 2 Partial bounded orthogonal deterministic sensing matrix generation.

**Input:** Control parameters such as the shifting value $S$, the multipliers $a, b$, numbers of bit $B_i$, and $\hat{\Psi}$.

**Output:** a $M \times N$ size partial bounded orthogonal deterministic sensing matrix $\Psi$ and sequence of integer $M_\Psi$.

1. Using the modified MLCG (12) expression generate a sequence of integers $M_\Psi$ of size $M$.
2. Build a sub-matrix from $\hat{\Psi}$ by selecting rows combinations whose indices belongs to $M_\Psi$.
3. Define this sub-matrix as $\Psi$ which is the proposed deterministic sensing matrix given in (5).
4. Repeat steps 1 through 3 to generate others sensing matrices by using different parameters of MLCG.

By applying different combinations of $S, a, b, and B_i$ on the equivalent form of the DCT matrix that we already at hand, many partial bounded orthogonal sensing matrices can be generated. Now, $S, a, b, and B_i$ can be considered as an additional encryption key for the proposed CS-system.

Taking the outcome of the above two algorithms, the closed-form of the proposed sensing matrix becomes,

$$
\psi_{k,n} = \sum_{n=1}^{N} \psi_n \cos \left( \frac{\pi}{N} \left( n + \frac{1}{2} \right) k \right)
$$

$$
k \epsilon = [M_\Psi]
$$

Each of the sensing matrices generated by algorithm-II has an equal probability, which is equal to $1/(N-M)!$ to exist in $\hat{\Psi}$ as a sub-matrix. As anticipated, this is a relatively higher probability than the minimum value, $1/N!$ according to the RIP principle.

### 3.2 The theoretical computational complexity of the algorithms 1 & 2

The proposed algorithms that generate the sensing matrix perform mainly two tasks. The first one is row scrambling of the DCT matrix, and the second is the rows set extraction at high probability to construct the sensing matrix. For a DCT matrix of $N$ by $N$ size, there are $N!$ permutation options to scramble the matrix row sequence. Only $M$ numbers of rows are enough to construct the sensing matrix. And the total number of sensing matrix can be constructed became $N!/(N-M)!$.

Suppose the image size to be compressed and encrypted is $m \times m$ and one of the image patch sizes is $l \times l$, the total number of the image patch becomes $m/l \times m/l = L$. In order to compress the given $m \times m$ image, $L$ number of sensing matrices are required. There are many ways to generate this number of sensing matrices. Because of the careful selection of parameters in MLCG Ref [8, 21] to avoid unnecessary repetitions the number of times for which Algorithm 2 to run must be restricted. Let us assume the number of times for Algorithm 2 to run for each matrix generated by Algorithm 1 to extract $L$ sensing matrices is $u$. In this case, one of the parameters in Algorithm 2, such as $S$, must vary $u$ times. Now the number of times for Algorithm 1 to run becomes $L/u$. Therefore, Algorithm 2 must be run $u$ time to generate $L$ sensing matrix for every matrix generated by Algorithm 1.
Fig. 1 Logistic map function plot for the growth rate (GR) between 2.5 and 4: (a) Normal plot (b) Plot based on ascending order of the population
Fig. 2 DCT matrix row scrambling process according to algorithm-1: (a) step-1 and 2 (b) step-3 (c) step-4 and (d) step-5
Therefore, the total number of operations now becomes \( L/u \times L = L \) only, and it is the less complex computation as compared to the sensing matrix generation reported in Ref. [6, 58]. While doing all computation, only one dictionary matrix is enough as it is only used to transform input data to sparse. If the type of sensing matrix is a random matrix similar to the matrix proposed in Ref. [16], the number of computations will not be matrix row index wise; instead, the calculation must be done for every single entry of the sensing matrix. Consequently, the number of computations may be raised to \( m \times m = l^2 \times L \), which is significantly larger than \( L \).

### 3.3 Dictionary learning

As seen in the mathematical formulation of CS, at least one dictionary matrix is required to test the proposed sensing matrix works properly. A data store containing one hundred plain fingerprints (Fig. 3) collected from public database known as NIST database (https://www.nist.gov/itl/iad/image-group/nist-special-database-302) is organized to construct this dictionary matrix. 8 by 8, 16 by 16, and 24 by 24 overlapping image patches were generated from the data set. Not all patches are used to train the dictionary. Rather, we use one reference patch which is resides within at least one of the fingerprint image feature, and the rest are selected based on their SSIM [69] with respect to the selected reference patch.

According to Ref. [44], a typical fingerprint image has some minutiae characteristics, as depicted in Fig. 4a. The dictionary training has been done by taking those characteristics under consideration. Some of the features of fingerprint or minutiae are ridge ending, bifurcation, crossover, enclosure, etc. The applied comparison yields about 3000 fingerprint patches to train each dictionary. Then using the K-SVD algorithm, we trained three different dictionary matrices for purposes of signal transformation to sparse form. The plot of these matrices is shown in the figure Fig. 5.

The \( K - SV D \) algorithm solves optimization problem (3) iteratively. The algorithm starts by solving the \( L_0\)-problem to get the suitable coefficient matrix, \( X \). Since the method has used the matching pursuit to solve the \( L_0 \) problem, the resulting, \( X \) is a sparse signal. Keeping this \( X \) value as fixed and updating the column of \( D \) is the next step.

\[
\min \| Z - DX \|_F^2 \quad \text{s.t.} \quad \| x_i \|_0 < K
\]  

As can be seen, updating all the dictionary columns once is impossible. Only one column or k-th column can be updated from the fixed value of \( X \) vector by solving (15).

\[
\| Z - DX \|_F^2 = \left\| Z - \sum_{j=1}^{K} d_j x_j^T \right\|_F^2 = \left\| \left( Z - \sum_{j \neq k} d_j x_j^T \right) - d_k x_k^T \right\|_F^2 = \| E_k - d_k x_k^T \|_F^2
\]

Where, \( d_k \) is the k-th column of the dictionary to be updated, \( F \) represents Frobenius norm, and \( E_k \) which is equal to \( \left( Z - \sum_{j \neq k} d_j x_j^T \right) \). From (15), one can easily understand that \( d_k \) can be solved and updated by minimizing \( E_k \) using singular value decomposition. The trained dictionaries are plotted as shown in Fig. 5.
3.4 The workflow to study the performance of the proposed sensing matrix

The proposed method used to examine the performance of the proposed sensing matrix indirectly (because it is not possible to study its performance alone) is given by the block diagram in Figs. 6 and 7. The first step in the workflow is to divide the plain fingerprint image into several small non overlapping blocks or patches. Each image patch is then converted to a one-dimensional vector or $y$ followed by transformation to its equivalent sparse form or $x$. In the computation of sparse representation of the input signal, the dictionary matrix or $\Phi$ is required, which is already trained using the K-SVD algorithm. Once the sparse representation of the signal is completed, then it will be ready to go for compression.

On the other hand, the combination of the encryption keys and DCT matrix builds a sensing matrix $\Psi$ for the specific image patch being processed. The next critical stage is the simultaneous compression and encryption of the sparse form of selected image patch by multiplying it with the sensing matrix. Finally, the compressed and encrypted Cipher fingerprint image patch is ready from the sender side. The above process is only for a single image patch. It must be repeated until all image patches are compressed and transferred.
Fig. 4  (a) Features of fingerprint or minutiae (b) Reference patch coordinate identification
Recovery of the cipher image back to its original form will be done on the receiver side. However, the process on the receiver side is generally reverse of the sender; the sequence followed to recover the original signal is quite different.

The encryption key and the dictionary used must be transferred along with the whole cipher image blocks. It meant that, single set of encryption keys and dictionary matrix is enough for all sparse representation of image blocks. The prime dual interior point solver first performs the sparse recovery from the compressed cipher image data. Then the recovered sparse vector of the image patch is multiplied by the dictionary matrix to generate the original image patch.

As it can be observed, the proposed approach has two flexible features. The first one is the number of dictionaries used throughout the process. Only one dictionary and encryption key vector is required to transform all the patches of the given fingerprint image to their corresponding sparse and compressed form. The other is various options for the sparse representation of the image patch. Suppose the image is required to be fully converted; the whole proposed process needs to be completed. On the other hand, if it is required to be recorded, the compressed version of the sparse representation of the image is enough. In the latter case, by taking the non-zero entries of the image, one can store the compressed version of the sparse image data.

The simulation and the analysis work has been done by running the MATLAB 12a script of the proposed system on a machine built with Intel(R) Core(TM) i5-7500 CPU,
at 3.40GHz CPU speed, 1GB Intel(R) UHD Graphics adapter, and 4GB Random Access Memory (RAM). The dictionary is not part of this study because it is already proposed in the cited work. The elapsed time to carry out the above-detailed task is 18.897523 seconds.

4 Analysis and result for the proposed sensing matrix

This section presents the comprehensive experimental study of the proposed system using two sensing matrices. The first one is a random matrix, and the other is the proposed sensing matrix. The practical test to check the validity of the above numerical formulation has been done using one ample fingerprint image taken public data sources from the NIST database (https://www.nist.gov/itl/iad/image-group/nist-special-database-302).

4.1 Assessment parameters for CS system performance

Three assessment parameters were used to check the methodology’s effectiveness in transforming and securely transferring of the input signals. These are the Root Mean Square Error (RMSE), Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index of Image (SSIM). The mathematical expression for SSIM of an image \((a)\) with respect to its reference image \((b)\) of sizes \((N, N)\) [69] is

\[
SSIM(a, b) = \frac{(2ma_m + B_1)(2\nu_{ab} + B_2)}{(\nu_a^2 + \nu_b^2 + B_2)(m_a^2 + m_b^2 + B_1)}
\]  

(16)

Where \(m_a\) and \(\nu_a^2\) are the average and variances value of image \(a\) respectively. Similarly, \(m_b\) and \(\nu_b^2\) are the average and variances value of image \(b\) respectively. The other variable used in (16) \(\nu_{ab}\) is the covariance values of image \(a\) and \(b\). Furthermore, the remaining terms \(B_1\) and \(B_2\) are variables to stabilize the division with a leak denominator.

In this paper, apart from using as a parameter to study the scheme’s performance, the SSIM is used to select a set of image patch data to train the dictionary matrix. An image patch that imbedded within at least one fingerprint feature is designated as a reference to choose the other training image patches. The SSIM based selection can be applied by setting the threshold value and rejecting the signal whose structural similarity index value is far below the adjusted threshold.

The PSNR of \(m\) by \(n\) image with respect to reference image \(a\) of a similar dimension is

\[
PSNR(a, b) = 10 \times \log \frac{(PV)^2}{MSE(a, b)}
\]

(17)
Where $PV$ is the Peake Value of the pixel in the image and $MSE(a, b)$ is the Mean Square Error of the image with respect to reference image $a$ and expressed as

$$MSE(a, b) = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (a(i, j) - b(i, j))^2$$ (18)

4.2 Data modeling for CS input

Using the K-SVD algorithm, we construct the dictionary matrix to model the data with lengths 64, 256, and 576 by another data of 121,400 and 1089 length vectors with a few non-zero components, respectively. The method that has been used to get the compressible model of this data is employing the OMP algorithm upon the measurement data.

As depicted in Fig. 8a and b, our result will be better if we restrict the number of sparse coefficient at low. This result indicates that the methodology supports a high compressional ratio like 1:16, which is pretty good in terms of the possibility of a secured recovery.

To make it more clear, let say we have 8 by 8 pixel image patch which is one of the image block obtained after the whole image is divided in to several non-overlapping image blocks. The length of this image data was $8 \times 8 = 64$ before sparse modeling. However, the length of this image data would be raised to some higher value which is equal to 121 with four non-zero entries after sparse modeling applied. The advantage is however, by preserving the index or location of those non zero entries [59] compression is possible. In this case 4:64 or 1:16 compression ratio can be achieved from our data set if the interest of the user is purely compression. The compression is not lossless but as can be justified latter the number of minutia of fingerprint image will not be affected by this type of lossy compression.

4.3 Compressing and encryption

Compressing the sparse data is the easiest step in the proposed architecture. It is implemented by multiplying the sparsely represented input data by the proposed sensing matrix. This stage allows the encryption and compression of the sparse data to be accomplished simultaneously. As long as the sensing matrix is secret from third-party users, the compressed data remains safe from any access by external agents. The security key must be transferred securely instead of the whole sensing matrix for decryption and decompression from the sender side.

According to the workflow provided by the block diagram (Figs. 6 and 7), a simultaneous compression and encryption of the sparse data of the fingerprint image patch is achieved by simply multiplying the sparse modeled data by sensing matrix. The sparse data whose length was 121 would undergo a simultaneous compression and encryption when it is multiplied by a 25 by 121 size sensing matrix. Its size obviously reduced to 25 from 121 which is approximately 1:5 encrypted compression ratio. The advantage is however, no additional task is required for encryption as it already accomplished at the same time with compression.

4.4 Decompression and decryption

This step or simultaneous decompression and decryption is the reverse process of the CS and takes place at the receiver side. The sensing matrix keys, the encrypted and compressed data, and the dictionary matrix must be delivered to the receiver to carry out this task appropriately. Then the decompression and decryption are possible by numerically solving the
Fig. 8  RMSE, PSNR, and SSIM Vs number of sample plot for: (a) random sensing matrix and (b) proposed sensing matrix
\( L_1 \) optimization problem, (7). Solving the \( L_1 \) problem using the already described primal-dual interior-point method has been applied to recover the sparse data, not the original. The numerical solution of the \( L_1 \) problem is obtained by running the MATLAB script of \( L_1 \)-magic package [9]. Finally, the recovered sparse data must be multiplied by the dictionary matrix to get the original data according to (1).

Unlike the previous section, we fix the numbers of the sparse coefficient constant to study the rate of change of signal assessment parameters with the number of samples taken from the sparse signal. The result Fig. 9 shows that still, we have a high recovery rate at a small number of samples. The obtained results are better than the recommended compression rate for wireless transmission, which is between 20 and 25dB [37, 67]. Our conclusion from this analysis is that the proposed sensing matrix performance is almost the same as the random matrix, which possesses the RIP feature [10].

This experimental performance study includes both compressing and recovery of sparse data uses only one sensing matrix for all image patches for the sake of consistency of the overall research. The performance study has also been done separately due to the high computational cost and a bit longer time requirement complete it. Hence, its complexity analysis in terms of time does not included. Furthermore, the task has been done in batch mode to manage the long-time requirement to run the MATLAB script.

5 Security threat model and analysis

For the safe flow of data from one state to another or from sender to receiver, the communication must be secured by encrypting the data. This section will identify the expected potential threat that will impact the security of the proposed architecture.

The critical elements of the system responsible for the loss of data if potential adversaries access them are the dictionary, the sensing matrix, and the encryption keys. The last element cannot be a problem if the encryption keys are securely transferred to a legitimate user. However, the first two elements need further analysis because the attacker has several alternative ways to construct them. The dictionary matrix can be constructed using the same algorithm used in the proposed design from different training signals or fingerprint data. However, any other outputs from such a dictionary matrix may not be simultaneously sparse with the patch being sparsely modeled by the one used in the dictionary built by our data set. Alternatively, the sparse solution is always unique for the fixed input signal and coherent dictionary matrix once it is computed [3, 18]. Hence, our analysis will focus only on the sensing matrix.

When the attacker continuously sends a randomly chosen plane text or image to the oracle and gets back the cipher version of the text or image, there will be a probability for the attacker to gain a piece of knowledge about how the system encrypts its data. This attack is known as Chosen Plain text Attack (CPA). Like other works [48, 71], the proposed encryption scheme in this paper is related but not directly to the sensing matrix. As already pointed out in [6, 58], such a kind of compressed system is not secured against CPA. We can eliminate this vulnerability by effectively extracting different sensing matrices with high probability using the modified MLCG (12) from a single DCT matrix (9) which is designed based on a chaotic model (11). The performance of the sensing matrix from a security point of view will be discussed in the next section.
Fig. 9  RMSE, PSNR, and SSIM Vs number of sample plot for: (a) random sensing matrix nd (b) Proposed sensing matrix
6 Discussion on the performance and reliability of the proposed system

This section will identify the sensitive part of the system that makes it vulnerable to attack. Furthermore, it will explain how the system will respond by analyzing the security threat model and the reliability of the design approach.

We begin by demonstrating how the input data is compressed and then recovered diagrammatically in Fig. 10. Additionally, we have assured that the system does not damage most of the fingerprint minutiae by showing that it preserves most of them in Fig. 11. To generate output like Fig. 10, Algorithm 1 was used 128 times by incrementing the growth rate $r$, 128 times by fixed positive step value equal to 0.0020. Algorithm 2 was used 32 times for every single run of algorithm-I by incrementing the newly introduced shift constant $S$, 32 times with unit step value. Finally, a total of $128 \times 32 = 4096$ sensing matrices are generated using only eight parameters or keys. First, the 512 by 512-pixel image is divided into 4096 non-overlapping eight by eight image patches. Then, for each patch different sensing matrix is used to compress and encrypt it. The parameters used to generate all the 4096 sensing matrices are summarized in Table 1.

6.1 Randomness test via correlation analysis

The correlation analysis that has been performed on the proposed scheme involves the evaluation of the correlation between several randomly selected pairs of adjacent pixels that are aligned horizontally, vertically, and diagonally. For a particular fingerprint image with given pixel coordinates $(x, y)$ for each randomly selected pixel, the correlation expression is given as,

$$
Corr (x, y) = \frac{T \sum_{i=1}^{T} (x_i y_i) - \sum_{i=1}^{T} x_i \sum_{i=1}^{T} y_i}{\sqrt{\left( T \sum_{i=1}^{T} x_i^2 - \left( \sum_{i=1}^{T} x_i \right)^2 \right)^2 \left( T \sum_{i=1}^{T} y_i^2 - \left( \sum_{i=1}^{T} y_i \right)^2 \right)^2}}
$$  \hspace{1cm} (19)

Equation (19) is effectively utilized for total pixel pairs $T$ of 3000 from both plain and compressed images. The distribution plot is shown in Fig. 12. Comparison of the figure in

![Fig. 10](image_url) (a) The original, (b) the compressed also encrypted and (c) the recovered fingerprint image using sensing matrices-I parameters

\[ \text{ Springer} \]
plain and cipher versions of images shows that the cipher version is possessed the expected randomness.

6.2 Histogram analysis for encrypted image

The histogram analysis is helpful to identify which data of the securely compressed image is easily visible for the attacker. There are several entries in the proposed sensing matrix whose values are negative. Take this as a one advantage, and the nearly uniform distribution of values as depicted on the histogram comparison figure, Fig. 13, further indicates that the system is safe from histogram-based attacks.

6.3 Key sensitivity test

Key sensitivity test used to examine how the proposed sensing matrix responds to a slight change in the magnitude of some of the keys used to construct the proposed sensing matrix. Keeping other keys that affect the construction of the sensing matrix the same except one

| Sr. No | Parameter | Description                     | Value |
|--------|-----------|---------------------------------|-------|
| 1      | \( r \)   | Growth rate initial value       | 3.57  |
| 2      | \( N \)   | The DCT row width               | 121   |
| 3      | \( z_0 \) | Initial value for logistic map  | 0.5   |
| 4      | \( A \)   | MLCG Constant                   | 5     |
| 5      | \( k_1 \) | MLCG initial value              | 3     |
| 6      | \( b \)   | MLCG increment value            | 35    |
| 7      | \( B_t \) | Number of bits for MLCG         | 6     |
| 8      | \( \alpha \) | The Sensing matrix row width  | 25    |
Fig. 12  Correlation plots (a) for plain and (b) compressed fingerprint image (Note: The two bottom plots are for vertical and diagonal pairs of pixels)
testing parameter, we have performed an experimental test and observed how the sensing matrix row sequence generated. The test is performed first using the growth rate of the Logistic map and changed by $2.6333 \times 10^{-6}\%$. The experiment has been done for two sensing matrices key vectors, provided in Table 2. The first is the key set already used in Table 1, and the other is the same key set with a slight change in the growth rate of the Logistic map. The output assessment parameters show a high deviation from the expected value as depicted in the Table 2.

The deviation of the output from expected values is due to the difference in DCT matrix rows sequence as a result of the change in the growth rate parameter. The growth rate, which is equal to 3.7504, selects the rows of the DCT matrix according to the sequences of 33, 117, 5, 47, 95, 27, 108... for the first sensing matrix. On the other hand, the other growth rate value, which is equal to 3.7505, defer by only $2.6333 \times 10^{-6}\%$ produces the sequences of 34, 120, 2, 48, 96, 26, 106... for the second sensing matrix. Therefore, the partial bounded orthogonal sensing matrix with those selected rows from DCT produces different cipher data. The same precision has been used in all sequence and initial values.
Table 2  Key sensitivity test study

| Sr. No | Parameter | Type | Description                           | Value          | Sensing matrix-I | Sensing matrix-II |
|--------|-----------|------|----------------------------------------|----------------|------------------|-------------------|
| 1      | \( r \)   | Input| Growth rate initial value              | 3.5704         | 3.7505           |                   |
| 2      | \( N \)   | Input| The DCT row width                     | 121            | 121              |                   |
| 3      | \( z_0 \) | Input| Initial value for logistic map        | 0.5            | 0.5              |                   |
| 4      | \( A \)   | Input| MLCG Constant                         | 5              | 5                |                   |
| 5      | \( k_1 \) | Input| MLCG initial value                    | 3              | 3                |                   |
| 6      | \( b \)   | Input| MLCG increment value                  | 35             | 35               |                   |
| 7      | \( B_t \) | Input| Number of bits for MLCG               | 6              | 6                |                   |
| 8      | \( \Psi \) | Input| The Sensing matrix row width          | 25             | 25               |                   |
| 9      | PSNR      | Output| Peak Signal to Noise Ratio            | 27.6475        | 4.5839           |                   |
| 10     | SSIM      | Output| Structural Similarity Index of Image  | 0.9632         | 0.0295           |                   |
| 11     | RMSE      | Output| Root Mean Squared Error               | 0.0391         | 0.5115           |                   |

computations. This behavior of the sensing matrix shows that the proposed scheme is a key sensitive one.

The recovered image using sensing matrix-1 is already shown in Fig. 9, whereas the other image that uses sensing matrix-2 is given in Fig. 14. As clearly depicted in Fig. 14, there are no apparent fingerprint minutiae in the output image. It looks like a cipher image. Therefore, comparison of Figs. 9, 10, 11, 12, 13 and 14 show how the scheme is key-sensitive type one.

Furthermore, to check whether the choice encryption keys does not affect the optimality of the design, one more performance test also conducted for the rest of keys other than logistics map variables. The test that involves variation of the MLCG which is responsible for sensing matrices generation has summarized in Table 3 with same notation that used for Table 2.

![Fig. 14](image)

(a) The original, (b) the compressed also encrypted and (c) the recovered fingerprint image using sensing matrices-II parameters
Table 3  Performance study of the proposed CS system based on input parameters variation

|      | PSNR   | SSIM   | RMSE  |
|------|--------|--------|-------|
| $r = 3.5704$, $N = 121, A=5, k_1=3, b =35, B_t=6$, $\Re_l=25$, reference $z_0=0.5$ |
| $z_0$ |        |        |       |
| 0.5  | 27.6475| 0.9632 | 0.0391|
| 0.7  | 11.0989| 0.6586 | 0.1697|
| 0.9  | 4.8154 | 0.0329 | 0.5022|
| $r = 3.5704$, $N = 121, z_0=0.5$, $A=5, b =35, B_t=6$, $\Re_l=25$, reference $k_1=3$ |
| $k_1$ |        |        |       |
| 3    | 27.6475| 0.9632 | 0.0391|
| 5    | 4.4970 | 0.0321 | 0.5140|
| 7    | 4.6043 | 0.0269 | 0.5110|
| $r = 3.5704$, $N = 121, z_0=0.5$, $A=5, k_1=3, B_t=6$, $\Re_l=25$, reference $b=35$ |
| $b$  |        |        |       |
| 35   | 27.6475| 0.9632 | 0.0391|
| 33   | 4.7310 | 0.0262 | 0.5025|
| 37   | 4.7140 | 0.0308 | 0.5028|
| $r = 3.5704$, $N = 121, z_0=0.5$, $A=5, k_1=3, b =35, B_t=6$, reference $A=25$ |
| $A$  |        |        |       |
| 5    | 27.6475| 0.9632 | 0.0391|
| 2    | 4.4911 | 0.0341 | 0.5009|
| 8    | 4.1463 | 0.0394 | 0.5103|

6.4 Key space of the proposed system

Based on our study about the brute force security attack occurrence that has been seen in the ECG signal encryption [15], a repeated trial must be done in our proposed scheme to see the effect of this attack. This means that the adversary must construct the sensing matrix by combining rows using its technique with a significantly very high number of trials. According to [27, 48], the attacker must attempt a maximum of $2^{100}$ rows combinations of the DCT matrix, which is supposed to exceed the keyspace of the system. In this regard, the keyspace of one of the 4096 proposed sensing matrices can be computed from the DCT matrix with row dimensions 121. 25 of them is enough to construct a single sensing matrix.

$$Key\ Space = (121!) \times \left(\frac{121!}{25! (121 - 25)!}\right) = 42.55^{255} \cong 2^{1225} \gg 2^{100} \quad (20)$$

The term under the first bracket is the number of options for an attacker to construct the masked DCT, whereas the second matrix is the number of options to construct the sensing matrices, and their product gives large enough value to turn the brute force attack into infeasible for the proposed system. Since there is an additional 4095 sensing matrices, the computed keyspace value might be higher.

6.5 Differential attack

Types of attack that we have already studied before requiring analysis involve the whole signal or image at a time. However, differential attack analysis is more focused on a specific
single pixel. Therefore, we only take one of the 4096 image contains the pixel that we want to study. Because different sensing matrices were used to resist brute force and chosen-ciphertext treat. Therefore, our analysis will be based on a selected specific block using NPCR (Net Pixel Change Rate) and UACI (Unified Average Changing Intensity), which are given by (21) and (22) respectively.

\[ \text{NPCR} = \frac{\sum_{i,j} D(i, j)}{H \times W} \times 100\% \] (21)

\[ \text{UACI} = \frac{1}{H \times W} \sum_{i,j} \left| c(i, j) - c'(i, j) \right| \times 255 \times 100\% \] (22)

Where \( H \) and \( W \) are the height and width of the image block. And \( c(i, j) \) is the plain image block whereas \( c'(i, j) \) also the encrypted image of the block with one of the pixels changed. The value of \( D(i, j) \) is one if there is pixel difference between \( c(i, j) \) and \( c'(i, j) \) otherwise it is zero. Our samples taken from eight by eight blocks are 100\% for NCPR and 0.92\% for UACI. The result shows that all sixty-four-pixel values undergo changes that make the design scheme superior against others scheme by ensuring resistive to any attack.

6.6 Entropy analysis

The entropy of our compressed system is the measurement of the system’s ability to generate random encrypted output. The entropy \( H \) of the image with probability \( P \) and total number \( N \) is mathematically expressed and given by (23).

\[ H = \sum_{i=1}^{N} P_i \log_2 P_i \] (23)

Since our system is designed based on the DCT matrix, many entries have negative values. Hence to compute the logarithmic content of (23), we took the absolute value of the cipher image, and we have verified that the system has 7.20, which is 90\% of the ideal value.

6.7 A comparative analysis of the proposed sensing matrix and its application

This section summarizes different CS system types and their corresponding sensing matrix used to build them. As mentioned in this work, a CS system can perform simultaneous compression and encryption of signals. Sensing matrices, in general, are of two types. The first is a random matrix, and the other is a deterministic sensing matrix. Random sensing matrices need many memory resources to store, but they are easy to generate. The drawback of this sensing matrix is that there is no efficient algorithm for verifying its RIP conditions. Some of the sensing matrices discussed in this section are deterministic sensing matrices.

Although Ref. [16] describes the derivation of a sensing matrix known as Structurally Random Matrix (SRM), which closely works like the presented work, no evidence shows that the resulting sensing matrix obeys RIP. However, the proposed matrix in Ref. [16] is still can be used for a CS system without involving a security feature. The same type of sensing matrix has been used in the compressed sensing system design presented in [19]. Those sensing matrices cannot be applied if the desired system needs a security feature. Because, as already discussed in the system’s algorithmic complexity, such metrics must be generated in high numbers that match the number of image patches to be encrypted.
While Ref. [22, 39, 50, 64, 66] uses a predictable random or pseudo-random matrix, unlike the previous two systems, they are generated from hardware like FPGA (Field Programmable Gate Array) and microcontroller. Since they are deterministic, they can generate multiple sensing matrices. This ability of the sensing matrix generator makes them suitable to build a secured CS system. However, slight limitations that already mentioned in that work, on the pseudo-random number generator like the Linear Feedback Shift Register (LFSR) will make them less reliable for fingerprint images. If the current state of the LFSR is known, the next state is easily predictable. That would probably affect the scheme’s security that uses LFSR for sensing matrix generation.

Chaotically generated sensing matrixes are the other option for a deterministic sensing matrix generation. Ref. [20, 27] uses that approach to generate chaotically controlled sensing matrices. The sensing matrix generated by the chaotic system has proven to obey RIP as mentioned in Ref. [64] and has no issue with the recovery of sparse signal compressed by such sensing matrices. This makes them suitable to construct an effective CS system like fingerprint imaging. However, their drawback is that to resist a security trait, the sensing matrix must be repeatedly generated in an element-wise fashion, and the respective keys must be transmitted with every signal component to be encrypted. This shows that this matrix construction approach is a little bit expensive computationally. This weakness has been eliminated by a partial bounded orthogonal sensing matrix derived from a chaotically scrambled DCT matrix in the presented work.

In the presented work, unlike Refs. [20, 27], element-wise computation of sensing matrix is not applied. By combining chaotic sequence with pseudo-random number generator, MLCG, a partial bounded orthogonal sensing matrix extracted by selecting the specific set of rows of DCT matrix accordingly. In the analysis part, we have demonstrated that multiple sensing matrices are generated for compression and recovery that take place at the senders and receivers side, respectively. The compressed sensing system tasks are completed using about eight parameters plus the dictionary matrix.

7 Conclusion

This work aimed at an optimized compressed sensing and secured transmission of fingerprint images. We have designed a novel deterministic sensing matrix based on a partial bounded orthogonal matrix derived from a (DCT) matrix to establish security. A novel and computationally low complex algorithmic approach has been proposed to construct a sensing matrix that fulfills RIP conditions. A modified MLCG has been employed to build the sensing matrix by extracting them from the chaotically scrambled rows of the DCT matrix at confirmed high probability. The comprehensive simulation-based analysis of our proposed compressed sensing system shows that the deterministic sensing matrix performs equally with the non-deterministic or random matrix. The results obtained from the simulation of the proposed system confirmed that a better PSNR than the recommended value for wireless transmission is archived using a sample below 20% without losing a significant number of fingerprints minutiae. The response of the designed CS system was also examined using various security threat models and found to be resistive against any potential adversaries. The randomness test has been performed using correlation analysis and entropy calculations on the proposed system. On the other hand, 100% for NCPR and 0.92% for UACI values prove the scheme’s resistance against different kinds of security attacks.
**Author Contributions** Workneh Wolde Hailemariam: Design methodology, simulation work, writing and editing the manuscript draft. Pallavi Gupta: supervision.

**Declarations**

**Conflict of Interests** The authors declare that they have no conflict of interest.

**References**

1. Achanta SDM, Karthikeyan T, Vinoth Kanna R (2020) A wireless IOT system towards gait detection technique using FSR sensor and wearable IOT devices. Int J Intell Unmanned Syst 8(1):43–54. https://doi.org/10.1108/IJJUS-01-2019-0005
2. Achanta SDM, Karthikeyan T, Vinothkanna R (2019) A novel hidden Markov model-based adaptive dynamic time warping (HMDTW) gait analysis for identifying physically challenged persons. Soft Comput 23(18):8359–8366. https://doi.org/10.1007/s00500-019-04108-x
3. Aharon M, Elad M, Bruckstein AM (2006) On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them. Linear Algebra Appl. https://doi.org/10.1016/j.laa.2005.06.035
4. Aharon M, Elad M, Bruckstein A (2006) K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans Signal Process 54:4311–4322. https://doi.org/10.1109/TSP.2006.881199
5. Allen JB, Rabiner LR (1977) A unified approach to short-time fourier analysis and synthesis. Proc IEEE 65:1558–1564. https://doi.org/10.1109/PROC.1977.10770
6. Alvarez G, Li S (2006) Some basic cryptographic requirements for chaos-based cryptosystems. Int J Bifurc Chaos. https://doi.org/10.1142/S0218127406015970
7. Antonini M, Barlau M, Mathieu P, Daubechies I (1992) Image coding using wavelet transform. IEEE Trans Image Process 1(2):205–220
8. Bakiri M, Guyeux C, Couchot JF, Oudjida AK (2018) Survey on hardware implementation of random number generators on FPGA: theory and experimental analyses. Comput Sci Rev 27:135–153. https://doi.org/10.1016/j.cosrev.2018.01.002
9. Candès E, Romberg J (2005) 1-magic: Recovery of Sparse Signals via Convex Programming. Caltech. https://candes.su.domains/software/l1magic/atlast. Accessed 10 Dec 2021
10. Candès EJ (2008) The restricted isometry property and its implications for compressed sensing. Comptes Rendus Math. https://doi.org/10.1016/j.crma.2008.03.014
11. Candès EJ, Donoho DL (2004) New tight frames of curvelets and optimal representations of objects with piecewise C 2 singularities. Commun Pure Appl Math 57:219–266. https://doi.org/10.1002/cpa.10116
12. Candès EJ, Romberg J, Tao T (2006) Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Tran Inf Theory. https://doi.org/10.1109/TIT.2005.862083
13. Chao L, Han J, Yan L, Sun L, Huang F, Zhu ZB, Wei S, Ji H, Ma D (2020) Fast compressed sensing analysis for imaging reconstruction with primal dual interior point algorithm. Opt Lasers Eng 129. https://doi.org/10.1016/j.optlaseng.2020.106082
14. Chen G, Mazzioni M (2010) Multiscale geometric wavelets for the analysis of point clouds. In: 2010 44th Annual Conference on Information Sciences and Systems, CISS. pp 1–6. https://doi.org/10.1109/CISS.2010.5464843
15. Djelouat H, Amira A, Bensaali F, Boukhennoufa I (2020) Secure compressive sensing for ECG monitoring. Comput Secur 88:101649. https://doi.org/10.1016/j.cose.2019.101649
16. Do TT, Gan L, Nguyen NH, Tran TD (2012) Fast and efficient compressive sensing using structurally random matrices. IEEE Trans Signal Process 60(1):139–154. https://doi.org/10.1109/TSP.2011.2170977
17. Donoho DL (2006) Compressed sensing. IEEE Trans Inf Theory. https://doi.org/10.1109/TIT.2006.871582
18. Elad M, Bruckstein AM (2002) A generalized uncertainty principle and sparse representation in pairs of bases. IEEE Trans Inf Theory. https://doi.org/10.1109/TIT.2002.801410
19. Fan H, Li M, Mao W (2017) VQ-Based compressive sensing with high compression quality. Electron Lett 53(17):1196–1198. https://doi.org/10.1049/el.2017.1321
20. Fan H, Zhou K, Zhang E, Wen W, Li M (2020) Subdata image encryption scheme based on compressive sensing and vector quantization. Neural Comput Applic 32(16):12771–12787. https://doi.org/10.1007/s00521-020-04724-x
21. Fishman GS, Moore LRIII (1986) Erratum: an exhaustive analysis of multiplicative congruential random number generators with modulus $2^{31} - 1$, SIAM J Sci Stat Comput. https://doi.org/10.1137/0907072
22. Gangopadhyay D, Allstot EG, Dixon AMR, Natarajan K, Gupta S, Allstot DJ (2014) Compressed sensing analog front-end for bio-sensor applications, IEEE J Solid-State Circ. https://doi.org/10.1109/JSSC.2013.2284673
23. Hashad FG, Zahran O, El-Rabaie ESM et al (2019) Fusion-based encryption scheme for cancelable fingerprint recognition. Multimed Tools Appl 78:27351–27381. https://doi.org/10.1007/s11042-019-7580-x
24. Hopper T, Brislawn C, Bradley J (1993) –WSQ gray-scale fingerprint image compression specification, Federal Bureau of Investigation, Criminal Justice Information Services, Washington, DC, USA, Tech. Rep. IAFIS-IC-0110-V2
25. Hopper T, Preston F (1992) Compression of grey-scale fingerprint images. In: Proceedings DCC ’92. Data Compression Conference. pp 309–318. https://doi.org/10.1109/DCC.1992.227450
26. Hsiung YS, Lu MSC (2011) A CMOS capacitive pressure sensor chip for fingerprint detection. In: 2011 16th International Solid-State Sensors, Actuators Microsystems Conference TRANSDUCERS’11. pp 24–27
27. Hu G, Xiao D, Wang Y, Xiang T (2017) An image coding scheme using parallel compressive sensing for simultaneous compression-encryption applications. J Vis Commun Image Represent 44:116–127. https://doi.org/10.1016/j.jvcir.2017.01.022
28. Javidi B (1999) Noise performance of double-phase encryption compared to XOR encryption. Opt Eng. https://doi.org/10.1117/1.602074
29. Javidi B (2000) Optical encryption using a joint transform correlator architecture. Opt Eng. https://doi.org/10.1117/1.1304844
30. Jiang X, Tang HY, Lu Y, Ng EJ, Tsai JM, Boser BE, Horsley DA (2017) Ultrasonic fingerprint sensor with transmit beamforming based on a PMUT array bonded to CMOS circuitry, IEEE Trans Ultrasonics, Ferroelectrics Frequency Control 64(9):1401–1408. https://doi.org/10.1109/UFFC.2017.2703606
31. Jung SM, Nam JM, Yang DH, Lee MK (2005) A CMOS integrated capacitive fingerprint sensor with 32-bit RISC microcontroller. IEEE J Solid-State Circ 40(8):1745–1750
32. Kharratzadeh M, Sharifnassab A, Babaie-Zadeh M (2017) Invariancy of sparse recovery algorithms. IEEE Trans Inf Theory 63:3333–3347. https://doi.org/10.1109/TIT.2017.2686428
33. Komarinski P (2005) Automated fingerprint identification systems (AFIS). Elsevier Science Technology
34. Kreutz-delgado K, Murray JF, Sejnowski TJ (2003) Dictionary learning algorithms for sparse representation kenneth. Neural Comput 15. https://doi.org/10.1116/089976603762552951
35. Kwon K, Nam I, Lee K (2016) A three-terminal n+-p-n+ silicon CMOS light-emitting device for the new fully integrated optical-type fingerprint recognition system. J Disp Technol 12(1):77–81. https://doi.org/10.1009/JDT.2015.2456641
36. Li R (2020) Fingerprint-related chaotic image encryption scheme based on blockchain framework Multimed Tools Appl. https://doi.org/10.1007/s11042-020-08802-z
37. Li X, Cai J (2007) Robust transmission of JPEG2000 encoded images over packet loss channels. In: Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, ICME. p 2007. https://doi.org/10.1109/icme.2007.4284808
38. Liu JC, Hsiung YS, Lu MSC (2012) A CMOS micromachined capacitive sensor array for fingerprint detection. IEEE Sens J 12(5):1004–1010
39. Liu D, Wang Q, Zhang Y, Liu X, Lu J, Sun J (2019) FPGA-Based real-time compressed sensing of multichannel EEG signals for wireless body area networks. Biomed Signal Process Control 49:221–230. https://doi.org/10.1016/j.bspc.2018.12.019
40. Mairal J, Bach F, FRANCISBACH J, Sapiro G (2010) Ponce JEANPONCE online learning for matrix factorization and sparse coding. J Mach Learn Res 11:19–60. http://www.jmlr.org/papers/volume11/mairal10a/mairal10a.pdf
41. Mairal J, Bach F, Sapiro G (2010) Ponce JEANPONCE online learning for matrix factorization and sparse coding, J Mach Learn Res. https://doi.org/10.1145/1756006.1756008
42. Mallat S (2009) A Wavelet Tour of Signal Processing. https://doi.org/10.1016/B978-0-12-374370-1.X0001-8
43. Mallat SG, Zhang Z (1993) Matching pursuits with time-frequency dictionaries, IEEE Trans Signal Process. https://doi.org/10.1109/78.258082
44. Maltoni D, Maio D, Jain AK, Prabhakar S (2009) Handbook of Fingerprint Recognition. Springer Science & Business Media. p 216. ISBN 978-1-84882-254-2
45. May RM (1976) Simple mathematical models with very complicated dynamics. Nature. https://doi.org/10.1038/261459a0
46. Mehta G, Dutta MK, Karasek J, Kim PS (2013) An efficient and lossless fingerprint encryption algorithm using Henon map & Arnold transformation. In: 2013 International Conference on Control Communication and Computing (ICCC). pp 485–489. https://doi.org/10.1109/ICCC.2013.6731703

47. Needell D, Tropp JA (2009) Cosamp: iterative signal recovery from incomplete and inaccurate samples. Appl Comput Harmon Anal. https://doi.org/10.1016/j.acha.2008.07.002

48. Orsdemir A, Altun HO, Sharma G, Bocko MF (2008) On the security and robustness of encryption via compressed sensing. In: Proceedings of the IEEE Military Communications Conference, MILCOM. https://doi.org/10.1109/MILCOM.2008.4753187

49. Pati YC, Rezaiifar R, Krishnaprasad PS (1993) Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: Conference Rec Asilomar Conference on Signals, Systems, and Computers. https://doi.org/10.1109/acssc.1993.342465

50. Polania LF, Carrillo RE, Blanco-Velasco M, Barner KE (2015) Exploiting prior knowledge in compressed sensing wireless ECG systems. IEEE J Biomed Heal Inform. https://doi.org/10.1109/JBHI.2014.2325017

51. Polat Ö, Kayhan SK (2018) High-speed FPGA implementation of orthogonal matching pursuit for compressed sensing signal reconstruction. Comput Electr Eng 71:173–190. https://doi.org/10.1016/j.compeleceng.2018.07.017

52. Rani M, Dhok SB, Deshmukh RB (2018) A systematic review of compressive sensing: concepts, implementations, and applications. IEEE Access. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2018.2793851

53. Ravelomanantsoa A, Rabah H, Rouane A (2015) Compressed sensing: a simple deterministic measurement matrix and a fast recovery algorithm. IEEE Trans Instrum Meas 64:3405–3413. https://doi.org/10.1109/TIM.2015.2459471

54. Rudelson M, Vershynin R (2008) On sparse reconstruction from Fourier and Gaussian measurements. Commun Pur Appl Math 61(8):1025–1045. https://doi.org/10.1002/cpa.20227

55. Ruelle D, Takens F (1971) On the nature of turbulence. Commun Math Phys. https://doi.org/10.1007/BF01646553

56. Sato N, Shigematsu S, Morimura H, Yano M, Kudou K, Kamei T, Machida K (2005) Novel surface structure and its fabrication process for MEMS fingerprint sensor. IEEE Trans Electron Devices 52:1026–1032. https://doi.org/10.1109/TED.2005.846342

57. (2001) ScientificWorking Group on Friction Ridge Analysis, Study and Technology (SWGFAST): Friction Ridge Digital Imaging Guidelines, 1 edn. http://www.theiai.org/guidelines/swgfast/. Accessed 12 Dec 2021

58. Shah AA, Parah SA, Rashid M, Elhoseny M (2020) Efficient image encryption scheme based on generalized logistic map for real time image processing. J Real-Time Image Process 17:2139–2151. https://doi.org/10.1007/s11354-020-01008-4

59. Shao G, Wu Y, Yong A, Liu X, Guo T (2014) Fingerprint compression based on sparse representation. IEEE Trans Image Process 23(2):489–501. https://doi.org/10.1109/TIP.2013.2287996

60. Sharma A, Shwetank A, Praveena C (2020) A novel image compression based method for multispectral fingerprint biometric system. Procedia Comput Sci 171:1698–1707. (Elsevier)

61. Shen Q, Liu W, Lin Y, Zhu Y (2019) Designing an image encryption scheme based on compressive sensing and non-uniform quantization for wireless visual sensor networks. Sensors (Switzerland) 19(14). https://doi.org/10.3390/s19143081

62. Sivapalan S, Rana RK, Chen D, Sridharan S, Denmon S, Fookes C (2011) Compressive sensing for gait recognition. In: Proceedings - 2011 international conference on digital image computing: techniques and applications, DICTA 2011 (pp 567–571). IEEE Computer Society. https://doi.org/10.1109/DICTA.2011.101

63. Skodras A, Christopoulos C, Ebrahimi T (2001) The JPEG 2000 still image compression standard. IEEE Signal Process Mag 18(5):36–58

64. Sun C, Li W, Chen W (2017) A compressed sensing based method for reducing the sampling time of a high resolution pressure sensor array system. Sensors (Switzerland) 17(8). https://doi.org/10.3390/s17081848

65. Tang Y, Zhao M, Li L (2020) Secure and efficient image Compression-Encryption scheme using new chaotic structure and compressive sensing. Secur Commun Netw 2020. https://doi.org/10.1155/2020/6665702

66. Tawfic I, Kayhan S (2015) Compressed sensing of ECG signal for wireless system with new fast iterative method. Comput Methods Programs Biomed. https://doi.org/10.1016/j.cmpb.2015.09.010

67. Thomos N, Boulgouris NV, Strintzis MG (2006) Optimized transmission of JPEG2000 streams over wireless channels. IEEE Trans Image Process. https://doi.org/10.1109/TIP.2005.860338
68. Vidyasagar M (2016) A tutorial introduction to compressed sensing. In: 2016 55th IEEE Conference on Decision and Control (CDC). pp 5091–5104. https://doi.org/10.1109/CDC.2016.7799048
69. Wang Z, Bovik AC, Sheikh HR (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process. https://doi.org/10.1109/TIP.2003.819861
70. Xie Y, Yu J, Guo S, Ding Q, Wang E (2019) Image encryption scheme with compressed sensing based on new three-dimensional chaotic system. Entropy 21(9). https://doi.org/10.3390/e21090819
71. Zhang Y, Zhang LY, Zhou J, Liu L, Chen F, He X (2016) A review of compressive sensing in information security field. IEEE Access. https://doi.org/10.1109/ACCESS.2016.2569421
72. Zhang Y, Zheng CH, Tanno N (2002) Optical encryption based on iterative fractional Fourier transform. Opt Commun. https://doi.org/10.1016 S0030-4018(02)01113-6
73. Zhao C, Wu X, Huang L, Yao Y, Chang YC (2014) Compressed sensing based fingerprint identification for wireless transmitters. Sci World J 2014. https://doi.org/10.1155/2014/473178
74. Zhao C, Wu X, Huang L, Yao Y, Chang YC (2014) Compressed sensing based fingerprint identification for wireless transmitters. Sci World J 2014. https://doi.org/10.1155/2014/473178

Publisher’s note  Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.