

## RESEARCH ARTICLE

# Secure Medical Image Communication Using Fragile Data Hiding Based on Discrete Wavelet Transform and $A_5$ Lattice Vector Quantization

EHSAN AKHTARKAVAN<sup>1,2</sup>, BABAK MAJIDI<sup>1</sup>, AND AMIRSINA MANDEGARI<sup>1</sup><sup>1</sup>Department of Computer Engineering, Faculty of Engineering, Khatam University, Tehran 1991633357, Iran<sup>2</sup>Pasargad Institute for Advanced Innovative Solutions, Tehran, Iran

Corresponding author: Ehsan Akhtarkavan (e.akhtarkavan@khatam.ac.ir)

**ABSTRACT** Secure communication of medical images is essential to telemedicine. Message Authentication Codes (MAC) can be embedded inside medical images to protect their integrity. Fragile watermarking algorithms are suitable options since they can be used to detect any tampering attempt. In this paper, a novel fragile data-hiding algorithm based on Integer-to-Integer Discrete Wavelet Transforms (IIDWT) and  $A_5$  Lattice Vector Quantization (LVQ) is proposed. In the proposed data-hiding algorithm, a combination of the medical image Metadata and a MAC is embedded into the medical image. The Metadata includes information about the patient such as name, family, birthday, the place where it is created such as the name of the hospital, and the physician's name. To preserve the privacy of the patients and the physician/hospital, the Metadata is then replaced with fake information. The receiver can extract the Metadata and the MAC. If the extracted MAC is the same as the expected MAC, the integrity of the medical image is guaranteed. Otherwise, a tampering attempt is detected. The proposed algorithm can embed 50% more data than similar algorithms in medical images while keeping the Peak Signal to Noise Ratio (PSNR) in acceptable ranges. Furthermore, the proposed algorithm is applied to a dataset of medical images and high PSNR values above 53.88 dB are experienced.

**INDEX TERMS** Data hiding, privacy, medical image privacy, fragile data hiding, integer discrete wavelet transforms, lattice vector quantization.

## I. INTRODUCTION

The developments in multimedia and communication technologies have provided more and more services such as telemedicine, but on the other hand, digital content is prone to manipulation. Thus, the integrity and authenticity of data and software often need to be protected. Over the past decades, medical image privacy protection based on data hiding or watermarking has been a common method [1], [2]. Specifically, authentication information can be hidden in the marked images and retrieved when needed. In addition to authentication, the data hidden in medical images may include Electronic Patient Records (EPR) [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Giovanni Dimauro<sup>1</sup>.

To employ watermarking schemes as an efficient way of protecting medical images, the embedded data must not cause significant distortion to the medical image. In addition, the watermarking must be invisible. A detailed description of watermarking schemes used for medical data authentication is presented in [4].

Although many watermarking algorithms have been proposed in the literature, the embedding capacity and imperceptibility of the embedding algorithms need to be improved. Lattice Vector Quantization (LVQ) [5] has been used for multiple descriptions image coding [6], [7] and multiple descriptions video coding [8]. However, LVQ is also used for data hiding [9], [10].

In this paper, a new fragile high-capacity data-hiding algorithm based on similar sub-lattices of  $A_5$  LVQ is proposed to protect the integrity and privacy of digital medical images.

The new proposed algorithm is an improved version of the watermarking algorithm proposed by the authors [11] based on similar sub-lattices of  $A_4$  LVQ, with higher capacity as well as higher imperceptibility. The proposed algorithm can embed more than 20 KB of data in a  $512 \times 512$  grayscale image, which is more than other watermarking algorithms while keeping the quality of the cover image at an acceptable level of Peak Signal to Noise Ratio (PSNR).

The proposed algorithm is applied to medical images and the results show that even after embedding the payload which is composed of the medical image Metadata, the patient's photo, and the watermark image, the quality of the marked cover image in terms of PSNR does not fall below 53 dB, which is completely acceptable for a semi-reversible data-hiding algorithm. The contributions of the proposed algorithm are as follows:

- 1) A systematic procedure for finding the similar sub-lattices of  $A_5$  with arbitrary index,  $N$  is presented in this manuscript.
- 2) The proposed data-hiding algorithm is based on  $A_5$ -LVQ can embed a high amount of data.
- 3) For the same embedding capacity, the proposed algorithm demonstrates higher perceptual quality in terms of PSNR compared to similar algorithms. This high capacity of embedding is consistent among different medical images.
- 4) The proposed data-hiding algorithm can hide the data inside the entire cover image and it is capable of extracting the embedded data without requiring the original image.

The rest of this paper is organized as follows. A review of the related works is provided in Section II. The proposed lattice vector quantization algorithm as well as the proposed data-hiding algorithm is introduced in Section III. The simulation study and the experimental results are presented in Section IV. Finally, Section V concludes the paper.

## II. RELATED WORKS

Data-hiding algorithms are categorized based on their application as well as the employed embedding technique. If the main objective of the data-hiding algorithm is to preserve the confidentiality of the embedded data, it is usually called steganography. But, if the security of the cover media is the main goal, it can be called watermarking.

Steganography algorithms are usually designed for grayscale images, and in the case of color images, the algorithms are applied to the 3 color channels equally. However, the relationship between color planes should be incorporated into steganography. In [12], a method, of amplifying channel modification probabilities (ACMP) is proposed, that can be used to generalize any grayscale steganography algorithm to a color steganography algorithm. Steganography algorithms are also divided into single-image and multiple-image methods. Dividing the payload into multiple images increases the security of the algorithm. The payload distribution

can be based on the texture of the image and distortion distribution [13].

Watermarking techniques are also divided into fragile watermarking methods and robust watermarking methods. Robust watermarking is suitable for ownership protection, since the algorithm is robust against possible attacks, and the embedded data is recovered. Fragile watermarking is suitable for the detection of tampering attempts [14].

In telemedicine secure communication of medical images is necessary. Digital medical images are prone to different attacks. One attacker may try to identify the patient or the physician, to compromise their privacy. On the other hand, one may try to tamper with the content of the image to cause misdiagnosis or to do fake insurance claims.

Zarrabi et al. present a blind diagnostically-lossless watermarking (BlessMark) [2]. In this research the region of interest (ROI) map is generated by a deep network and the watermark is only embedded in the region of non-interest (RONI) and the ROI is kept intact. The authors claim that BlessMark is a blind watermarking scheme since neither the ROI map nor the original cover image is required on the receiver side to extract the watermark. The proprietary deep network weights, which are used to find the ROI map are not publicly known, like a secret key. Therefore, the confidentiality of the embedded information is preserved. Fares et al. [15] proposed a blind and robust watermarking algorithm based on discrete cosine transform (DCT) and Schur decomposition with the results of 47-49 dB PSNR values.

The presented algorithm by Gao et al. in [16], separates an original medical image into ROI and RONI then it estimates the pixels concentration rate (PCR) of the ROI part. If the PCR is above a threshold, the ROI histogram is stretched, but if the PCR is less than the threshold, the ROI histogram is stretched without shifting. Finally, it embeds the secret data in the ROI. In addition, the RONI is pre-processed to embed secret data into it.

YANG et al. presented a framework for secure medical image communication based on Contrast-Enhancement Reversible Data Hiding (RDH) and Homomorphic Encryption [17]. The presented algorithm first embeds the payload in the medical image using a lesion-extraction-based RDH, then encrypts the marked medical image by a symmetric key Homomorphic cipher.

Kahlessenane et al. [18] presented a medical image watermarking method in the frequency domain which uses discrete wavelet transform (DWT), non-subsampled contourlet transforms, non-subsampled Shearlet transforms, and DCT. Robustness against various attacks is tested in this research. Alshanbari [19] used LZW (Lempel-Ziv-Welch) based fragile medical image watermarking. The results are proven to be robust against various attacks.

Hemdan [20] proposed a combination of Wavelet Fusion (WF), Singular Value Decomposition (SVD), and Multi-Level Discrete Wavelet Transform (MDWT) for medical image watermarking. The algorithm is proven to be able to

TABLE 1. Imperceptibly comparisons of the related state-of-the-art methods.

Reference	Method	PSNR (dB)
K. Fares et al. [15]	Discrete Cosine Transforms and Schur decomposition	47-49
Gao et al. [16]	RDH with automatic contrast enhancement for medical images	25-26
Gao et al. [17]	Contrast-Enhancement RDH and Homomorphic Encryption	25-31
F. Kahlessenane et al. [18]	Discrete wavelets transform, non-subsampled contourlet transform, non-subsampled shearlet transform, and discrete cosine transform	42-49
H. S. Alshanbari [19]	LZW (Lempel–Ziv–Welch)	47-49
E. E.D. Hemdan [20]	Wavelet Fusion, Singular Value Decomposition, and Multi-Level Discrete Wavelet Transform	36-55
N. Zermi et al. [21]	Discrete Wavelet Transform and Singular Value Decomposition	44
R. Thanki and A. Kothari [22]	Ridgelet transform and singular value decomposition	43-57
Bhardwaj et al. [23]	Efficient reversible and secure patient data hiding, a fragile watermark as well as the EPR is embedded in the medical image	48.13
P. V. Sanivarapu [24]	Schur decomposition and Quantum Hilbert Image Scrambling	17-37
S. P. Vaidya [25]	Lifting Wavelet Transform and Discrete Wavelet Transform	36
P. Singh et al. [26]	Discrete Wavelet Transform-Singular Value Decomposition	43
N. Agarwal and P. K. Singh [27]	Discrete Cosine Transforms and the genetic algorithm	52-53

resist severe attacks [20]. Zermi et al. [21] presented an algorithm that uses DWT decomposition and SVD for medical image watermarking. The PSNR values in the range of 44 are reported as the results.

Thanki and Kothari [22] used ridgelet transform and SVD and the Arnold scrambling-based encryption for medical image watermarking.

An enhanced RDH scheme is presented in [23]. In this work, a fragile watermark as well as the EPR is embedded in the medical image. Any kind of tampering attack would be detected since it will destroy the fragile watermark. During the embedding process underflow (intensity values <0) and overflow (intensity values >255) may occur. The authors have used Paillier cryptosystem to avoid this problem [23].

Sanivarapu [24] proposed a watermarking algorithm for medical images based on a 2-level DWT for the creation of sub-bands which are then partitioned into blocks. Furthermore, Schur decomposition is used for the creation of Authenticated Block Bits (ABB). The results show PSNR values larger than 30 dB.

Vaidya [25] presented a hybrid transform-based medical image watermarking which embeds the patient's fingerprint in the medical images using lifting wavelet transform (LWT) and DWT. The experimental results show the PSNR values in the range of 36 dB for watermarked medical images.

Singh et al. [26] used Least Significant Bit (LSB) substitution and pseudo-random key encryption and Discrete Wavelet Transform-Singular Value Decomposition (DWT-SVD) for watermarking. The experimental results show PSNR values in the range of 43 dB. Agarwal and Singh [27] used discrete cosine transform (DCT) and the genetic algorithm for RGB, YCbCr, and YIQ image watermarking. An increase in PSNR value led to 52-53 dB reported. Anand and Singh [4] surveyed various watermarking techniques in the medical domain.

### III. MATHEMATICS OF $A_5$ LATTICES

An  $n$ -dimensional lattice  $\Lambda$  in  $\mathbb{R}^n$  is denoted by a set of basis vectors,  $\Lambda = (b_1, b_2, \dots, b_n)$ . The lattice points are

generated using a generator matrix. The generator matrix is composed of the basis vectors of the lattice. The generator matrix of the lattice  $\Lambda$  with the basis vectors  $b_1 = (b_{11}, b_{12}, \dots, b_{1m}), \dots, b_n = (b_{n1}, b_{n2}, \dots, b_{nm})$  is given as [28]:

$$G = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{pmatrix} \quad (1)$$

The Gram matrix,  $A$ , of a lattice  $\Lambda$  is defined as  $A = GG^t$ , where  $G^t$  is the transposed matrix of  $G$ . The Gram matrix  $A$  determines the linear independence of the basis vectors, that is, they are linearly independent if and only if the determinant of  $A$  is non-zero. Two lattices are called equivalent if they are with proportionate Gram matrices. The determinant of a lattice  $\Lambda$  is also equal to the determinant of the  $A$  [28]:

$$\det \Lambda = \det(\check{A}) \quad (2)$$

If the generator matrix is square then (2) is written as [28]

$$\det \Lambda = (\det(G))^2 \quad (3)$$

Thus, the volume of the fundamental parallelotope of a lattice  $\Lambda$  is also calculated as [28]:

$$\text{vol} = \det(G) = \sqrt{\det(\Lambda)} \quad (4)$$

Assume that  $\Lambda$  is an  $L$ -dimensional lattice with the generator matrix  $G$ . A sub-lattice  $\Lambda' \subset \Lambda$  with generator matrix  $G'$  is said to be geometrically similar to  $\Lambda$  if and only if  $G' = cUGB$ , for nonzero scalar  $c$ , an integer matrix  $U$  with  $\det U = \pm 1$ , and a real orthogonal matrix  $B$  (with  $BB^t = I$ ) [28].

The index  $N$  is defined as the ratio of the fundamental volume of the sub-lattice  $\Lambda'$  to the fundamental volume of

the lattice  $\Lambda$ . The fundamental volume of the lattice ( $vol$ ) is equal to the determinant of the generator. Thus,  $N$  is calculated by

$$N = \frac{vol'}{vol} = \sqrt{\frac{\det(\Lambda')}{\det(\Lambda)}} = \frac{\det(\mathbf{G}')}{\det(\mathbf{G})} \quad (5)$$

The root lattice  $A_n$  is a subset of points with  $n+1$  coordinates, such that the sum of their coordinates is zero. Therefore, the lattice  $A_n$  can be defined as:

$$A_n = \left\{ (x_0, x_1, \dots, x_n) \in \mathbb{Z}^{n+1} : x_0 + x_1 + \dots + x_n = 0 \right\} \quad (6)$$

A sub-lattice  $\Lambda' \subset \Lambda$  with generator matrix  $\mathbf{G}'$  is said to be geometrically similar to  $\Lambda$  if and only if  $\mathbf{G}' = c\mathbf{U}\mathbf{G}\mathbf{B}$ , for nonzero scalar  $c$ , an integer matrix  $\mathbf{U}$  with  $\det \mathbf{U} = \pm 1$ , and a real orthogonal matrix  $\mathbf{B}$  (with  $\mathbf{B}\mathbf{B}^t = \mathbf{I}$ ) [28].

In other words, the existence of  $\Lambda'$  can be determined (in principle) by searching through  $\Lambda$  to see if it contains a set of vectors with Gram matrix  $c\mathbf{A}$ . For small  $k$  and  $c$ , this is quite feasible [29]. In lattice vector quantization (LVQ), the lattice points are used as the quantization bins. During the LVQ process, every point  $p \in \mathbb{R}^5$  is mapped to a certain lattice point  $\lambda$ , that is closer to  $p$  than any other lattice point. The region around every  $\lambda$ , which includes all the points that are mapped onto  $\lambda$  by LVQ is called its Voronoi region,  $Voro_\lambda$ .

Sub-lattice  $\Lambda' \subset \Lambda$  is considered a clean sub-lattice if all the points of the  $\Lambda$  reside only inside the Voronoi region of the sub-lattice points rather than on the boundary of the Voronoi region [29]. It means that the lattice points are not shared between the Voronoi regions of adjacent sub-lattice points.

### A. FINDING SIMILAR SUB-LATTICES OF $A_5$

According to (6), the  $A_5$  root lattice is defined as:

$$A_5 = \left\{ (x_0, x_1, x_2, x_3, x_4, x_5) \in \mathbb{Z}^6 : \sum_{i=0}^5 x_i = 0 \right\} \quad (7)$$

The  $A_5$  root lattice space is spanned by the multiplication of  $\mathbb{Z}_5$  points and (8)

$$\mathbf{G}_{A_5 \times 6} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \quad (8)$$

Every  $A_n$  lattice has a  $n \times n$  square generator in addition to the rectangular  $n \times (n+1)$  the generator is usually called a triangular generator. It is difficult to find the similar sub-lattices of  $A_5$  using the rectangular description  $A_{5 \times 6}$ . Therefore, the square description of  $A_5$  in  $\mathbb{R}^5$  space is usually used instead. The square generator of the  $A_5$  is

given as:

$$\mathbf{G}_{A_5 \times 5} = \begin{pmatrix} 1.414 & 0 & 0 & 0 & 0 \\ -0.707 & 1.224 & 0 & 0 & 0 \\ 0 & -0.816 & 1.154 & 0 & 0 \\ 0 & 0 & -0.866 & 1.118 & 0 \\ 0 & 0 & 0 & -0.894 & 1.095 \end{pmatrix} \quad (9)$$

In general, it is not possible to find clean any sub-lattice of  $A_n$  with any arbitrary  $N$ . In other words, for every  $A_n$  lattice, only certain values of  $N$  are possible. In the case of root  $A_5$  lattice, similar sub-lattice generators can only be found for:

$$N = \{1, 32, 1024, 32768, 1048576, \dots, 32^k, k \in \mathbb{Z}\} \quad (10)$$

This sequence of integer numbers is named A000302 in [30]. The Gram matrices of the generators of the similar sub-lattices,  $A'_{A_5 \times 5}$  and the Gram matrix of the lattice,  $A_{A_5 \times 5}$  are proportional. The Gram matrix of  $\mathbf{G}_{A_5 \times 5}$  is:

$$\check{A}_{A_5 \times 5} = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix} \quad (11)$$

Suppose the generator of a clean sub-lattice is

$$\mathbf{G}' = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \\ \mathbf{b}_5 \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} \end{pmatrix} \quad (12)$$

Then, the Gram matrix of the generator of the clean sub-lattice  $A'_{A_5 \times 5}$  would be:

$$\check{A}'_{A_5 \times 5} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \\ \mathbf{b}_5 \end{pmatrix} \times \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \\ \mathbf{b}_5 \end{pmatrix}^t = \begin{pmatrix} 2d & -d & 0 & 0 & 0 \\ -d & 2d & -1 & 0 & 0 \\ 0 & -d & 2d & -d & 0 \\ 0 & 0 & -d & 2d & -d \\ 0 & 0 & 0 & -d & 2d \end{pmatrix} \quad (13)$$

According to (12), the basis vectors of the generator of the sub-lattice must have the following properties:

$$\mathbf{b}_i \cdot \mathbf{b}_j = \begin{cases} +2d & i = j \\ -di - j & = \pm 1 \\ 0 & otherwise \end{cases} \quad (14)$$

where  $d$  can be 1, 4, 16, 64, 256, 1024, 4096, ... In other words, to find a clean sub-lattice generator with an arbitrary index  $N$ , first  $A_5$  lattice is generated by the multiplication of

$Z_5$  by (8). Any set of 5 points in  $A_5$  lattice can be used as the basis vectors of a potential generator of the sub-lattice. But only those potential generators with a Gramm matrix  $A'$  that fulfills the requirements of (14) can generate a clean sub-lattice of  $A_5$ . Therefore, the  $A_5$  lattice must be searched to find sets of 5 lattice points such that, if used as the basis vectors of the generator of the sub-lattice, they satisfy the properties of (13).

Although the  $A_5$  lattice includes an infinite number of points, it is not required to search all the  $R^5$  space because the lattice points that are possible basis vectors reside in the proximity of the origin, for example with Euclidean distance less than 10 it is possible to find sub-lattice generators with  $N < 1048690$ .

In this way, a portion of  $A_5$  lattice was generated by the multiplication of a portion of the  $Z_5$  points and (8). The selected portion of  $Z_5$  includes all 5-dimensional integer vectors that range from  $[-5 - 5 - 5 - 5 - 5]$  to  $[5 5 5 5 5]$ , therefore it includes  $11^5$  elements and consequently the corresponding  $A_5$  lattice will have the same number of elements.

However, searching among all the 5-combinations of a set with  $11^5$  elements,  $(161051!)/(5! \times 161046!)$  will be infeasible. To minimize the searching time, a progressive procedure is defined. Assume that we are going to find sub-lattice generators with  $N = 1ord = 1$ . According to our experiments, there are 30 lattice points  $\lambda_i$  in the selected portion of  $A_5$  that hold the following property:  $\lambda_i \cdot \lambda_i = 2d = 2$ .

TABLE 2. A5 lattice points that satisfy  $\lambda_i \cdot \lambda_i = 2$ .

$i$	$\lambda_i$	$i$	$\lambda_i$
1	(-0.707,-0.408,-1.155, 0, 0)	16	(0, 0, 0, -0.894, 1.095)
2	(-0.707,-0.408,-0.289, 1.118, 0)	17	(0, 0, 0.866, 1.118, 0)
3	(-0.707,-0.408,-0.289, 0.224, 1.095)	18	(0, 0, 0.866, 0.224, 1.095)
4	(-0.707,-1.225, 0, 0, 0)	19	(0, -0.816, 0.289,-0.224,-1.095)
5	(-1.414, 0, 0, 0, 0)	20	(0, -0.816, 0.289,-1.118, 0)
6	(0.707,-0.408,-1.155, 0, 0)	21	(0, -0.816, 1.155, 0, 0)
7	(0.707,-0.408,-0.289, 1.118, 0)	22	(-0.707, 1.225, 0, 0, 0)
8	(0.707,-0.408,-0.289, 0.224, 1.095)	23	(-0.707, 0.408, 0.289,-0.224,-1.095)
9	(0.707,-1.225, 0, 0, 0)	24	(-0.707, 0.408, 0.289,-1.118, 0)
10	(0, 0.816,-1.155, 0, 0)	25	(-0.707, 0.408, 1.155, 0, 0)
11	(0, 0.816,-0.289, 1.118, 0)	26	(1.414, 0, 0, 0, 0)
12	(0, 0.816,-0.289, 0.224, 1.095)	27	(0.707, 1.225, 0, 0, 0)
13	(0, 0, -0.866,-0.224,-1.095)	28	(0.707, 0.408, 0.289,-0.224,-1.095)
14	(0, 0, -0.866,-1.118, 0)	29	(0.707, 0.408, 0.289,-1.118, 0)
15	(0, 0, 0, 0.894,-1.095)	30	(0.707, 0.408, 1.155, 0, 0)

Therefore, these lattice points are potential basis vectors of the sub-lattice generators with index  $N = 1$ , presented in Table 2. This matrix is called  $A_{5_{N1}}$ . Although, searching among all the 5-combinations of a set with 30 elements  $(30!)/(5! \times 25!)$  is very faster than the previous set with 161051 members, but it is still impossible. Each row of Table 2 can be used as the first basis vector of the sub-lattice generator. Next, to find the second basis vector of the sub-lattice generator, the mutual dot product between the 30 rows of  $A_{5_{N1}}$  are computed. Among  $30 \times 30$  couples, there are 120 couples  $[\lambda_i, \lambda_j]$  such that  $\lambda_i \cdot \lambda_j = -1$ . This matrix

is called  $A_{5_{N1_{ij}}}$ . Several example couples of  $A_{5_{N1_{ij}}}$  are presented in Table 3.

TABLE 3. Example couples of lattice points satisfying  $\lambda_i \cdot \lambda_j = -1$ .

COUPLE	$[\lambda_i, \lambda_j]$
1	$[\lambda_1, \lambda_{17}] = [(-0.707, -0.408, -1.155, 0, 0), (0, 0, 0.866, 1.118, 0)]$
2	$[\lambda_1, \lambda_{18}] = [(-0.707, -0.408, -1.155, 0, 0), (0, 0, 0.866, 0.224, 1.095)]$
3	$[\lambda_2, \lambda_{14}] = [(-0.707, -0.408, -0.289, 1.118, 0), (0, 0, -0.866, -1.118, 0)]$
4	$[\lambda_3, \lambda_{13}] = [(-0.707, -0.408, -0.289, 0.224, 1.095), (0, 0, -0.866, -0.224, -1.095)]$
5	$[\lambda_3, \lambda_{15}] = [(-0.707, -0.408, -0.289, 0.224, 1.095), (0, 0, 0, 0.894, -1.095)]$

In the third step, for every couple of  $A_{5_{N1_{ij}}}$ , a suitable lattice point  $\lambda_k$  from  $A_{5_{N1}}$  must be selected. Among these  $120 \times 30$  possible tuples, there are only 600 triple like  $[\lambda_i, \lambda_j, \lambda_k]$  such that  $\lambda_i \cdot \lambda_k = 0, \lambda_j \cdot \lambda_k = -1$ . These 600 triples form a matrix called  $A_{5_{N1_{ijk}}}$ . Two examples are triples of  $A_{5_{N1_{ijk}}}$  are presented in Table 4. These triples all satisfy the requirements of (15).

TABLE 4. Example triples of lattice points satisfying (12).

COUPLE	$[\lambda_i, \lambda_j, \lambda_k]$
1	$[\lambda_1, \lambda_{17}, \lambda_6] = [(-0.707, -0.408, -1.155, 0, 0), (0, 0, 0.866, 1.118, 0), (0.707, -0.408, -1.154, 0, 0)]$
2	$[\lambda_1, \lambda_{18}, \lambda_{10}] = [(-0.707, -0.408, -1.155, 0, 0), (0, 0, 0.866, 0.224, 1.095), (0, 0.816, -1.154, 0, 0)]$

For each row of  $A_{5_{N1_{ijk}}}$  such as  $[\lambda_i, \lambda_j, \lambda_k]$ , the following dot multiplications hold:

$$\begin{matrix}
 \text{dot} & \lambda_i & \lambda_j & \lambda_k \\
 \lambda_i & 2 & -1 & 0 \\
 \lambda_j & -1 & 2 & -1 \\
 \lambda_k & 0 & -1 & 2
 \end{matrix} \tag{15}$$

In the fourth step, for every triple of  $A_{5_{N1_{ijk}}}$ , suitable lattice points  $\lambda_l$  from  $A_{5_{N1}}$  are selected. Among these  $600 \times 30$  possible quadruples, there are only 3960 quadruples  $[\lambda_i, \lambda_j, \lambda_k, \lambda_l]$  such that  $\lambda_i \cdot \lambda_l = 0, \lambda_j \cdot \lambda_l = 0$ , and  $\lambda_k \cdot \lambda_l = -1$ . These 3960 quadruples form a matrix called  $A_{5_{N1_{ijkl}}}$ . Two examples of quadruples of  $A_{5_{N1_{ijkl}}}$  are presented in Table 5.

For each quadruple of  $A_{5_{N1_{ijkl}}}$  such as  $[\lambda_i, \lambda_j, \lambda_k, \lambda_l]$ , the following dot multiplications hold:

$$\begin{matrix}
 \text{dot} & \lambda_i & \lambda_j & \lambda_k & \lambda_l \\
 \lambda_i & 2 & -1 & 0 & 0 \\
 \lambda_j & -1 & 2 & -1 & 0 \\
 \lambda_k & 0 & -1 & 2 & -1 \\
 \lambda_l & 0 & 0 & -1 & 2
 \end{matrix} \tag{16}$$

Finally, for every quadruple point in  $A_{5_{N1_{ijkl}}}$ , possible lattice points are found such as  $[\lambda_i, \lambda_j, \lambda_k, \lambda_l, \lambda_m]$ , that satisfy the requirements of (14) and therefore they form a sub-lattice

TABLE 5. Two example couples of  $A_{5\_N1\_ijkl}$ .

COUPLE	$[\lambda_i, \lambda_j, \lambda_k, \lambda_l]$
1	$[-0.707, -0.408, -1.155, 0, 0]$ ,
	$(0, 0, 0.866, 1.118, 0)$ ,
	$(0.707, -0.408, -1.154, 0, 0)$ ,
	$(0, -0.816, 1.155, 0, 0)$ ]
2	$[-0.707, -0.408, -1.155, 0, 0]$ ,
	$(0, 0, 0.866, 0.224, 1.095)$ ,
	$(0, 0.816, -1.154, 0, 0)$ ,
	$(0, 0, 0, 0.866, 1.118, 0)$ ]

generator of index  $N = 1$ . According to our experiments, 720 distinct sub-lattice generators with index  $N = 1$  were found.

$$\begin{matrix}
 \text{dot} & \lambda_i & \lambda_j & \lambda_k & \lambda_l & \lambda_m \\
 \lambda_i & 2 & -1 & 0 & 0 & 0 \\
 \lambda_j & -1 & 2 & -1 & 0 & 0 \\
 \lambda_k & 0 & -1 & 2 & -1 & 0 \\
 \lambda_l & 0 & 0 & -1 & 2 & -1 \\
 \lambda_m & 0 & 0 & 0 & -1 & 2
 \end{matrix} \tag{17}$$

This means that each row of  $A_{5\_N1\_ijklm}$  form a new generator of the  $A_5$  sub-lattice with index  $N = 1$ . For example, the 3<sup>rd</sup> quintuple gives the following generator, (18), as shown at the bottom of the page.

Since  $G'_{A5\_N1\_3}$  and  $G_{A5\_5 \times 5}$  have the same Gramm matrices, their relative index is  $N = 1$ . The same procedure is

repeated for other values of  $d = 4, 16, 64, \text{and } 256$ , and generators with indexes of  $N = 32, 1024, 32768, \text{and } 1048576$  were calculated. Example generators are reported in (19) to (22), as shown at the bottom of the page.

Fast quantizing algorithms are a family of lattice vector quantization algorithms presented in [28] for different root lattices. The fast quantizing algorithm first projects the  $n$ -dimensional input vector onto  $n + 1$  dimensional vectors on  $\sum_{i=1}^{n+1} x_i = 0, x_i \in R$  hyper-plane using a matrix-multiplication [28]. Then, using a manipulation the projected point is mapped onto a lattice point. The transformation matrix determines on which lattice the input points must be mapped. For example, if the points are going to be quantized on  $G_{A5\_5 \times 5}$  then  $T_{A5\_5 \times 5}$  must be used. The transformation matrix  $T_{A5\_5 \times 5}$  is calculated using the relation between the triangular generators  $G_{A5\_5 \times 5}$  and the  $(n+1)$ -dimensional definition  $G_{A5\_5 \times 6}$  [31]:

$$T_{A5\_5 \times 5} = (G_{A5\_5 \times 5})^{-1} G_{A5\_5 \times 6} \tag{23}$$

#### IV. PROPOSED MEDICAL IMAGE DATA HIDING BASED ON DWT AND $A_5$ LVQ ALGORITHM

In this part of the paper, the proposed data-hiding algorithm is presented. For every data-hiding algorithm whether it is fragile or not, two different functions must be implemented,

$$G'_{A5\_N1\_3} = \begin{pmatrix} \lambda_{29} \\ \lambda_{11} \\ \lambda_{19} \\ \lambda_8 \\ \lambda_{25} \end{pmatrix} \begin{pmatrix} -0.707 & 0.408 & 0.2887 & -1.118 & 0 \\ 0 & 0.816 & -0.288 & 1.118 & 0 \\ 0 & -0.816 & 0.288 & -0.223 & -1.095 \\ 0.707 & -0.408 & -0.288 & 0.223 & 1.095 \\ -0.707 & 0.408 & 1.154 & 0 & 0 \end{pmatrix} \tag{18}$$

$$G'_{A5\_N32} = \begin{pmatrix} -0.707 & -1.224 & -1.732 & -1.341 & -1.095 \\ -0.707 & -1.224 & 1.732 & 1.341 & 1.095 \\ -0.707 & 2.041 & -1.154 & 0.894 & -1.095 \\ 0 & 0 & 0 & -1.788 & 2.190 \\ 0 & 0 & 1.732 & -0.447 & -2.190 \end{pmatrix} \tag{19}$$

$$G'_{A5}{}^{/N1024} = \begin{pmatrix} -3.535 & -2.857 & -2.020 & -1.565 & -2.191 \\ 1.414 & 2.449 & -1.732 & -1.341 & 4.381 \\ 0 & -3.265 & 1.154 & 4.472 & 0 \\ 3.535 & -0.408 & -0.288 & -3.801 & -2.191 \\ -3.535 & 0.408 & 3.752 & -0.670 & 2.191 \end{pmatrix} \tag{20}$$

$$G'_{A5-N32768} = \begin{pmatrix} -7.071 & -4.899 & -5.196 & -4.024 & -3.286 \\ 7.071 & 1.632 & -0.577 & -4.024 & 7.668 \\ -8.485 & 4.899 & 5.196 & 2.236 & 0 \\ 2.828 & -6.532 & -4.618 & 7.155 & 2.190 \\ 1.414 & -3.265 & 8.082 & -6.260 & -3.286 \end{pmatrix} \tag{21}$$

$$G'_{A5\_N1048576} = \begin{pmatrix} -10.606 & -6.940 & -8.371 & 3.354 & -16.431 \\ 10.606 & 4.490 & -13.279 & -7.602 & 12.049 \\ 4.242 & -17.146 & 13.856 & 1.788 & -2.190 \\ -18.384 & 6.532 & -3.175 & 1.118 & 10.954 \\ 13.435 & 3.674 & -2.598 & 16.770 & -5.477 \end{pmatrix} \tag{22}$$

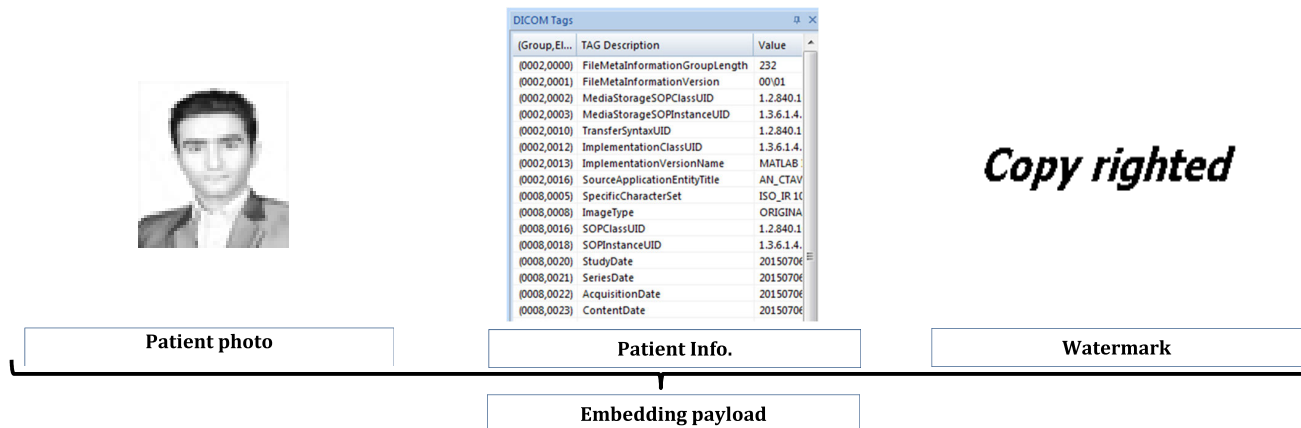


FIGURE 1. The payload data includes patient’s photo, Meta data, and watermark image.

that is embedding function and extracting function. These functions are described in the following subsections.

TABLE 6. Selected Metadata fields to be anonymized and embedded in the cover image [32].

Field name	Original info.	Fake info.
StudyDate	'20150706'	'13960101'
SeriesDate	'20150706'	'13960101'
AcquisitionDate	'20150706'	'13960101'
ContentDate	'20150706'	'13960101'
PatientBirthDate	'20120706'	'13960101'
PatientSex	'F'	'O'
PatientName.FamilyName	'MAMASHLI'	'NIYAZMAND'
PatientName.GivenName	'HELMA'	'PARISA'
PatientID	'3388517'	'11111111'
PatientAge	'003Y'	'113Y'
TransferSyntaxUID	'1.2.840.10008.1.2.1'	'1.2.840.10008.1.2.1'
InstitutionName	'MILAD HOSPITAL'	'HOTEL HOSPITAL'

A. EMBEDDING FUNCTION

Digital medical image files usually have a DICOM (Digital Imaging and Communication in Medicine) format. DICOM makes medical images exchange easier and more independent of the imaging equipment manufacturer. The image data type is usually uint16 which means every pixel has a positive number between 0 and 65536. Besides the image data, the DICOM file format includes other information useful to describe the images such as the type of patient diagnostic image (MRI/CT), the patient’s demographic data, the place where it is created such as the name of the hospital, and the physician’s name, called Metadata [32].

The medical images that are used in our experiments include 95 fields, which include 2 KB of Metadata. The details of the selected Metadata are provided in Table 6. In the proposed data-hiding algorithm, the patient’s photo and the watermark logo are also embedded in the cover image. The image inside the DICOM file is used as the cover image, in which the sensitive data are embedded.

To describe the proposed embedding algorithm, assume the cover image has a common size of 512 × 512. It is important to minimize the distortion due to the embedding process, therefore the embedding process does not affect all pixels of the image, but it only selects several blocks of the image and embeds the data in those blocks. The number of required blocks depends on the amount of the payload. Since the algorithm is going to embed the payload using the DWT coefficients and A<sub>5</sub> LVQ, the blocks must be 10 × 10 pixels so that DWT coefficients become 5 × 5 blocks. Thus, to create 10 × 10 pixels blocks, the 512 × 512 cover image is cropped and one 510 × 510 image is formed and then it is reshaped to 2601 blocks.

Assume a given 10 × 10 block of the image is selected for embedding. In the second step of the embedding process, this block is transformed by an IIDWT. This is important to mention that the DWT must be Integer-to-Integer. If a non-integer DWT is used instead, then the cover image would have fractions after the embedding. Since the data type of the cover image is an unsigned integer then any floating-point value is rounded and consequently, the embedded data is lost. The IIDWT transforms the input 10 × 10 block into four 5 × 5 sub-bands, namely approximation coefficients or LL, horizontal sub-band (HL), vertical sub-band (LH), and diagonal sub-band (HH). Embedding inside the approximation coefficients injects a considerable amount of distortion therefore it is usually kept intact. The coefficients of the HL, LH, and HH sub-bands are only used for embedding.

Suppose a vector of the HL is selected for the embedding V. The main idea of the embedding process is to replace V with V' according to a mapping table and recover the embedded data by having V' and the mapping table. The set of Z<sub>5</sub> points in the vicinity of V are suitable candidates. However, the mapping table must not be sent along V', thus in the receiver, it will not be possible to recover the data using V' without the mapping table. The mapping table can be reconstructed in the receiver if the same set of Z<sub>5</sub> points can be formed, which requires to have a consistent relation between

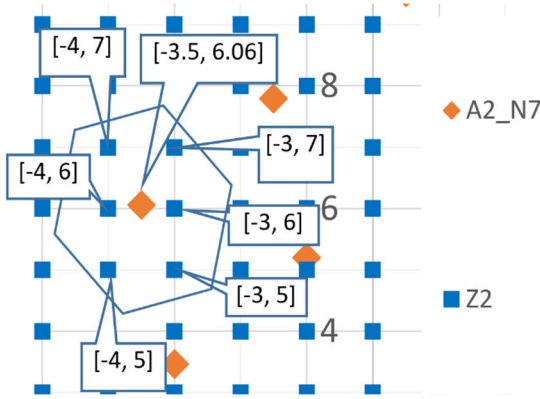


FIGURE 2. Similar sublattice of  $A_2$  with index  $N = 7$  is shown with red diamond and  $Z_2$  points are shown with blue squares.

the  $Z_5$  points in the set and  $V$ . The lattice vector quantization can be used to form the required relation.

The set of  $Z_5$  points found within the Voronoi region of any given  $A_5$  sub-lattice point  $\lambda'$ , includes  $Z_5$  points that are quantized  $\lambda'$  by  $A_5$ -LVQ. If the set of the  $Z_5$  points that are quantized to a fixed point such as  $\lambda' = LVQ_{A_5}(V, N)$  is formed, then it is possible to reconstruct the same set  $Z_5$  points in the receiver. This set,  $Z_{voro}$ , is defined as:

$$Z_{voro}(V, N) = \{ \forall v | v \in Z_5 \text{ and } \lambda' = LVQ_{A_5}(v, N) \} \quad (24)$$

The process of finding the  $Z_5$  points that reside within the Voronoi region of a given sub-lattice point are as follows. First, the DWT coefficient vector  $V$  is quantized onto  $\lambda'$ , a sub-lattice of  $A_5$  with a given index  $N$ . Then,  $\lambda'$  is rounded to the nearest  $Z_5$  and added to each row of a combination matrix  $[-e, -e, -e, -e, -e]$  to  $[+e, +e, +e, +e, +e]$ , where  $e$  is determined by  $N$ . The result is the set of all  $Z_5$  points around  $\lambda'$ . Finally, among these points, only those that are lattice vector quantized on  $\lambda'$  are selected and then  $Z_{voro}(V, N)$  is formed.

For  $N = 1$  the value of  $e$  can be as small as 1 and for  $N = 1048576$  the value of  $e$  must be as high as 4. When  $e = 2$ , there are  $5^5$  rows in the combination matrix. The embedding capacity depends on the number of  $Z_5$  points found inside  $Z_{voro}(V, N)$ , which is controlled by  $N$ . The most important property of  $Z_{voro}(V_{HL}, N)$  is that:

$$\begin{aligned} Z_{voro}(V_1, N) &= Z_{voro}(V_2, N) \\ \Leftrightarrow V_1 \in Z_{voro}(V_2, N) \wedge V_2 \in Z_{voro}(V_1, N) \end{aligned} \quad (25)$$

In other words, if any member of  $Z_{voro}(V, N)$  is sent instead of  $V$ , such as  $V'$ , the receiver repeats the process and finds  $Z_{voro}(V', N)$  that is the same as  $Z_{voro}(V, N)$ . Thus, it is no longer required to send the mapping table with  $V'_{HL}$ .

It is impossible to depict the embedding process in 5 dimensions. However, to visualize the process, it is possible to simulate the process in 2 dimensions, that is,  $A_2$  is used instead of  $A_5$  and  $Z_2$  is used instead of  $Z_5$ .

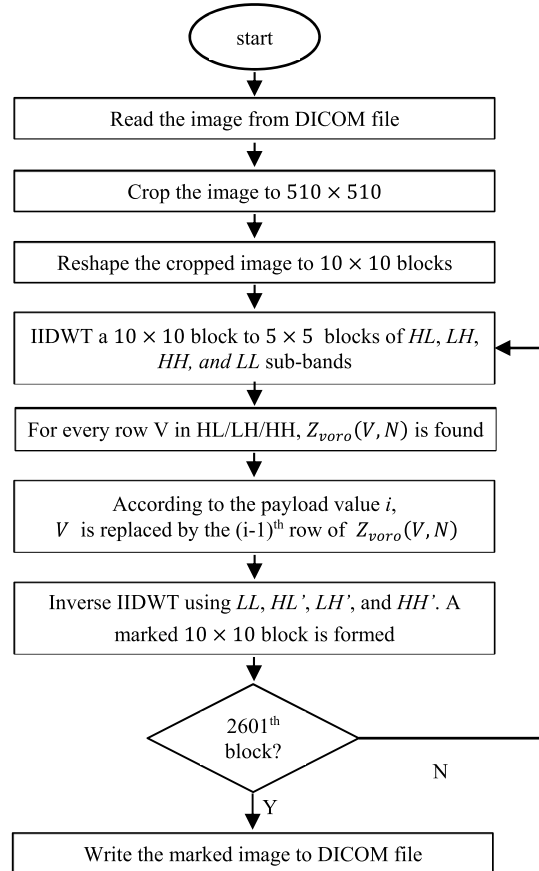


FIGURE 3. The embedding phase of the proposed algorithm.

In FIGURE 2, a sub-lattice of  $A_2$  with index  $N = 7$  is depicted with red diamonds and  $Z_2$  points are shown with blue squares. Suppose the DWT coefficient vector is  $V_{Z_2D} = [-4, 7]$ , and  $V_{Z_2D}$  is lattice vector quantized onto  $\lambda' = [-3.5, 6.06]$ . The set of  $Z_2$  points that are also quantized by LVQ onto  $\lambda'$  is  $Z_{voro}(V_{Z_2D}, N) = \{N\} = [-4, 6], [-4, 7], [-3, 7], [-3, 6], [-3, 5], [-4, 5]$ .

Since these points are all within the Voronoi of the same sub-lattice point  $\lambda'$ , the same  $Z_{voro}$  will be calculated for each of them. Among these six points, we can select four points as the embeddingbins =  $\{[-4, 6], [-4, 7], [-3, 7], [-3, 6]\}$ , as a consequence, the embedding capacity related to  $V_{Z_2D}$  is  $\log_2 4, 2$  Bits Per 2 Pixels (BPP), or 1 BPP.

According to our experiments, the proposed data-hiding algorithm shows a very high capacity when the index  $N > 1024$ . For example, if a given vector of DWT coefficient  $V = [63, 82, -75, 83, 27]$  is quantized onto a sub-lattice of  $A_5$  with a given index  $N = 1$ , it will be mapped on to  $\lambda' = [62.932, 82.058, -74.478, 82.734, 27.386, ]$ . There are only three  $Z_5$  points found in the Voronoi of  $\lambda'$  which form  $Z_{voro}$ , provided in Table 7.

The mapping table can be as simple as possible. For example, if  $Z_{voro}$  in Table 7 is considered, according to Table 7 the capacity is 1 bit per 5 pixels. It means that if the payload data is 0,  $V' = [63, 82, -74, 83, 27]$  is sent, and if the payload data is 1,  $V' = [63, 82, -75, 83, 27]$  is selected.



TABLE 7. The set of  $Z_{voro}$  corresponding  $V = [63,82,-75,83,27]$ .

$V$	$Z_{voro}(V, 1)$	Selected if
	$[63, 82, -74, 83, 27]$	Payload = 0
$[63, 82, -75, 83, 27]$	$[63, 82, -75, 83, 27]$	Payload = 1
	$[63, 82, -74, 83, 28]$	-

The embedding phase continues by repeating the same procedure for every row in  $HL$ ,  $LH$ , and  $HH$  sub-bands. For a given  $510 \times 510$  image there are 2601 blocks of the size  $10 \times 10$  and as a consequence, there are  $2601 \times 3$  sub-bands of the size  $5 \times 5$ . The next step of the embedding phase is to do the inverse IIDWT transform and prepare  $10 \times 10$  blocks of the embedded cover image. Finally, the embedded image and anonymized Metadata are used to generate a DICOM image to send to the destination. The embedding procedure is depicted in FIGURE 3.

**B. EXTRACTING FUNCTION**

On the receiver side, the extracting process is applied to the received DICOM file. The data extraction process is very similar to the embedding process. First, the cover image is read from the DICOM file. Next, the cover image is reshaped to the form of 2601 blocks of  $10 \times 10$  dimensions. Every  $10 \times 10$  block is transformed to the wavelet domain using the same IIDWT. The result for every block would be four  $5 \times 5$  sub-bands,  $LL$ ,  $HL$ ,  $LH$ , and  $HH$ . As mentioned in IV-A there is no data embedded in the approximation sub-band,  $LL$ . Thus,  $HL$ ,  $LH$ , and  $HH$  sub-bands are used to extract data. Every  $5 \times 5$  sub-band includes 5 rows and consequently 5 embedded data.

Assume that a row  $V'$  is received. To extract the data embedded in  $V'$ , it is required to find its corresponding  $Z_{voro}(V', N)$ , the set of all  $Z_5$  points that are in the Voronoi of the same sub-lattice point  $\lambda'$ . As mentioned in IV-A, members of the  $Z_{voro}(V', N)$ , set have two properties, they are all close to the input vector and they are all quantized to the same sub-lattice point, and consequently, they are all within the Voronoi of the same sub-lattice point. Thus, the following relation can be used instead of (24).

$$Z_{voro}(V', N) = \{v' | v' \in Z_5 \text{ and } \lambda' = LVQ\_A5(v', N)\} \tag{26}$$

In other words, it doesn't matter which member of  $Z_{voro}(V', N)$  has been sent and received, since  $Z_{voro}(V')$  is the same as  $Z_{voro}(V', N)$  for every one of the rows. Now the input  $V'$  is searched in the  $Z_{voro}(V', N)$  and its index in  $Z_{voro}(V', N)$  is found. The embedded data is decoded by subtracting 1 from the index of  $V'$  in  $Z_{voro}(V', N)$ . The extraction procedure is depicted in FIGURE 4.

According to the example provided in IV-A, if  $V' = [63, 82, -74, 83, 27]$  is received, the extracted data bit is 0, whereas if  $V' = [63, 82, -75, 83, 27]$  is received, the extracted data bit is 1. The extracting phase continues by

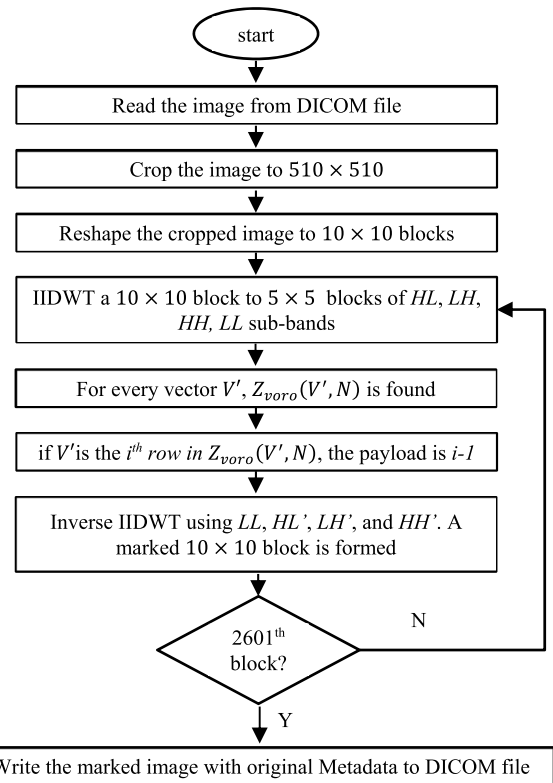


FIGURE 4. The extracting phase of the proposed algorithm.

TABLE 8. The decoding procedure for  $V'_{HL}$ .

$V'$	$Z_{voro}(V', 1)$	Extracted bit
$[63, 82, -75, 83, 27]$	$[63, 82, -74, 83, 27]$	0
	$[63, 82, -75, 83, 28]$	
$[63, 82, -75, 83, 27]$	$[63, 82, -74, 83, 27]$	1
	$[63, 82, -75, 83, 27]$	

repeating the same procedure for every row in  $HL$ ,  $LH$ , and  $HH$  sub-bands. For a given  $510 \times 510$  image there are 2601 blocks of size  $10 \times 10$  and as a consequence, there are  $2601 \times 3$  sub-bands of the size  $5 \times 5$ . By repeating the algorithm for the rest of the blocks, the embedded payload, including the Metadata, the patient's photo, and the watermark image is extracted. The final step of the extracting phase is to replace the fake Metadata with the extracted original Metadata and generate a reconstructed DICOM image.

**V. EXPERIMENTAL RESULTS**

In this section, the performance of the proposed data-hiding algorithm is evaluated and compared against several recent algorithms proposed in the literature. The experimental results are divided into four parts. In section V-A the impact of changing the index  $N$  on the embedding capacity for random vectors is provided. In section V-B, the impact of changing the index  $N$  on the total amount of embedding capacity on medical images is provided. In sections V-C the Metadata of the patient, the patient's photo, and the watermark image are

embedded in selected medical images with DICOM format. In these three experiments, a database of 20 medical DICOM files has been used. Finally, the proposed data-hiding algorithm is compared with similar medical image watermarking algorithms in sections V-D. In this experiment, 14 images from [33], 12 images from [34], and 24 images from [35] were used.

The most common measure for imperceptibility analysis in the literature is the PSNR. PSNR measures the amount of distortion between two images. The value of PSNR can be calculated as:

$$P(I1, I2) = 10 \log_{10} \left( N_{rows} \times N_{cols} \times \frac{\max(I1)^2}{MSE} \right) \quad (27)$$

where  $I1$  is the first image,  $I2$  is the second image,  $N_{rows}$  is number rows,  $N_{cols}$  number of columns,  $\max(I1)$ , is the maximum level of intensity, and MSE is the Mean Squared Error defined as follows:

$$MSE(I1, I2) = \frac{1}{N_{rows} * N_{cols}} \sum_i \sum_j (I1(i, j) - I2(i, j))^2 \quad (28)$$

The second imperceptibility measure that we use is the number of pixels changed ratio (NPCR). To calculate NPCR the number of pixels that are changed after embedding is divided by the number of all pixels. The structural similarity index measurement (SSIM) is another common measure of distortions between an image and its altered version.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\delta_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\delta_x^2 + \delta_y^2 + C_2)} \quad (29)$$

where  $\mu_x$  is the average of  $x$ ,  $\mu_y$  is the average of  $y$ ,  $\delta_x^2$  is the variance of  $x$ ,  $\delta_y^2$  is the variance of  $y$ ,  $\delta_{xy}$  is the covariance of  $x$  and  $y$ ,  $C_1 = (k1L)^2$ -Constant to avoid instability when  $\mu_x^2 + \mu_y^2$  is close to zero,  $k1 = 0.01$ ,  $L = 255$ ,  $C_2 = (k2L)^2$ -Constant to avoid instability when  $\delta_x^2 + \delta_y^2$  is close to zero,  $k2 = 0.03$ . The PSNR is not an objective measure of perceptual quality whereas SSIM is a common measure of perceptual quality. The embedding capacity of data-hiding algorithms are usually measured by the value of BPP. To calculate BPP, the number of embedded bits is divided by the number of pixels used.

### A. THE IMPACT OF INDEX $N$ ON THE EMBEDDING CAPACITY

The lattice vector quantization onto the points of a sub-lattice depends on the index of the generator which has generated the sub-lattice. The number of lattice points and  $Z_5$  points within the Voronoi of any sub-lattice point are increased when the index  $N$  increases. In this part of the experiment, we are going to check the impact of the index  $N$  on the performance of the data-hiding algorithm in terms of the embedding capacity and the perceptual quality of the cover image.

In section IV-A,  $V = [63, 82, -75, 83, 27]$  as a random vector was used to generate a sample  $Z_{voroi}(V, N)$  for index  $N = 1$ . However, if the same  $V$  is quantized onto a sub-lattice of  $A_5$  lattice with a given index  $N = 32$ , a different  $Z_{voroi}(V, N)$  with a larger number of points will be found. If the index of the sub-lattice is  $N = 32$  then the result of the quantization of  $V$  will be  $\lambda' = [62.932, 82.874, -75.632, 82.734, 27.386]$  and there are eighty  $Z_5$  points inside the Voronoi of  $\lambda'$ . The embedding capacity for a given vector depends on the coarse degree of quantization. The embedding capacity for a given vector  $V$  is calculated as

$$Capacity(V, N) = \lfloor \log_2(\text{rowsin}Z_{voroi}(V, N)) \rfloor \quad (30)$$

The number of points in  $Z_{voroi}$  is also dependent on the input vector. In other words, the embedding capacity per vector is not the same for different vectors. For example, if  $Z_{voroi}$  in Table 7 is considered, the capacity is 1 bit per 5 pixels or 0.2 BPP. To show the relation between the input vector and the number of points in  $Z_{voroi}(V, N)$ , 1000 sample random vectors are generated and the corresponding embedding capacity is depicted in Table 9.

**TABLE 9. Average Number of  $Z_5$  points inside the Voronoi of 1000 random  $V$ , for different indexes  $N = 1, 16, 256, 4096, 65536, 1048576$ .**

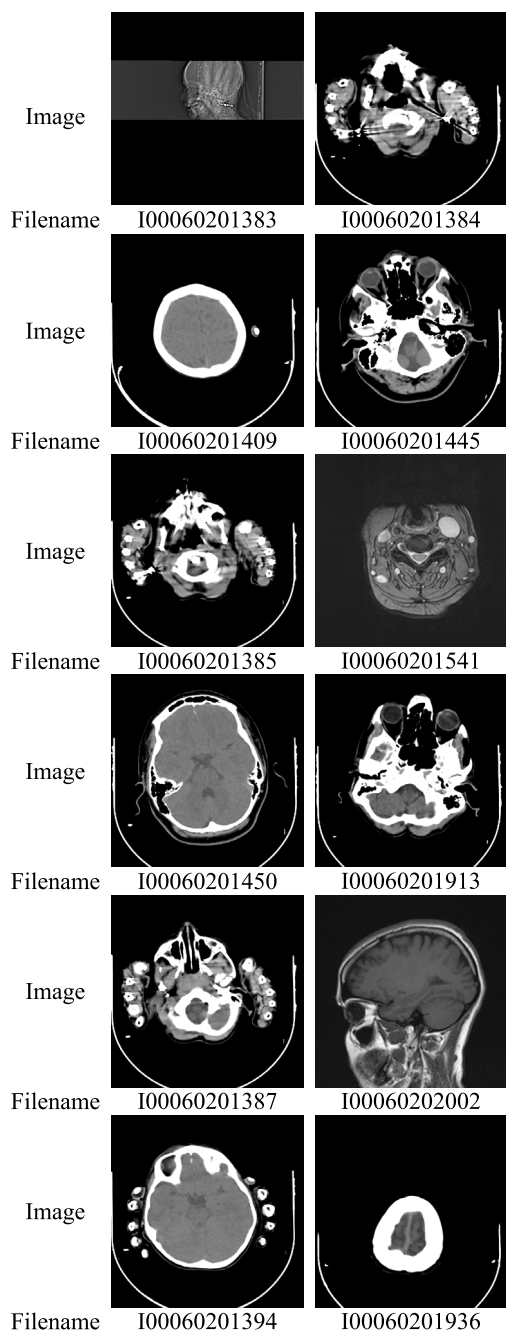
N	Average number of points in $Z_{voroi}(V, N)$	Standard Deviation of points in $Z_{voroi}(V, N)$	Average Capacity (bits)
1	2.67	0.74	1
32	79.73	4.59	6
1024	2510.72	26.90	11
32768	68955.30	276.52	16
1048576	743088.22	1705.61	19

According to Table 9, the embedding capacity for the same random vector increases when the index is increased and it remains almost constant for the same index across different vectors, especially for  $N > 32$ . The total embedding capacity for a given  $512 \times 512$  image is calculated as follows. There are 2601 blocks of the size  $10 \times 10$  in a cropped image of the size  $510 \times 510$  and as a consequence, there are  $2601 \times 3$  sub-bands of the size  $5 \times 5$ . The total capacity of the proposed embedding algorithm in a  $512 \times 512$  image will be determined by:

$$totalcapacity = \sum_{i=1}^{2601 \times 3 \times 5} Capacity(V_i, N) \quad (31)$$

It is also possible to trade the quality for a higher capacity. In other words, if the approximation coefficients are also used for the embedding, then the capacity will increase but the quality will decrease. Therefore, the total capacity of the proposed embedding algorithm in a  $512 \times 512$  image will be determined by:

$$totalcapacity = \sum_{i=1}^{2601 \times 4 \times 5} Capacity(V_i, N) \quad (32)$$



**FIGURE 5.** Twelve Selected medical images (out of 20 experimented) with 512 widths and 512 heights. To avoid confusion between similar images, the filenames are included too.

**B. THE IMPACT OF N ON THE TOTAL EMBEDDING CAPACITY OF MEDICAL IMAGES**

In this section the impact of the selected index for the  $A_5$  sub-lattice vector quantization on the total embedding capacity of the proposed algorithm is studied. According to the embedding algorithm described in section V-A, the cover image is first divided into  $10 \times 10$  blocks and then these blocks are transformed into wavelet domains.

The embedding process is done on 3 sub-bands out of 4 available sub-bands and the approximation coefficients are

**TABLE 10.** The impact of index n on the embedding capacity and imposed distortion for embedding in the first block of 20 images.

N	16	256	4096	65536	1048576
Average PSNR (dB)	129.12	127.24	124.72	122.98	122.98
Average embedding capacity (bits)	119	216	312	408	487

**TABLE 11.** The impact of index N on the average embedding capacity and PSNR for embedding in the first 128 blocks of 20 images.

N	32	1024	32768	1048576
Average PSNR	109.53 dB	104.89 dB	99.815 dB	96.63 dB
Average embedding capacity	7872.25 bits	17298.65 bits	26897.1 bits	28797.05 bits

left intact. Therefore, for every block, there are 3 sub-bands each one having 5 rows, a total of 15 rows per block. The number of bits per block is dependent on the value of the row vector and is not consistent for all vectors. In this experiment, 20 medical images have been selected for embedding random numbers. Selected medical images are illustrated in FIGURE 5.





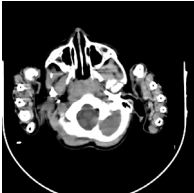
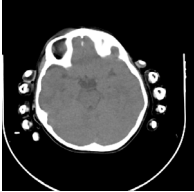
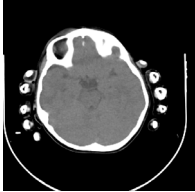
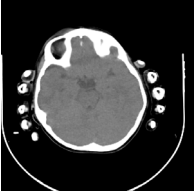
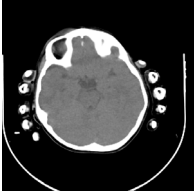
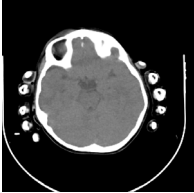
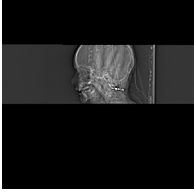
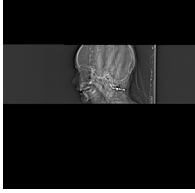
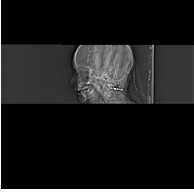
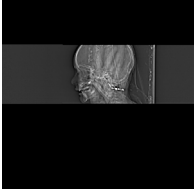
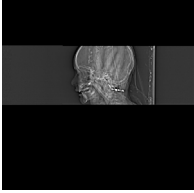
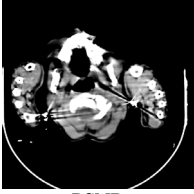
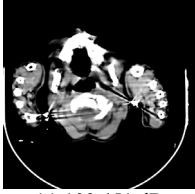
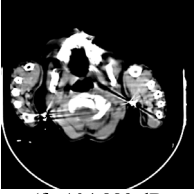
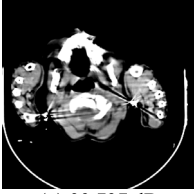
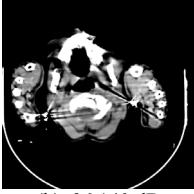





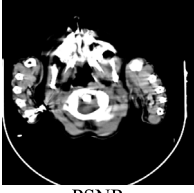
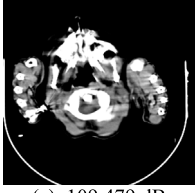
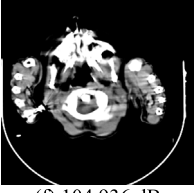
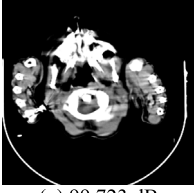
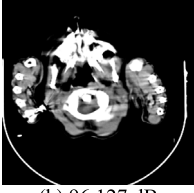
For every row vector  $V_Z$  the set of  $Z_{voro}$  is calculated and a random number in the range of  $[1, Capacity(V_Z, N) - 1]$  is embedded. According to our experimental results, presented in Table 10, the embedding capacity is directly related to the value of the index  $N$  as well as the number of embedded blocks. In other words, increasing the index or the number of embedded blocks improves the embedding capacity. However, increasing the embedding capacity imposes higher distortions and lower PSNR values.

According to Table 10, the distortion between the embedded image and the original image increases as the value of index  $N$  is increased and consequently, the PSNR value decreases. On the other hand, the embedding capacity is increased as the value of index  $N$  is increased. Increasing the number of embedded blocks can impact both the distortion and the embedding capacity.

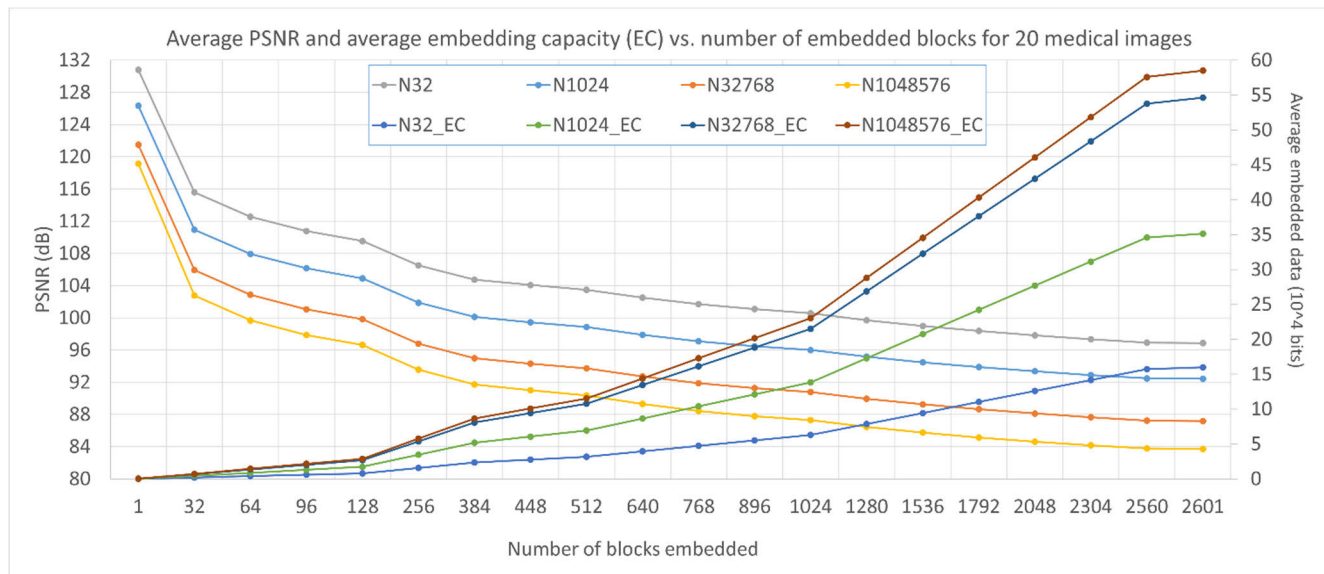
Embedding in the first 128 blocks decreases the PSNR values and increases the average embedding capacity for the same index  $N$ . Details of the average PSNR values and the average embedding capacities for embedding in the first 128 blocks of the 20 images are shown in Table 6.

To demonstrate the impact of the index  $N$  and the number of embedded blocks on the perceptual quality of the cover images, the original images and the cover images with 128 embedded blocks are provided in FIGURE 6 with corresponding PSNR values and SSIM values.

The maximum embedding capacity of the proposed embedding algorithm based on the similar sub-lattices of  $A_5$  is achieved for the index  $N = 1048576$  and the number of embedded blocks = 2601. The average PSNR values of the 20 embedded medical images as well as the corresponding average embedding capacities are plotted as a function of

Filename	Original image	$N = 32$	$N = 1024$	$N = 32768$	$N = 1048576$
100060201387					
	PSNR	(a) 109.503 dB	(b) 104.869 dB	(c) 99.767 dB	(d) 96.575 dB
	Embedded bits	7869 bits	17319 bits	26898 bits	28795 bits
100060201394					
	PSNR	(e) 109.566 dB	(f) 104.985 dB	(g) 99.727 dB	(h) 96.548 dB
	Embedded bits	7838 bits	17370 bits	26878 bits	28918 bits
100060201383					
	PSNR	(e) 109.327 dB	(f) 104.854 dB	(g) 100.018 dB	(h) 97.313 dB
	Embedded bits	7830 bits	17377 bits	26941 bits	28680 bits
100060201384					
	PSNR	(e) 109.451 dB	(f) 104.889 dB	(g) 99.737 dB	(h) 96.140 dB
	Embedded bits	7867 bits	17208 bits	26861 bits	28873 bits
100060201445					
	PSNR	(e) 109.511 dB	(f) 104.945 dB	(g) 99.853 dB	(h) 96.641 dB
	Embedded bits	7954 bits	17230 bits	26902 bits	28826 bits
100060201385					
	PSNR	(e) 109.479 dB	(f) 104.936 dB	(g) 99.723 dB	(h) 96.127 dB
	Embedded bits	7846 bits	17328 bits	27033 bits	28840 bits
	SSIM	1.0000	1.0000	1.0000	1.0000

**FIGURE 6.** Two original medical images and their embedded cover images after embedding in the first 128 blocks are provided with the PSNR values, the number of embedded bits, and SSIM values.



**FIGURE 7.** The vertical axis on the left shows the average PSNR and the vertical axis on the right shows average embedding capacity for 20 medical images vs. number of embedded blocks.

the number of embedded blocks in **FIGURE 7** for different index values. According to **FIGURE 7**, it is obvious that the distortion between the embedded image and the original image and the average or embedding capacity increases as the number of embedded blocks is increased, despite the value of index  $N$ . Besides, according to **FIGURE 7** for the same number of embedded blocks, the distortion between the embedded image and the original image increases as index  $N$  is increased.

**C. EMBEDDING THE METADATA IN THE SELECTED MEDICAL IMAGES**

As stated in section **IV-A**, the entire Metadata of the selected medical images includes less than 2 KB of data. The watermark image, which is shown in **FIGURE 8**, is a binary image with dimensions of  $180 \times 50$ . Thus,  $180 \times 50 \div 4 = 2250$  bytes in the payload data are reserved for the watermark.

*Copy righted*

**FIGURE 8.** The watermark image.

The patient photo is a gray-scale image with  $50 \times 50$  pixels, and 5000 bytes of payload data must be reserved for it. Less than 6 KB will be embedded in a typical medical image. The experimental results of embedding the patient Metadata, his/her photo, and the watermark image in the 20 selected medical images are provided in **TABLE 12**.

According to **TABLE 12** the number of required blocks to be used in the embedding process of the Metadata, patients' photo, and the watermark image, is constant for each index  $N$  for different image, and PSNR values range from 93.573 dB to 101.076 dB.

**TABLE 12.** The experimental results of embedding the Metadata, patients' photos, and the watermark in the 20 selected medical images.

N	Average PSNR values	
32	PSNR (dB)	101.076 dB
	required blocks	1792
1024	PSNR (dB)	100.114 dB
	required blocks	768
32768	PSNR (dB)	96.783 dB
	required blocks	512
1048576	PSNR (dB)	93.573 dB
	required blocks	512

As demonstrated in Section **IV**, the proposed data-hiding algorithm based on  $A_5$  lattice vector quantization is capable of embedding a high amount of data. Furthermore, for the same embedding capacity, the proposed algorithm demonstrates higher perceptual quality in terms of PSNR compared to similar algorithms. Also, this high capacity of embedding is consistent among different medical images. Finally, the proposed data-hiding algorithm can hide the data inside the entire cover image and it is capable of extracting the embedded data without requiring the original image.

**D. COMPARISON WITH RELATED WORKS**

The results of the proposed algorithm are compared with the related works on three medical image datasets. The medical image datasets comprise an MRI image dataset [33], a radiographic image dataset [34], and an ultrasound image dataset [35]. The datasets are in BMP format. The comparison of the PSNR, SSIM, NPCR, and BPP values between the proposed algorithm index  $N = 1024$ , with similar methods proposed from 2017 to 2021 is presented in **TABLE 13**.

According to our experiments, as the embedded payload is increased, that is higher BPP, the PSNR values decrease.

TABLE 13. Imperceptibility results in comparison with related work.

Images		Ours	[21]	[36]	[37]
CT	PSNR	58.45	55.85	63.67	44.19
	SSIM	0.9987	0.9997	1.00	0.9913
	NPCR	10.18%	50.48%	-	0.9959
	BPP	0.771	0.3974	-	-
X-Ray		Ours	[21]	[38]	[20]
	PSNR	55.63	57.04	53.47	55.12
	SSIM	0.9693	0.9998	0.9736	1.00
	NPCR	18.73%	50.48%	-	-
Ultrasound		Ours	[21]	[19]	[39]
	PSNR	53.88	44.91	48.15	56.36
	SSIM	0.9378	0.9961	-	-
	NPCR	27.62%	69.56%	-	-
	BPP	0.774	0.3130	-	0.09

TABLE 14. Speed of the proposed algorithm.

	Embedding time (ms)	Extracting time (ms)	Embedding time (ms) [40]	Extracting time (ms) [40]
CT Images	209.36	208.23	306.0945	334.2892
X-Ray	212.29	202.21	418.3546	308.6833
Ultrasound	211.09	201.15	320.7444	295.8444

This is logical because to increase the payload we have to embed data in more blocks or use a larger index  $N$  and consequently impose more distortion.

### E. COMPUTATIONAL COMPLEXITY AND SPEED OF THE PROPOSED ALGORITHM

To calculate the speed and computational complexity of the proposed algorithm the simulations are performed on the same computer system with a processor Intel(R) Core (TM) i9-10850K CPU @ 3.60GHz 3.60 GHz, with 64 GB of RAM, Windows 10, and Matlab R2019b for the same three datasets and the results are reported in TABLE 14.

### VI. CONCLUSION

In this paper, a novel data-hiding algorithm based on IIDWT and the  $A_5$  lattice vector quantization is proposed that can embed the payload data inside the entire image. The proposed data-hiding algorithm is a fragile data-hiding algorithm that enables us to detect any tampering efforts. The proposed data-hiding algorithm embeds the data by finding the  $Z_{voro}$ , the set of  $Z_5$  points that coexist in the same Voronoi as the current vector of the DWT coefficients. Then the vector is substituted by a proper member of the  $Z_{voro}$ . The embedded data is recovered in the receiver by doing the same procedure and finding the index of the received vector in the same  $Z_{voro}$ .

The simulation results show that the proposed algorithm outperforms similar proposed algorithms. The proposed data-hiding algorithm is applied to 62 BMP medical images from an MRI image dataset [33], a radiographic image dataset [34], and an ultrasound image dataset [35]. The experimental results show that the proposed algorithm can embed more data than similar algorithms in the standard test images while keeping the PSNR above 53 dB. In addition, the proposed algorithm is applied to 20 DICOM medical images to embed

the Metadata of the file together with the patient's photo and a watermark as MAC. The fragility of the algorithm enables us to detect any tampering efforts and keep the integrity of the system. In addition, the PSNR value which are above 53 dB guarantees no adverse effect in the process of diagnosis. For the future works of the proposed algorithm the  $A_6$ ,  $A_7$ , and  $A_8$  similar sub-lattice are considered for increasing the capacity of the data hiding in various multimedia applications. Furthermore, other applications of the proposed algorithms for the Internet of Things and smart cities are under investigation.

### REFERENCES

- [1] X. Liu, J. Lou, H. Fang, Y. Chen, P. Ouyang, Y. Wang, B. Zou, and L. Wan, "A novel robust reversible watermarking scheme for protecting authenticity and integrity of medical images," *IEEE Access*, vol. 7, pp. 76580–76598, 2019, doi: 10.1109/access.2019.2921894.
- [2] H. Zarrabi, A. Emami, P. Khadivi, N. Karimi, and S. Samavi, "Bless-Mark: A blind diagnostically-lossless watermarking framework for medical applications based on deep neural networks," *Multimedia Tools Appl.*, vol. 79, nos. 31–32, pp. 22473–22495, Aug. 2020, doi: 10.1007/s11042-020-08698-9.
- [3] C. K. Tan, J. C. Ng, X. Xu, C. L. Poh, Y. L. Guan, and K. Sheah, "Security protection of DICOM medical images using dual-layer reversible watermarking with tamper detection capability," *J. Digit. Imag.*, vol. 24, no. 3, pp. 528–540, Jun. 2011.
- [4] A. Anand and A. K. Singh, "Watermarking techniques for medical data authentication: A survey," *Multimedia Tools Appl.*, vol. 80, no. 20, pp. 30165–30197, Aug. 2021, doi: 10.1007/s11042-020-08801-0.
- [5] J. Conway and N. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 227–232, Mar. 1982.
- [6] E. Akhtarkavan and M. F. M. Salleh, "Multiple descriptions coinciding lattice vector quantizer for wavelet image coding," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 653–661, Feb. 2012, doi: 10.1109/TIP.2011.2164419.
- [7] E. Akhtarkavan and M. F. M. Salleh, "Multiple description lattice vector quantization using multiple A4 quantizers," *IEICE Electron. Exp.*, vol. 7, no. 17, pp. 1233–1239, 2010.
- [8] E. Akhtarkavan, M. Salleh, and O. Sidek, "Multiple descriptions video coding using coinciding lattice vector quantizer for H.264/AVC and motion JPEG2000," *World Appl. Sci. J.*, vol. 21, no. 2, pp. 157–169, 2013. [Online]. Available: [https://www.idosi.org/wasj/wasj21\(2\)13/2.pdf](https://www.idosi.org/wasj/wasj21(2)13/2.pdf)
- [9] P. Rahmani and G. Dastghaibifard, "Two reversible data hiding schemes for VQ-compressed images based on index coding," *IET Image Process.*, vol. 12, no. 7, pp. 1195–1203, Jul. 2018, doi: 10.1049/iet-ipr.2016.0618.
- [10] Z. Pan and L. Wang, "Novel reversible data hiding scheme for two-stage VQ compressed images based on search-order coding," *J. Vis. Commun. Image Represent.*, vol. 50, pp. 186–198, Jan. 2018.
- [11] E. Akhtarkavan, B. Majidi, M. F. M. Salleh, and J. C. Patra, "Fragile high capacity data hiding in digital images using integer-to-integer DWT and lattice vector quantization," *Multimedia Tools Appl.*, vol. 79, nos. 19–20, pp. 13427–13447, May 2020, doi: 10.1007/s11042-020-08662-7.
- [12] X. Liao, Y. Yu, B. Li, Z. Li, and Z. Qin, "A new payload partition strategy in color image steganography," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 3, pp. 685–696, Jan. 2020, doi: 10.1109/TCSVT.2019.2896270.
- [13] X. Liao, J. Yin, M. Chen, and Z. Qin, "Adaptive payload distribution in multiple images steganography based on image texture features," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 897–911, Mar. 2022, doi: 10.1109/TDSC.2020.3004708.
- [14] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—A survey," *Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.
- [15] K. Fares, A. Khaldi, K. Redouane, and E. Salah, "DCT & DWT based watermarking scheme for medical information security," *Biomed. Signal Process. Control*, vol. 66, Apr. 2021, Art. no. 102403.
- [16] G. Gao, S. Tong, Z. Xia, B. Wu, L. Xu, and Z. Zhao, "Reversible data hiding with automatic contrast enhancement for medical images," *Signal Process.*, vol. 178, Jan. 2021, Art. no. 107817, doi: 10.1016/j.sigpro.2020.107817.

- [17] Y. Yang, X. Xiao, X. Cai, and W. Zhang, "A secure and high visual-quality framework for medical images by contrast-enhancement reversible data hiding and homomorphic encryption," *IEEE Access*, vol. 7, pp. 96900–96911, 2019, doi: [10.1109/ACCESS.2019.2929298](https://doi.org/10.1109/ACCESS.2019.2929298).
- [18] F. Kahlessenane, A. Khaldi, R. Kafi, and S. Euschi, "A robust blind medical image watermarking approach for telemedicine applications," *Cluster Comput.*, vol. 24, no. 3, pp. 2069–2082, Sep. 2021.
- [19] H. S. Alshambari, "Medical image watermarking for ownership & tamper detection," *Multimedia Tools Appl.*, vol. 80, no. 11, pp. 16549–16564, May 2021.
- [20] E. E.-D. Hemdan, "An efficient and robust watermarking approach based on single value decompression, multi-level DWT, and wavelet fusion with scrambled medical images," *Multimedia Tools Appl.*, vol. 80, no. 2, pp. 1749–1777, Jan. 2021.
- [21] N. Zermi, A. Khaldi, M. R. Kafi, F. Kahlessenane, and S. Euschi, "Robust SVD-based schemes for medical image watermarking," *Microprocessors Microsystems*, vol. 84, Jul. 2021, Art. no. 104134, doi: [10.1016/j.micpro.2021.104134](https://doi.org/10.1016/j.micpro.2021.104134).
- [22] R. Thanki and A. Kothari, "Multi-level security of medical images based on encryption and watermarking for telemedicine applications," *Multimedia Tools Appl.*, vol. 80, no. 3, pp. 4307–4325, Jan. 2021.
- [23] R. Bhardwaj and A. Singh, "An efficient reversible and secure patient data hiding algorithm for E-healthcare," *Multimedia Tools Appl.*, vol. 80, nos. 21–23, pp. 31687–31703, Sep. 2021, doi: [10.1007/s11042-021-10892-2](https://doi.org/10.1007/s11042-021-10892-2).
- [24] P. V. Sanivarapu, "Adaptive tamper detection watermarking scheme for medical images in transform domain," *Multimedia Tools Appl.*, vol. 81, pp. 11605–11619, Feb. 2022.
- [25] S. P. Vaidya, "Fingerprint-based robust medical image watermarking in hybrid transform," *Vis. Comput.*, pp. 1–16, Jan. 2022.
- [26] P. Singh, K. J. Devi, H. K. Thakkar, and K. Kotecha, "Region-based hybrid medical image watermarking scheme for robust and secured transmission in IoMT," *IEEE Access*, vol. 10, pp. 8974–8993, 2022.
- [27] N. Agarwal and P. K. Singh, "Discrete cosine transforms and genetic algorithm based watermarking method for robustness and imperceptibility of color images for intelligent multimedia applications," *Multimedia Tools Appl.*, vol. 81, pp. 19751–19777, Jan. 2022.
- [28] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd ed. New York, NY, USA: Springer-Verlag, 1998.
- [29] J. H. Conway, E. M. Rains, and N. J. A. Sloane, "On the existence of similar sublattices," *Can. J. Math.*, vol. 51, no. 6, pp. 1300–1306, Dec. 1999.
- [30] J. H. Conway. (2023). *The On-Line Encyclopedia of Integer Sequences*. [Online]. Available: <https://oeis.org/search?q=1%2C16%2C256%2C4096&sort=&language=english&go=Search>
- [31] E. Akhtarkavan, "Multiple-description lattice vector quantization for image and video coding based on coinciding similar sublattices of an," Ph.D. dissertation, School Elect. Electron. Eng., Univ. Sci. Malaysia, George, Malaysia, 2012. [Online]. Available: <https://ee.eng.usm.my/eeacad/fadzli/abstract/Phd-06-12-Abstract.pdf>
- [32] M. Mustra, K. Delac, and M. Grgic, "Overview of the DICOM standard," in *Proc. 50th Int. Symp. (ELMAR)*, vol. 1, Sep. 2008, pp. 39–44.
- [33] K. S. Mader. (2022). *CT Medical Images CT Images From Cancer Imaging Archive With Contrast and Patient*. [Online]. Available: <https://www.kaggle.com/datasets/kmader/sim-medical-images>
- [34] T. Rahman, M. E. H. Chowdhury, and A. Khandakar. *COVID-19 Chest X-Ray Image Dataset*. [Online]. Available: <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>
- [35] M. Zukal, R. Beneš, P. Číka, and K. Říha. *Ultrasound Image Database*. [Online]. Available: [http://splab.cz/wp-content/uploads/2013/11/us\\_images.zip](http://splab.cz/wp-content/uploads/2013/11/us_images.zip)
- [36] U. Verma and N. Sharma, "Hybrid mode of medical image watermarking to enhance robustness and imperceptibility," *Int. J. Innov. Technol. Exploring Eng.*, vol. 9, no. 1, pp. 351–359, Nov. 2019.
- [37] A. Anand and A. K. Singh, "An improved DWT-SVD domain watermarking for medical information security," *Comput. Commun.*, vol. 152, pp. 72–80, Feb. 2020.
- [38] F. N. Thakkar and V. K. Srivastava, "A blind medical image watermarking: DWT-SVD based robust and secure approach for telemedicine applications," *Multimedia Tools Appl.*, vol. 76, no. 3, pp. 3669–3697, 2017, doi: [10.1007/s11042-016-3928-7](https://doi.org/10.1007/s11042-016-3928-7).
- [39] N. A. Memon and A. Alzahrani, "Prediction-based reversible watermarking of CT scan images for content authentication and copyright protection," *IEEE Access*, vol. 8, pp. 75448–75462, 2020.
- [40] G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni, and W. Su, "Distortionless data hiding based on integer wavelet transform," *Electron. Lett.*, vol. 38, no. 25, pp. 1646–1648, Dec. 2002.



**EHSAN AKHTARKAVAN** was born in Tehran, Iran, in 1981. He received the B.Sc. degree in hardware computer engineering from Islamic Azad University, Tehran, in 2004, the M.S. degree in computer system architecture from Islamic Azad University, Arak, Iran, in 2007, and the Ph.D. degree from the School of Electrical and Electronics Engineering, University of Science Malaysia, Penang, Malaysia. He worked as a Network Administrator and IT Manager at the Academic Center for Education, Culture and Research, Tehran, until July 2005. He is currently an Assistant Professor with the Department of Computer Engineering, Khatam University, Tehran, and a Computer Engineering Researcher at the Pasargad Institute for Advanced Innovative Solutions (PIAIS), Tehran. His current research interests include digital image watermarking, multiple descriptions image coding, and multiple descriptions video coding.



**BABAK MAJIDI** was born in Tehran, Iran, in 1977. He received the B.S. and M.S. degrees in computer engineering from the University of Tehran, Iran, in 2000 and 2003, respectively, and the Ph.D. degree in computer engineering from the Swinburne University of Technology, Melbourne, Australia, in 2014.

From 2014 to 2021, he was an Assistant Professor with the Department of Computer Engineering, Khatam University, Tehran, where is an Associate Professor, since 2021. He is the author of more than 50 research articles. His research interests include machine learning, smart environments, and eco-cyber-physical systems.



**AMIRSINA MANDEGARI** was born in Tehran, Iran, in 1997. He received the B.Sc. degree in computer science from the University of Qom, Qom, Iran, in 2021. He is currently pursuing the master's degree in artificial intelligence and robotics engineering with Khatam University, Tehran.

He has been a Research Assistant at the Robotics and Artificial Intelligence Laboratory, Khatam University, since 2022.