Int. j. inf. tecnol.
https://doi.org/10.1007/s41870-024-01787-0

ORIGINAL ARTICLE

Hyperchaos-based cryptosytem to secure WSN data transfer for smart home application

Samir Benzegane[1] · **Mounir Bouhedda**[1] · **Noureddine Batel**[1]

Received: 4 October 2023 / Accepted: 6 February 2024
© The Author(s), under exclusive licence to Bharati Vidyapeeth's Institute of Computer Applications and Management 2024

Abstract The purpose of this work is to introduce a new crypto-system for the protection of information transfer within the context of the intelligent home implementation. A Hyperchaos-based Key Stream Generator (H-KSG) based on Chen's Hyperchaos scheme is used to produce high-performance dynamic key chips with high quality randomness as confirmed by conventional National Institute of Standards and Technologies (NIST) statistical tests. The advanced crypto-system was implemented on microcontroller-based ARM architecture and tested for smart home applications. Efficiency and high safety level of the proposed system are shown by the outcomes of cryptanalysis and the required safety tests. The results of cryptanalysis and the necessary security performance tests show the efficiency and the high level of security of the proposed system.

Keywords Cryptosystem · Chen system · H-KSG · Hyperchaos · NIST,WSN

1 Introduction

Smart home is one of the Internet of Things (IoT) applications and targets technology where all home-based devices interact with each other to achieve automation [1, 2]. One of the most relevant solutions used in the intelligent home based IoT is the Wireless Sensors Network (WSN). Single components (sensors and actuators) are networked together via a wireless medium in order to perform an intelligent control system. This latter gets the important information after analysis to expand the comfort, save energy and ensure more safety in the home. Three layers exist in each sensor node of the WSNs; sensor layer for sensing the different environmental quantities, processing layer for calculating of the acquired information and a communication layer for sending to the actuators and exchanging the collected sensors' data [3–7].In a low-security smart home system, user privacy information can be exposed to intruders or attackers with ease. It is, in fact, normal to worry about the personal information and to ensure that these data are safe and protected in all cases [8–11].Sensors and actuators send and receive a variety of values and information in a smart home. Data reliability and security play a significant role. It is therefore important to protect the smart home system through cryptography. Several systems have therefore been proposed to address this challenging issue, including traditional encryption processes as well as security practices developed for WSNs where the data are encrypted by each node individually and then decrypted by the respective receiver[12–17].The use of chaotic theory for the application of the encryption method was recently devoted to an increase in attention due to the principal benefit of these encryptions by the fact that unauthorized users do not know the way to produce a chaotic signal like noise [18–26]. Chaotic systems have ergodic, high sensitivity to originals and random behavior, linked to uncertainty and communication properties which in all cryptosystems are considered essential [20, 21, 27].Chaos systems, depending on the time range, can be described by

✉ Mounir Bouhedda
bouhedda.m@gmail.com

Samir Benzegane
samir_bz@hotmail.Fr

Noureddine Batel
batelnour@gmail.com

[1] Laboratory of Advanced Electronic Systems (LSEA), University of Medea, Algeria, Nouveau Pôle Urbain, 26000 Medea, Algeria

Published online: 30 March 2024

difference equations (discrete-time systems) or differential equations (continuous-time systems). Examples of the discrete time systems are the logistic map and the Henon map, while the Chen system and the Rossler system are examples of the continuous time systems.

In Table 1, a literature review is presented, with efforts made to distill insights, identify trends, and unravel the intricacies of prior research, thus laying the groundwork.

In this paper, an implementation of a Hyperchaos-based cryptosystem for securing data in smart home automation was developed, utilizing the Hyperchaotic Chen system. To ensure statistical key generation, a four-signal generator was employed simultaneously. The Chen system was implemented using the fourth-order Runge–Kutta (RK-4) method for resolving chaos systems. Data encoding involved base-64 encoding in double precision format, followed by the application of a nonlinear Boolean function. Key performance was validated using NIST statistical tests. An Arduino Due board, WI-FI shield, and DHT22 sensor were used for constructing a WSN-based platform. H-KSG was employed for key generation, and performance and security analyses, including assessments of key space and resistance to attacks, were conducted. Compliance with WSN safety requirements was verified, and comparisons with other algorithms in statistical and differential analysis were performed.

The paper is arranged accordingly. Section 2 gives a description of the proposed hyperchaotic keystream. An exhaustive security analysis is performed in Sect. 3. Section 4 describes the architecture system and the performance and security analysis for the implemented algorithms. Conclusion and future scope are finally drawn in Sect. 5.

2 Proposed H-KSG

The proposed H-KSG principle scheme is illustrated by the bloc diagram of Fig. 1. It consists of a hyperchaotic Chen generator [21] given by(1) and implemented using RK-4 method with encoding the fraction part p on 48 bits double precision floating point data format and removing the integer part n.

$$\begin{cases} \dot{x} = a(y-x) \\ \dot{y} = x(b-z) - cy \\ \dot{z} = xy + (d-y+x)z - w \\ \dot{w} = ew + (y-x)z \end{cases} \tag{1}$$

Then we apply the four nonlinear Boolean functions used in hash functions algorithms Message Digest 5 (MD5) [36] and Secure Hash Algorithm-1(SHA-1) [37] as coupling functions according to (2).

$$\begin{aligned} F(X_1, X_2, X_3) &= (X_1 \wedge X_2) \vee (\neg X_1 \wedge X_3) \\ G(X_1, X_2, X_4) &= X_1 \oplus X_2 \oplus X_4 \\ P(X_1, X_3, X_4) &= (X_1 \wedge X_2) \vee (\neg X_4 \wedge X_3) \\ I(X_2, X_3, X_4) &= X_3 \oplus (X_2 \wedge \neg X_4) \end{aligned} \tag{2}$$

where $\wedge$, $\vee$, $\neg$ and $\oplus$ are respectively AND, OR, NOT and XOR logic operators.

The hyperchaotic generator generates four hyperchaotic 48-bit keys K_1, K_2, K_3, and K_4 encryption keys at the same time as illustrated in Fig. 1. The quality of randomness of proposed H-KSG must be evaluated. The standard NIST (SP 800–22) statistical test suite is used to perform statistical testing with generated 48 bits encryption keys for this purpose. Each test has been conducted 300 times on 1 Mbit substrings in this test suite. A series of 100 is generated for each test, and a test sequence passes when $P_value \geq \alpha = 0.01$ where α indicates the level of significance of the test. With a value of $\alpha = 0.01$, 1% of the 100 sequences would fail. The sequence proportion that passes a test is $P_value \geq \alpha$ divided by 100. For each statistical test, the minimum rate for passes is approximately 0.972766 for sample size equal to 300 binary sequences except the random excursion variant test [38]. From the results of Table 2, we note that the encryption keys generated by the H-KSG pass all kinds of the NIST test. Thus, we can say that our H-KSG generates a good randomness encryption keys.

Perturbation techniques are applied in order to improve the dynamic complexity of H-KSG and enhance the performance of the produced sequences, as depicted in Fig. 2. The main idea is to encrypt the clear-text with the keys of our H-KSG generator. The encrypted message is looped back (feedback) to the input to disturb the generator and thus complicate its chaotic dynamics to the maximum. In addition, this process will ensure the phenomenon of avalanche and diffusion. Indeed, the main text or encrypted image alteration is triggered by a small modification to clear text or picture.

3 Security analysis

The primary measure, called security, for determining a cryptosystem's efficiency is its ability to withstand unauthorized access to information. A cryptosystem's effectiveness is its immunity to different attacks including brute force attack, differential attacks and statistical attacks. In this context, we performed some of the best known and used security tests such as key space, entropy calculation, autocorrelation, histograms and key sensitivity. These tests have

Table 1 Literature review

References	Title	Year	Implementation	Main Findings	Relevance to our work
[28]	A Secure Chaos-based Lightweight Cryptosystem for the Internet of Things	2023	Hardware (FPGA)	Evaluated the keystream a secure chaos-based generated by theirs proposed schemes with batteries of the NIST statistical tests Achieved higher resistance against statistical and differential attacks	Evaluation of the performance, security and reliability of the encryption scheme
[29]	A new combination chaotic system and its application in a new bit-level image encryption scheme	2022	Software		
[30]	A stable meaningful image encryption scheme using the newly-designed 2D discrete fractional-order chaotic map and Bayesian compressive sensing	2022	Software		
[31]	An image encryption algorithm based on new chaos and diffusion values of a truth table	2021	Software		
[32]	Image Encryption Scheme Based on Newly Designed Chaotic Map and Parallel DNA Coding	2023	Software		
[33]	Color Image Encryption Algorithm Based on a Chaotic Model Using the Modular Discrete Derivative and Langton's Ant	2023	Software		
[34]	Low Power Cryptography Solution Based on Chaos Theory in Wireless Sensor Nodes	2019	Hardware microcontroller	Evaluate and compare the existing cryptographic algorithms for sensors such as Skipjack and RC5 with their proposed algorithm based chaos	Hardware implementation and running time comparison
[35]	Speed and power consumption comparison between DES and AES algorithm in arduino	2019	Hardware microcontroller	Implementation and comparing DES and AES cryptographic algorithms in telemedicine context	

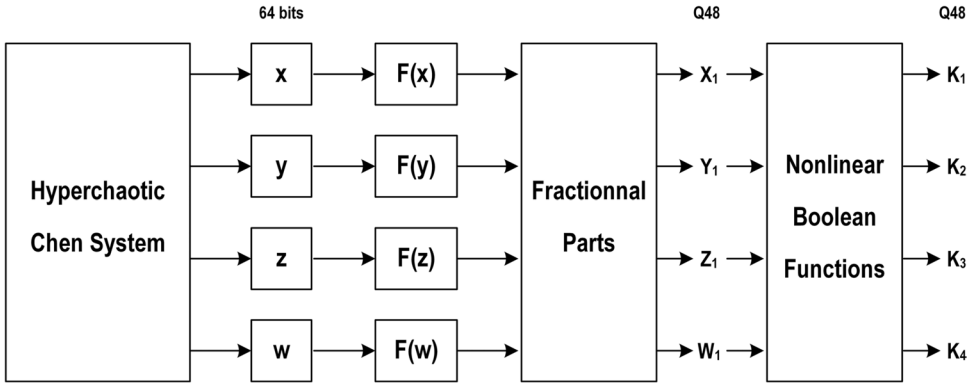
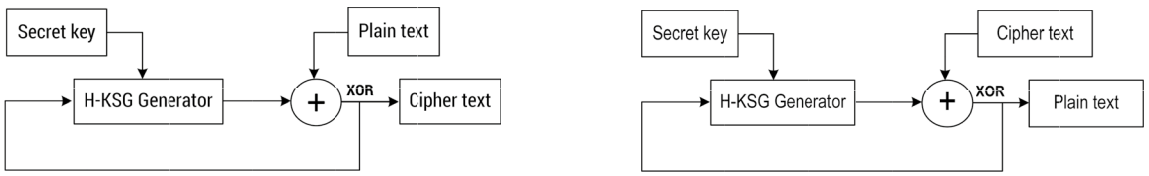


Fig. 1 Proposed H-KSG hyperchaotic generator keys

Table 2 NIST tests results

Tests		Results in %			
N°	Test type	K_1	K_2	K_3	K_4
1	Frequency	0.9933	0.9967	0.9933	0.9967
2	Block Frequency	0.9900	0.9900	0.9933	0.9967
3	Cumulative Sums Up	0.9867	0.9967	0.9833	1.0000
4	Cumulative Sums Down	0.9900	0.9933	0.9833	0.9967
5	Runs	0.9867	0.9867	0.9933	0.9867
6	Long Runs of One's	0.9933	0.9867	0.9967	0.9867
7	Rank	0.9867	0.9933	0.9933	0.9800
8	Discrete Fourier Transform	1.0000	1.0000	0.9933	0.9967
9	Non-overlapping Template Matching	0.9833	0.9967	0.9867	0.9833
10	Overlapping Template Matching	0.9833	0.9832	0.9833	0.9833
11	Maurer's "UniversalStatistical"	0.9933	0.9933	0.9900	0.9767
12	ApproximateEntropy	0.9833	0.9933	0.9833	0.9867
13	Random Excursions	0.9948	0.9944	0.9895	0.9896
14	Random Excursions Variant	0.9948	0.9888	0.9948	0.9844
15	Serial 1	0.9900	0.9833	0.9867	0.9833
16	Serial 2	0.9933	0.9967	0.9900	0.9867
17	Linearcomplexity	0.9967	0.9967	0.9933	0.9867



(a) Encryption process (b) Decryption process

Fig. 2 Cryptosystem scheme using perturbation

been applied using our cryptosystem of Fig. 2, to different reference images such as Lena, Barbara, Peppers, Baboon and Airplane [39].

3.1 Histogram analysis

A cryptosystem attack based on statistical analysis is one of the most frequent attacks [18, 40]. If the encrypted histogram image is distributed consistently, a cryptosystem is considered strongly against these attacks. The obtained result using the proposed cryptosystem is shown in Fig. 3. We notice that the histograms of the encrypted images

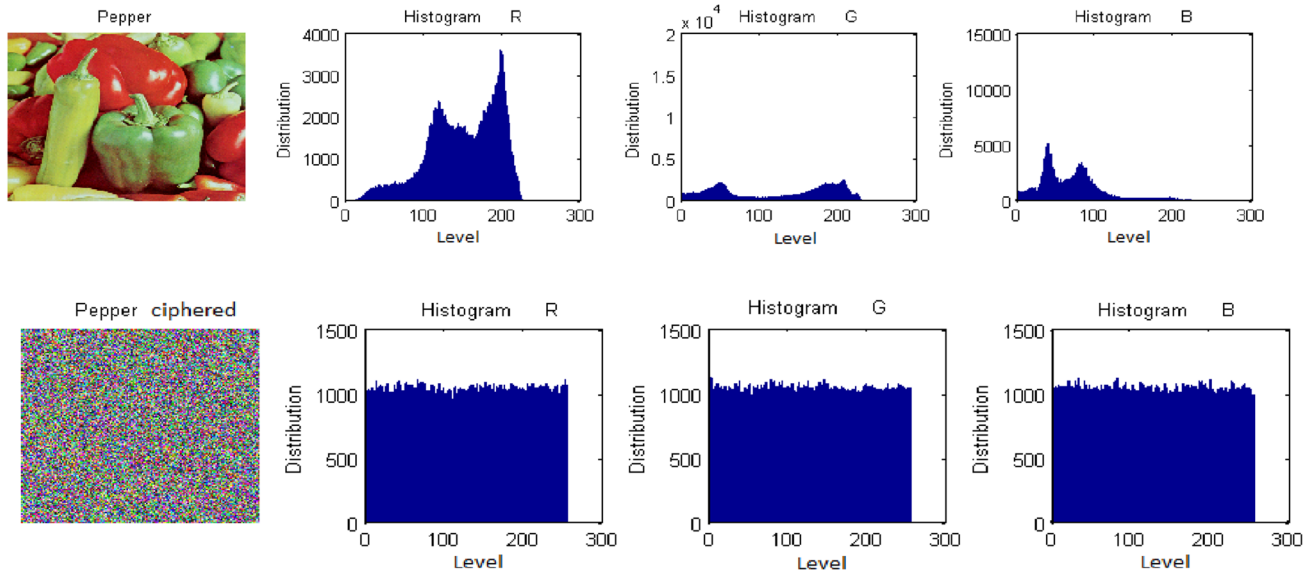


Fig. 3 Results of the histogram analysis

Table 3 Entropy calculation results

Image	Entropy (Original Image)	Entropy (Ciphered-Image)
Airplane	6.66390	7.99978
Baboon	7.69575	7.99871
Lena	5.68222	7.99980
Peppers	7.66982	7.99974

have a flat amplitude level compared to those of the original images. That confirms their random aspects and shows that the proposed cryptosystem is robust against the histogram attack.

3.2 Information entropy analysis

The entropy information of an image is a measure of the degree of randomness in the image. The Shannon entropy of a source M with symbols m (m = 256 Gy levels) is calculated by the following formula [26]:

$$H(M) = \sum_{i=1}^{m} p(m_i) \log_2 \left(\frac{1}{P(m_i)} \right) \text{ bits} \tag{3}$$

where $p(m_i)$ is the probability of the m_i symbol. According to (3), entropy H(M) is maximum in the case of uniform distribution and is expressed by (4) [26].

$$H_{\max} = \log_2(m) = \log_2(256) = 8 \text{ bits} \tag{4}$$

Thus, if the value of the entropy is very close to 8, it means that the distribution of gray levels in the image is almost uniform, this means that the cryptosystem is statistically resistant to attack. The entropy values calculated for some images are shown in Table 3. We realize that the entropy values are very close to 8 for our systems, proving that the encryption system is resistant to an entropy attack.

3.3 Key space analysis

The key space for the encryption method is the total number of keys that can be used. The main space size H should be large enough to avoid brute attacks. According to [41, 42] it should be greater than 2^{128}. Thus, the proposed H-KSG based on the Chen generator has five control parameters a, b, c, d and e as well as four initial conditions x_0,y_0, z_0, w_0. We can add the integration step h of the RK-4 method used to solve the equation system. Thus, we obtain a total of ten parameters coded on 64 bits. Therefore, formula (5) gives the H-key space of the proposed cryptosystem. The key space of our cryptosystem is thus adequate to withstand brutal attacks.

$$H(a, b, c, d, e, h, x_0, y_0, z_0, w_0) = \left(2^{64}\right)^{10} = 2^{640} \tag{5}$$

3.4 Sensitivity analysis

Robust image encryption should be sensitive to any secret key changes. Any key change must generate an encoded picture entirely different. Therefore, key sensitivity for a cryptosystem

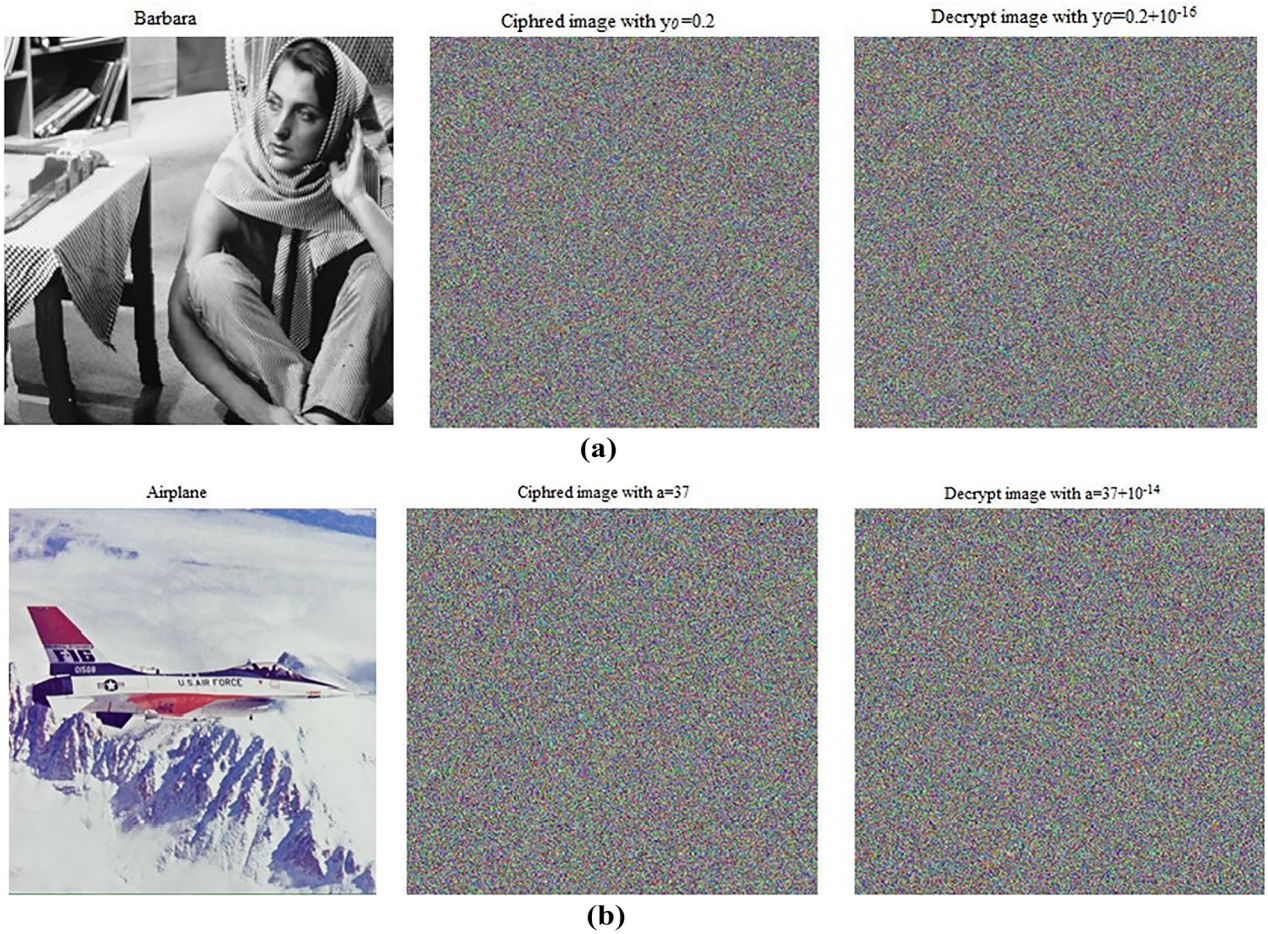


Fig. 4 Image encryption sensitivity

is important. The minor change to the decryption key leads to major improvements to the decryption result in a chaos-based encryption because of its properties [43]. We consider the two images (Barbara and Airplane), we evaluated the criterion of the sensitivity of the key by sensitivity to initial and parameters conditions. Sensitivity test results to the initial conditions given by Fig. 4 shows that decryption with the value $y_0 = 0.2 + 10^{-16}$ of the encrypted image (Barbara) with the value $y_0 = 0.2$ does not allow a correct reconstitution of the original image. Similarly for the sensitivity test to the parameters, for a decryption with the value of $a = 37 + 10^{-14}$ of the encrypted image (plane) with the value $a = 37$. Therefore, our cryptosystem is very sensitive to the slightest variation in the initial conditions or in the control parameters.

3.5 Correlation coefficient analysis

The correlation coefficient determination among the adjacent pixels helps to measure the performance of the encryption cryptosystem. In a clear image each pixel is strongly correlated to the adjacent pixel, allowing the value of a pixel to be determined from those around pixels. When two variables are closely connected, the correlation factor is close to 1 or – 1. The best way to create an encrypted image is to have zero correlation coefficients horizontally, vertically and diagonally. Formulas (6), (7), (8) and (9) are used for the calculation of the coefficient r[33].

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{6}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x_i)\right)^2 \tag{7}$$

$$\operatorname{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x_i)\right)\left(y_i - E(y_i)\right) \tag{8}$$

$$r_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)D(y)}} \tag{9}$$

Table 4 Correlation coefficients

Algorithm	Image	Correlation original image			Correlation ciphered image		
		H	V	D	H	V	D
ProposedH-KSG	Lena	0.97527	0.98530	0.96483	– 0.00022	– 0.00043	– 0.00162
[28]	Lena	0.9850	0.9719	0.9593	0.0093	0.0032	– 0.0019
[29]	Lena	0.96850	0.96252	0.96462	0.0005	– 0.0025	0.0028
[30]	Lena	0.9859	0.9741	0.9618	0.0052	00.0112	0.0034
ProposedH-KSG	Barbara	0.89274	0.95421	0.88393	– 0.00003	0.00129	– 0.00016
[28]	Barbara	0.9542	0.8927	0.8839	0.0022	0.0053	– 0.0005
[31]	Barbara	0.9588	0.8991	0.8733	– 0.0002	0.0081	– 0.0011

x and y are two adjacent pixels of the image. Table 4 shows the correlation coefficients of adjacent pixels, calculated for certain images with different cryptosystems in three different directions (horizontal, vertical and diagonal) using our proposed system. The results obtained are comparable. The correlation distributions in the original image (a) and the encrypted one (b) with the H-KSG algorithm are displayed in Fig. 5. It is found that the surrounding pixels in the single image are too much correlated whereas the encrypted image has a small correlation of the nearby pixels.

3.6 Differential analysis

The high sensitivity to small changes in the original image is a valuable feature of cryptosystems. An attacker can make a slight change to the encrypted image (such as a single pixel) and observe the results changes. He can thus consider a meaningful relationship between the clear image and the encrypted image. If a small change in the clear image will cause the encrypted frame to change significantly (avalanche effect), the differential attack is useless. Two metrics, introduced by Eli Biham and Adi Shamir [44], are commonly used to measure cryptosystem resistance to differential attacks. They are namely the Number of Pixel Change Rate (NPCR) and the Unified Average Changing Intensity (UACI) defined by (10) and (11) respectively.

$$NPCR = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} D(i,j)}{M \times N} \times 100 \tag{10}$$

$$UACI = \frac{1}{M \times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\frac{|I_1(i,j) - I_2(i,j)|}{255} \times 100 \tag{11}$$

where M and N are the width and height of the image respectively.$I_1(i,j)$ and $I_2(i,j)$ are pixel values at the position (i, j) of the two encrypted pictures, whose image originals differ by one pixel only. D is a matrix that have the same size as I_1 and I_2 equal and calculated with (12).

$$D(i,j) = \begin{cases} 1 & \text{if } I_1(i,j) \neq I_2(i,j) \\ 0 & \text{else} \end{cases} \tag{12}$$

The NPCR indicates the number of ciphered image pixels changing rate, while changing one single pixel. The UACI calculates the average intensity of variations between the simple image and the ciphered image. The NPCR has an optimal value of 99.61%, while UACI is 33.46% [32]. Table 5 gives the results of the differential analysis of the proposed cryptosystem. For Barbara and Lena images, the analysis consists in fact to encrypt two different clear images only by one pixel. The first is a clear original image and the second is obtained with the same image after making a change in the bit of the first pixel. The two images are encrypted with the same key to produce the corresponding encrypted images I_1 and I_2.

The results presented in Table 5 show the robustness in differential attacks of the proposed cryptosystem. It is obvious, indeed, that both NPCR and UACI values are very close to the 99.61% and 33.46%, which confirms the high sensitivity of the proposed cryptosystem towards minor changes in the original image that have a significant impact on the encrypted image.

4 Methods and implementation

The WSN-system developed consists of the collection of data that can be shared between source and destination nodes safely and efficiently. Every sensor's specific information is also stored on the server and the data can be analyzed correctly when validated. For home automation systems, secure data transfer is therefore important.

All security constraints such as confidentiality, authentication, transparency, trustworthiness and freshness of data must be met [45, 46]. This research meets network security

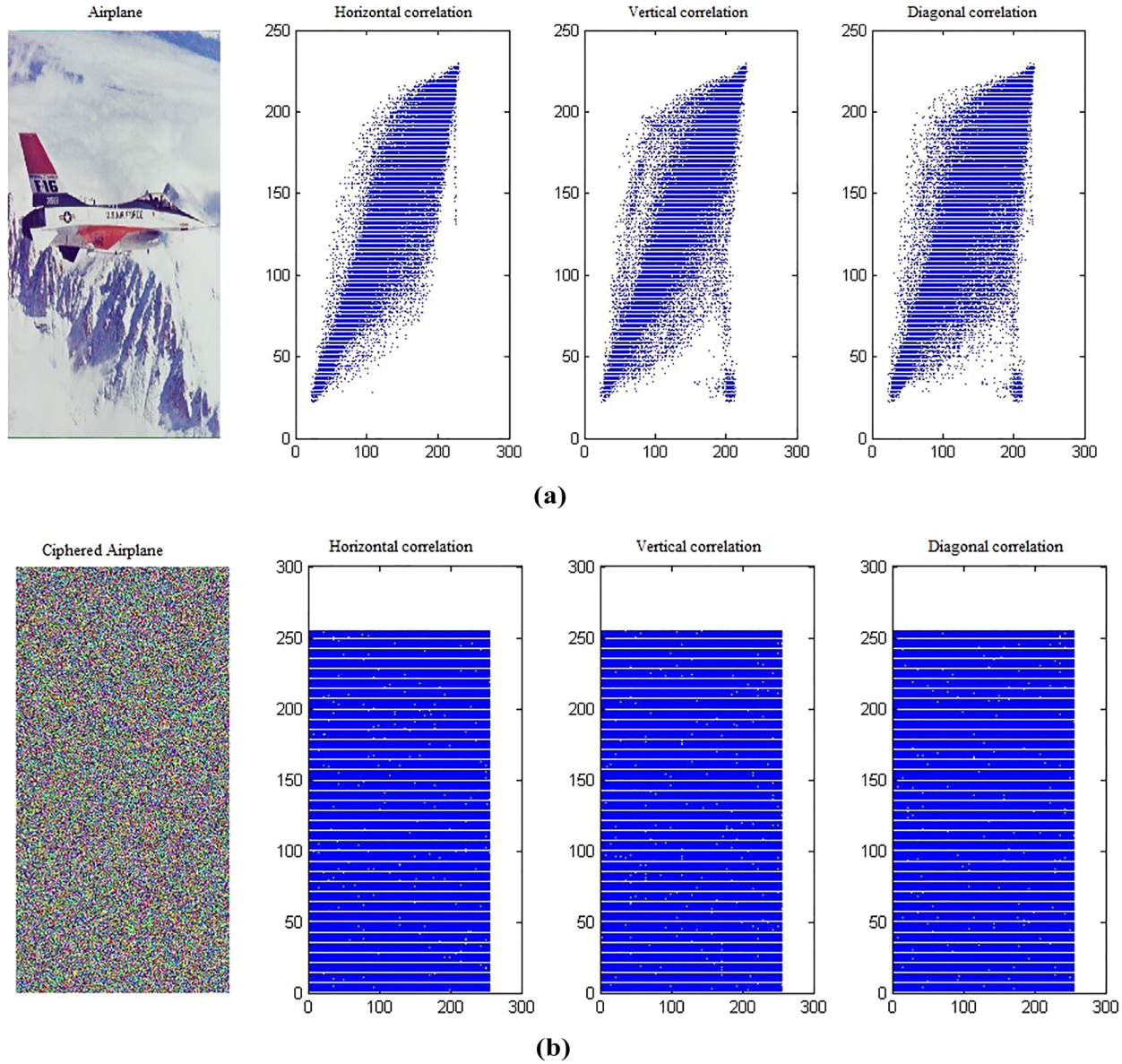


Fig. 5 Analysis of the correlation of adjacent pixels

requirements by designing a stable WSN-based platform powered by the proposed H-KSG to meet data security constraints. In this analysis, the DHT22 temperature and humidity sensor was used to measure ambient temperature and humidity.

Table 5 UACI and NPCR values for Lena, Peppers and Barbara images

Algorithm	Image	UACI (%)	NPCR (%)
Proposed H-KSG	Lena	33.4051	99.5970
[32]	Lena	33.4618	99.6085
Proposed H-KSG	Peppers	33.4784	99.6050
[33]	Peppers	33.4211	99.6185
Proposed H-KSG	Barbara	33.4701	99.6269
[28]	Barbara	33.5432	99.6001

4.1 Proposed architecture system

The different components of the sensor nodes are shown in Fig. 6.The humidity and temperature sensor (DHT22) is presented in Fig. 6c, while Fig. 6b shows the Wi-Fi module (Arduino Wi-Fi shield). These two components were both integrated on an Arduino Due board (Fig. 6a) to build the wireless sensor node. The Arduino Due is an Atmel SAM3X8E ARM Cortex-M3 CPU-based microcontroller

board. In smart home-based applications, it is important that sensor nodes are synchronized with information received from a remote server located at a large distance with reasonable speed and reliability. Therefore, the Wi-Fi Shield was utilized in our system to allow arduino Due to connect and transmit information over a wireless network. This module also has a network Internet Protocol (IP) stack capable of transmitting data from both TCP/IP and UDP protocol.

As Fig. 7 shows, two types of entities constitute the proposed WSN architecture. A system node that is used to monitor environmental parameters (humidity and temperature), and then sends the data via a Wi-Fi network to the server through a trustworthy TCP/IP protocol and the proposed encryption algorithm.

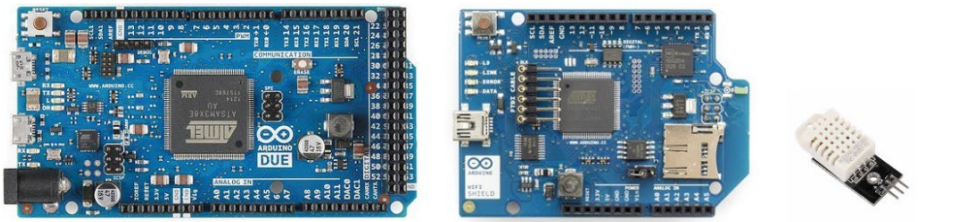
The server application written in C# using the.NET Framework achieves sensed data storage after proper decryption and authentication. The data collected is handled with an SQL database language, allowing various operations and analyzes of stored data to be carried out and then taking the necessary corrective action for automation.

4.2 Proposed security method

There are two stages of the proposed protection approach. The first level is to secure and control the Wi-Fi network, and the second level is to fix data security.

4.2.1 Protection and administration Wi-Fi network

We have hidden the Service Set Identifier (SSID) and used the Media Access Control (MAC) address filtering, in order to prevent externals accessing the Wireless Local Area Network (WLAN). Indeed,a network can protect itself by refusing to give its SSID following a request for the presence of a client. The network appears as unnamed. While the client has to know in advance the name of the network and the password for accessing the network. Each device with a network adapter has a unique MAC address that identifies it on the network. The MAC address filtering router will first check the MAC address of the user against the accepted MAC address and will only enable the device to enter the Wi-Fi network if its MAC address has been explicitly approved. So, the server establishes and recognizes a connection for the node through a secure TCP/IP protocol after proper validation of the SSID, password and MAC address. If no match exists, the connection request will be refused.



a) *Arduino Due board* b) *WIFI Shield* c) *DHT 22 Sensor*

Fig. 6 Components of a node sensor



Fig. 7 Software and Hardware implementation of WSNs

4.2.2 Set-Up of the proposed encryption algorithm

In this section, we present our proposal for a hyperchaotic encryption algorithm that consists of using a synchronous stream cipher mode. The purpose of stream ciphers is to imitate the Vernam Cipher by continuously producing and synchronizing new keys between the sender and the receiver[47].

The keystream is created independently of the plain text or the encrypted message in a synchronous stream Cipher. A pseudorandom generator, parameterized by secret key, produces the keystream generator (in our case H-KSG).In synchronous stream cipher mode sender and receiver have to be synchronized in order to decrypt successfully. One way to reach synchronization is to send an Initialization Vector (*Init_Vect*) before each encrypted frame with the encrypted message.

Since a nonrandom Initialization Vector (*Init_Vect*) weakens the encryption and allows dictionary and differential attacks to have a better chance to decipher the encrypted message, we use an *Init_Vect* with higher randomicity and which also changed every new transmission. In fact, to be safe, a stream cipher should not be using the same key to encrypt various messages.In order to hide the *Init_Vect* in the transmitted frame and increase the performances of the proposed algorithm,we include a shuffling process array which performs an efficient mixture of a frame's segment [48].In addition, a secure communication in Smart Home means that legitimate sensors can be identified in the home environment. An anonymous sensor node can embody communication in the home and access private data. Protection systems need to classify these anonymous nodes in order to have a special and secure ID that can be found specific to the server on each device within the network. For the encryption process, the variables in upper case X_0, Y_0, Z_0 and W_0 and the parameters a, b, c, d and e will represent the secret key whereas the variables in lower case x_n, y_n, z_n and w_n will represent the values corresponding to the last iteration of the H-KSG generator (latest solution). Therefore, the secret key, the initial vector nonce *Init_Vect* are the inputs of the hyperchaotic generator H-KSG.We also want to mention that the K_1 encryption keys will be used to encrypt the temperature, K_2 for the humidity, K_3 for the ID and finally K_4 for Cyclic Redundancy Check (CRC) which is used as a Message Authentication Codes (MAC).All of the temperature, humidity, ID and CRC data are coded in two Bytes. The details of the encryption and decryption process are shown respectively in Algorithm 1 and Algorithm 2.

Algorithm 1 : Encryption process

Input: temperature **T** and humidity **H**
1: CRC = (ID ⊕ T ⊕ H) // ⊕ denotes a xor operation and ID represents a unique and a secret identification code of each nodes.
2: V = ID ||T ||H|| CRC. // || means string concatenation. The array V forms the plaintext.
3: **Init_Vect'=[x_n, y_n, z_n,w_n]** // a last iteration for H-KSG generator
4: **Init_Vect =Init_Vect'⊕** [X_0⊕a, Y_0⊕b, Z_0⊕c, W_0⊕d] // X_0, Y_0,Z_0,W_0, a, b, c and d are a secret Key.
5: **C=V ⊕ K** // using **Init_Vect** as nonce for **H-KSG** generator. K represents a key stream.
6: A=**Init_Vect'** || **C** // The array A forms a final ciphered array.
7: A ←Shuffling array process [49].
Output : send **A** using TCP/IP protocol to the server

Algorithm 2 : Decryption process

Input :Array **A**
1: A←inverse array Shuffling process.
2: Extract Init_Vect'i+1=**[x_n, y_n, z_n,w_n]**. // Through comparing **Init_Vect'i+1** and the last saved *Init_Vect'I* the server verify the data freshness.
3: **If** (Init_Vect'i+1==Init_Vect'i**)** // the server discards the message and returns to step1.
else Init_Vect =Init_Vect'i+1⊕ [X_0⊕a, Y_0⊕b, Z_0⊕c, W_0⊕d] // X_0, Y_0,Z_0,W_0, a, b, c and d are a secret key.
4: M=A⊕K. // M plain array and K represents a key stream.
5: Compute the CRC' for a plain array M.
6: Extract ID and CRC from M.
7: **If** (CRC'≠ CRC) or ID is no recorded in the data base the server discards the message.
else the server save **Init_Vect'i+1** and erase the last **Init_Vect'i** .
Output : display and records **T** and **H** data.

Table 6 Comparison of running time

Algorithm	Packet Length (Bytes)			Platform
	16	32	512	
Proposed H-KSG	0.800 ms	1.50 ms	25.60 ms	Arduino DUE
Skipjack [34]	–	6.62 ms	103.45 ms	Sun Small Pro-grammable Object Technology (SunSPOT)
RC5 [34]	–	14.96 ms	235.39 ms	
MBCC [34]	–	34.02 ms	533.29 ms	
BCC [34]	–	220.74 ms	3515.43 ms	
AES-128 [35]	5.856 ms	–	–	Arduino UNO
DES [35]	39.040 ms	–	–	

It should also be noted that at each power-up of all sensors nodes, an initialization procedure of the H-KSG generator is done asgivenby the Algorithm 3. This ensures that a start of an encryption process will not be the same for each node, since the values of the **Init_Vect'i** = $[x_n, y_n, z_n, w_n]$ (initial conditions) will always be different for each node's H-KSG generator.

Algorithm 3 :The H-KSG generator initialization

Inputs: $X_0, Y_0, Z_0, W_0, a, b, c$ and d .// A secret Key
1: **randomSeed**(analogRead(A0)).// Arduino's function to initialize a random number generator with a fairly random input, such as analogRead () on an unconnected pin A0
2: **N**=random(1,1000).// Arduino's function to generate pseudo-random numbers
2: Iterate H-KSG generator **N** times
Output : **IV'**= [x_n, y_n, z_n, w_n]. // IV' initial conditions

4.3 Analysis and performances of the implementation

4.3.1 Confidentiality

In the proposed scheme communication between the nodes and the server is encrypted using a synchronous stream cipher mode with a large space key ($H = 2^{640}$ bits).Then the probability of an eavesdropper cracking the key by brute force is $1/2^{640}$. Furthermore the stream cipher keys are only used for one time encryption as a result even if the attacker successfully cracks the secret key for one transmission, the attacker cannot crack all the other keys with in non-negligible probability, therefore the proposed cryptosystem provides a high confidentiality of data.

4.3.2 Authentication and non-repudiation

The verification of the valid nodes in smart home environments allows the system to test whether the information was sent from the correct source node. Each node in the network during the installation period has given a secret and unique ID in the proposed scheme. Also, during the configuration, the server records the ID of all network nodes in the database.The attackers may be able to add a malicious node to intercept communication in a smart home, but the ID of a malicious node cannot be compared to the list of permitted and valid nodes and can be easily identified by the controller. In fact, the valid ID makes it possible to ensure that the source nodes do not deny validity when transmitting their messages.

4.3.3 Integrity

Because an attacker can always change information to alter its original meaning by inserting fragments of fake information in the message transmitted. Integrity is a major factor in smart home security and plays a major role.Each node computes the Message Authentication Code (MAC) by computing the CRC and sends the Message together with a message, to achieve integrity in the proposed system.The server calculates the message CRC and compares it with the received CRC when it receives the packets. A malicious node trying to change a message can be easily found by the process of shuffling array, since the same CRC cannot be found and calculated.

4.3.4 Data freshness

Data freshness ensures that the database is fresh and no intruder will recreate the old information. To do this, the server for each reception will compare **Init_Vect'** in the received message and the last saved **Init_Vect'** in the proposed scheme. If there are equals, the server finds the message to be old and non-valid, and the message is denied.

4.3.5 Resilience

WSN resilience is an issue of safety which means that it is very important to ensure protection over the rest of the network if a node is compromised by an attacker. In the proposed solution, only the key for the corresponding node is exposed when a node is compromised.

4.3.6 Time Complexity

Table 6 presents various comparisons between our algorithm with other related encryption algorithms in terms of execution time. From the stated information in Table 6, we can deduce that the encryption speed of proposed cryptosystem is better than those listed.

5 Conclusion and future scope

In this paper, we proposed a secure and efficient cryptosystem based in Hyperchaos to secure data within the context of smart home automation applications, by using a H-KSG.The proposed approach is based on hyperchaotic Chen system. The fundamental idea is to exploit Chen's four hyperchaotic signals to generate encryption keys with a good statistical performances. The randomness of the encryption keys has been extensively tested using NIST testing strategy. According to the completed NIST test, the proposed Hyperchaos based H-KSG provides highly random bit sequences which serve as keystream for encryption and decryption process. In addition, a statistical and differential cryptanalysis examined various aspects of our cryptosystem such as randomness, histogram, entropy, key sensitivity and key space.In this study, it was verified that the developed WSN platform meets all important safety requirements including network security (access control) and data security (confidentiality, authentication and non-repudiation, integrity, resilience and freshness). It can be seen that our proposed cryptosystem has excellent running speed compared to other encryption algorithms. The hyperchaos-based cryptosystem proposed in this work represents an initial foray into leveraging complex dynamics for smart home security. Considerable opportunities exist to build upon this foundation through evaluating alternative hyperchaotic systems, implementing the approach in physical smart home testbeds, developing optimized hardware implementations like FPGA platforms, comprehensively analyzing attack resistance, exploring the use of the stream cipher scheme to secure real-time video and audio data, applying the encryption protocol to other domains like vehicular communications, integrating complementary security primitives, and assessing impacts on quality of service. By thoroughly investigating these promising research avenues, we can further refine hyperchaos cryptography to provide maximally secure, performant, and holistic data protection tailored to the demands of emerging smart home technologies and applications.

Acknowledgements This work is supported by the Directorate General for Scientific Research and Technological Development (DGRSDT) of the Algerian Ministry of Higher Education and Scientific Research (MESRS).

Data availability Not applicable.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Sudha KS, Jeyanthi N (2022) Similarity Learning-Based Supervised Discrete Hash Signature Scheme for authentication of smart home scenario. Int J Inf Technol 14:3377–3389
2. Poh GS, Gope P, Ning J (2019) PrivHome: privacy-preserving authenticated communication in smart home environment. IEEE Trans Depend Secure Comput 18:1095–1107
3. Stolojescu-Crisan C, Crisan C, Butunoi B-P (2021) An IoT-based smart home automation system. Sensors 21:3784
4. Jenal M, Omar AN, Hisham MAA, Noh WNWM, Razali ZAI (2022) Smart home controlling system. J Electron Volt Appl 3:92–104
5. Ghayvat H, Mukhopadhyay SC, Liu J, Babu A, Elahi E, Gui X (2015) Internet of Things for smart homes and buildings: opportunities and challenges. Australian J Telecommun Digit Econ 3:33. https://doi.org/10.18080/ajtde.v3n4.23
6. Kolanur CB, Banakar RM, Rajneesh G (2021) Design of IoT based platform development for smart home appliances control. J Phys Conf Ser 1969:12052
7. Yasmin N, Gupta R (2023) Modified lightweight cryptography scheme and its applications in IoT environment. Int J Inf Technol 15:4403–4414
8. Abi Sen AA, Eassa FA, Jambi K, Yamin M (2018) Preserving privacy in internet of things: a survey. Int J Inf Technol 10:189–200
9. Santoso FK, Vun NCH (2015) Securing IoT for smart home system. In: 2015 International Symposium on consumer electronics (ISCE). pp 1–2. IEEE
10. Wheelus C, Zhu X (2020) IoT network security: Threats, risks, and a data-driven defense framework. IoT 1:259–285

11. Abu-Tair M, Djahel S, Perry P, Scotney B, Zia U, Carracedo JM, Sajjad A (2020) Towards secure and privacy-preserving IoT enabled smart home: architecture and experimental study. Sensors 20:6131
12. Liu Y, Zhang J, Zhan J (2021) Privacy protection for fog computing and the internet of things data based on blockchain. Cluster Comput 24:1331–1345
13. Gulganwa P, Jain S (2022) EES-WCA: energy efficient and secure weighted clustering for WSN using machine learning approach. Int J Inf Technol 14:135–144
14. Mrabet H, Belguith S, Alhomoud A, Jemai A (2020) A survey of IoT security based on a layered architecture of sensing and data analysis. Sensors 20:3625
15. Ghani A, Mansoor K, Mehmood S, Chaudhry SA, Rahman AU, Najmus Saqib M (2019) Security and key management in IoT-based wireless sensor networks: An authentication protocol using symmetric key. Int J Commun Syst 32:e4139. https://doi.org/10.1002/dac.4139
16. Mamvong JN, Goteng GL, Zhou B, Gao Y (2020) Efficient security algorithm for power-constrained IoT devices. IEEE Internet Things J 8:5498–5509
17. Rahman Z, Yi X, Billah M, Sumi M, Anwar A (2022) Enhancing AES using chaos and logistic map-based key generation technique for securing IoT-based smart home. Electronics (Basel). 11:1083
18. Sadoudi S, Benzegane S, Djeddou M (2017) 10th Chaotic modeling and simulation International Conference (CHAOS2017). In: Hyperchaos-Based Stream Cypher for Smarphone, Barcelona, Spain
19. Umar T, Nadeem M, Anwer F (2023) Modified chaotic quadratic map with improved robust region. Int J Inform Technol 16:131–136
20. Vidal G, Baptista MS, Mancini H (2014) A fast and light stream cipher for smartphones. Eur Phys J Spec Top 223:1601–1610. https://doi.org/10.1140/epjst/e2014-02185-y
21. Kocarev L (2001) Chaos-based cryptography: a brief overview. IEEE Circuits Syst Mag 1:6–21. https://doi.org/10.1109/7384.963463
22. Maurya R, Rao GVE, Rajitha B (2022) Visual cryptography for securing medical images using a combination of hyperchaotic-based pixel, bit scrambling, and DNA encoding. Int J Inf Technol 14:3227–3234
23. Meshram C, Lee C-C, Bahkali I, Imoize AL (2023) An efficient fractional chebyshev chaotic map-based three-factor session initiation protocol for the human-centered IoT architecture. Mathematics 11:2085
24. Abdullah HA, Abdullah HN, Mahmoud Al-Jawher WA (2020) A hybrid chaotic map for communication security applications. Int J Commun Syst 33:e4236. https://doi.org/10.1002/dac.4236
25. Hsueh J-C, Chen VH-C (2019) An ultra-low voltage chaos-based true random number generator for IoT applications. Microelectronics J 87:55–64. https://doi.org/10.1016/J.MEJO.2019.03.013
26. Dhall S, Pal SK, Sharma K (2018) Cryptanalysis of image encryption scheme based on a new 1D chaotic system. Signal Process 146:22–32. https://doi.org/10.1016/j.sigpro.2017.12.021
27. Alvarez G, Li S (2006) Some basic cryptographic requirements for chaos-based cryptosystems. Int J Bifur Chaos 16:2129–2151
28. Hadj Youssef W, Abdelli A, Kharroubi F, Dridi F, Khriji L, Ahshan R, Machhout M, Nengroo SH, Lee S (2023) A secure chaos-based lightweight cryptosystem for the Internet of Things. IEEE Access 11:123279–123294
29. Zhou W, Wang X, Wang M, Li D (2022) A new combination chaotic system and its application in a new Bit-level image encryption scheme. Opt Lasers Eng 149:106782
30. Zhu L, Jiang D, Ni J, Wang X, Rong X, Ahmad M, Chen Y (2022) A stable meaningful image encryption scheme using the newly-designed 2D discrete fractional-order chaotic map and Bayesian compressive sensing. Signal Process 195:108489
31. Wang X, Zhang M (2021) An image encryption algorithm based on new chaos and diffusion values of a truth table. Inf Sci (N Y) 579:128–149
32. Zhu S, Deng X, Zhang W, Zhu C (2023) Image encryption scheme based on newly designed chaotic map and parallel DNA coding. Mathematics 11:231
33. Moya-Albor E, Romero-Arellano A, Brieva J, Gomez-Coronel SL (2023) Color image encryption algorithm based on a chaotic model using the modular discrete derivative and Langton's ant. Mathematics 11:2396
34. Sharafi M, Fotouhi-Ghazvini F, Shirali M, Ghassemian M (2019) A low power cryptography solution based on chaos theory in wireless sensor nodes. IEEE Access 7:8737–8753
35. Sukiatmodjo A, Setianto YD (2019) Speed and Power Consumption Comparison between DES and AES Algorithm in Arduino. Scientific Journal of Informatics 6:45–53. https://doi.org/10.15294/SJI.V6I1.17838
36. Rivest R (1992) The MD5 Message-Digest Algorithm. https://tools.ietf.org/html/rfc1321Accessed 11 May 2023
37. Pieprzyk J, Sadeghiyan B (1993) Design of hashing algorithms. Springer-Verlag
38. Bassham LE, Rukhin AL, Soto J, Nechvatal JR, Smid ME, Leigh SD, Levenson M, Vangel M, Heckert NA, Banks DL (2010) A Statistical Test Suite for random and pseudorandom number generators for cryptographic applications | NIST. Special Publication (NIST SP) - 800–22 Rev 1a
39. Liu X, Yan G (2019) Plaintext-related image encryption scheme using hyperchaotic system and DNA computing. J Electron Imaging 28:43027
40. Belhorma B, Bouhedda M, Bengherbia B, Benzineb O (2021) DNA-Based Cryptosystem for Connected Objects and IoT Security. In: Advanced smart computing technologies in cybersecurity and forensics., Kaushik K, Tayal,S, Bhardwaj A, Kumar M, pp. 23–46. CRC Press
41. Nigel PS, Rijmen V, Warinschi B, Watson G (2014) Algorithms, Key Sizes and Parameters Report. https://www.enisa.europa.eu/publications/algorithms-key-size-and-parameters-report-2014Accessed 20 Nov 2023
42. Wang J, Liu L (2022) A novel chaos-based image encryption using magic square scrambling and octree diffusing. Mathematics 10:457
43. Trujillo L, Álvarez-Hernández G, Maldonado Y, Vera C (2020) Comparative analysis of relocation strategies for ambulances in the city of Tijuana. Mexico Comput Biol Med 116:103567. https://doi.org/10.1016/J.COMPBIOMED.2019.103567
44. Biham E, Shamir A (1991) Differential cryptanalysis of DES-like cryptosystems. J Cryptol 4:3–72. https://doi.org/10.1007/BF00630563
45. Mosenia A, Jha NK (2017) A comprehensive study of security of Internet-of-Things. IEEE Trans Emerg Top Comput 5:586–602. https://doi.org/10.1109/TETC.2016.2606384
46. Jing Q, Vasilakos AV, Wan J, Lu J, Qiu D (2014) Security of the Internet of Things: perspectives and challenges. Wireless Netw 20:2481–2501. https://doi.org/10.1007/s11276-014-0761-7

47. El-Razouk H, Reyhani-Masoleh A, Gong G (2014) New implementations of the WG stream cipher. IEEE Trans Very Large Scale Integr (VLSI) Syst 22:1865–1878. https://doi.org/10.1109/TVLSI.2013.2280092
48. Almasalha F, Hasimoto-Beltran R, Khokhar A, Almasalha F, Hasimoto-Beltran R, Khokhar AA (2014) Partial encryption of entropy-coded video compression using coupled chaotic maps. Entropy 16:5575–5600. https://doi.org/10.3390/e16105575

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.