Efficient Biomedical Signal Security Algorithm for Smart Internet of Medical Things (IoMTs) Applications

Achraf Daoui 1, Mohamed Yamni 2, Hicham Karmouni 1, Mhamed Sayyouri 1, Hassan Qjidaa 2, Saad Motahhir 1, Ouazzani Jamil 3, Walid El-Shafai 4,5,*, Abeer D. Algarni 6, Naglaa F. Soliman 6 and Moustafa H. Aly 7

1 Engineering, Systems and Applications Laboratory, National School of Applied Sciences, Sidi Mohamed Ben Abdellah University, BP 72, My Abdallah Avenue Km. 5 Imouzzer Road, Fez 30000, Morocco
2 CED-ST, STIC, Laboratory of Electronic Signals and Systems of Information LESSI, Faculty of Science Dhar El Mahrez, University Sidi Mohamed Ben Abdellah, Fez 30000, Morocco
3 Systems and Sustainable Environment Laboratory (SED), Faculty of Engineering Sciences (FSI), Private University of Fez (UPF), Fez 30000, Morocco
4 Security Engineering Lab, Computer Science Department, Prince Sultan University, Riyadh 11586, Saudi Arabia
5 Department Electronics and Electrical Communications Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf 32952, Egypt
6 Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
7 Electronics and Communications Engineering Department, College of Engineering and Technology, Arab Academy for Science, Technology and Maritime Transport, Alexandria 1029, Egypt

* Correspondence: welshafai@psu.edu.sa or walid.elshafai@el-eng.menofia.edu.eg

Abstract: Due to the rapid development of information and emerging communication technologies, developing and implementing solutions in the Internet of Medical Things (IoMTs) field have become relevant. This work developed a novel data security algorithm for deployment in emerging wireless biomedical sensor network (WBSN) and IoMTs applications while exchanging electronic patient folders (EPFs) over unsecured communication channels. These EPF data are collected using wireless biomedical sensors implemented in WBSN and IoMTs applications. Our algorithm is designed to ensure a high level of security for confidential patient information and verify the copyrights of bio-signal records included in the EPFs. The proposed scheme involves the use of Hahn’s discrete orthogonal moments for bio-signal feature vector extraction. Next, confidential patient information with the extracted feature vectors is converted into a QR code. The latter is then encrypted based on a proposed two-dimensional version of the modified chaotic logistic map. To demonstrate the feasibility of our scheme in IoMTs, it was implemented on a low-cost hardware board, namely Raspberry Pi, where the quad-core processors of this board are exploited using parallel computing. The conducted numerical experiments showed, on the one hand, that our scheme is highly secure and provides excellent robustness against common signal-processing attacks (noise, filtering, geometric transformations, compression, etc.). On the other hand, the obtained results demonstrated the fast running of our scheme when it is implemented on the Raspberry Pi board based on parallel computing. Furthermore, the results of the conducted comparisons reflect the superiority of our algorithm in terms of robustness when compared to recent bio-signal copyright protection schemes.

Keywords: biomedical sensors; chaotic system; discrete moments; electronic patient folders; encryption; IoMTs; modified logistic map; WBSNs; zero-watermarking

1. Introduction

Recently, the development and implementation of e-health solutions have become current trends in scientific research. Indeed, these solutions promote the integration of
the latest information and communication technologies in the healthcare sector [1]. E-health applications include teleradiology [2], telediagnosis [3], teleophthalmology [4], telemedicine [5], image classification via Convolutional Neural Networks (CNN) [6], cell segmentation and tracking [7,8], face recognition [9], etc. These applications are very promising because they can significantly improve the access, equity, and quality of health services by linking institutions and professionals in the healthcare sector. In addition, they minimize geographical and physical barriers. However, technologies for transmitting and accessing medical information raise critical issues that need to be urgently addressed, including security-related issues [1]. In addition, unauthorized access to private information, such as laboratory tests, pathology history, radiology, physical examinations, etc., can negatively affect patients’ privacy [10]. Moreover, medical-related identity crime is a growing threat that already costs billions of dollars each year, and tampering with medical information can put a person’s health at risk due to misdiagnosis, delayed treatment, or incorrectly prescribed medications [1].

Healthcare professionals currently exchange electronic patient folders (EPFs) over the Internet worldwide via applications such as telemedicine [11]. An EPF contains information about the patient’s health, such as laboratory tests, disease history, radiology, physical examinations, etc. [12]. In addition, the EPF also contains medical images, bio-signals, diagnostic reports, etc. [1]. However, due to the use of many routers when transmitting EPFs via the IoMTs, EPFs become vulnerable to cyber-attacks [13]. In addition, medical multimedia (2D and 3D images, bio-signals, etc.) can be distorted when EPFs are exchanged via IoMTs [14]. Moreover, the transmission of EPFs can be unsecured, which causes a serious risk since EPFs contain confidential medical information about the patients [15]. To overcome this issue, researchers proposed the use of certain techniques to achieve integrity and tamper resistance of EPFs’ content. Some of these techniques include watermarking [16,17], zero-watermarking [18,19], and encryption schemes [20,21].

In general, watermarking schemes proceed in two phases: the first one is embedding secret binary information called watermarks in the original (host) signal. The second phase consists of extracting the embedded information from the host signal to confirm the originality of the copyrighted signal. However, the watermark insertion into bio-signals, especially ECGs, can lead to visual artifacts in this signal. Such artifacts can lead to misdiagnosis, and the corresponding treatment can have fatal consequences for the patient [22]. To avoid this problem, researchers are using zero-watermarking methods, which allow protection of the copyrights of bio-signals without introducing any modification of these signals. Indeed, Ali et al. [18] proposed a zero-watermarking scheme to protect patient identifiers (IDs) and detected vocal cord disorders. Experimental results showed that Ali et al.’s scheme could successfully assure bio-signals’ imperceptibility and exhibited good reliability in ID extraction. However, since Ali et al.’s algorithm is based on a spatio-temporal descriptor, namely the local bit pattern (LBP), the robustness of this algorithm is weak in noisy environments. To address this problem, Daoui et al. proposed in [23] a zero-watermarking scheme in the transform domain using the sine cosine algorithm (SCA) [24] and Krawtchouk–Charlier fractional-order transform. The simulation results showed that the proposed scheme provides both imperceptibility of bio-signals and high robustness against different types of signal-processing attacks, including noise attacks. Daoui et al. recently proposed in [19] a novel scheme for biomedical zero-watermarking using quaternion Shmaliy moments and short-time Fourier transform (STFT). This scheme could provide good robustness to different types of signal-processing attacks. Encryption schemes can also be implemented to provide a high degree of security when exchanging bio-signals via unsecured communication channels [20,25–28].

By analyzing the different bio-signal security schemes, it could be concluded that these schemes are more focused on either protecting the bio-signal copyrights or ensuring a strong security level when communicating bio-signals via unsecured communication channels such as IoMTs. However, these security schemes usually ignore the problem of protecting confidential patient information, such as personal information, identifiers, laboratory test
results, medical history, radiologist reports, physical examinations, etc. This information is included with the bio-signal records in the EPFs. Then, the EPFs can be communicated between healthcare analysis service centers through unsecured communication channels such as IoMTs. This problem can have a harmful impact on patients' privacy if their confidential information is misused by unauthorized users of IoMTs. To overcome this limitation, a new security scheme is presented in the present work. The proposed scheme is used to both: (i) protect the copyrights of the bio-signals without modifying these signals during the process of watermark embedding, which ensures perfect imperceptibility, and (ii) ensure a strong security level of confidential patient information, which is stored in a plain text file. For this purpose, a new 2D chaotic system was used for encrypting the private patient plain text information.

The proposed scheme involves the selection of four frames from the original bio-signal. Then, the feature vectors are extracted from these frames using Hahn’s discrete orthogonal moments. Next, the extracted feature vectors are binarized and then encrypted based on the proposed 2D chaotic system that is called the modified sine cosine logistic map. Then, a text file containing the encrypted feature vectors is produced. Meanwhile, confidential patient information is stored in another text file, which is combined with the text file containing the encrypted feature vectors in order to generate a single text file. This text file is then converted into a QR code which is encrypted via the proposed chaotic system with the control parameters of this system used as security keys. Finally, the encrypted QR code and the original bio-signal are included in the EPF, which can be safely transmitted through an unsecured communication channel using IoMTs.

The analysis of bio-signal security schemes also indicated that most of these schemes are designed to be implemented on personal computers (PCs (laptop or desktop)). However, the use of PCs in the development of solutions in the Internet of Things (IoT) presents some major drawbacks, namely: (i) the large size and weight of the PCs, which limits their portability. As a result, PC-based applications are rarely deployed in IoTs [29,30]. (ii) In addition, PCs are expensive in terms of energy consumption, especially if they are used for long periods, which reduces their energy efficiency. (iii) Furthermore, PCs are generally costly, which limits their use in IoT-based projects and smart cities that require low-cost solutions [31]. To overcome these limitations, the proposed security scheme was implemented on a Raspberry Pi-embedded board, which is considered as one of the most appropriate hardware for IoMTs applications [32]. In fact, Raspberry Pi boards offer the following benefits: (i) low cost and (ii) reduced dimensions, since the average dimensions of various Raspberry Pi models are 85.60 mm (length) × 56.5 mm (width) × 17 mm (height). Therefore, these boards are desirable for their portability. In addition, (iii) Raspberry Pi boards are widely appreciated for their low power consumption. Indeed, Raspberry Pi 4 model B consumes between 3.8 W and 6.0 W. This makes Raspberry Pi boards efficient in terms of energy consumption. (iv) Furthermore, parallel computing is an extra advantage offered by Raspberry Pi boards. This advantage can be used to considerably accelerate the execution time of applications that are designed to run in parallel [33].

We can summarize the main contributions of our work as follows:

(i) Proposing a new 2D chaotic system based on the existing modified logistic map and then demonstrating the chaotic behavior of the proposed system.

(ii) Proposing a new security system to protect the confidentiality of electronic patient data (bio-signal records and patient personal information) in IoMTs.

(iii) Implementing the proposed security system on a Raspberry Pi board to take advantage of its multiple benefits (low cost, multi-core for parallel computing, low power consumption, portability, etc.).

(iv) Providing simulation and comparison results on a PC and a Raspberry Pi 4 Model B-embedded board to confirm the good robustness and the cost efficiency of the proposed system in IoMTs.

The rest of this work is organized in the following manner. In the second section, we outline the preliminaries employed within our work. The third section deals with
the proposed two-dimensional sine cosine chaotic map. The fourth section describes the proposed security scheme for bio-signals and patient data privacy preservation. The fifth section outlines details on the proposed rapid implementation of our scheme on Raspberry Pi. Simulation results and comparisons are discussed in the sixth section. Finally, conclusions and future work are given in the last section.

2. Preliminaries

2.1. Hahn Moments

The one-dimensional Hahn moments (1DHMs) are defined for a discrete one-dimension (1D) signal of the function \( f(x), x = 0, 1, \ldots, N - 1 \) by the following matrix product [34]:

\[
HM^{(a,b)} = H^{(a,b)} \times f
\]  
(1)

where \( H^{(a,b)} \) is the normalized HP matrix that is computed up to the polynomial order \( n \) with \( n < N \) and \((a,b)\) are the local parameters of HPs with \( b > -1 \) and \( a > -1 \).

The computation of 1DHMs up to the last order \( N \) using (1) requires the generation of an HP matrix \((H^{(a,b)})\) of size \( N \times N \). Then, (1) is used to compute the 1DHMs of the input signal \( f \) of size \( N \times 1 \). This process is time-consuming, especially for large-sized signals \((N > 1000 \text{ samples})\). To avoid this limitation, a fast method presented in [23,35] can be used. This method proceeds as follows: (i) converting the 1D signal \((f)\) of size \( N \times 1 \) into a 2D matrix denoted \( F \) of size \((K,K) = (\sqrt{N} \times \sqrt{N})\). (ii) Next, generating two HP matrices each of size \( K \times K \) rather than the generation of an HP matrix of size \( N \times N \) when using the conventional method (see (1)). (iii) Then, computing Hahn moments of \( F \)-matrix in the 2D domain (2DHMs) by using (2). This method considerably reduces the computation time of the input signal moments when compared to the conventional method. More details regarding this method can be found in [34].

\[
HM^{(a,1\alpha,2\beta)} = H_1^{(a,\alpha,1)} \times F \times (H_2^{(2\beta)})^T
\]  
(2)

where \( H_1^{(a,\alpha,1)} \) and \( H_2^{(2\beta)} \) represent HP matrices, each of size \( n \times K \) with \( n \leq K \) and \( K = \sqrt{N} \). These matrices are computed for different values of the local HP parameters \((a,\alpha,1,2\beta)\) with \( a,\alpha > -1 \) and \( b,\beta > -1 \). The symbol \( (.)^T \) denotes the operator of the matrix transpose.

In order to reconstruct the \( F \)-matrix from the computed 2DHMs, the following formula is used [34]:

\[
\hat{F} = (H_1^{(a,\alpha,1)})^T \times HM^{(a,1B,2\beta)} \times H_2^{(2\beta)}
\]  
(3)

where \( \hat{F} \) represents the reconstructed form of an \( F \)-matrix of size \((K,K) = (\sqrt{N} \times \sqrt{N})\). The \( \hat{F} \) matrix is then reshaped into a 1D vector of size \( N \times 1 \) for producing the reconstructed signal denoted \( \hat{f} \). This reconstructed signal is subject to certain degradation compared to its original version \((f)\). This degradation can be assessed by the mean square error (MSE) criterion given in (4) as follows [34]:

\[
MSE = \frac{1}{N} \sum_{x=0}^{N-1} (f(x) - \hat{f}(x))^2
\]  
(4)

If \( MSE = 0 \), both \( f \) and \( \hat{f} \) signals are considered identical.

Based on (1)–(2), we can notice that HMs are defined from HP matrices and the signal function \((f)\). The generation of an HP matrix starts with the computation of the initial values \( \tilde{H}_0^{(a,b)}(x) \) and \( \tilde{H}_1^{(a,b)}(x) \) by using the following equations [34]:

\[
\tilde{H}_0^{(a,b)}(x) = \sqrt{ \frac{(N-x)(\beta+x)}{x(N+a-x)}} \tilde{H}_0^{(a,b)}(x-1), x = 1,2,\ldots,N-1
\]  
(5)
The strengths of the QR code include (i) high data-encoding capacity, (ii) readability from any angle between 0 and 360 degrees, (iii) small dimensions and small size, (iv) strong resistance to noise and damage, and (v) ease of access by users with smartphones or other devices equipped with a digital camera. For these reasons, industries widely use QR codes in a variety of services and goods that they produce.

Indeed, Chen et al. [37] used QR codes to protect the copyright of images. Thulasidharan et al. presented in [38] a QR code-based blind digital watermarking scheme for protecting the copyright of images. Mathivanan et al. [39] used a QR code to secure remote transmission of patient diagnostic details with the ECG signal between medical centers. In [40], Mathivanan et al. used a QR code-based method for encrypting ECG bio-signals in order to secure the transmission of such signals via the Internet. Finally, Daoui et al. [41] used a QR code in robust encryption with a zero-watermarking scheme for image copyright protection. In this work, a QR code was employed to contain information about the protected image.

### 2.2. QR Code

The QR code is a type of two-dimensional barcode invented initially by Denso Wave Institute [36]. This code has good features that make it very useful in various applications. The strengths of the QR code include (i) high data-encoding capacity, (ii) readability from any angle between 0 and 360 degrees, (iii) small dimensions and small size, (iv) strong resistance to noise and damage, and (v) ease of access by users with smartphones or other devices equipped with a digital camera. For these reasons, industries widely use QR codes in a variety of services and goods that they produce.

Currently, QR codes are increasingly being used in the field of information security. Indeed, Chen et al. [37] used QR codes to protect the copyright of images. Thulasidharan et al. presented in [38] a QR code-based blind digital watermarking scheme for protecting the copyright of images. Mathivanan et al. [39] used a QR code to secure remote transmission of patient diagnostic details with the ECG signal between medical centers. In [40], Mathivanan et al. used a QR code-based method for encrypting ECG bio-signals in order to secure the transmission of such signals via the Internet. Finally, Daoui et al. [41] used a QR code in robust encryption with a zero-watermarking scheme for image copyright protection. In this work, a QR code was employed to contain information about the protected image.

### 2.3. Internet of Medical Things (IoMTs)

Advances in information and communication technology inspire medical services to move towards the use of virtual systems, cloud storage, remote access to patient information through distance applications, etc. Modern technology of the IoTs applied in medical services has proven to be extremely beneficial for the healthcare industry and for the public life of individuals [42]. The orientation of medical services and enterprises towards IoMTs applications aims to provide better-quality, cost-effective, and accessible healthcare. In addition, IoMT equipment helps improve the efficiency and quality of medical services and real-time diagnosis through sophisticated health information technology (health IT) ecosystems [43]. In addition, IoMTs includes the use of wearable sensor devices and security systems [44].
2.4. Modified Logistic Map

The modified logistic map (MLM) is a new chaotic system presented by Daoui et al. in [32]. This system is designed for use in security systems and is based on modifying the original logistic map (LM) [36]. The MLM provides better security and improved chaotic dynamics compared to the original LM. The mathematical model of the MLM is given by [41]:

\[
L_j = a(1 - b|\cos(j)|)L_{j-1}(1 - L_{j-1}), \text{ with } j = 1, \ldots, n \text{ and } 0 < L_0 < 1
\]  

(9)

where \(a\) and \(b\) are the control parameters of the MLM. The symbol \(|\cdot|\) represents the absolute value, and the symbol \(\cos(\cdot)\) is the cosine function.

The MLM exhibits chaotic behavior for \(3.56 \leq a \leq 4\) and \(0 \leq b \leq 0.25\). The latter represents the control parameters of the MLM. These parameters are real values that cause a large variation in the MLM output values \(L_j\) if they are changed by a minor variation of the order \(10^{-15}\). For this reason, \(a\) and \(b\) parameters can be used as security keys when designing MLM-based security schemes.

In the next section, we present a two-dimensional (2D) version of the MLM to achieve higher safety for the proposed security scheme.

3. Proposed 2D Chaotic Sine Cosine Logistic Map

Starting from the idea that a chaotic system with a large number of parameters can offer a good security level if the parameters are very sensitive to any small variation, we propose a 2D version of the MLM, which is called a 2D sine cosine logistic map (2DSCLM). The mathematical model of the 2DSCLM is given by:

\[
\begin{align*}
    x_i &= a(1 - b|\cos(j)|)x_{i-1}(1 - x_{i-1}) \text{ with } i = 1, \ldots, n \text{ and } 0 < x_0 < 1 \\
    y_j &= \delta(1 - \varphi|\sin(x_i)|)y_{j-1}(1 - y_{j-1}) \text{ with } j = 1, \ldots, n \text{ and } 0 < y_0 < 1
\end{align*}
\]  

(10)

where \(a, \delta \in [3.54 \ldots 4]\) and \(\beta, \varphi \in [0 \ldots 0.25]\).

In Figure 1, the bifurcation diagram of the 2DSCLM is shown. For this purpose, the values of \(a\) and \(\delta\) parameters were increased by a step of \(10^{-4}\) with \(a, \delta \in [3.54 \ldots 4]\), \(\beta = \varphi = 0.02\), and \(x_0 = y_0 = 0.5\). Then, the produced \(x\) and \(y\) sequences were displayed for 5000 iterations. The results in Figure 1 illustrate that \(x\) and \(y\) sequences are characterized by bifurcations \(a \in [3.57 \ldots 4]\) and \(\delta \in [3.61 \ldots 4]\). These results indicated that the 2DSCLM exhibits chaotic behavior.

![Figure 1](image-url)
To further validate the chaotic behavior of the 2DSCLM, the Lyapunov exponent (LE) of 2DSCLM sequences \((x, y)\) can be computed via the following formula [37]:

\[
LE = \lim_{n \to \infty} \left[ \frac{1}{n} \sum_{i=1}^{n} \log_2 \left| \frac{dx_{i+1}}{dx_i} \right| \right] 
\]

(11)

where \(\log_2\) represents the log base 2 function.

Figure 2 shows the LE values of \(x\) and \(y\) sequences produced via the 2DSCLM. From this figure, we can see that the values of LE became positive for \(\alpha \in [3.57...4]\) and \(\delta \in [3.61...4]\), indicating chaos in 2DSCLM sequences. In other words, the 2DSCLM model strongly depends on its control parameters and initial conditions. This property is fundamental for any chaotic system.

![Figure 2. LE of x (a) and y (b) sequences produced by 2DSCLM for \(\alpha, \delta \in [3.54...4]\), \(\beta = \phi = 0.02\), and \(x_0 = y_0 = 0.5\).](image)

To further validate the chaotic behavior of the 2DSCLM, \(x\) and \(y\) sequences were computed for specified control parameters and initial values. Then, we slightly modified only one parameter by \(10^{-15}\), and then \(x^*\) and \(y^*\) sequences were computed using (9). Finally, the original sequences \((x, y)\) with \((x^*\) and \(y^*)\) were generated using the varied parameter, and the difference in absolute value between these sequences are displayed in Figure 3, which shows 500 iterations of \(x\) and \(y\) sequences obtained using the parameters \(\{\alpha, \beta, \delta, \phi, x_0, y_0\} = \{3.99, 0.021, 3.99, 0.021, 0.5, 0.5\}\) as well as \(x^*\) and \(y^*\) sequences obtained when varying the parameter \(\beta\) by \(\Delta = +10^{-15}\). The results displayed in this figure clearly indicate the high sensitivity of the 2DSCLM to a slight variation of its control parameters, which confirms the chaotic character of the proposed system. It can also be noticed that the first 50 iterations of the original \((x, y)\) and modified sequences \((y^*\text{ and } y^*)\) are identical. Therefore, this shortcoming of the 2DSCLM should be considered when integrating the proposed chaotic map into information security systems.
Figure 3. The first 500 iterations of $x$, $x^*$, $y$, and $y^*$ sequences that are obtained using the parameters $\{\alpha, \beta, \delta, \phi, x_0, y_0\} = \{3.99, 0.021, 3.99, 0.021, 0.5, 0.5\}$ and $\{\alpha, \beta^*, \delta, \phi, x_0, y_0\} = \{3.99, 0.021 + \Delta, 3.99, 0.021, 0.5, 0.5\}$ of 2DSCLM, with $\Delta = 10^{-15}$.

After validating the chaotic behavior of the 2DSCLM, it can be used in a robust security scheme designed to ensure a high level of security when communicating EPFs via an insecure communication channel in IoMTs.

4. Proposed Security Algorithm for Patient Data in WBSN and IoMTs Applications

The proposed scheme for protecting both bio-signal copyrights and confidential patient information is summarized in Figure 4. Detailed information about this scheme is discussed in the following subsections.
The proposed scheme for protecting both bio-signal copyrights and confidential patient information is summarized in the following subsections.

The proposed security algorithm.

4.1. Pre-Processing Phase

In this phase, the original signal is partitioned into equal-sized frames, as shown in Figure 5, and then four frames with different indices are arbitrarily selected. The objective of selecting four frames was to improve the security level of our scheme. Indeed, if one frame is attacked by a cutting (cropping) attack, we can use the remaining frames to validate the copyright of the protected bio-signal. Then, a 1D to 2D conversion of each frame is performed. This conversion is useful to ensure fast feature vector extraction from each frame [25]. In this step, the indices of the selected frames are saved in the indices vector noted (Ind).

Figure 4. The proposed security algorithm.

Figure 5. Pre-processing phase of the bio-signal to be copyrighted.
4.2. Feature Vector Extraction and Encryption Phase

This phase consists of several successive steps as follows:

**Step 1:** Normalize the amplitude of each input signal frame between 0 and 1. This step was designed to resist the amplitude scaling attack of the original signal during its transmission via IoMTs.

**Step 2:** For each normalized 2D frame of size $N \times N$, use (2) to compute 2DHMs up to the last order $(N, N)$. Finally, we obtain four 2D matrices noted $M_i, i = 1, \ldots, 4$, each of size $N \times N$.

In this step, the 2DHM parameter values (see (2)) are given as a security key denoted $KEY1$.

**Step 3:** Compute the value $Th$, which represents the average value of each $M_i, i = 1, \ldots, 4$ matrix. Then, binarize the elements of these matrices according to the following formula

$$MB_i(x, y) = \begin{cases} 
1, & \text{if } M_i(x, y) \geq Th \\
0, & \text{otherwise}
\end{cases} ; x, y = 1, 2, \ldots, N$$

(12)

**Step 4:** Reshape $MB_i, i = 1, \ldots, 4$ matrices into $VB_i, i = 1, \ldots, 4$ vectors, each of length $L = N \times N$, which represent the feature vectors of the selected frames.

**Step 5:** To provide a better security level for our scheme, the proposed 2DSCLM is used to encrypt $VB_i, i = 1, \ldots, 4$ vectors. To this end, two chaotic sequences denoted $X$ and $Y$, each of size $L = N \times N + 50$, are generated via the 2DSCLM. Then, the first 50 values of $X$ and $Y$ are removed to ensure a very low correlation between the values of the produced sequences. Next, we determine the average values $T_1$ and $T_2$ of $X$ and $Y$ sequences, respectively. Finally, the $X$ and $Y$ sequences are binarized based on the next equations:

$$XB(i) = \begin{cases} 
1, & \text{if } X(i) \geq T_1 \\
0, & \text{otherwise}
\end{cases} ; i = 1, 2, \ldots, N \times N$$

(13)

$$YB(j) = \begin{cases} 
1, & \text{if } Y(j) \geq T_2 \\
0, & \text{otherwise}
\end{cases} ; j = 1, 2, \ldots, N \times N$$

(14)

In the current stage, the parameters $\{\alpha, \beta, \delta, \phi, x_0, y_0\}$ of the 2DSCLM are given as a security key denoted $KEY2$.

**Step 6:** In this step, the bitwise exclusive OR (XOR) operator is applied to $XB$ and $YB$ for generating the following binary chaotic vector denoted $D$:

$$D = XB \oplus YB$$

(15)

where $\oplus$ designates the XOR operator symbol.

**Step 7:** In this step, the feature vectors $VB_i, i = 1, \ldots, 4$ are encrypted as follows:

$$E_i = VB_i \oplus D \text{ with } i = 1, \ldots, 4$$

(16)

**Step 8:** Regroup $E_i$ binary sequences into 64-bit groups and then use the IEEE Standard format [45] to represent these groups using their integer format. Next, four sequences noted $FV_i, i = 1, \ldots, 4$ of integer values are obtained. These are then saved in a text file named “Feature_vectors.txt”. This step was designed to optimize the length of the data to be included in the text file.
4.3. QR Code Generation

In this phase, confidential patient information is saved in a text file named “Patient_Info.txt”. Then, this file is concatenated with the “Feature_vectors.txt” file to produce a single text file named “FILE.txt”, which is converted into a QR code. Figure 6 shows an example of a QR code obtained with the proposed method. When this code is scanned, confidential patient information is obtained with the feature vectors. We used the “qrcode” 7.3.1” python module for producing the QR code image in our scheme.

![Figure 6. QR code of size 282 × 282, which includes the encrypted bio-signal feature vectors and confidential patient information.](image)

4.4. Visual Encryption of QR Code Image

The purpose of the present phase is the visual encryption of the QR code image of size $M \times M$. For this purpose, use the 2DSCLM for generating two chaotic sequences denoted $SH$ and $SW$, each of size $L = 50 + M$. After eliminating the first 50 values of these sequences, they are normalized and rounded in the interval $[-M, M]$. Next, $SH$ and $SW$ sequences are employed to confuse the QR code pixels as described in the next steps.

**Step 1:** Use the elements of $SH$ to confuse each line of the QR code image via a circular shifting by $k$-positions as follows:

$$QR^*(i, :) = \text{circshift}(QR(i, :) , k_i) , i = 1, 2, \ldots, M$$

with $k_i = SH(i) , i = 1, 2, \ldots, M$  

where $Y = \text{circshift}(A,k)$ is a function that circularly shifts the elements of $A$ array by $k$-positions [40].

**Step 2:** Use $SW$ elements to confuse each row of the $QR^*$ image by $k$-positions via the next circular shifting operation:

$$QR^{**}(i, :) = \text{circshift}(QR^*(i, :) , k_i), j = 1, 2, \ldots, M$$

with $k_i = SW(i) , i = 1, 2, \ldots, M$  

It should be noted that the parameters $\{x^*, \beta^*, \delta^*, \varphi^*, x_{0}^*, y_{0}^*\}$ of the 2DSCLM are given as a security key denoted $KEY3$.

**Step 3:** Generate two chaotic sequences $(X$ and $Y$), each of size $L = M \times M + 50$, using the proposed chaotic system, in which $M \times M$ represents the QR code dimensions. Then, the first 50 values of the generated sequences are eliminated. Next, the median values $M_1$ and $M_2$ of $X$ and $Y$ sequences are calculated, respectively. Finally, the created sequences are binarized as follows:

$$B_1(i) = \begin{cases} 1 & \text{if } X(i) \geq M_1 \\ 0 & \text{otherwise} \end{cases} \quad i = 1, 2, \ldots, M \times M$$
\[ B_2(j) = \begin{cases} 1 & \text{if } Y(j) \geq M_2 \\ 0 & \text{otherwise} \end{cases} ; j = 1, 2, \ldots, M \times M \] (20)

The parameters of the 2DSCLM are given in the current step as a security key denoted \( KEY_4 = \{ \alpha^{**}, \beta^{**}, \delta^{**}, \phi^{**}, x_0^{**}, y_0^{**} \} \).

**Step 4:** In the current step, the XOR operator is used as follows:

\[ E = B_1 \otimes B_2 \quad k_i = SW(i), \quad i = 1, 2, \ldots, M \] (21)

**Step 5:** The purpose of this step is the diffusion of the QR** image using the \( E \) binary matrix as follows:

\[ E_{QR} = QR^{**} \otimes E \] (22)

Finally, the produced \( E_{QR} \) image, which is called the zero-watermark image, is attached to the EPF. The EPF can now be safely transmitted via IoMTs.

Figure 7 shows an example of a produced QR code using our scheme; with encrypted and decrypted versions. The security keys used during the encryption and decryption phases are

\[
\begin{align*}
KEY &= \{ \{ a_1, b_1, a_2, b_2 \}, \{ a, b, \delta, \phi, x_0, y_0 \}, \{ \alpha^*, \beta^*, \delta^*, \phi^*, x_0^*, y_0^* \}, \{ \alpha^{**}, \beta^{**}, \delta^{**}, \phi^{**}, x_0^{**}, y_0^{**} \} \} \\
&= \{ 2.16, 5.22, 6.98, 3.38 \}, \{ 3.97, 0.020, 4, 0.045, 0.8, 0.6 \}, \{ 3.99, 0.020, 3.98, 0.030, 0.5, 0.6 \}, \{ 3.98, 0.022, 3.99, 0.034, 0.6, 0.5 \}.
\end{align*}
\]

It can be seen from Figure 8b that the encrypted QR code completely masks the visual information of the original QR code (Figure 7a). As a result, no information can be retrieved from the encrypted QR code using a QR code scanner. We can also observe that the QR code image (Figure 7c) is correctly retrieved in the decryption phase using correct security keys. This finding shows good security provided by our scheme. To further support this finding, we slightly modified one of the KEY parameters in the decryption phase, and then we display in Figure 8 the decrypted QR code when using the wrong security key. The results presented in this figure confirm the strong sensitivity of our scheme to its security KEY parameters because the QR code is unrecovered when using a wrong security key, even if one security parameter is changed by a very slight variation (\( \pm 10^{-15} \)).

![Figure 7](image-url)

**Figure 7.** (a) Original QR code, (b) its encrypted form, and (c) its decrypted form using correct KEY.
4.5. Patient Information Recovery and Copyright Verification

The process of the current phase involves the execution of consecutive steps, which are summarized in Figure 9. Details about these steps are provided below.

**Step 1:** In this step, we apply the reverse procedure of visual image encryption given in Section 4.4 to retrieve the original QR code image. In this step, it is necessary to use the correct KEY for decrypting the received zero-watermark.

**Step 2:** Use a QR code reader to retrieve the information stored in the decrypted image. For this purpose, we used python’s “qrcode 7.3.1” module.

**Step 3:** Retrieve confidential patient information and the original binary feature vectors stored in the QR code.

**Step 4:** In this step, use the Ind vector to relocate the frames in the received bio-signal to extract their feature vectors.

**Step 5:** Use (2) to compute the 2DHMs of each frame of size $N \times N$ to obtain four feature matrices denoted $M_i^*, i = 1, \ldots, 4$. Then, binarize these matrices by the same method given in step 3 (Section 4.2). Then, reshape the binary matrices into four feature vectors denoted $VB_i^*, i = 1, \ldots, 4$.

**Step 6:** Encrypt $VB_i^*, i = 1, \ldots, 4$ by the same procedure mentioned in steps 5–7 (Section 4.2). By applying this step, the encrypted feature vectors $E_i^*, i = 1, \ldots, 4$ are produced. Then, use the bit error rate (BER) criterion to compare the encrypted feature vectors with the same procedure mentioned in steps 5–7 (Section 4.2). By applying this step, the encrypted feature vectors $E_i^*, i = 1, \ldots, 4$ are produced. Then, use the bit error rate (BER) criterion to compare the encrypted feature vectors with the original feature vectors stored in the QR code.

It is worth mentioning that the security KEY, as well as the indices vector (Ind), is communicated between the sender and the recipient via a reliable communication channel such as short message service (SMS).

The next phase is intended to verify the transmitted bio-signal’s authenticity and recover confidential patient information in the EPF.

**Figure 8.** (a) The original QR code and its decrypted versions using the following wrong KEY parameters: (b) $a_1 = a + \Delta$, (c) $\beta_1 = \beta + \Delta$, (d) $\delta_1 = \delta - \Delta$, (e) $\varphi_1 = \varphi + \Delta$, (f) $x_0^1 = x_0 - \Delta$, and (g) $y_0^1 = y_0 - \Delta$, with $\Delta = 10^{-15}$.
vectors with \( E_i, i = 1, \ldots, 4 \). Next, the achieved BER values are stored in a \( BER_V = \{BER_1, \ldots, BER_4\} \) vector where the BER criterion is computed by:

\[
BER = \frac{1}{L} \sum_{i=1}^{L} |E_i^*(i) - E_i(i)|
\]

(23)

**Step 7:** Find the minimum BER value (\( Min_{BER} \)) of \( BER_V \) and compare this value with the threshold \( T_{BER} = 0.05 \). If \( Min_{BER} < T_{BER} \), the physician/doctor can proceed with analyzing the received bio-signal as it is considered original; otherwise, they inform the sender to retransmit the bio-signal while increasing the security level.

![Figure 9. The proposed patient data recovery and bio-signal copyright verification.](image)

To implement our scheme in real-world scenarios, it is necessary that it satisfies some fundamental requirements; namely (i) high degree of security, (ii) good robustness against various signal-processing attacks (noise, filtering, compression, common signal processing, etc.), (iii) fast execution time, (iv) ease to implementation on hardware, (v) low power consumption, and (vi) portability.

We assumed that our scheme satisfies requirement (i), since it involves the use of the proposed 2DSCLM chaotic system that is very sensitive to its initial parameters. For requirement (ii), it was assumed to be satisfied, since our scheme involves the use of the 2DHM descriptor that belongs to the family of discrete orthogonal moments, which is widely used in signal-processing applications for its good robustness against different types of attacks (noise, geometric attacks, compression, filtering, etc.). To investigate requirement (iii), we ran our scheme using four frames of electro-radiogram (ECG) signals, each frame with size \( L = 1600 \) samples. These frames were arbitrarily selected from a test signal called “Record_100” in the MIT-BIH arrhythmia database [41]. Figure 10 shows the test signal frames used in the current runtime test. Then, all the steps shown in Figure 5 were executed on a PC with the following characteristics: 4 GB of RAM and a CPU of 2.4 GHz. The programming language used to implement our scheme is Python 3.7. All the steps of our scheme were executed 100 times, the average execution time of each step was computed and is reported in Table 1, and the execution time percentage of each step in our scheme is shown in Figure 11.
Figure 10. Four test frames, each of size L = 1600, of ECG signal named “Record_100” that was selected from dataset presented in [41].

It appears from the results of Table 1 and Figure 11 that the feature vector extraction and encryption and the QR code encryption steps consumed almost 85% of the total execution time within the proposed scheme. Therefore, reducing the execution time of these steps is necessary before implementing the suggested scheme on low-cost hardware of limited performance. For this purpose, we can implement these steps based on parallel computing.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Runtime in Seconds (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-processing</td>
<td>0.0710</td>
</tr>
<tr>
<td>Feature vectors extraction and encryption</td>
<td>1.4880</td>
</tr>
<tr>
<td>Concatenation and QR code generation</td>
<td>0.3800</td>
</tr>
<tr>
<td>QR code encryption</td>
<td>0.9820</td>
</tr>
<tr>
<td>Total runtime in seconds</td>
<td>2.9210</td>
</tr>
</tbody>
</table>
5. Raspberry Pi-Based Implementation of the Suggested Algorithm

In this section, we present an implementation of the proposed scheme on a Raspberry Pi-embedded system that supports parallel computing on its quad-core processors.

5.1. Parallel Computing of the Feature Vector Extraction and Encryption

Parallel computing is a very efficient solution to accelerate the computation time of algorithms in different areas. A thorough evaluation of our scheme showed that the feature vector extraction and encryption and the QR code encryption steps can be parallelized. Figure 12 shows the difference between sequential and parallel feature vector computation using multi-core processors.

![Diagram showing sequential and parallel feature vector extraction](image)

**Figure 11.** Percentage corresponding to the execution time of the proposed scheme’s steps.

**Figure 12.** Feature vectors extraction using (a) the sequential approach, and (b) the parallel one.

In the sequential computation on a multi-core CPU (Figure 12a), we could observe that all instructions were executed on a single processor and that only one instruction was executed at a given instant. This implementation type limits the full exploitation of the available computing resources on multi-core CPUs. To solve this limitation, parallel computing can be exploited (Figure 12b). This type of computation requires initially splitting the computational problem into different parts that can be computed in parallel, and then the instructions of the individual parts are executed simultaneously (in parallel) on the available multi-processors. Consequently, the runtime can be improved by a factor...
of up to \( \chi \) compared to the sequential calculation of the same problem, where \( \chi \) factor is defined as follows [46]:

\[
\chi = \frac{T_1}{T_N}
\]

(24)

where \( T_1 \) is the sequential execution time of a given problem and \( T_N \) is the execution time when \( N \) parallel computations are performed to solve the same problem.

We use the Message Passing Interface (MPI) Python module [47] to implement the proposed parallel computing, which enables Python software to run on multi-core CPUs.

To test the performance of parallel computing in the feature vector extraction and encryption task, we used four frames of the ECG signal shown in Figure 6 for different lengths (\( L = 256 \), \( L = 1024 \), \( L = 4096 \), and \( L = 8100 \)). Then, the extraction and the encryption of the four feature vectors were performed by the conventional method using serial computing and by using the proposed parallel commuting method. We used a PC equipped with 4 GB of RAM and 2.4 GHz quad-core CPU to conduct the present test. Python 3.7 was used as a programming language to implement the feature vector extraction and encryption algorithm. Each method was executed 100 times, and the average execution time of both methods is shown in Figure 12.

To compare the extraction time of the four feature vectors using the proposed method with that of the conventional one, the following execution time improvement ratio (ETIR (%)) criteria were used:

\[
ETIR(\%) = \left( 1 - \frac{\tau_p}{\tau_s} \right) \times 100
\]

(25)

where \( \tau_p \) represents the time consumed by using the parallel computation method on the PC quad-cores and \( \tau_s \) corresponds to the time consumed when using the sequential method.

The results presented in Table 2 clearly show that the parallel computing method improved the execution time by more than 72% compared to the conventional method for different feature vector lengths. This result encouraged us to implement our scheme on low-cost multi-core CPUs such as Raspberry Pi. This hardware can meet the requirements of ease of implementation since it supports Python language programming and offers advantages of low power consumption and portability.

Table 2. Comparison in terms of execution time between the sequential and parallel methods for the feature vectors extraction and encryption problem.

<table>
<thead>
<tr>
<th>Length of the Feature Vector</th>
<th>Conventional Serial Method</th>
<th>Parallel Computing Method</th>
<th>ETIR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L = 256 )</td>
<td>0.1240</td>
<td>0.0325</td>
<td>73.7903</td>
</tr>
<tr>
<td>( L = 1024 )</td>
<td>0.5602</td>
<td>0.1520</td>
<td>72.8668</td>
</tr>
<tr>
<td>( L = 4096 )</td>
<td>2.1220</td>
<td>0.5810</td>
<td>72.6202</td>
</tr>
<tr>
<td>( L = 8100 )</td>
<td>2.6210</td>
<td>0.7024</td>
<td>73.2011</td>
</tr>
</tbody>
</table>

5.2. Parallel Encryption of the QR Code

The QR code encryption step can also be implemented based on parallel computing, as illustrated in Figure 13.
Figure 13. Proposed parallel encryption of the QR code using multi-core CPUs.

The QR code encryption phase can be implemented in parallel using multi-core CPUs by following the next steps:

(i). Divide the QR code into a set of equal-sized blocks where the number of these blocks is proportional to the available CPUs.

(ii). Use the method described in Section 4.4 to encrypt the QR code blocks on the available CPUs in parallel, as shown in Figure 13.

(iii). Concatenate the encrypted blocks to generate the encrypted QR code.

In order to recover the original QR code in the decryption phase, we applied the reverse steps from (iii) to (i). To show the ETIR(%) of parallel QR code encryption in comparison to the conventional encryption method, we used the QR code shown in Figure 8a, which was resized to 200 × 200, 400 × 400, and 600 × 600 pixels, respectively. The codes were then encrypted using the conventional and parallel methods based on a quad-core PC equipped with 4 GB of RAM and 2.4 GHz CPU. The encryption process of each QR code was then run 100 times, and the average running time was determined and is presented in Table 3. From this table, we can clearly note that the encryption time of the QR code improved by ETIR > 71% when using the parallel encryption method instead of the serial method. This outcome motivated us to implement our scheme on low-cost multi-core hardware such as Raspberry Pi.

Table 3. Comparison in terms of execution time in seconds (s) between the conventional and parallel encryption of QR codes.

<table>
<thead>
<tr>
<th>QR Code Size</th>
<th>Conventional Encryption in (s)</th>
<th>Parallel Encryption in (s)</th>
<th>ETIR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 × 200</td>
<td>0.9615</td>
<td>0.2767</td>
<td>71.2253</td>
</tr>
<tr>
<td>400 × 400</td>
<td>1.8845</td>
<td>0.5276</td>
<td>72.0016</td>
</tr>
<tr>
<td>600 × 600</td>
<td>2.8852</td>
<td>0.8248</td>
<td>71.4129</td>
</tr>
</tbody>
</table>

5.3. Raspberry Pi Hardware Board

Raspberry Pi is a nano-computer running on the Linux operating system that can be used to control electronic components for physical computing, exploring the Internet of Things (IoTs) applications, etc. Raspberry Pi is widely utilized in a wide range of application domains, including signal processing [48–50] and the IoTs [51–53]. Since Raspberry Pi is a Linux-based operating ecosystem, it offers the advantage of the Python programming language and its strong packages, especially the Message Passing Interface (MPI) [54] for parallel computing and QR code python module. Moreover, a recent version of the Raspberry Pi series, namely Raspberry Pi 4 model B [55], is equipped with a quad-core 64-bit CPU, which is very useful for running parallel algorithms. Raspberry Pi 4B has
three RAM options: 1 GB, 2 GB, and 4 GB. To connect the Raspberry Pi 4 model B, we can select the wireless connection mode via Gigabit Ethernet, Dual Band Wireless LAN port 2.4/5.0 GHz, or Bluetooth 5.0. Figure 14 shows the Raspberry Pi 4B hardware model with its various physical devices.

In our work, we used Raspberry Pi 4 model B equipped with 2 GB of RAM and 1.5 GHz CPU for implementing our security scheme. The used programming language is Python, and the employed software packages are the Message Passing Interface (MPI) for Python [51,54] and the QR code python module.

6. Simulation Results

In this section, we present the simulation findings to confirm the efficiency of the proposed scheme implemented on Raspberry Pi 4 model B.

6.1. Analysis of the Used Security Key Sensitivity and Space Size

A security scheme is considered strong if (i) it is highly sensitive to any minor variation in its security keys [56] and (ii) the size of the latter exceeds $2^{100}$ [54], which is the minimum recommended key size to avoid brute force attacks. Our scheme satisfies requirement (i) because previous experiments in this work demonstrated this criterion (see Figure 9). It remains satisfying requirement (ii) for validating the highest security level offered by our scheme.

The global security key used in our scheme is

$$KEY = \{KEY1, KEY2, KEY3, KEY4\} = \{(a_1, b_1, a_2, b_2), \{a, \beta, \delta, \psi, x_0, y_0\}, \{a^*, \beta^*, \delta^*, \psi^*, x_0^*, y_0^*\}, \{a^{**}, \beta^{**}, \delta^{**}, \psi^{**}, x_0^{**}, y_0^{**}\}\}.$$ 

It is worth mentioning that the real-type elements of KEY1 are not very sensitive to a slight variation. Therefore, we only considered the components of KEY2, KEY3, and KEY4 in the calculation of the security KEY size. If we consider that floating-point arithmetic can return inaccurate results from 15 digits after the decimal point, then the size of the security KEY used in our system is about $(10^{15})^{18} = 10^{270}$. This size is larger than the recommended minimum size for security keys [57]. As a result, our security system can successfully resist brute force attacks.

6.2. Accuracy of the Extracted Feature Vectors

Before testing the robustness of our scheme against different types of attacks, we validated the accuracy of the extracted feature vectors, as these are the main elements designed to resist signal-processing attacks. For this purpose, we used a set of biomedical
signals selected from the database in [58] in the rest of our work. Next, the selected signals were normalized to the range.

To conduct the present test, we used three test bio-signals: the first one is an electroencephalography (EEG) signal frame labeled “chb01_01_edfm” with size L = 1024 (Figure 15a). The second is a frame of ECG signal labeled “Record_108” of size L = 4096 (Figure 15b), and the third frame is an electrogastrogram (EHG) signal frame labeled “tpehg1007” of size L = 6400 (Figure 15c). Next, we used (2) to calculate the 2DHMs of the test frames. Then, (3) was used to obtain the reconstructed signal frames. Finally, to quantify the reconstruction error between the original signal frames and their reconstructed versions, we used the MSE criterion defined by (4). The results of this test are shown in Figure 15. This figure shows that the MSE values tended to be zero for the three reconstructed signal frames, demonstrating the high precision of 2DHMs. This represents the bio-signal feature vectors in our work. This result confirmed that the extracted feature vectors are stable and accurate.

![Figure 15. (a–c) Original and reconstructed bio-signal frames using 2DHMs.](image)

It is worth mentioning that the optimal parameters of 2DHMs were obtained by using the method presented by Daoui et al. in [23], which is based on the sine cosine algorithm (SCA) [24].

6.3. Robustness against Common Signal-Processing Effects

In this section, the robustness of our scheme is analyzed against common signal-processing attacks.
6.3.1. Robustness in Noisy Environments

Bio-signals can be affected by noise during transmission between medical services in the IoMTs ecosystem. Therefore, it is necessary to test the robustness of the proposed scheme in noisy environments. For this purpose, we used an ECG test signal named “Record_112”. This signal was then affected by a “Gaussian” noise as below [59]:

\[ S_N = S_O + kG \]  

(26)

where \( S_O \) denotes the original bio-signal, \( S_N \) is the signal contaminated by “Gaussian” noise noted \( G \), and \( k \) represents the noise strength.

The test signal was contaminated by Gaussian noise using different values of \( k \). Then, the proposed method was used to construct the zero-watermark image and then to check the signal copyrights according to the diagrams presented in Figures 4 and 9, respectively. In Figure 16, the results of the current test are displayed, and Figure 17 shows an example of 512-bit sequences of the encrypted feature vector \((E_1)\) and the extracted feature vector \((E_1^*)\) from the original signal and those contaminated by “Gaussian” noise. These binary sequences were used in the BER calculation for an ECG signal contaminated by “Gaussian” noise with \( k = 0.06 \). From Figure 16, we notice that the BER value increased with the increase in \( k \)-strength. However, the BER values remained low and tended to be zero, indicating that our scheme is resistant in noisy environments. This result can be explained by the fact that orthogonal moment is known by the property of noise resistance, especially at lower orders.

**Figure 16.** Noisy ECG signal frame of size \( N = 4096 \) with Gaussian noise of \( k \)-strength and the corresponding BER values.
Figure 17. Example of binary sequences of size $L = 512$ used in the BER calculation for an ECG signal contaminated by “Gaussian” noise with $k = 0.06$.

6.3.2. Robustness against Signal Cropping

The removal of one or more parts of the original signal is known as the “signal cropping” effect. This issue can occur when transmitting signals between different storage devices in IoMTs. Our scheme is evaluated against such attacks in the next experiment. To this end, a bio-signal named “emg_healthy” of electromyograms (EMG) was used. First, this signal was cropped at arbitrary indices, as shown in Figure 18. Then, our method was used to verify the copyrights of the protected signal. From the results displayed in Figure 18, it is clear that the BER values of the cropped signals remained equal to 0, indicating that our method is highly robust against cropping attacks. Furthermore, this result can be justified by the fact that our method was originally designed to extract four feature vectors from the original signal, which makes our scheme able to resist cropping attacks even if parts of the signal are removed.

Figure 18. Original EMG signal and its cropped versions with the corresponding BER values.
6.3.3. Robustness to Signal Compression

Bio-signal records are usually of long duration, which can sometimes exceed 12 h. Consequently, these records occupy important storage space. For this reason, medical service practitioners may use compression methods to archive the bio-signal recordings.

In the current test, the robustness of our method was tested against bio-signal compression. To perform this test, a mechanomyogram (MMG) signal was selected and then compressed by the method presented in [60] for different compression ratios (CRs). Next, our scheme was applied to the decompressed signal, and the BER values were computed and are reported in Figure 19. From the results displayed in this figure, it is noticeable that the BER value increased proportionally with the increase in CR. However, the BER values remained low and close to zero. Thus, our scheme shows a significant resistance against signal compression.

![Original MMG signal and its compressed versions with the corresponding BER values.](image)

**Figure 19.** Original MMG signal and its compressed versions with the corresponding BER values.

6.3.4. Robustness to Signal Filtering

Because of the noise that can affect a signal during its transmission, the bio-signal can be filtered for noise removal before being used for diagnostic purposes. In this test, the robustness of the proposed scheme against signal filtering was checked. For this, a test signal named “118e06” of ECG type (Figure 20) was used. This signal was then filtered by the “Median filter” of orders 5 and 10. This record was also filtered by the “Average filter” with a window of size 10. Next, the BER value was calculated for each filtered signal. From the results displayed in Figure 20, we can observe that the BER values remained close to
zero. This finding means that the ECG signal copyrights were validated even if this signal was filtered by various filters.

Figure 20. Original ECG signal and its filtered versions with the corresponding BER values.

6.4. Robustness to Geometric Attacks

Geometric transformations such as circular shifting translation and amplitude scaling can be used by an attacker when they try to recognize the security keys of a security system. In this section, the ability of the proposed scheme to resist geometric attacks is investigated.

6.4.1. Robustness to a Translation by Circular Shifting

In this test, an EEG-like signal called “chb01_01_edf” was circularly shifted to the right and then to the left by the factors \( k = +20 \) and \( k = -20 \), respectively. The proposed method was then used to check the copyright of the shifted signal. The results of the test are shown in Figure 21. From the results obtained in this figure, it is noticeable that the BER values tended to be zero, which indicates good resistance of the suggested method to signal-shifting attacks. This outcome can be explained by the fact that our method extracts feature vectors from large-sized frames. Therefore, signal shifting by a few signal samples presents a low influence on the extracted feature vector. In the present test, the used frames are of size \( L = 4096 \). Therefore, a shift in \( k = \pm 20 \) samples led to a slight variation in the coefficients of the extracted feature vectors. However, if the circular shifting is significant \( (k = \pm 200 \text{ samples}) \), the signal copyrights could not be deleted by our scheme. This is a weak point of our scheme to be addressed in future work.
Circular right shifting by \( k = +20 \), BER = 0.0186

Circular left shifting by \( k = -20 \), BER = 0.0185

**Figure 21.** Original EEG signal and its circularly shifted versions with the corresponding BER values.

### 6.4.2. Robustness against Signal Amplitude Scaling

The multiplication of a constant value \( \delta \) by the amplitude of a signal function \( f(t) \) enables generation of a scaled signal \( f^\delta(t) = \delta f(t) \), while the values on the time axis remain unchanged. In this test, we examined the effectiveness of the proposed method in resisting scaling attacks of signal amplitude. For conducting the present test, an ECG signal called “Record_232” was used, which was distorted by scaling attacks of factors \( \delta = 0.5 \) and \( \delta = 1.5 \). Then, the copyrights of the attacked signals were checked via our security scheme. The results of the actual test, presented in Figure 22, showed that the BER values were equal to zero. This indicates that our scheme can effectively resist the amplitude scaling attack. This finding is because, in the feature vector calculation stage, the signal amplitude must first be normalized between 0 and 1, which allows for avoidance of amplitude scaling attacks.

**Figure 22.** Original ECG signal and its amplitude scaled versions with the corresponding BER values.
6.5. Robustness Comparison with Existing Watermarking and Zero-Watermarking Schemes

This section compares the proposed scheme with some recent bio-signal watermarking and zero-watermarking schemes presented in [10,17–19,23,61,62] in terms of robustness against different types of attacks (see Table 4).

It is worth mentioning that the present comparison was performed on a PC (4 GB of RAM and 2.4 GHz CPU) using Python as the programming language. The reason for performing the current comparison on a PC was that the compared schemes were not natively implemented on a Raspberry Pi board. The bio-signals used in the present comparison were the same ones used in Section 6.2 through 6.4. Next, each zero-watermarking technique was applied to the test signals, and the average BER value corresponding to the extracted watermark was computed and then is reported in Table 4. The results presented in this table indicated, on the one hand, that the transform-based watermarking and zero-watermarking schemes presented in [17,19,23,61,62] are more robust to the variety of attacks compared to the schemes implemented in the spatial domain [10,18]. On the other hand, we can clearly notice the superiority of the proposed scheme designed in this work over the compared schemes. This superiority can be explained by the fact that our scheme was originally developed to provide strong resistance against different types of attacks, especially cropping, scaling, and noise attacks.

Table 4. Comparison of the average BER corresponding to the retrieved watermarks using various bio-signals zero-watermarking methods.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Circular right shifting by factor $k = +20$</td>
<td>0.0181</td>
<td>0.0723</td>
<td>0.0953</td>
<td>0.0192</td>
<td>0.0190</td>
<td>0.0494</td>
<td>0.0236</td>
<td>0.0401</td>
</tr>
<tr>
<td>Circular left shifting by factor $k = -20$</td>
<td>0.0179</td>
<td>0.0719</td>
<td>0.0955</td>
<td>0.0189</td>
<td>0.0191</td>
<td>0.0556</td>
<td>0.0633</td>
<td>0.0523</td>
</tr>
<tr>
<td>Scaling of the amplitude by the factor $\delta = 1.5$</td>
<td>0</td>
<td>0.1258</td>
<td>0.2369</td>
<td>0.0201</td>
<td>0.0182</td>
<td>0.1625</td>
<td>0.1409</td>
<td>0.1368</td>
</tr>
<tr>
<td>Scaling of the amplitude by the factor $\delta = 0.5$</td>
<td>0</td>
<td>0.2067</td>
<td>0.2336</td>
<td>0.0302</td>
<td>0.0201</td>
<td>0.0955</td>
<td>0.1002</td>
<td>0.0905</td>
</tr>
<tr>
<td>Signal cropping in three arbitrary indices</td>
<td>0</td>
<td>0.0523</td>
<td>0.0662</td>
<td>0.0255</td>
<td>0.0126</td>
<td>0.0892</td>
<td>0.0705</td>
<td>0.0806</td>
</tr>
<tr>
<td>Signal compression with CR = 75%</td>
<td>0.0110</td>
<td>0.0145</td>
<td>0.0136</td>
<td>0.0160</td>
<td>0.0201</td>
<td>0.0533</td>
<td>0.0602</td>
<td>0.0449</td>
</tr>
<tr>
<td>Gaussian noise of strength $k = 0.04$</td>
<td>0.0031</td>
<td>0.0065</td>
<td>0.0082</td>
<td>0.0042</td>
<td>0.0038</td>
<td>0.0081</td>
<td>0.0056</td>
<td>0.0092</td>
</tr>
<tr>
<td>Average filtering of window $w = 5$</td>
<td>0.0093</td>
<td>0.0102</td>
<td>0.0211</td>
<td>0.0096</td>
<td>0.0094</td>
<td>0.0105</td>
<td>0.0202</td>
<td>0.0098</td>
</tr>
<tr>
<td>Gaussian noise with $k = 0.06$ + compression with CR = 50% + amplitude scaling by $\delta = 1.25$</td>
<td>0.0109</td>
<td>0.0209</td>
<td>0.0311</td>
<td>0.0185</td>
<td>0.0119</td>
<td>0.0205</td>
<td>0.0198</td>
<td>0.0159</td>
</tr>
</tbody>
</table>

After validating the strong robustness of the proposed scheme against different types of signal-processing attacks, the efficiency of our scheme in terms of execution speed is investigated in the next section.

6.6. Runtime Study of the Proposed Scheme on Raspberry Pi

In this section, we test the execution time of our scheme that was implemented on a Raspberry Pi 4 model B board. This hardware board is equipped with 2 GB RAM and a 1.5 GHz quad-core CPU. To perform the experiments in the present runtime analysis, we used all 48 ECG recordings from the MIT-BIH arrhythmia database, which is largely used in medical research. The length of the frames used to generate the feature vectors and the size of the corresponding QR code are indicated in Table 5. Then, our scheme was executed on Raspberry Pi 4 model B for the ECG recordings using two execution modes: sequential and parallel execution. Our scheme was run 10 times, and the average execution time for
the test recordings was computed and is reported in Table 4. From this table, we can notice that the parallel computing on Raspberry Pi significantly improved our scheme’s global execution time by an ETIR > 65% compared to the sequential execution. This improvement allows for saving important energy in the case of large databases due to the reduction in the Raspberry Pi board’s running time. Thus, our security scheme suggestion is cost-effective regarding energy consumption and can be efficiently deployed in the IoMTs and smart city projects.

Table 5. Comparison in terms of the average execution time of the proposed scheme using the sequential and parallel methods for feature vector extraction and encryption.

<table>
<thead>
<tr>
<th>Length of the Feature Vector</th>
<th>QR Code Size</th>
<th>Total Execution Time in (s) by the Sequential Method</th>
<th>Total Execution Time in (s) by the Parallel Method</th>
<th>ETIR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L = 256</td>
<td>260 × 260</td>
<td>71.5690</td>
<td>24.1107</td>
<td>66.3258</td>
</tr>
<tr>
<td>L = 1024</td>
<td>564 × 564</td>
<td>150.0288</td>
<td>49.3229</td>
<td>67.1244</td>
</tr>
<tr>
<td>L = 4096</td>
<td>1036 × 1036</td>
<td>241.0704</td>
<td>82.0035</td>
<td>65.9836</td>
</tr>
</tbody>
</table>

7. Conclusions

In this work, a security system was proposed for deployment in IoMTs to ensure the confidentiality of patient information and verify the originality of the bio-signal recordings included in the EPF. First, our system was developed using Hahn’s discrete orthogonal moments, 2DSCLM, and QR code. Then, our scheme was implemented on the Raspberry Pi 4 model B hardware of low cost and low power consumption. A parallel implementation of the feature vector extraction and QR code encryption phases was also presented with the proposed system to fully exploit the available resources of Raspberry Pi hardware. This implementation considerably reduced the total execution time of our scheme compared to its sequential implementation by an ETIR > 65%. The achieved simulation and comparison outcomes showed, on the one hand, the high-security level of our scheme and its good robustness against different types of attacks, and on the other hand, its fast execution on the Raspberry Pi board. The latter guarantees the power efficiency of our system in IoMTs. In the future, we will improve our scheme so that it can be applied to various types of biomedical multimedia, such as 2D and 3D medical images, and we will exploit the parallel computing on a cluster of Raspberry Pi boards to provide a fast execution time of our scheme when it is applied to large databases.


Funding: This work is supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R66), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Acknowledgments: The authors would like to acknowledge the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R66), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.
References


12. El-Shafai, W. Joint adaptive pre-processing resilience and post-processing concealment schemes for 3D video transmission. *3D Res.* 2015, 6, 10. [CrossRef]


