POLITECNICO DI TORINO
Repository ISTITUZIONALE

Real time hybrid medical image encryption algorithm combining memristor-based chaos with DNA coding

Original
Real time hybrid medical image encryption algorithm combining memristor-based chaos with DNA coding / Demirkol Ahmet, S., Sahin Muhammet, E., Karakaya, B., Ulutas, H., Ascoli, A., Tetzlaff, R.. - In: CHAOS, SOLITONS AND FRACTALS. - ISSN 0960-0779. - ELETTRONICO. - 183:(2024), pp. 1-14. [10.1016/j.chaos.2024.114923]

Availability:
This version is available at: 11583/2988105 since: 2024-04-25T18:43:54Z

Publisher:
Elsevier

Published
DOI:10.1016/j.chaos.2024.114923

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

11 November 2026

Chaos, Solitons and Fractals 183 (2024) 114923

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

journal homepage: www.elsevier.com/locate/chaos

Real time hybrid medical image encryption algorithm combining memristor-based chaos with DNA coding

Ahmet Samil Demirkol [a,*], Muhammet Emin Sahin [b], Baris Karakaya [c], Hasan Ulutas [b], Alon Ascoli [d], Ronald Tetzlaff [a]

[a] Chair of Fundamentals of Electrical Engineering, Technische Universität Dresden, Germany
[b] Department of Computer Engineering, Yozgat Bozok University, Turkey
[c] Department of Electrical and Electronics Engineering, Firat University, Turkey
[d] Department of Electronics and Telecommunications of Politecnico di Torino, Italy

ARTICLE INFO

Keywords:
Image encryption
DNA coding
Diffusion-confusion
Chaos
Locally active memristor model
Memristor-based chaotic circuit

ABSTRACT

Image encryption is a commonly used method to secure medical data on a public network, playing a crucial role in the healthcare industry. Because of their complex dynamics, memristors are often used in developing novel chaotic systems that can improve the efficiency of encryption algorithms based on chaos. In this work, we propose a novel locally active memristor-based chaotic circuit model and present a real time hybrid image encryption application developed on a PYNQ-Z1 (Python Productivity for Zynq) low-cost FPGA board using Jupiter programming environment. The proposed hybrid algorithm combines memristor-based chaos with a DNA (deoxyribonucleic acid) encryption algorithm exploiting diffusion-confusion technique. We initially present a new compact and inductorless chaotic circuit, derive the model equations, and then verify its chaotic dynamics numerically through the investigation of the phase portraits, Lyapunov exponents and the bifurcation diagrams. We further implement the chaotic circuit experimentally with discrete elements. The randomness of the chaotic sequence is improved using Trivium and von Neumann post-processor algorithms and assessed through the NIST tests. Finally, the performance of the encryption algorithm is evaluated through various metrics, including histogram and correlation analyses, differential attack, information entropy, as well as data-loss and noise attack, demonstrating its security and suitability for real-time encryption systems.

1. Introduction

As a result of the rapid development of technology and its integration into our daily lives, it is becoming easier to access consumer electronic devices leading to an increase in internet usage. Following the progressively increased network bandwidths due to the downscaling in semiconductor technology and corresponding developments in networking protocols, there has been a noticeable increment in the data transfer rates of communication networks which, in parallel, results in increased rates for the transfer of digital images through public networks in an unsecure internet environment. Therefore, multiple encryption systems have been implemented to ensure secure communication with extensive data sets, including multimedia data [1]. In particular, the inability of traditional methods to provide sufficient security for communication with big data prompted research into the integration of diverse disciplines in this area [2–4]. Among these disciplines, chaos has come to the foreground due to its appropriateness for encryption, and many chaos based application areas have been developed [5,6]. In order to ensure information security, various encryption methods such as AES (Advanced Encryption Standard), RSA (Rivest, Shamir, Adleman), blowfish, DNA, and chaos based methods have been widely used in the literature [7–11] for encrypting different types of data including text, image, audio, video and neural data [12–16]. Because of its delicate reliance on initial conditions, low predictability, complex dynamics and deterministic nature allowing systematic hardware implementations, chaos-based encryption stands out as one of the most widely adopted encryption algorithms favored by numerous researchers where chaotic systems are employed as sources of randomness in both discrete and continuous time domains [17–22].

Additionally, there has been a continuous effort towards developing new chaotic models and circuit realizations of various chaotic generators [23–27]. In this context, there is a substantial amount of work

* Corresponding author.
E-mail address: ahmet_samil.demirkol@tu-dresden.de (A.S. Demirkol).

https://doi.org/10.1016/j.chaos.2024.114923
Received 19 February 2024; Received in revised form 20 April 2024; Accepted 20 April 2024
0960-0779/© 2024 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

presented in the literature on the implementation of chaotic circuits suitable for random bit generation. Some of these designs may contain inductors [28] or several active elements [29], which can pose limitations on the utilization of these circuits. The use of nanoscale memristors, which employ an internal state variable and are endowed with unique dynamic properties, has recently become a popular approach for the design of chaotic circuits [30–32]. Early memristor-based chaotic circuit implementations have utilized nonvolatile memristors as a replacement of nonlinear devices in established chaotic systems where the memristors are typically realized by model-based emulators [33,34]. A similar approach has been adopted for volatile memristor based chaotic circuits employing locally active memristor models whereas circuit implementations utilize complicated emulator circuits [35]. Featuring internal device dynamics, several hyperchaotic systems using memristor models have been implemented [36–39]. As a recent important achievement, chaotic circuits utilizing commercially available memristors have been presented in [40,41]. However, there is still a requirement for compact circuit implementations solely based on memristors.

Many chaos-based cryptosystems do not meet adequate security standards due to their inherently limited throughput rates, requirements for small key gaps or lengthy iterations, and flaws in the keyflow generation [42]. Therefore, algorithms that offer enhanced randomness are favored in the chaos-based implementations of TRNGs (True Random Number Generators). In fact, there are many applications in the literature with embedded system hardware boards where chaotic systems and algorithms can be implemented in the same environment [43–45].

Considering recent studies, in [46], authors employ a simplified multi-piecewise-linear memristor model to design a family of multi-butterfly chaotic systems capable of producing attractors in multiple dimensions. Compensating the expenses associated with a complex hardware implementation, the proposed approach demonstrated a significant enrichment in the nonlinear dynamics of the system, making it suitable for image encryption purposes. In order to reduce the hardware design complexity, the authors in [47] suggested a simplified piecewise memristor model, which was then employed for creating a grid multi-scroll attractor chaotic system within a Hopfield neural network framework. Additionally, a medical image encryption scheme utilizing the proposed chaotic system was demonstrated on the Raspberry Pi, programmed in Python language, and successfully implemented under the Message Queue Telemetry Transmission protocol for machine-to-machine communication. Concentrating on a novel strategy, authors in [48] introduced a brain-like system with coexisting chaotic and hyperchaotic dynamics, and initial condition-dependent attractor offset effects, through aa memristive-coupled neural network model based on two sub-neural networks and one multistable memristor synapse. The hyperchaotic system proposed was employed in a biomedical image encryption system implemented on an FPGA platform, showcasing superior performance in resisting cryptographic attacks. Taking into account the computational costs, authors in [49] proposed a novel image encryption scheme, utilizing a hyperchaotic system and DNA computing for achieving high plaintext sensitivity. The method employs a computationally efficient tent map-based keystream selection method while ensuring high security through DNA encoding. The advantages of DNA cryptography in enhancing image security for industrial applications are highlighted in [50], where a combination of various chaotic maps (tent, circle, Chebyshev, 3D logistic) is utilized to create a significantly large key space. This approach enhances security against attacks when compared to traditional methods. In [51], authors presented a color image encryption platform based on the dynamic DNA encoding and the 4D memristive hyperchaotic system. The key feature of the proposed algorithm is the dynamic DNA mechanism based on hyperchaos in encoding, confusion, and diffusion processes which results in strong resistance against various attacks. Similar works combining memristor-based chaos and DNA coding for image encryption can be found in [52–54].

In this work, we propose a compact and locally active memristor-based chaotic circuit and combine a diffusion-confusion technique with DNA coding for a real-time image encryption application on a PYNQ embedded FPGA platform. PYNQ, an open-source framework by Xilinx [55], aims to facilitate interactive testing, fast design iteration, and rapid prototyping on System on Chip (SoC) FPGAs. It enables hardware/software co-design development while ensuring the seamless integration of Python language with the Zynq FPGA board. The memristor employed is realized through a minimal transistor-based circuit realization, demonstrating a S-shaped DC characteristic on the I-V plane, and exhibiting locally active dynamics around its negative differential resistance (NDR) region. The proposed chaotic circuit has a very compact form, being composed of two memristors and is realized experimentally with off-the-shelf elements. The chaotic dynamics of the circuit are analyzed numerically through the investigation of phase portraits, Lyapunov exponents, and bifurcation diagrams. The digitalized state variable (i.e. the capacitor voltage V_{C1}) of the chaotic circuit is used for generating true random bits after a post-processing step aimed to increase the randomness of the chaotic bit sequences. In the post-processing step, the Trivium stream cipher algorithm and von Neumann corrector are employed while, at the encryption phase, a DNA coding algorithm in combination with a diffusion-confusion technique is applied, with the chaotic dynamics of the proposed memristive circuit playing a key role in increasing the sensitivity of the encrypted image with respect to the input image. The main benefits related to the use of DNA in cryptography are examined, particularly the large key space, low computational power, and the opportunity, it naturally offers, to apply various cryptographic techniques to the sequences it allows to generate. The randomness of the chaotic sequences generated by the proposed memristor-based chaotic circuit is evaluated rigorously using National Institute of Standards and Technology (NIST) tests [56]. After the post-processing phase, the randomness of the chaotic sequences is re-evaluated. The resulting key sequence, i.e. the true random bit (TRB) stream, passed the NIST statistical tests demonstrating an acceptable randomness. Subsequently, the random key sequence is utilized during the confusion and diffusion phases of the encryption procedure. To ensure a high security level for the encrypted data, a hybrid coding method, based on DNA sequences with randomness boosted through the chaotic dynamics of a memristive circuit, is introduced. The encrypted data is then subjected to multiple security checks, including important metrics, namely histogram analysis, NPCR (Number of Pixel Change Rate) and UACI (Unified Averaged Changed Intensity), so as to validate the efficacy of the proposed method, before the decryption process.

The main contributions of this paper are highlighted as below.

(1) We propose a **novel compact and inductorless** chaotic circuit based on a simple circuit topology. The proposed circuit employs the recently proposed **locally active 2T1RC one-port** as the only nonlinear element together with a DC voltage source, a bias resistor and two parallel capacitors.
(2) The chaotic circuit is **realized experimentally** with off-the-shelf elements demonstrating very good match with the analytical model predictions. The **chaotic dynamics** of the circuit are analyzed numerically through the investigation of phase portraits, Lyapunov exponents, and bifurcation diagrams.
(3) The proposed approach combines the widely-used **diffusion-confusion** technique **with DNA coding** while the chaotic sequences are utilized as the source of randomness. In order to improve the randomness of these sequences, the **Trivium stream cipher** algorithm is applied as a post-processor. This integration maintains the **design complexity** and **computational cost** of the proposed encryption algorithm at a competitive level, making it suitable for real-time applications.
(4) The **chaos-enhanced image encryption and decryption processes**, post-processor algorithms, statistical tests, and attacks are implemented on the **PYNQ-Z1 FPGA board** and the Jupyter

programming environment, which uses the **Python** language, so as to demonstrate the efficiency of the proposed system in a real-time application.

Fig. 1 offers a comprehensive overview of the proposed system, illustrating its key stages and components. The paper is organized as follows. Section 2 outlines the innovative memristor-based chaotic circuit. This section demonstrates the presence of chaos in the circuit by showcasing generated waveforms, computing Lyapunov exponents, and illustrating a bifurcation diagram. Additionally, it includes details on DNA coding. Section 3 delves into the outcomes of security tests conducted through computer simulations and elucidates the operational concepts of the encryption system. The robustness of the proposed design is thoroughly analyzed in this section. Finally, Section 4 offers a concise discussion and conclusions.

2. Material and methods

2.1. The locally active memristor model

In this work, as a volatile memristive system, we refer to a recently introduced 2-transistor-1-resistor-1-capacitor (2T1RC) circuit [57], which is serially connected with an additional linear resistor R_s element and illustrated in Fig. 2a. The DC I-V characteristic of the 2T1RC circuit, including the influence of the resistor R_s on the locus, is depicted in Fig. 2b. The 2T1RC circuit is originally derived from the capacitorless 2T1R circuit [58] which represents the minimal implementation of a transistor-based current controlled resistor. Due to the S-shaped DC I-V curve with a zero-crossing property, the circuit shares similar characteristics with threshold switches, or namely, locally active memristors, which fall into the class of volatile resistive switching memories. The locally active dynamics essentially occur due to the existence of the negative differential resistance (NDR) region on the DC I-V curve and can be utilized to induce complex behaviours in otherwise-dumb circuits and systems. With the inclusion of an external capacitor dominating over the device parasitics, the 2T1RC circuit features an accurate yet simple model structure employing a single state variable which enables the control of the switching speed of the circuit. Given its compact form and ease of electronic implementation, the 2T1RC circuit may help the robust realization of inductorless chaotic oscillators, to be shown in the

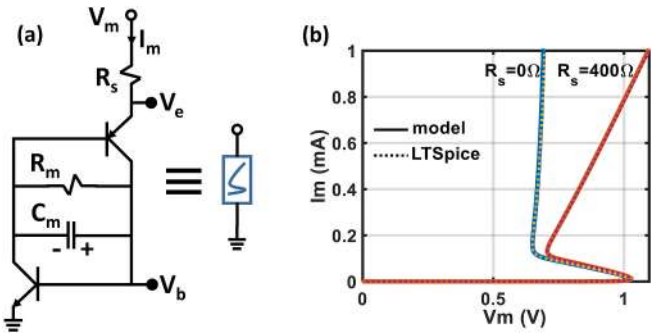

Fig. 2. (a) 2T1RC circuit and its single element representation. (b) The mathematical model given by Eqs. (3)–(5) demonstrates a very good match with the LTSpice simulation results, both with and without the inclusion of the R_s element.

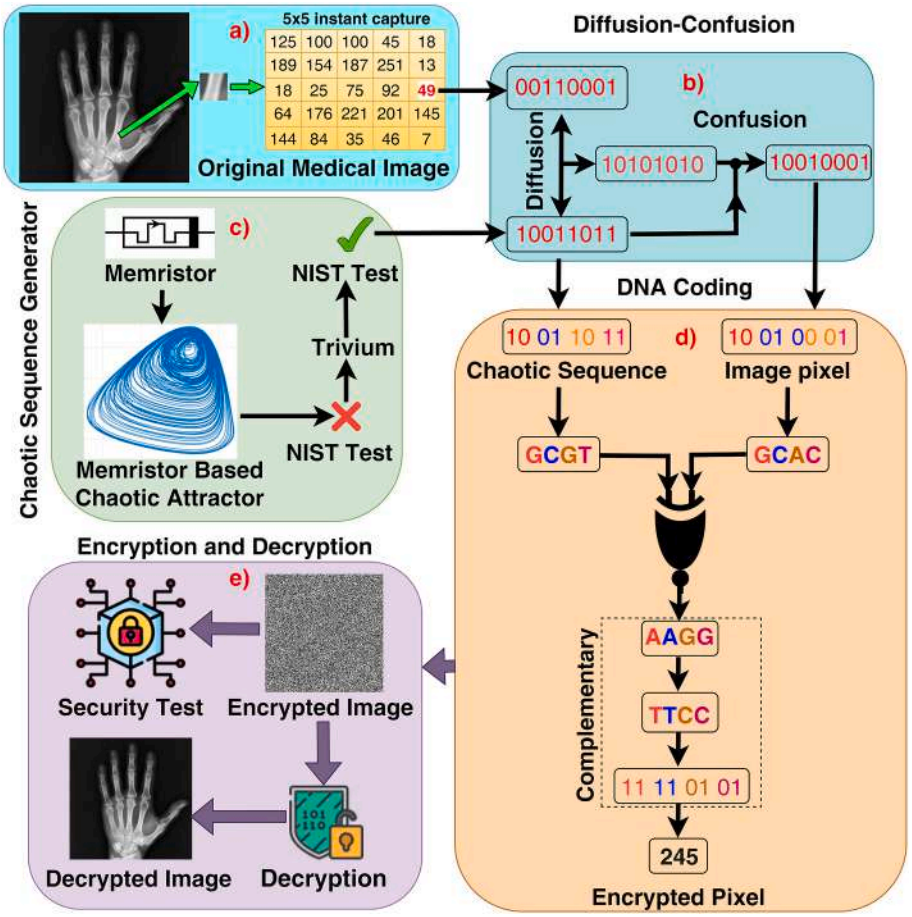

Fig. 1. (a) The pixel values of the input image expressed in binary format (b) The diffusion-confusion operation (c) The key stream generated by the chaotic circuit (d) DNA coding (e) Encryption, and decryption operations for an instant selected pixel.

following section.

We introduce the model equations of the 2T1RC circuit from Fig. 2a, initially assuming $R_s = 0\Omega$, i.e. $V_m = V_e$, in Eqs. (1) and (2), which are adopted from [57] for convenience.

$$I_m = I_S\left[\left(1+\frac{1}{\beta_F}\right)e^{(V_m+V_{Cm})/2V_T} - e^{V_{Cm}/V_T} - \frac{1}{\beta_F}\right] \tag{1}$$

$$C_m\frac{dV_{Cm}}{dt} = I_S\left[\left(1-\frac{1}{\beta_F}\right)e^{(V_m+V_{Cm})/2V_T} - \left(1+\frac{2}{\beta_R}\right)e^{V_{Cm}/V_T} + \left(\frac{1}{\beta_F}+\frac{2}{\beta_R}\right)\right] - \frac{V_{Cm}}{R} \tag{2}$$

Here, we refer to the Ebers-Moll equations for modeling properly the large-signal behavior of bipolar transistors, assuming the same model parameter values for the npn and pnp transistors, namely $\beta_f = 250, \beta_r = 3, V_T = 0.0259\,V, I_s = 10^{-14}\,A$, and $R_m = 5k\Omega$. The DC I-V characteristic obtained by solving Eqs. (1) and (2) is shown in Fig. 2b and highlighted through the $R_s = 0\Omega$ curve. In addition, we simulate the 2T1RC circuit in LTSpice and as shown in Fig. 2b, the result of numerical simulations is in very good agreement with the LTSpice simulation result.

In order to include the effect of R_s, we reformulate Eqs. (1) and (2) and obtain Eqs. (3)–(5) as given below where W(·) stands for the *Lambert W* function characterized by the property $W(x) \bullet e^{W(x)} = x$.

$$I_m = I_S\left[\left(1+\frac{1}{\beta_F}\right)e^{(V_e+V_{Cm})/2V_T} - e^{V_{Cm}/V_T} - \frac{1}{\beta_F}\right] \tag{3}$$

$$C_m\frac{dV_{Cm}}{dt} = I_S\left[\left(1-\frac{1}{\beta_F}\right)e^{(V_e+V_{Cm})/2V_T} - \left(1+\frac{2}{\beta_R}\right)e^{V_{Cm}/V_T} + \left(\frac{1}{\beta_F}+\frac{2}{\beta_R}\right)\right] - \frac{V_{Cm}}{R_m} \tag{4}$$

$$V_e = V_m + R_S \bullet I_S\left(e^{V_{Cm}/V_T} + \frac{1}{\beta_F}\right) - 2V_T \bullet W\left(\frac{R_S \bullet I_S}{2V_T}\left(1+\frac{1}{\beta_F}\right) \bullet e^{\left(V_{Cm}+V_m+R_S \bullet I_S\left(e^{V_{Cm}/V_T}+\frac{1}{\beta_F}\right)\right)/2V_T}\right) \tag{5}$$

Here we note that, for $R_s = 0\Omega$, Eq. (5) implies $V_m = V_e$ which simply reduces Eqs. (3)–(5) to (1)–(2). Likewise, Fig. 2b depicts the result of numerical simulations after solving Eqs. (3)–(5), which matches very well with the LTSpice simulation result for $R_s = 400\Omega$.

2.2. Memristor-based chaotic circuit

An early and well-known example of an inductorless chaotic circuit is the "Poor Man's Chaos" circuit ([59] pp. 239) which is depicted in Fig. 3a. The chaotic circuit employs neon-lamp based elements serving as threshold switching devices, which are the only nonlinear elements in the circuit. Inspired from this topology, we propose the circuit given in Fig. 3b where we utilize 2T1RC circuits in place for the neon lamps. The other elements in the circuit are a DC voltage source, a bias resistor and capacitors with different values appearing in parallel with the locally active elements.

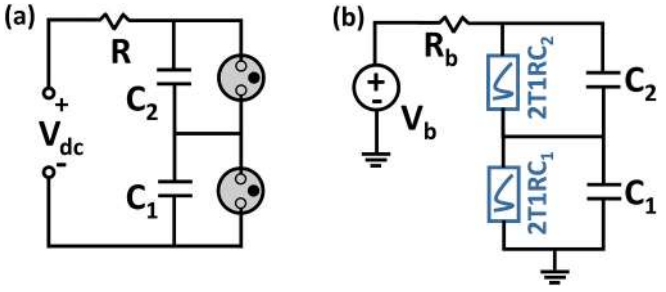

Fig. 3. (a) "Poor Man's Chaos" circuit, originally employing neon lamps as threshold switching devices (adopted from [59]). (b) The proposed memristor based chaotic circuit where neon lamps in the circuit from (a) are replaced with 2T1RC circuits serving as locally active memristors.

The model equations of the proposed chaotic circuit are given in Eqs. (6)–(13) as below.

$$I_{m1} = I_S\left[\left(1+\frac{1}{\beta_F}\right)e^{(V_{e1}+V_{Cm1})/2V_T} - e^{V_{Cm1}/V_T} - \frac{1}{\beta_F}\right] \tag{6}$$

$$I_{m2} = I_S\left[\left(1+\frac{1}{\beta_F}\right)e^{(V_{e2}+V_{Cm2})/2V_T} - e^{V_{Cm2}/V_T} - \frac{1}{\beta_F}\right] \tag{7}$$

$$V_{e1} = V_{C1} + R_S \bullet I_S\left(e^{V_{Cm1}/V_T} + \frac{1}{\beta_F}\right) - 2V_T \bullet W\left(\frac{R_S \bullet I_S}{2V_T}\left(1+\frac{1}{\beta_F}\right) \bullet e^{\left(V_{Cm1}+V_{C1}+R_S \bullet I_S\left(e^{V_{Cm1}/V_T}+\frac{1}{\beta_F}\right)\right)/2V_T}\right) \tag{8}$$

$$V_{e2} = V_{C2} + R_S \bullet I_S\left(e^{V_{Cm2}/V_T} + \frac{1}{\beta_F}\right) - 2V_T \bullet W\left(\frac{R_S \bullet I_S}{2V_T}\left(1+\frac{1}{\beta_F}\right) \bullet e^{\left(V_{Cm2}+V_{C2}+R_S \bullet I_S\left(e^{V_{Cm2}/V_T}+\frac{1}{\beta_F}\right)\right)/2V_T}\right) \tag{9}$$

$$C_{m1}\frac{dV_{Cm1}}{dt} = I_S\left[\left(1-\frac{1}{\beta_F}\right)e^{(V_{e1}+V_{Cm1})/2V_T} - \left(1+\frac{2}{\beta_R}\right)e^{V_{Cm1}/V_T} + \left(\frac{1}{\beta_F}+\frac{2}{\beta_R}\right)\right] - \frac{V_{Cm1}}{R_{m1}} \tag{10}$$

$$C_{m2}\frac{dV_{Cm2}}{dt} = I_S\left[\left(1-\frac{1}{\beta_F}\right)e^{(V_{e2}+V_{Cm2})/2V_T} - \left(1+\frac{2}{\beta_R}\right)e^{V_{Cm2}/V_T} + \left(\frac{1}{\beta_F}+\frac{2}{\beta_R}\right)\right] - \frac{V_{Cm2}}{R_{m2}} \tag{11}$$

$$C_1\frac{dV_{C_1}}{dt} = (V_b - V_{C1} - V_{C2})/R_b - I_{m1} \tag{12}$$

$$C_2\frac{dV_{C_2}}{dt} = (V_b - V_{C1} - V_{C2})/R_b - I_{m2} \tag{13}$$

In order to observe chaotic behavior, we set element values such that $R_{m1} = R_{m2} = 5k\Omega$, $C_{m1} = C_{m2} = 1nF$, $C_1 = 2.2nF$, $C_2 = 3nF$, $R_s = 400\Omega$, $V_b = 3V$, and $R_b = 40k\Omega$. The projections of the chaotic attractors obtained by numerically solving Eqs. (6)–(13) in MATLAB are introduced in the first column of Fig. 4, respectively on the $V_{b1} - V_{C1}$, $V_{b1} - (V_{C1} + V_{C2})$, $V_{b2} - (V_{C1} + V_{C2})$, and $V_{C1} - (V_{C1} + V_{C2})$ planes. Similarly, we simulate the proposed circuit in LTSpice for the same set of element values while we use 2N2222A and 2N2907A devices as npn and pnp type bipolar transistors, respectively. The results of LTSpice simulations are depicted in the middle column of Fig. 4, for the same family of attractors considered in the first column. Finally, we build the proposed circuit using discrete components with the same element values as in theory, while we present the corresponding experimental chaotic attractors in the right column of Fig. 4. It can be deduced from Fig. 4 that the attractors observed experimentally match well with the ones observed in numerical and LTSpice simulations.

2.3. Dynamical analysis

Among several approaches, Lyapunov exponents and bifurcation diagrams are efficient and robust tools for the identification of chaotic dynamics for nonlinear systems under investigation. In order to evaluate the sensitivity of the proposed chaotic system to its initial conditions, we recur to the computation of the Lyapunov exponents. We depict, in Fig. 5, the Lyapunov exponents of the proposed chaotic system obtained for the set of initial conditions $[V_{C1}, V_{C2}, V_{Cm1}, V_{Cm2}] = [0.1, 0, 0, 0]$

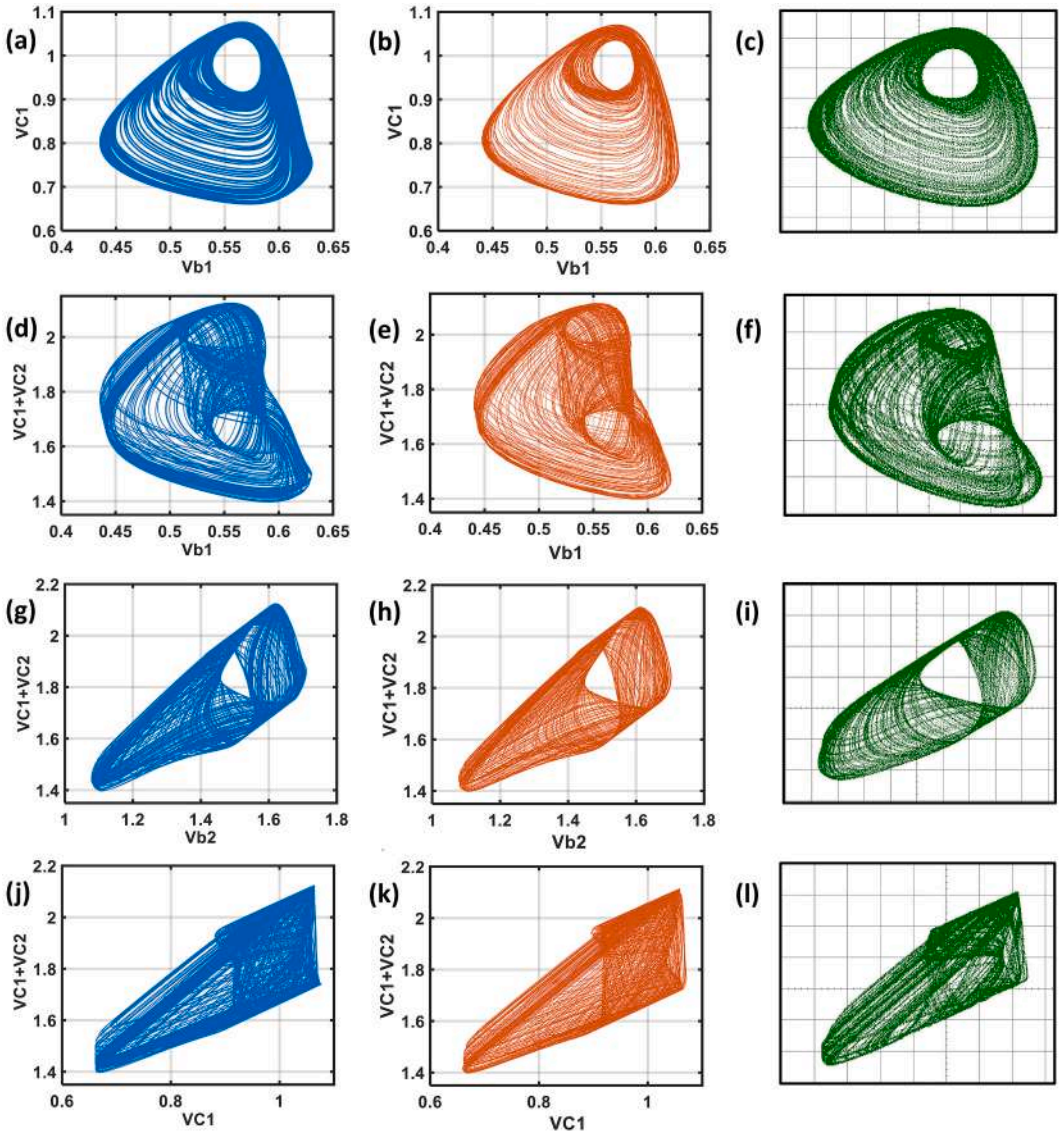

Fig. 4. Phase portraits obtained from the proposed system. The first, second, and the third columns respectively show the results of the numerical simulations in MATLAB, the results of the circuit simulations in LTSpice, and the experimental results obtained from the discrete circuit implementation of the proposed circuit.

where $L_1 > 0$, $L_2 = 0$, and $L_3, L_4 < 0$. As a result, the system can be described as chaotic while also being bounded.

Similarly, we examine the change in the chaotic dynamics of the proposed system with respect to the model parameters. In particular, we derive separately, two bifurcation diagrams assuming C_2 and R_S as the control parameters. The results are depicted in Fig. 6a and b respectively.

2.4. DNA coding algorithm

DNA-based cryptographic algorithms leverage the unique structure of biological DNA molecules, which consist of four different nitrogen bases: adenine (A), guanine (G), cytosine (C), and thymine (T). One key advantage of DNA encryption systems is their ability to offer large data capacity, parallel processing capabilities, and remarkably low power consumption, making them particularly attractive for certain applications [60–64]. DNA encryption algorithms typically involve both DNA sequence operations (such as addition, subtraction, and exclusive-OR (XOR)) and non-DNA sequence operations. In the non-DNA sequence operation, the image is initially transformed into a binary *mxn* matrix. This matrix is then subjected to diffusion-confusion operation using chaotic sequences, which introduces complexity and randomness into the data, enhancing the security of the encryption process. Subsequently, in the DNA sequence operation, the permuted image is encoded into a DNA sequence. This encoding process is often intricate and involves the representation of the binary data using the four DNA bases, which is explained below.

A 0 and a 1 exhibit a complementary relationship, similar to the pairs 00 and 11 or 10 and 01. In the DNA encoding process, the binary pairs 00, 01, 10, and 11 corresponding to each 8-bit pixel are linked to 2-bit nucleotide bases A, C, G, and T through the Watson-Crick base pairing, as outlined in Table 1. For instance, when considering the decimal number 123 in binary format, i.e. $(01111011)_2$, the corresponding DNA codes is GTCT according to rule-1, or CTGT according to rule-2, or GACA according to rule-3, and so on [64]. Advancements in DNA computing have led to the emergence of algebraic and biological operators. These operators are employed to manipulate various DNA sequences through operations such as DNA addition, subtraction, XOR, and their complementary operations. The XOR operation, short for Exclusive OR, is a logical operation that compares two values. It returns "true" if the values are different, and "false" if they are the same. The results of the application of the XOR operation on DNA sequences are provided in Table 2.

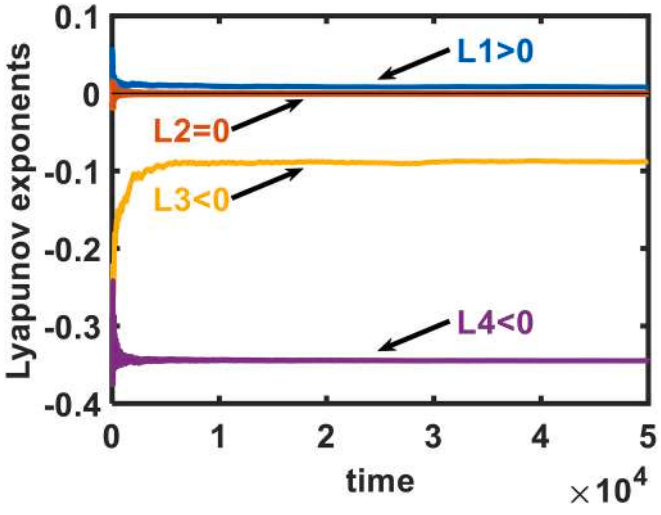

Fig. 5. Lyapunov exponents of the proposed chaotic system where $L_1 > 0$, $L_2 = 0$, and L_3, $L_4 < 0$, obtained for the set of initial conditions $[V_{C1}, V_{C2}, V_{Cm1}, V_{Cm2}] = [0.1, 0, 0, 0]$.

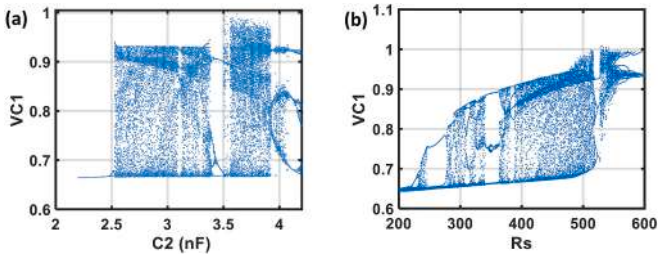

Fig. 6. Bifurcation diagrams obtained from the proposed model where the bifurcation parameter is, in (a), the parallel capacitor C_2 and, in (b), the series resistor R_s.

Table 1
Rules of DNA standards.

Rule	1	2	3	4	5	6	7	8
00	A	A	T	T	G	G	C	C
01	G	C	G	C	A	T	A	T
10	C	G	C	G	T	A	T	A
11	T	T	A	A	C	C	G	G

Table 2
DNA-XOR operation.

XOR	A	G	C	T
A	A	G	C	T
G	G	A	T	C
C	C	T	A	G
T	T	C	G	A

For example, according to Rule 2, the XOR of G (10) and T (11) results in C (01). During coding, the nucleotides are thus substituted using the DNA rules such as the one from Table 2. The rule Tables 1 and 2 demonstrates that DNA encoding and decoding can significantly alter the pixel values, indicating its potential for achieving a good form of encryption.

2.5. Post-Processor algorithms

2.5.1. von Neumann post processor

It is the first and oldest algorithm proposed by von Neumann to eliminate bit string correlation. In this post-processing algorithm, generated bit strings are partioned into 2 bit pairs, and the output is set to 1 if the bit pair is (1,0) or to 0 if the bit pair is (0,1). For the (0,0) and (1,1) pairs, there is no output generated.

2.5.2. The synchronous stream cipher algorithm: TRIVIUM

The Trivium stream cipher algorithm is specifically designed to generate a sequence of random bits, known as the key stream. The primary objective behind the development of this algorithm is to create a simple encryption system that doesn't compromise security, speed, or adaptability. It can generate up to 2^{64} random bits by employing input bit streams (an 80-bit secret key and an 80-bit initial value (IV)) to set an internal state s composed of 288 bits and utilizing two distinct phases for s [65]. First, the cipher's internal state s is initialized using the secret key bits and initial value bits. Then, the internal state, comprising 288 bits, is repeatedly updated and re-used to produce the key stream bit (z_i). The procedure and pseudo-code of the algorithm is detailed in Table 3.

3. Proposed chaos-enhanced image encryption system

In this section we present a novel hybrid encryption algorithm that combines memristor based chaotic dynamics with the robust encryption capabilities of the DNA coding method. The incorporation of memristor-based chaos serves as the basis for generating a complex and

Table 3
The procedure of the pseudo-code of the Trivium Stream Cipher algorithm.

Step 1: Key and IV (initial value) setup

Description: The algorithm is initialized by loading an 80-bit key and an 80-bit IV into the 288-bit initial state s, and setting all remaining bits to 0, except for s_{286}, s_{287}, and s_{288}.

```
s = zeros(288,1);
s(1:80) = key(1:80);
s(94:173) = initialvalue(1:80);
s(286:288)= [1];
```

Step 2: Rotate the output of Step 1 (initial state s (288-bit)) to initialize the state
Description: The initial state is rotated over 4 full cycles without generating key stream bits z_i.

```
for i = 1:1:4*288

a = and (s(91), s(92));
b = xor (s(66), a);
c = xor (s(93), b);
t1 = xor (s(171), c);

a = and (s(175), s(176));
b = xor (s(162), a);
c = xor (s(177), b);
t2 = xor (s(264), c);

a = and (s(286), s(287));
b = xor (s(243), a);
c = xor (s(288), b);
t3 = xor (s(69), c);

s(1:93) = [t3;s(1:92)];
s(94:177) = [t1;s(94:176)];
s(178:288) = [t2;s(178:287)];
end
```

Step 3: Key stream generation using the output of Step 2 (initial state s (288-bit))
Description: The key stream generation involves an iterative process wherein the values of 15 specific state bits are extracted. These extracted values are utilized to perform two main tasks: first, to update 3 bits of the state, and second, to compute i^{th} bit of the key stream, denoted as z_i. The state bits are then rotated by applying *Step 2* and the process repeats itself until the requested $N \leq 2^{64}$ bits of the key stream have been generated.

```
for i = 1:1:N

t1 = xor (s(66), s(93));
t2 = xor (s(162), s(177));
t3 = xor (s(243), s(288));

z(i) = xor (t1, t2, t3);

e = and (s(91), s(92));
f = xor (s(171), e);
t1 = xor (t1, f);

e = and (s(175), s(176));
f = xor (s(264), e);
t2 = xor (t2, f);

e = and (s(286), s(287));
f = xor (s(69), e);
t3 = xor (t3, f);

s(1:93) = [t3;s(1:92)];
s(94:177) = [t1;s(94:176)];
s(178:288) = [t2;s(178:287)];

end
```

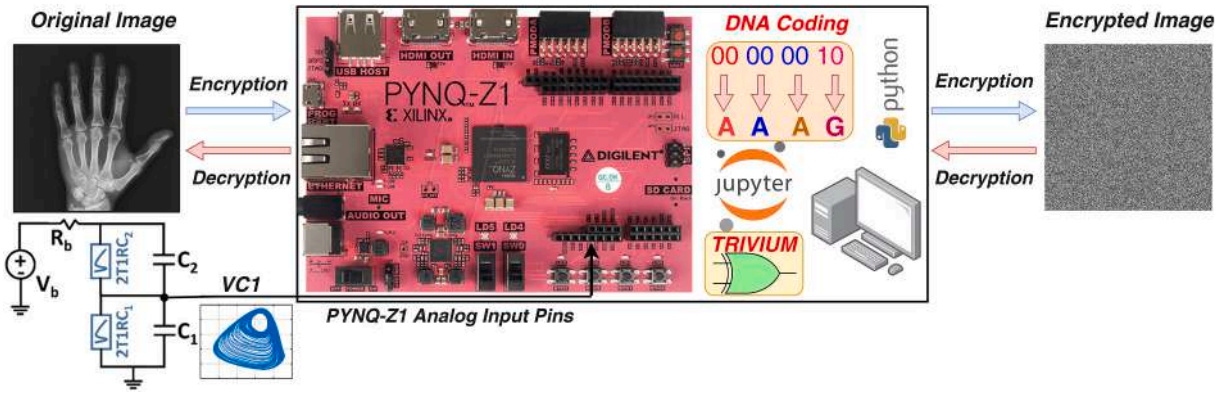

Fig. 7. The flowchart of the encryption system, implementing chaos-based DNA coding and decoding strategies, respectively.

unpredictable entropy source for encryption. Meanwhile, the DNA encryption algorithm, utilizing the diffusion-confusion technique, offers a highly secure and efficient approach for image encryption. The block diagram illustrating the proposed scheme can be found in Fig. 7, while a comprehensive explanation of both the encryption and decryption procedures is provided in Table 4.

4. Experimental results

Our experimental setup, consisting of a locally active memristor based chaotic system, the embedded FPGA board PYNQ-Z1, and a computer, generates true random bits and encrypts a medical image on board, as depicted in Fig. 8. The real-time encryption and decryption processes, post-processor algorithms, statistical tests and differential attacks are implemented on PYNQ-Z1 FPGA board and Jupiter programming environment by using the Python language. The output of the chaotic circuit, specifically the capacitor voltage V_{C1}, is transmitted to the PYNQ-Z1 FPGA board via the analog input pin. The analog signal is then quantized once to a 16-bit sequence corresponding to a decimal value ranging from 0 to 65,535. Following this step, the 16-bit quantized value is transformed into an 8-bit format to facilitate the execution of parallel post-processing algorithms for each image pixel intesity value, which is also mapped into an 8-bit long sequence.

Table 4
Procedure of the Encryption System.

Step	Description of the Encryption Step
1:	An 8-bit gray (*mxn*) plain image with *m* rows and *n* columns is applied to the encrypton system as the input.
2:	A uniform reshaping of the plain input images is carried out. grayimg = image.reshape(*mxn*)
3:	The uniformly-reshaped input image is converted to binary format. bitimage = uint8tobinary(grayimg)
4:	A Chaotic sequence, denoted as *chaoticseq*, is generated from the chaotic circuit in decimal format. Then, the *chaoticseq* data is converted into binary format and labeled as *bitSeq*. bitSeq = decimaltoBinary(chaoticseq)
5:	The binary chaotic sequence *bitSeq* is initally assessed through the NIST statistical tests. In case of any test failure, the *bitSeq* is applied to the post-processor algorithm which ensures *bitSeq* to pass the NIST tests. The *randbitSeq* is the bit sequence generated by post processing V_{C1} bit sequence with the Trivium stream cipher algorithm. The *randbitSeq* is taken as key since all the NIST test results are successful in scenario-3.
6:	An XOR operation is performed in the diffusion stage between the 8-bit sequence *bitimage*, extracted from the binary input image, and the the 8-bit-long chaotic bit string *randbitSeq* that passed all the NIST tests. diffimg = randbitSeq $\oplus$ bitimage
7:	A confusion process is performed by shuffling the output string of the diffusion stage *diffimg* as specified the index array *randbitidx* which is obtained by sorting the chaotic bit string *randbitSeq*. randbitidx = sort(randbitSeq) confimg = diffimg[randbitidx]
8:	The shuffled bit stream *confimg* and the *randbitSeq* are encoded into DNA sequences according to the Rule-2 from Table 1 (DNA_coded).
9:	A DNA-XOR operation is applied to the DNA sequences obtained in the previous step, as defined in Table 2 (DNA_XORed).
10:	The output of the DNA-XOR operation (DNA_XORed) is converted back to the bit string *encimgbit* according to the Rule-2 given in Table 1.
11:	The encryption process is completed by converting the bit sequence *encimgbit* from step 10 into a grayscale pixel intensity value (*encimg*) and assembling it in the form of an image of suitable dimensions. encimg = Binary2Uint8(encimgbit), encimg = encimg.reshape(m,n).
12:	The decryption process is performed by applying the encryption process in reverse order (*Step 11* → *Step 1*).

The randomness of the bit sequences produced by random number/bit generators is assessed using statistical tests. If the tested bit sequences successfully pass all these tests, they are considered to be random. The NIST 800–22 test [65] is widely utilized for randomness evaluation, with a recommendation to conduct the test with a minimum of 1 million bits of data. This test comprises 15 distinct sub-tests, with the requirement that the *p*-value exceeds 0.01 for each sub-test.

In this work, we explore three different scenarios in order to generate true random bit sequences for which the entropy source of randomness is the digitized state variable (specifically V_{C1}) of the memristor based chaotic circuit discussed in Section 2.2. In the first scenario, named as scenario-1, digitized V_{C1} values directly undergo the NIST statistical test suite. In scenario-2, the chaotic sequence obtained in the first scenario is fed to the Von Neumann post processor algorithm to enhance its randomness properties to improve to the results of the NIST test. Lastly, scenario-3 incorporates the Trivium algorithm, utilizing an 80-bit secret key and an 80-bit initial value (IV) derived from the bit sequence from the first scenario. In each scenario, a bit sequence of length 1,157,139 bits, is submitted to the NIST test. In scenario 2, the Von Neumann post-processor algorithm processes a longer 5,000,000-bit length sequence, originating from the digitized data series corresponding to the voltage V_{C1}, before outputting a 1,157,139-bit long string to be submitted to the NIST test.

Table 5 shows the NIST test results for the three different scenarios. It is noteworthy to observe that all NIST test results in scenario-3 have been successful. Therefore, during the encryption phase, the bit sequence generated from the post-processed bit sequence V_{C1} using the Trivium stream cipher algorithm will be utilized as the 8-bit long *key* (also referred to as *randbitSeq* in Fig. 7).

4.1. Performance analysis

Ensuring the confidentiality of patient data in medical images is crucial. In order to test the proposed system in real-time applications, its encryption processes have been applied to data formatted in traditional medical imaging modalities, including CT and X-ray images. In this work, the encryption system operates on grayscale input images with dimension $n \times m$, with $n = m = 500$. Fig. 9a and d display a hand X-ray

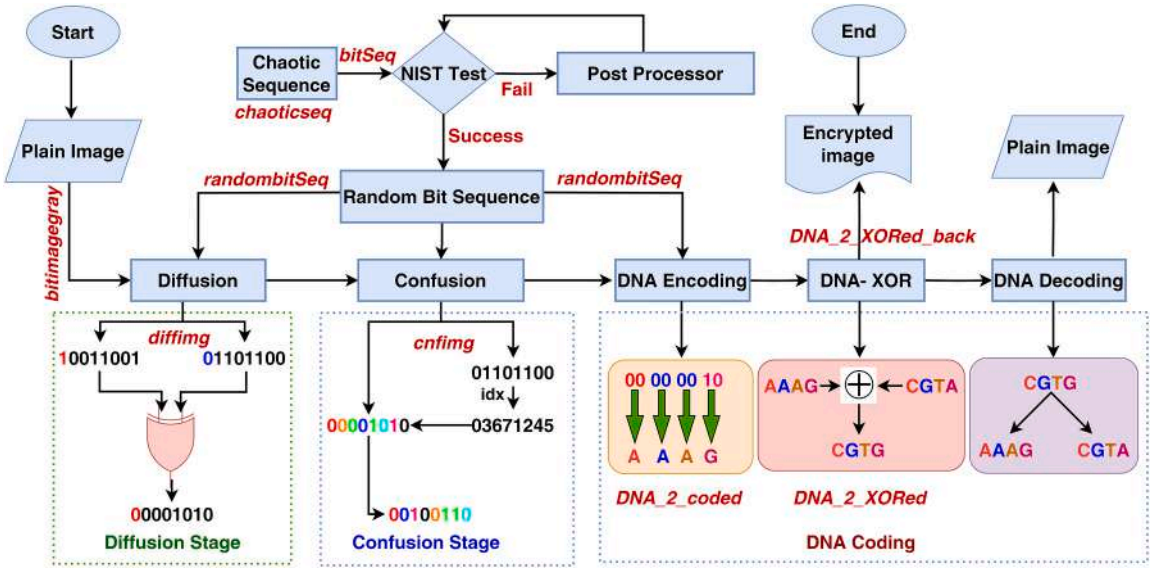

Fig. 8. Experimental setup for the proposed real-time image encryption system.

Table 5
NIST test results for the three different scenarios (a 1,157,139-bit-long sequence is tested).

Statistical test	Scenario-1 Chaotic sequence		Scenario-2 Chaotic sequence + von Neumann		Scenario-3 Chaotic sequence + Trivium algorithm	
	p-value	Result	p-value	Result	p-value	Result
Frequency (monobit) test	0.73499	Success	0.23815	Success	0.12103	Success
Frequency test within a block	0.34695	Success	0.92322	Success	0.33268	Success
Cumulative sums test - forward	0.94159	Success	0.35279	Success	0.08935	Success
Cumulative sums test - reverse	0.99170	Success	0.06130	Success	0.10484	Success
Runs test	–	Failed	0.32081	Success	0.60336	Success
Longest run of ones in a block	0.10876	Success	0.09028	Success	0.29256	Success
Binary matrix rank	0.86596	Success	0.60210	Success	0.57703	Success
Discrete fourier transform test	0.18800	Success	0.59035	Success	0.06735	Success
Overlapping template matching test	0.61932	Success	0.01704	Success	0.01796	Success
Non-overlapping template match test	0.04396	Success	0.13243	Success	0.31560	Success
Maurer's universal statistical test	–	Failed	–	Failed	0.30056	Success
Approximate entropy test	0.81123	Success	0.15919	Success	0.59411	Success
Random excursions test	0.55643	Success	0.42789	Success	0.35920	Success
Random excursions variants test	0.49145	Success	0.44567	Success	0.82569	Success
Linear complexity tests	0.41578	Success	0.89862	Success	0.28984	Success
Serial tests p-value1	–	Failed	0.36867	Success	0.99741	Success
p-value2	–	Failed	0.09708	Success	0.71589	Success

image and a kidney stone CT image, respectively. They act as inputs to the encryption system. Fig. 9b and e depict the encrypted versions of the input images, while Fig. 9c and f visualize how the decryption process reconstructs the input images. To assess the effectiveness of the proposed algorithm, a comprehensive evaluation is conducted by subjecting the system to rigorous testing using the two above mentioned medical images. Various performance analyses have been employed to measure the algorithm's robustness and security. These include encryption and decryption speed estimation, derivation and analysis of histograms, computation of correlation coefficients, calculation of information entropy, evaluation of robustness against differential attacks, execution of randomness tests, and study of sensitivity to data loss and noise attacks. By conducting these performance analyses, the proposed algorithm's effectiveness is quantitatively and qualitatively evaluated, demonstrating its suitability for medical image encryption and its ability to maintain data integrity and confidentiality, while exhibiting a good resistance against attacks.

4.1.1. Speed analysis

By measuring the encryption and decryption times, we can assess the computational efficiency of the proposed algorithm in converting the original images into their encrypted counterparts and in reverting them back to their original form. This information is crucial for assessing the potential of the proposed system in real time encryption applications. Table 6 provides the encryption and decryption speeds of the proposed encryption system. As the proposed system is implemented real-time on the PYNQ-Z1 FPGA board and Jupiter programming environment which uses the Python language, its response time justifies its adoption in real-time applications.

4.2. Histogram analysis

A histogram analysis is performed to examine the distribution of pixel intensity values across in the encrypted images, allowing for an evaluation of the preservation of details, pertaining to the original images, within the encrypted ones. A histogram chart can visually represent the distribution of pixel values across an image. It typically appears irregular and centered around certain regions of pixel intensity values in plain images. Encrypted images created using encryption algorithms are expected to exhibit a smoother distribution of pixel intesity values in their histograms. The encryption strength and the resistance to password attacks improve the smoother is the histogram of the encrypted image.

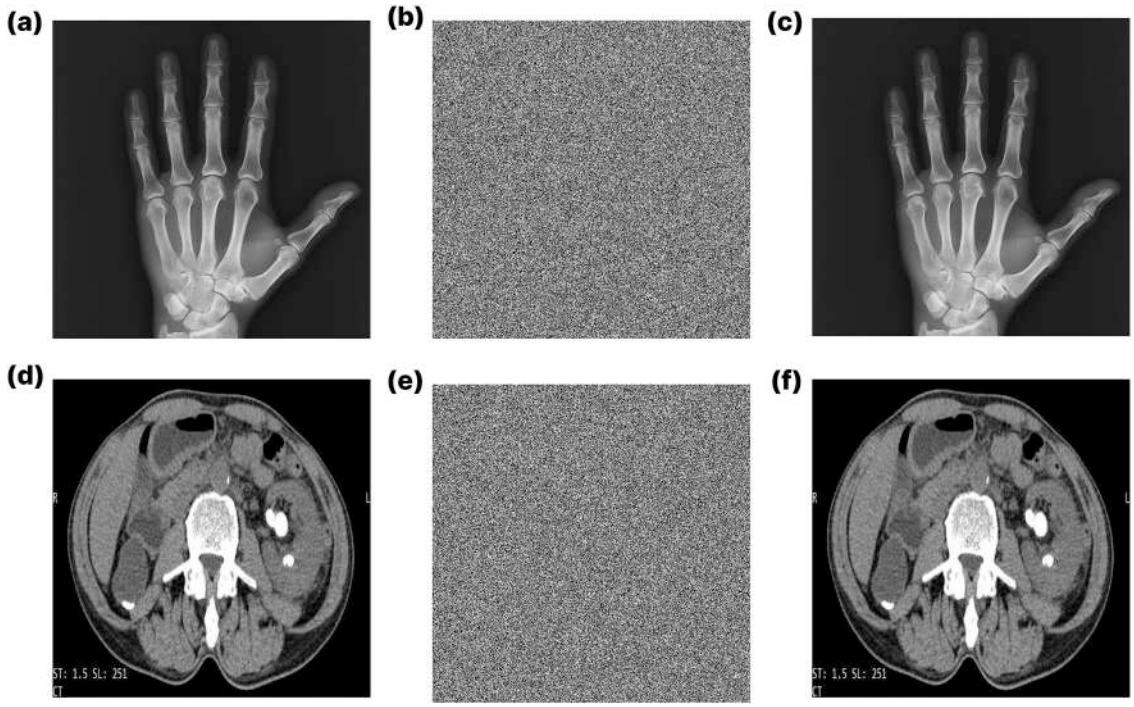

Fig. 9. Encryption and decryption of gray scale images. (a) Hand X-ray image, (b) encrypted hand X-ray image, (c) decrypted hand X-ray image, (d) kidney stone CT image, (e) encrypted kidney stone CT image and (f) decrypted kidney stone CT image.

Table 6
Encryption and decryption times.

Image	Encryption time (s)	Decryption time (s)
Hand X-ray	234.44	256.39
Kidney stone CT	241.83	253.73

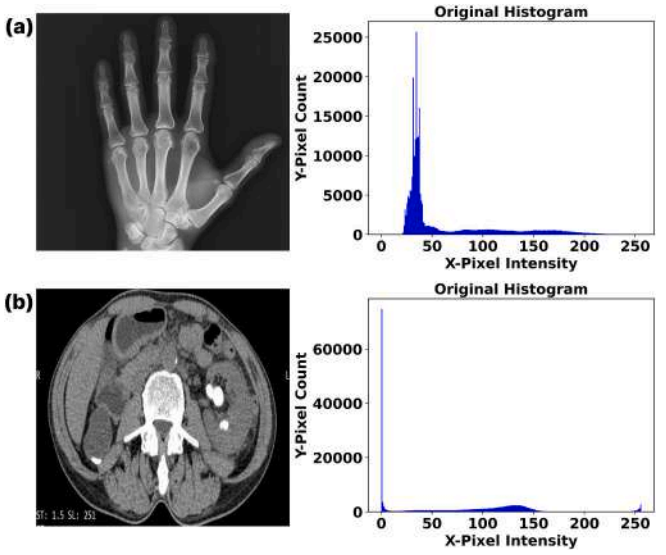

Fig. 10. Histogram plots of (a) a hand X-ray image, and a (b) kidney stone CT image.

The left and right panels in Figs. 10 and 11 display the histograms of plain and encrypted images, respectively. The plain images feature an uneven pixel value distribution as they are characterized by a high concentration of black pixels. In contrast, each of the encrypted images exhibits a uniform pixel count distribution across a wide range of pixel

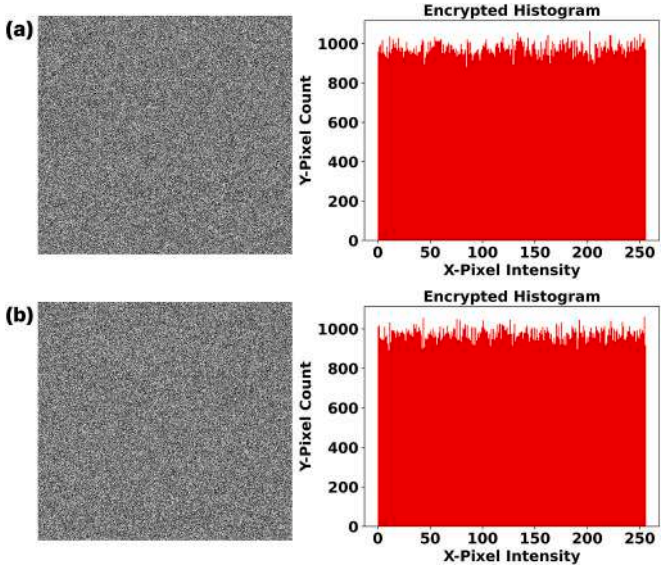

Fig. 11. Histogram plots of an (a) encrypted hand X-ray image, and an (b) encrypted kidney stone CT image.

intensity values, revealing the robustness of the encryption algorithm against differential attacks.

4.3. Correlation analysis

The purpose of a correlation coefficient analysis is to measure the level of correlation between neighboring pixels in both the original and encrypted images. A low correlation coefficient indicates on effective encryption as it reveals the capability of the encryption system to reduce the predictability of the pixel intensity values in the original images. This analysis involves examining the correlation between adjacent pixels in various directions, including the diagonal, horizontal, and

vertical ones. By assessing the pixel correlation distributions in the encrypted images, this technique helps evaluating the encryption effectiveness. Generally, the original image tends to have high correlation coefficients. Therefore, it is important to minimize the correlation between neighboring pixels as part of the encryption algorithm in order to enhance data security and privacy. [54,66]. Eqs. (14)–(16) are used to calculate the correlation coefficient according to

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{14}$$

$$\text{with } cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)), \text{ and} \tag{15}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i) \text{and } D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2. \tag{16}$$

In these equations, x_i and y_i denote the pixel intensity values of neighboring pixels, N stands for the overall count of chosen pixels, while $E(x)$ and $E(y)$ symbolize the mean and $D(x)$ and $D(y)$ represents the variance of the sampled selections. The correlation between the pixels of the encrypted image should ideally be very low, which is the case if the correlation coefficient r_{xy}, calculated via Eq. (14) for any group of neighboring pixels within the encrypted image itself, is close to 0 irrespective of its sign. Under these favorable circumstances, the encryption system protects the sensitive data against differential attacks. The top, center, and bottom panels of Fig. 12 and Fig. 13 refer to a graph or chart that displays the correlation coefficients for adjacent pixels in the horizontal, vertical, and diagonal directions, respectively for the hand X-ray image and the kidney stone CT image.

In the original image, the intensity values of neighboring pixels are close one to the other within a small range. However, in the encrypted image, the pixel intensity values are spread out across the entire admissible range. As a result, neighboring pixels might feature very different intensity values, indicating the successful randomization operated by the encryption process. This makes it challenging to identify any patterns or extract information from the encrypted image. Table 7 provides more details on the correlation measures of the proposed

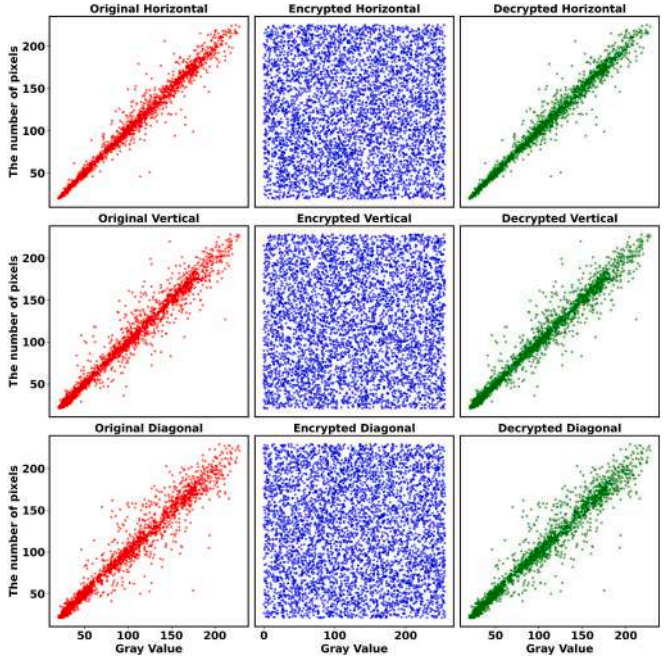

Fig. 12. Horizontal, vertical and diagonal correlation scatter plots for the original, encrypted and decrypted hand X-ray images.

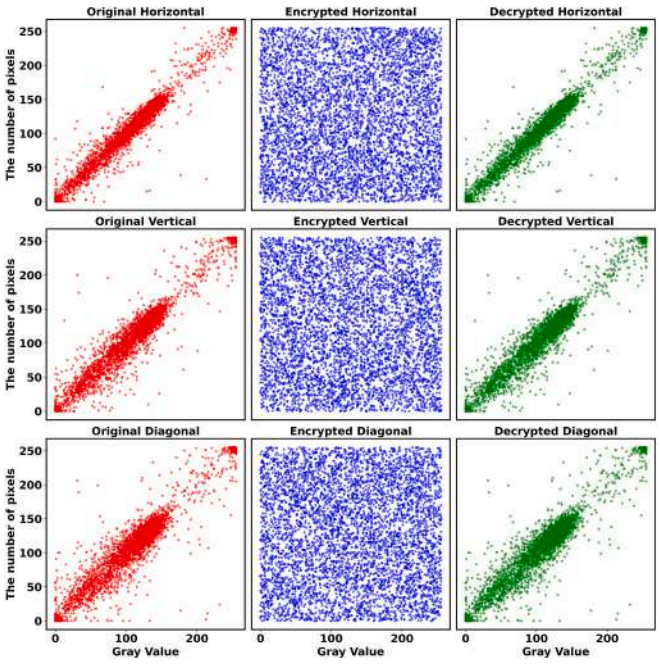

Fig. 13. Horizontal, vertical and diagonal correlation scatter plots for the original, encrypted and decrypted kidney stone CT images.

Table 7
Correlation coefficients obtained without/with DNA Coding.

Image	Class	Horizontal	Vertical	Diagonal
Hand X-ray	Original	0.9940	0.9893	0.9848
	Encrypted	0.0828 / -0.0099	0.1801 / 0.0154	0.0987 / 0.0146
	Decrypted	0.9940 / 0.9940	0.9893 / 0.9893	0.9848 / 0.9848
Kidney stone CT	Original	0.9888	0.9816	0.9732
	Encrypted	0.3187 / 0.0161	0.4838 / -0.0066	0.3132 / 0.0103
	Decrypted	0.9888 / 0.9888	0.9816 / 0.9816	0.9732 / 0.9732

system, for both scenarios, where DNA coding is omitted or included within the encryption and decryption processes. While there is no consensus in the literature regarding the impact of the characteristics of the plain image on the correlation analysis results, it is generally recognized that low correlation among neighboring pixels in the encrypted image is crucial for encryption. The results demonstrate the effectiveness of the encryption system in scrambling pixel values in medical images, particularly CT and X-ray images, which are characterized by high redundancy, and pixel correlation, and are thus vulnerable to unauthorized access. Decrypting the encrypted image without the correct key makes it difficult for a hacker to extract meaningful information from it.

4.3.1. *Differential attack*

A differential attack analysis is carried out to assess the algorithm's resistance against differential attacks. The concept of analyzing cipher texts for potential differential attacks was first proposed in [67]. The purpose of a differential attack is to predict the secret key by examining the encrypted images. In order to perform this test, the original image and n image, differing from the original one by one bit only, are encrypted to generate two separate encrypted images. The difference between these two encrypted images is then measured to find out the encryption method's effectiveness against differential attacks. Key metrics such as the numer of pixel change rate (NPCR) and the unified

averaged changed intensity (UACI) are utilized to determine the resistance of an encryption method to differential attacks. The NPCR calculates the total difference between two encrypted images on a pixel-by-pixel basis, assigning a value of one when each pair of pixels, sitting in the same position, exhibit differences. The UACI, on the other hand, allows the calculation of the sum of the differences between the intensity values of corresponding pixels from two encrypted images. The NPCR and the UACI are defined according to Eqs. (17)–(19):

$$D(i,j) = \begin{cases} 1, C_{o1}(i,j) \neq C_{o2}(i,j) \\ 0, C_{o1}(i,j) = C_{o2}(i,j) \end{cases} \text{ with} \tag{17}$$

$$UACI = \frac{1}{m \times n}\left[\sum_{ij}\frac{C_{o1}(i,j) - C_{o2}(i,j)}{255}\right] \times 100\% \text{and} \tag{18}$$

$$NPCR = \frac{\sum_{ij} D(i,j)}{m \times n} \times 100, \tag{19}$$

where C_{o1} and C_{o2} are two encrypted images of size *mxn*. For an 8-bit grayscale image, the number L of image grayscale levels is 256. It is known that an NPCR value >99.61 % and an UACI value close to 33.46 % ensure a high performance encryption process [68]. The proposed hybrid algorithm demonstrates an NPCR of 99.61 % and an UACI of 33.47 % while encrypting the hand X-ray image, and an NPCR of 99.58 % and an UACI of 33.46 % while encrypting the kidney stone CT image, as shown in Table 8. These results indicate that the encryption algorithm has been designed to withstand differential attacks.

4.3.2. Information entropy (IE)

The information entropy analysis is utilized to evaluate the randomness and complexity of the encrypted images. Higher entropy values indicate increased randomness and enhanced security. The information entropy (IE) [64] quantifies the random distribution of pixels in the encrypted image as,

$$IE = \sum_{i=0}^{L} p(v_i) log_2 \frac{1}{p(v_i)} \tag{20}$$

where v_i is the i^{th} gray value for a gray image with an L levels and $p(v_i)$ is the probability of the i^{th} pixel to attain the value of v_i. As the IE approaches to the value of 8, the randomness of the pixels increases, enhancing the security of the algorithm. The IE values calculated by applying the proposed hybrid algorithm are given in Table 9. The IE values for the encrypted images are almost exactly 8, showing that the proposed encryption method leads to secure results.

4.3.3. Data loss and noise attacks

During image transmission, images are prone to noise and partial data loss, making it important for an image encryption algorithm to be resistant to these factors [48,69]. A data-loss attack is performed on the encrypted images, damaging or deleting partially their data, to assess the ability of the decryption algorithm to recover the original plain images in this non-ideal case. In this study, cropping operation is performed on the encrypted medical images cutting away areas of various dimensions, particularly, 1/32, 1/16, and ¼ of their nxm size, respectively. After the cropping operation, the decryption process is applied to the corrupt encrypted images. The results of the decryption process are depicted in Fig. 14, indicating that the encryption algorithm successfully decrypted the cropped images with minimal distortion.

Table 8
NPCR and UACI values obtained without/with DNA Coding.

Image	UACI (%)	NPCR (%)
Hand X-ray	32.65 / 33.47	99.50 / 99.61
Kidney stone CT	34.68 / 33.46	96.16 / 99.58

Table 9
Information entropy of original and encrypted images obtained without/with DNA Coding.

Image	Original	Encrypted
Hand X-ray	6.079	7.963 / 7.999
Kidney stone CT	6.027	7.225 / 7.999

It is also crucial for an encryption algorithm to be able to withstand data losses caused by noise during network transmission or image storage. In order to investigate this aspect, the encrypted versions of the medical images are subjected to salt-and-pepper noise at 1 %, 10 %, and 20 % rates, and subsequently, and decrypted using the proposed encryption algorithm in reverse order. The results presented in Fig. 14 show that the proposed encryption algorithm effectively decrypted the noisy encrypted images. The decrypted images closely resemble the original plain images. This suggests that the encryption algorithm is highly effective withstanding noise attacks. Even in the case noisy sources corrupt the encrypted image, our decryption system is able to retrieve many details of the original plain image.

5. Discussions

This study introduces a chaos based real-time hybrid image encryption algorithm, which offers advantages over traditional methods. The chaotic circuit presented employs locally active memristors combined with passive circuit elements and does not require any further active components. Therefore, the proposed topology offers a highly effective solution in terms of area and power requirements. Additionally, the proposed memristive one-port (see Fig. 2(a)) can be substituted with any other locally active device, which potentially extends the class of chaotic circuits with a scalable physical implementation.

In order to carry out real-time encryption, we utilized the recently released PYNQ-Z1 development board from Xilinx. This allowed us to implement a hardware/software co-design approach. The PYNQ-Z1 FPGA board offers two options for real-time applications. Its hardware can be programmed using VHDL or Verilog. Alternatively, it can be utilized as a software-based All Programmable SoC (APSoC) without the need for employing its programmable logic circuits. PYNQ-Z1 is the first APSoC that can be programmed using Python, which makes it a cutting-edge choice for real-time applications. As a future work, we aim to optimize the performance, such as the computation time, of the overall implementation by improving the workload distribution between the hardware and software components.

In the encryption phase, we applied a combination of diffusion-confusion technique and DNA coding, incorporating chaotic sequences, extracted from a memristive circuit. The randomness of the data stream, generated by the circuit, was, as a result, significantly enriched. The performance of the proposed encryption algorithm was thoroughly tested by performing histogram and correlation analyses, by studying the response to differential attacks, by estimating the information entropy measure, and by evaluating the robustness to data loss and noise attacks. The results of these evaluations demonstrate the exceptional performance of the algorithm, making it suitable for real-time encryption systems. The adoption of the proposed encryption system not only enhances the security of medical images but may also contribute to the integrity and privacy of confidential data to be exchanged across communication links between physical systems in various fields. Our future research shall also focus on identifying the impact of each stage on the performance of the hybrid encryption system in order to optimize its performance. We will also work on incorporating improvements into the system, including the consideration of alternative encryption methods based on the overall performance requirements of the application being studied.

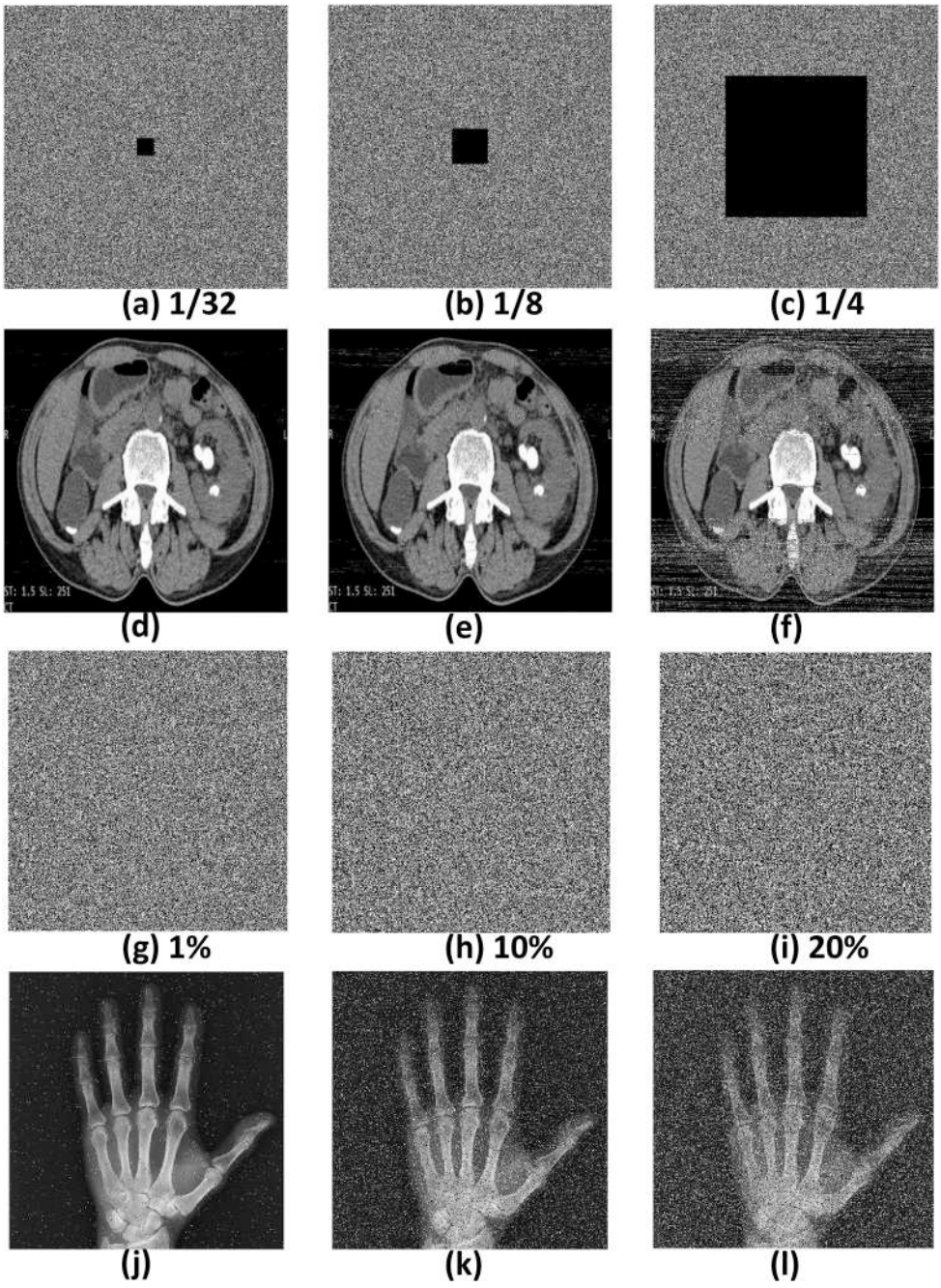

Fig. 14. The resilience of the proposed algorithm against partial data loss and noise attacks: (a) to (c) depict encrypted kidney images with 1/32, 1/16, and 1/4 data loss, respectively (d) to (f) display the corresponding decrypted images; (g) to (i) visualize encrypted images of the hand subjected to salt and pepper noise at 1 %, 10 %, and 20 % rates, respectively. (j) to (l) showcase the resulting decrypted images.

6. Conclusions

In this work, we presented a real-time hybrid image encryption algorithm combining memristor-based chaos with DNA coding. The proposed compact and inductorless chaotic circuit featured a simple topology without the need for active elements, resulting in a highly efficient solution in terms of power and area. Real-time encryption was achieved using the PYNQ-Z1 development board from Xilinx. Performance analysis results highlight the proposed algorithm's exceptional capabilities, making it a fitting choice for real-time encryption systems. Additionally, future research could explore further enhancements to optimize the algorithm's performance in various applications.

CRediT authorship contribution statement

Ahmet Samil Demirkol: Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Muhammet Emin Sahin:** Writing – original draft, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Baris Karakaya:** Writing – original draft, Methodology, Investigation, Formal analysis. **Hasan Ulutas:** Methodology, Investigation, Formal analysis. **Alon Ascoli:** Writing – review & editing, Methodology, Formal analysis. **Ronald Tetzlaff:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this the manuscript entitled, “Real time hybrid medical image encryption algorithm combining memristor-based chaos with DNA coding”.

Data availability

No data was used for the research described in the article.

Acknowledgment

This work was supported by TUBITAK 3501 (Grant number: 122E004) and the German Research Association (DFG) under the Project No. 411647366.

References

[1] Abbasi R, Bashir AK, Mateen A, Amin F, Ge Y, Omar M. Efficient security and privacy of lossless secure communication for sensor-based urban cities. IEEE Sensors J March 2024;24(5):5549–60. https://doi.org/10.1109/JSEN.2023.3305716.
[2] Hosny KM, Zaki MA, Lashin NA, Fouda MM, Hamza HM. Multimedia security using encryption: a survey. IEEE Access 2023;11:63027–56. https://doi.org/10.1109/ACCESS.2023.3287858.
[3] Sasikumar A, Ravi L, Kotecha K, Abraham A, Devarajan M, Vairavasundaram S. A secure big data storage framework based on blockchain consensus mechanism with flexible finality. IEEE Access 2023;11:56712–25. https://doi.org/10.1109/ACCESS.2023.3282322.
[4] Tran H-Y, Hu J. Privacy-preserving big data analytics a comprehensive survey. J Parallel Distrib Comput 2019;134:207–18. https://doi.org/10.1016/j.jpdc.2019.08.007.
[5] U. Zia, M. McCartney, B. Scotney, B. et al., “Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains,” in Int J Inf Secur, vol. 21, pp. 917–935, 2022, doi:https://doi.org/10.1007/s10207-022-00588-5.
[6] Zhang B, Liu L. Chaos-based image encryption: review, application, and challenges. Mathematics Jan. 2023;no. 11. https://doi.org/10.3390/math11112585.
[7] Liu X, Tong X, Zhang M, Wang Z. A highly secure image encryption algorithm based on conservative hyperchaotic system and dynamic biogenetic gene algorithms. Chaos Solitons & Fractals June 2023;171(113450). https://doi.org/10.1016/j.chaos.2023.113450.
[8] Çavuşoğlu Ü, Akgül A, Zengin A, Pehlivan I. The design and implementation of hybrid RSA algorithm using a novel chaos based RNG. Chaos, Solitons Fractals Nov. 2017;104:655–67. https://doi.org/10.1016/j.chaos.2017.09.025.
[9] Niu Y, Zhang J, Wang A, Chen C. An efficient collision power attack on AES encryption in edge computing. IEEE Access 2019;7:18734–48. https://doi.org/10.1109/ACCESS.2019.2896256.
[10] Alotaibi M. Improved blowfish algorithm-based secure routing technique in IoT-based WSN. IEEE Access 2021;9:159187–97. https://doi.org/10.1109/ACCESS.2021.3130005.
[11] Belazi A, Talha M, Kharbech S, Xiang W. Novel medical image encryption scheme based on chaos and DNA encoding. IEEE Access 2019;7:36667–81. https://doi.org/10.1109/ACCESS.2019.2906292.
[12] Wang X, Chen X. An image encryption algorithm based on dynamic row scrambling and zigzag transformation. Chaos Solitons & Fractals June 2021;147(110962). https://doi.org/10.1016/j.chaos.2023.113450.
[13] Wu R, et al. AEA-NCS: An audio encryption algorithm based on a nested chaotic system. Chaos, Solitons Fractals Dec. 2022;165(112770). https://doi.org/10.1016/j.chaos.2022.112770.
[14] Adeniyi AE, Misra S, Daniel E, Bokolo A. Computational complexity of modified blowfish cryptographic algorithm on video data. Algorithms Oct. 2022;15(373). https://doi.org/10.3390/a15100373.
[15] Aribilola I, Asghar MN, Kanwal N, Fleury M, Lee B. SecureCam: selective detection and encryption enabled application for dynamic camera surveillance videos. IEEE Trans Consum Electron May 2023;69(2):156–69. https://doi.org/10.1109/TCE.2022.3228679.
[16] A. Ascoli, V. Senger, R. Tetzlaff, N. Du, O. G. Schmidt and H. Schmidt, "BiFeO3 memristor-based encryption of medical data," 2016 IEEE international symposium on circuits and systems (ISCAS), Montreal, QC, Canada, 2016, pp. 1602–1605, doi: https://doi.org/10.1109/ISCAS.2016.7538871.
[17] Dehghani R, Kheiri Hossein. Chaotic-based color image encryption using a hybrid method of reversible cellular automata and DNA sequences. Multimed Tools Appl July 2023;83:17429–50. https://doi.org/10.1007/s11042-023-16118-x.
[18] Liu X, Sun K, Wang H, He S. A class of novel discrete memristive chaotic map. Chaos, Solitons Fractals Sep. 2023;174(113791). https://doi.org/10.1016/j.chaos.2023.113791.
[19] Demir K, Ergün S. Cryptanalysis of a random number generator based on continuous-time chaos. IET Circuits Devices Syst April 2020;14:569–75. https://doi.org/10.1049/iet-cds.2019.0356.
[20] Gong L-H, Luo H-X, Wu R-Q, Zhou N. New 4D chaotic system with hidden attractors and self-excited attractors and its application in image encryption based on RNG. Physica A Dec. 2021;591(126793). https://doi.org/10.1016/j.physa.2021.126793.
[21] Qobbi Y, Jarjar A, Essaid M, Benazzi A. Image encryption algorithm based on genetic operations and chaotic DNA encoding. Soft Comput Jan. 2022;26(12):5823–32. https://doi.org/10.1007/s00500-021-06567-7.
[22] Tsafack N, et al. A new chaotic map with dynamic analysis and encryption application in internet of health things. IEEE Access 2020;8:137731–44. https://doi.org/10.1109/ACCESS.2020.3010794.
[23] Tian K, Grebogi C, Ren H-P. Chaos generation with impulse control: application to non-chaotic systems and circuit design. IEEE Trans Circuits Syst I Regul Pap July 2021;68(7):3012–22. https://doi.org/10.1109/TCSI.2021.3075550.
[24] J. Petrzela, “Chaos in analog electronic circuits: comprehensive review, solved problems, open topics and small example,” in Mathematics, vol. 10, no. 4108, Nov. 2022, doi:https://doi.org/10.3390/math10214108.
[25] Vaidyanathan S, et al. A 5-D multi-stable Hyperchaotic two-disk dynamo system with no equilibrium point: circuit design, FPGA realization and applications to TRNGs and image encryption. IEEE Access 2021;9:81352–69. https://doi.org/10.1109/ACCESS.2021.3085483.
[26] Z. Faghani, F. Nazarimehr, S. Jafari, and J. C. Sprott, “A New Category of Three-Dimensional Chaotic Flows with Identical Eigenvalues,” in Int. J. of Bifurcation and Chaos, vol. 30, no. 2, Feb. 2020, doi:https://doi.org/10.1142/s0218127420500261.
[27] L.-H. Gong, R.-Q. Wu, and N. Zhou, “A new 4D chaotic system with coexisting hidden chaotic attractors,” in Int J Bifurcation Chaos, vol. 30, no. 10, Aug. 2020, doi:https://doi.org/10.1142/s0218127420501424.
[28] M. Yildirim and Fırat Kaçar, “Chaotic circuit with OTA based memristor on image cryptology,” in AEU - Int J Electron Commun, vol. 127, no. 153490, Dec. 2020, doi: https://doi.org/10.1016/j.aeue.2020.153490.
[29] Sahin ME, Cam G, Guler H, Hamamci SE. Application and modeling of a novel 4D Memristive chaotic system for communication systems. Circuits Syst Signal Process Jan. 2020;39(7):3320–49. https://doi.org/10.1007/s00034-019-01332-6.
[30] Muthuswamy B, Chua LO. Simplest chaotic circuit. Int J Bifurcation Chaos 2010;20(5):1567–80. https://doi.org/10.1142/S0218127410027076.
[31] A. Buscarino, L. Fortuna, M. Frasca, L. V. Gambuzza, “A chaotic circuit based on Hewlett-Packard memristor,” in Chaos, vol. 22, no. 2, June 2012, doi:https://doi.org/10.1063/1.4729135.
[32] Wu J, Wang G, Iu HH, Shen Y, Zhou W. A nonvolatile fractional order memristor model and its complex dynamics. Entropy 2019;21(955). https://doi.org/10.3390/e21100955.
[33] Wu R, Wang C. A new simple chaotic circuit based on Memristor. Int J Bifurcation Chaos 2016;26(1650145). https://doi.org/10.1142/s0218127416501455.
[34] Liang Y, Lu Z, Wang G, Dong Y, Yu D, Iu HH-C. Modeling simplification and dynamic behavior of N-shaped locally-active Memristor based oscillator. IEEE Access 2020;8:75571–85. https://doi.org/10.1109/ACCESS.2020.2988029.
[35] Jin P, Wang G, Liang Y, Chen L, Iu HH-C, Chua LO. Poor man’s Memristor: Chua corsage Memristor. IEEE Trans Circuits Syst II Express Briefs Aug. 2023;70(8):3139–43. https://doi.org/10.1109/TCSII.2023.3252524.
[36] Liang Y, Wang G, Chen G, Dong Y, Yu D, Iu HH-C. S-type locally active Memristor-based periodic and chaotic oscillators. IEEE Trans Circuits Syst I Regul Pap Dec. 2020;67(12):5139–52. https://doi.org/10.1109/TCSI.2020.3017286.
[37] Sahin ME, Demirkol AS, Guler H, Hamamci SE. Design of a hyperchaotic memristive circuit based on wien bridge oscillator. Comput Electr Eng 2020;88(106826). https://doi.org/10.1016/j.compeleceng.2020.106826.
[38] Bao B, Jiang T, Wang G, Jin P, Bao H, Chen M. Two-memristor-based Chua’s hyperchaotic circuit with plane equilibrium and its extreme multistability. Nonlinear Dyn Apr. 2017;89(2):1157–71. https://doi.org/10.1007/s11071-017-3507-0.
[39] X. Liu and J. Wang, “The Simplest Memristor Circuit With Hyperchaos,” in Frontiers in Physics, vol. 10, Jun. 2022, doi:https://doi.org/10.3389/fphy.2022.904200.
[40] Minati L, Gambuzza LV, Thio WJ, Sprott JC, Frasca M. A chaotic circuit based on a physical memristor. Chaos, Solitons Fractals Sep. 2020;138(109990). https://doi.org/10.1016/j.chaos.2020.109990.
[41] C. Volos, H. Nistazakis, V.-T. Pham and I. Stouboulos, "The first experimental evidence of chaos from a nonlinear circuit with a real memristor," 2020 9th international conference on modern circuits and systems technologies (MOCAST), Bremen, Germany, 2020, pp. 1–4, doi:https://doi.org/10.1109/MOCAST49295.2020.9200269.
[42] Shafique A, Ahmed J, Boulila W, Ghandorh H, Ahmad J, Rehman MU. Detecting the security level of various cryptosystems using machine learning models. IEEE Access 2021;9:9383–93. https://doi.org/10.1109/ACCESS.2020.3046528.
[43] Karakaya B, Gülten A, Frasca M. A true random bit generator based on a memristive chaotic circuit: analysis, design and FPGA implementation. Chaos, Solitons Fractals Feb. 2019;119:143–9. https://doi.org/10.1016/j.chaos.2018.12.021.
[44] D. A. Trujillo-Toledo et al., “Real-time RGB image encryption for IoT applications using enhanced sequences from chaotic maps,” in Chaos, Solitons & Fractals, vol. 153, p. 111506, Dec. 2021, doi:https://doi.org/10.1016/j.chaos.2021.111506.
[45] Liao TL, Wan PY, Yan J-J. Design and synchronization of Chaos-based true random number generators and its FPGA implementation. IEEE Access 2022;10:8279–86. https://doi.org/10.1109/ACCESS.2022.3142536.

[46] Lin H, Wang C, Du S, Yao W, Sun Y. A family of memristive multibutterfly chaotic systems with multidirectional initial-based offset boosting. Chaos, Solitons Fractals Jul. 2023;172(113518). https://doi.org/10.1016/j.chaos.2023.113518.
[47] Yu F, Shen H, Yu Q, Kong X, Sharma PK, Cai S. Privacy protection of medical data based on multi-scroll Memristive Hopfield neural network. IEEE Trans Netw Sci Eng 1 March–April 2023;10(2):845–58. https://doi.org/10.1109/TNSE.2022.3223930.
[48] Lin H, Wang C, Cui L, Sun Y, Xu C, Yu F. Brain-like initial-boosted hyperchaos and application in biomedical image encryption. IEEE Trans Industr Inform Dec. 2022;18(12):8839–50. https://doi.org/10.1109/TII.2022.3155599.
[49] Malik MGA, Bashir Z, Iqbal N, Imtiaz MA. Color image encryption algorithm based on hyper-Chaos and DNA computing. IEEE Access 2020;8:88093–107. https://doi.org/10.1109/ACCESS.2020.2990170.
[50] Singh AK, Chatterjee K, Singh A. An image security model based on Chaos and DNA cryptography for IIoT images. IEEE Trans Ind Informatics Feb. 2023;19(2):1957–64. https://doi.org/10.1109/TII.2022.3176054.
[51] Liu Z, Wu C, Wang J, Hu Y. A color image encryption using dynamic DNA and 4-D Memristive hyper-Chaos. IEEE Access 2019;7:78367–78. https://doi.org/10.1109/ACCESS.2019.2922376.
[52] C. Li, Z.-Y. Li, W. Feng, Y.-N. Tong, J.-R. Du, and Du Qu Wei, "Dynamical behavior and image encryption application of a memristor-based circuit system," in AEU Int J Electron Commun, vol. 110, no. 152861, Oct. 2019, doi:https://doi.org/10.1016/j.aeue.2019.152861.
[53] Dagadu JC, Li J, Aboagye EO. Medical image encryption based on hybrid chaotic DNA diffusion. Wirel Pers Commun Sep. 2019;108(1):591–612. https://doi.org/10.1007/s11277-019-06420-z.
[54] Sahin ME. Memristive chaotic system-based hybrid image encryption application with AES and RSA algorithms. Physica Scripta Jun. 2023;98(7). https://doi.org/10.1088/1402-4896/acdba0.
[55] "PYNQ-Python productivity for Zynq." http://www.pynq.io/ (accessed March 21, 2024).
[56] Rukhin AL, Soto J, Nechvatal JR, Smid ME, Barker EB. A statistical test suite for random and pseudorandom number generators for cryptographic applications. National Institute of Standards and Technology (U.S.); Dec. 2000. https://doi.org/10.6028/nist.sp.800-22r1a.
[57] A. S. Demirkol, M. M. Al Chawa, A. Ascoli, R. Tetzlaff, D. Bedau and M. Grobis, "A locally active device model based on a minimal 2T1R circuit," 2022 29th IEEE international conference on electronics, circuits and systems (ICECS), Glasgow, United Kingdom, 2022, pp. 1–4, doi:https://doi.org/10.1109/ICECS202256217.2022.9970873.
[58] L. Chua, Juebang Yu and Youying Yu, "Bipolar-JFET-MOSFET negative resistance devices," in IEEE Trans Circuits Syst, vol. 32, no. 1, pp. 46–61, January 1985, doi: https://doi.org/10.1109/TCS.1985.1085599.
[59] Sprott JC. Elegant chaos. World Scientific 2010. https://doi.org/10.1142/7183.
[60] Elsaid SA, Alotaibi ER, Alsaleh S. A robust hybrid cryptosystem based on DNA and Hyperchaotic for images encryption. Multimed Tools Appl Jun. 2022;82:1995–2019. https://doi.org/10.1007/s11042-022-12641-5.
[61] Iqbal N, et al. On the image encryption algorithm based on the chaotic system, DNA encoding, and castle. IEEE Access 2021;9:118253–70. https://doi.org/10.1109/ACCESS.2021.3106028.
[62] Mansoor S, Parah SA. HAIE: a hybrid adaptive image encryption algorithm using Chaos and DNA computing. Multimed Tools Appl Feb. 2023;82(19):28769–96. https://doi.org/10.1007/s11042-023-14542-7.
[63] Masood F, et al. A new color image encryption technique using DNA computing and Chaos-based substitution box. Soft Comput Dec. 2021;26(16):7461–77. https://doi.org/10.1007/s00500-021-06459-w.
[64] P. N. Lone, U. H. Mir and A. Gaffar, "Hyperchaotic image encryption using DNA coding and discrete cosine transform," in J Franklin Inst, vol. 360, no. 17, pp. 13318–13338, Nov. 2023, doi:https://doi.org/10.1016/j.jfranklin.2023.10.010.
[65] C. D. Canniere, "Trivium: a stream cipher construction inspired by block cipher design principles," Lect Notes Comput Sci, vol. 4176, pp. 171–186, Jan. 2006.
[66] X. Wang, W. Xue and J. An, "Image encryption algorithm based on tent-dynamics coupled map lattices and diffusion of household," in Chaos, Solitons Fractals, vol. 141, no. 110309, Dec. 2020, doi:https://doi.org/10.1016/j.chaos.2020.110309.
[67] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," J Cryptol, vol. 4, no. 1, pp. 3–72, Jan. 1991, doi:https://doi.org/10.1007/bf00630563.
[68] Y. Wang, K.-W. Wong, X. Liao, T. Xiang, and G. Chen, "A chaos-based image encryption algorithm with variable control parameters," Chaos, Solitons Fractals, vol. 41, no. 4, pp. 1773–1783, Aug. 2009, doi:https://doi.org/10.1016/j.chaos.2008.07.031.
[69] A. Mansouri and X. Wang, "A novel one-dimensional chaotic map generator and its application in a new index representation-based image encryption scheme," in Infor. Sciences, vo. 563, pp. 91–110, doi:https://doi.org/10.1016/j.ins.2021.02.022.