Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 6, pp. 5638–5648, DOI 10.12694/scpe.v25i6.3394

AN EFFICIENT CRYPTOGRAPHIC SCHEME BASED ON OPTIMIZED WATERMARKING SCHEME FOR SECURING INTERNET OF THINGS

ABHINAV VIDWANS* AND MANOJ RAMIYA†

Abstract. In this work, a new efficient cryptographic scheme based on the concept of chaotic map and optimized watermarking scheme is proposed. In the optimized watermarking scheme, a combination of discrete wave transformation (DWT), hessenberg decomposition (HD), and singular value decomposition (SVD) are used. In this, the host image is first broken down into several sub-bands using multi-level DWT, and the resulting coefficients are then fed into HD during the embedding phase. Simultaneous watermark operation is performed on SVD. Finally, the scale factor embeds the watermark into the host image. The Differential evolution method is used to find the best scaling factor for the optimized watermarking scheme. The resulting watermarked image is then encrypted by the session key based scheme. In this scheme for each image encryption, a new random session key will be produced. The presented approach uses 64-bit plaintext and a variable size key that will be decided at the time of encryption for encrypting an image. Since session keys change with each transmission, this approach does not involve extracting and remembering session keys in order to produce subsequent session keys. IoT devices are used to test the developed method for security. The experiment's findings shows that the suggested method works better than the current scheme in several aspects.

Key words: Encryption, Session Key, Decryption, Block cipher, Hybrid pseudo random number generator, DWT, HD, SVD

1. Introduction. Now a days, Digital images are used extensively in a variety of industries, such as finance, personal communication, and healthcar, thus it is becoming more and more important to ensure their safe transmission and storage. This is particularly true in terms of preserving them from tampering, interception, and unauthorized access [1, 2]. As a result, maintaining the security of digital images has become essential. Steganography and watermarking are two essential techniques for enabling secret communication. In this study, the simplest form of watermarking-based security is used.

Digital watermarking schemes can be categorized into a variety of techniques. Based on their domain, the techniques of image watermarking fall into one of two categories: transformed domains or spatial domains. Because spatial domain-based watermarking techniques are resistant to a wide range of attacks, they are not widely used. Rather, modified domain approaches are usually used in robust watermarking to ensure the image's resilience against several types of attacks. SVD, DFT, DCT, and DWT are a few transformed domain techniques via examples. This work employs the hybrid SVD and DWT techniques [3–8] in conjunction with Hessenberg decomposition. Of these strategies, the best assault resistance is reported in [9]. These techniques provide better security for securing digital images but don't provide an extra layer of security for securing confidential images. Because as the technology grows, the user needs extra security for secure transmission of images from sender to receiver.

Traditional symmetric-key image encryption approaches are frequently more expensive due to their mathematical complexity and the increased number of rounds required for encryption. Hence the author in [10-13] has discussed session key image encryption schemes having less number of encryption and decryption rounds. The work addresses several image encryption techniques that are based on hyperchaotic and genetic algorithms, such as the logistic map [14], the hyperchaos and Chinese remainder theorem [15–19], the hash key using the crossover operator, and the chaos [20], and chaos using the mutation and crossover operator [21]. These techniques also do not provide additional security in terms of adding one extra layer of security.

*Department of Computer Science and Engineering, Institute of Advanced Computing, Sage University, Indore, MP-India (vidwans.abhinav@gmail.com)
†Department of Computer Science and Engineering, Institute of Advanced Computing, Sage University, Indore, MP-India (manojramiya@gmail.com)

To reduce the complexity and increase one more level of security, the following improvements have been made to the existing scheme defined in [22]:

(a) One more level of security in the form of an optimized watermarking scheme is used in this work, where a differential evolution scheme is used to find the optimal scaling factor used in the watermarking scheme.
(b) In the existing scheme, only a 96-bit key was used to protect the images from intruders. In this work, a variable key size will be used that will be decided at the time of encryption.
(c) In the existing scheme, there was no comparison of the developed methodology on the IoT-based hardware. In this work, a comparative study has been done on the different IoT-based hardware.

This is the configuration of the complete work: The various terminologies used for the implementation of the work are explained in Section 2. The proposed methodology is given in section 3. Separate descriptions of the experimental results are given in Section 4. Before the entire project is wrapped up in section 6, IoT applications are covered in section 5.

2. Preliminaries. The different terminology utilized in this work for implementation will be explained in this part.

2.1. Hybrid Pseudo Random Generator. For any random generating algorithm, a seed value must be entered when the program is first started. Every trial will produce the same random number if arithmetic operations are applied to generate random numbers and the seed value remains constant. Thus, a millisecond system timestamp is employed as a seed value to solve this problem. Since a millisecond can alter drastically in a very short amount of time, this approach will always provide a different random number. The multiplicative congruence approach is applied here as well for arithmetic operations. Each trial results in an update of the input value for the subsequent iteration. Utilizing a random number generator, distinct crossover points could be generated, as the crossover function relies on two random crossover points. When it comes to producing distinct random numbers, it performs better than other similar pseudo-random number generators.

The hybrid pseudo-random generator approach is demonstrated in the subsequent steps [26]:

Step 1: Initialize the number of random integers (m) and the upper limit (p) with their initial values.
Step 2: Capture the timestamp and extract the last four digits from the microsecond section, assigning this value to 't1' as the seed.
Step 3: Set 'x' to 1.
Step 4: For each iteration from 1 to m:
  a. Calculate y using the formula x = $(m^2 + a * x)$ mod (p + 1), where 'a' is derived from t1 (1, 6).
  b. Record the floor value of 'x' in the array R at position (1, i).
  c. Modified the value of 't1'
  End the loop.
Step 5: Provide the resulting array R as the output.

2.2. Two-point crossover. This type of crossover uses two crossover points to further distort the resultant strings. The crossover point values are selected at random for each of the two crossover points to prevent infiltration.

For example: Let M = 0 1 1 0 1 1 0 and N = 1 0 0 0 1 1 0 be two parent strings with crossover points generated at random intervals of 2 and 4. After two-point crossover, the newly formed parents are:

M = 0 1 0 0 1 1 0 1; N = 1 0 1 0 1 1 1 0.

Here m1 = 0 1; m2 = 0 0; m3 = 1 1 0 1; n1 = 1 0; n2 = 1 0; n3 = 1 1 1 0

2.3. Differential Evolution Scheme. One well-liked evolutionary technique for global optimization issues is differential evolution (DE) [33]. It works especially well for multi-modal and continuous optimization. The Differential Evolution algorithm's fundamental steps are as follows:

Step 1: Define the problem definition, population size, search space dimensions, and initialize a population of potential solutions.
Step 2: Make a mutant vector for every member of the population by fusing three randomly chosen members and adjusting the mutation by a scaling factor.
Step 3: Create a trial vector by performing a binary crossover operation between the original individual and the mutant vector.

Step 4: Compare and examine the trial vector's fitness with the original individual. Replace the original individual with the trial vector if it proves to be superior.
Step 5: Select a termination condition, such as convergence requirements or reached the number of generations.
The best scaling factor value is found in step 5 of the optimized watermarking technique, as illustrated in section 3.1, using the DE algorithm in the proposed work. The initial population for this DE algorithm comprises values ranging from 0 to 1.

3. Proposed Methodology. The proposed methodology in this work is based on two security phases: The first security phase is based on an optimized watermarking scheme while the second security phase is based on the encryption phase.

3.1. Optimized watermarking Scheme. The following steps are used to perform an optimized watermarking scheme in this work-
Step 1: The host image is decomposed into the components of LH, LL, HH, and HL based on R-level DWT [23] where $R = log_2(\mathrm{m/n})$
Step 2: Perform HD [25] on the LL component of the host image by using the following equation-

$$PHP^T = HD(LL) \tag{3.1}$$

Step 3: Apply SVD [24] to H by using the following equation-

$$HU_w HS_w HV_w^T = SVD(H) \tag{3.2}$$

Step 4: Applied SVD on the secret image W by using this equation –

$$U_w S_w V_w^T = SVD(W) \tag{3.3}$$

Step 5: Calculate an embedded singular value by using this equation-

$$HS_w^* = HS_w + \theta S_w \tag{3.4}$$

Here θ is the scaling factor chosen by using the differential evolution method.
Step 6: Convert watermarked sub-band H^* by the inverse SVD, using the following equation-

$$H^* = HU_w HS_w^* HV_w^T \tag{3.5}$$

Step 7: Reconstructing a new approximation sub-band LL_ at low frequencies is done using the inverse HD, which is provided by -

$$LL^* = PH^*P^T \tag{3.6}$$

Step 8: Finally the watermarked image is obtained by performing the inverse R-level DWT.
After step 8, the watermarked image will be generated. Now this watermarked image will be encrypted by the below encryption scheme (described in section 3.2) used in this work

3.2. Encryption Scheme. The encryption scheme used in this work is based on three different phases as described in the [22].

3.2.1. Key Generation phase. The following procedures are employed in this phase to produce the key using a hybrid pseudo-random number generator-
Step 1: Define 56-bit key and the following key =01234567890123 is used for experiments in this work.
Step 2: Split the 64-bit into 8 different blocks.
Step 3: Generate 8-bit, 24-bit, and 40-bit random blocks using a hybrid pseudo-random number generator, discussed in section 2.4.
Step 4: With the help of the XOR operator and one user-defined function based on two-point crossover, discussed in section 2.5 to generate 64-bit, 80-bit, and the 96-bit key.
The generated keys will be used randomly at the time of encryption.

3.2.2. Encryption Phase. After the key generation, the next phase is encryption in which 4 encryption rounds and two swaps are used if using the 64-bit key for encryption during run time. 5 encryption rounds and 3 swaps will be used if using the 80-bit key for encryption during run time. Otherwise, 6 encryption rounds and 4 swaps will be used. One random function, XNOR, and XOR are included in each round (f). To increase the security of the cipher text, this random function is based on the two-point crossover operator. The following steps will be used for encryption-

Step 1: Generate 4 blocks by dividing the 64-bit plain text.
If using 64-bit key during run time then perform following steps:
Step 2: In round 1, perform XOR and two-point crossover operations between the first 16 bits of the key part and generated blocks in step 1, to generate 4 new different blocks.
Step 3: Perform swapping between newly generated blocks in step 2.
Step 4: Reiterate step 2 and step 3 for 2^{nd}, 3^{rd} and 4^{th} 16-bit key parts in round 2, 3 and 4.
Else if using 80-bit key during run time then perform following steps:
Step 5: In round 1, perform XOR and two-point crossover operations between the first 16 bits of key part and generated blocks in step 1, to generate 4 new different blocks.
Step 6: Perform swapping between newly generated blocks in step 2.
Step 7: Reiterate step 5 and step 6 for 2^{nd}, 3^{rd}, 4^{th} and 5^{th} 16-bit key parts in round 2, 3, 4, and 5.
Otherwise (if using 96-bit key during run time then perform following steps):
Step 8: In round 1, perform XOR and two-point crossover operations between the first 16 bits of key part and generated blocks in step 1, to generate 4 new different blocks.
Step 9: Perform swapping between newly generated blocks in step 2.
Step 10: Reiterate step 8 and step 9 for 2^{nd}, 3^{rd}, 4^{th}, 5^{th} and 6^{th} 16-bit key parts in round 2, 3, 4, 5, and 6.

3.2.3. Decryption Phase. In this phase, perform reverse operations implemented in the encryption phase to decrypt the image.

The following general steps are used to perform the proposed methodology-

Step 1: Import host image and secret image
Step 2: Perform an optimized watermarking scheme to generate the watermarked image.
Step 3: Use a watermarked image as an input to the used encryption scheme.
Step 4: Generate encrypted watermarked image at the sender side and sent for communication at the receiver side.
Step 5: The Receiver will decrypt the watermarked image by using the inverse process of encryption.
Step 6: Perform the reverse procedure of an optimized watermarking scheme to generate the original secret image.
Step 7: Received original secret image at the receiver side.

The following flow chart shows the overall scheme used in this work in Fig 3.1.

4. Experimental Results. This section describes the experimental results on a number of assessment factors that were used to gauge how effective the recommended method was. MATLAB version 2017 and a computer with a core i3 Processor and 2GB of RAM are used to accomplish the suggested encryption method. For this experiment, a symmetric 14-digit hexadecimal key with the value "01234567890123" is used. Figure 4.1 displays the four commonly used images for testing purposes, each measuring 256 by 256 pixels in both RGB and grayscale. Figure 4.2 illustrates how encryption works.

4.1. Evaluation Parameters. The effectiveness of the suggested encryption scheme is evaluated using the following evaluation criteria –

4.1.1. Clipping Attack. To evaluate the resilience of the method proposed in this study, a clipping attack is necessary because data loss can occur during transmission. To verify the validity of the recommended process, the encryption image is cut off at a random rectangle position, and the appropriate key is then used to decrypt the data. As seen in Figure 4.3, the recommended method is adequately guarded against a clipping assault.

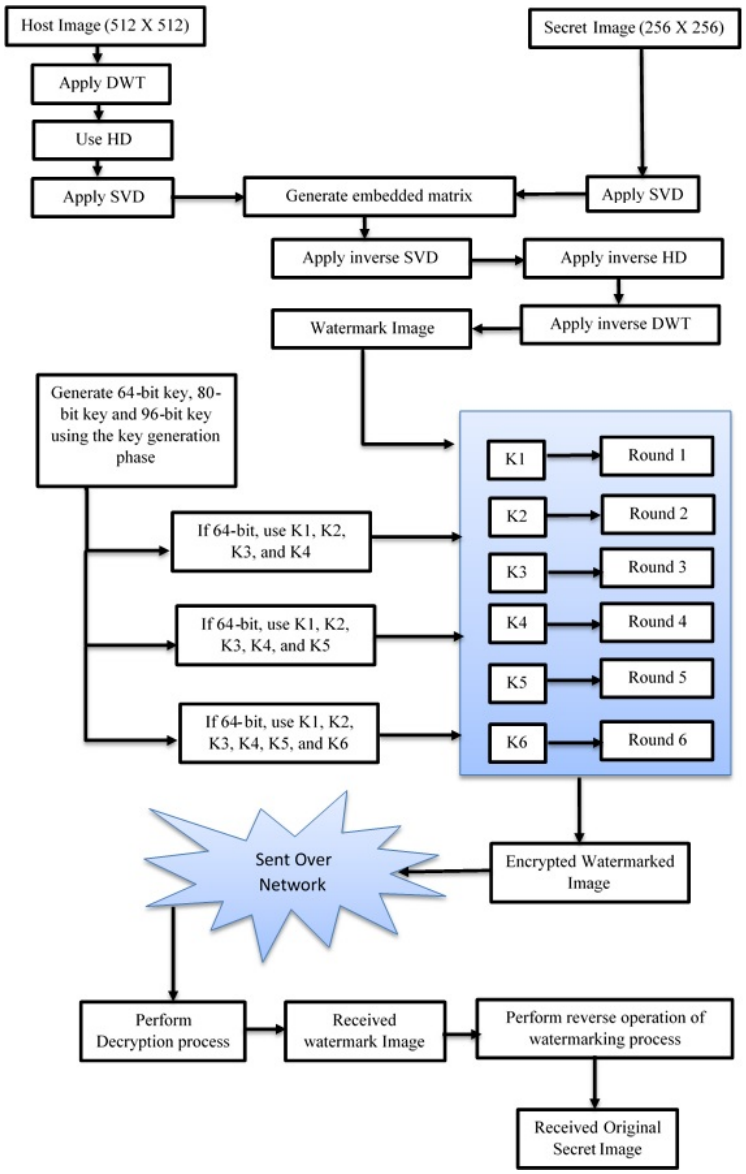

Fig. 3.1: Overall proposed encryption system

Fig. 4.1: Images used for experiments

Fig. 4.2: Proposed cryptographic system

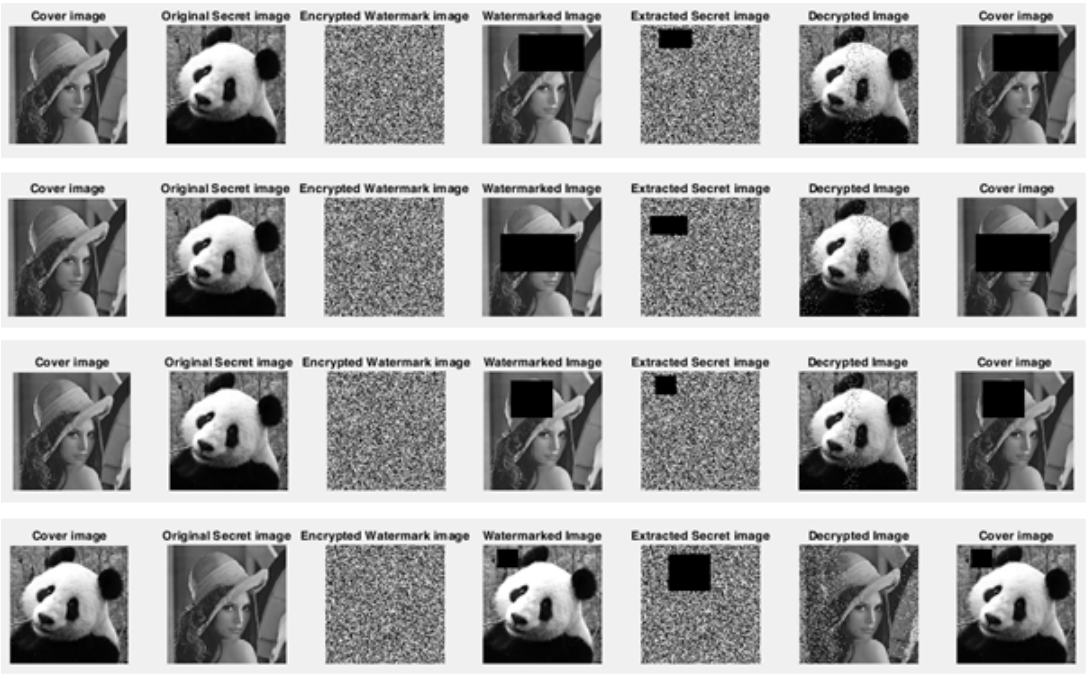

Fig. 4.3: Proposed encryption system after applying clipping attack

4.1.2. Salt & Pepper Noise Attack. Transmission noise typically deteriorates cipher images. With the right key and the input images, the cipher images can still be decrypted. The decrypted image quality appears to be significantly impacted by noise pixels, hence the technique for image encryption needs to be reasonably noise-resistant. After adding salt and pepper noise to the original image, this study employs the recommended technique to decode the original image. It is clear from Figure 4.4 that the recommended method is immune to the noise attack caused by salt and pepper.

4.1.3. NPCR and UACI. The percentage of unique pixels in the two images that differ from one another is determined by the NPCR. The higher the values of NPCR, the better an encryption scheme is. The following formulas are used for NPCR calculations-

$$NPCR = \frac{\sum_{j,i} D(j,i)}{HXW} * 100 \tag{4.1}$$

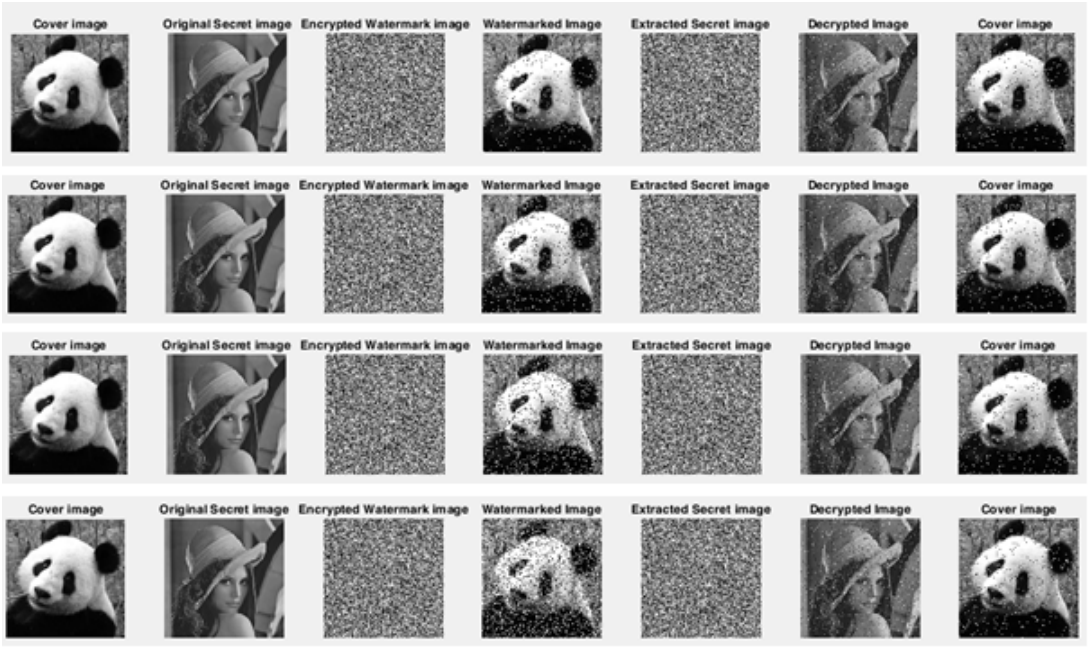

Fig. 4.4: Proposed encryption system after applying salt and pepper noise attack

Here, W and H represent the width and height of the image, respectively. D(j, i) can be defined as

$$D(j,i) = \begin{cases} 0, \text{if} C1(j,i) = C2(j,i) \\ 1, \text{otherwise} \end{cases}$$

4.1.4. Information Entropy. The Entropy of the message source can be calculated using the formula below:

$$E(n) = -\sum p(m) log_2 p(m) \tag{4.2}$$

when p(m) represents the probability of the source m. Entropy attacks cannot be carried out against the recommended algorithm. Table 1 compares the entropy values of the current work and the proposed work in percentage terms.

4.1.5. Histogram Analysis. It is crucial to compare the statistical likenesses of the encrypted and the original image that was utilized to prevent unauthorized access to user data. The histogram analysis quantifies these statistical parallels. The grayscale and color image histogram is displayed in Figure 4.5.

4.1.6. Correlation Analysis. In plain-image and cipher images, the correlation between two adjacent pixels that are split away vertically, horizontally, and diagonally can be calculated using the following formulas [10]:

$$COV(p,q) = E(p - E(M))(q - E(M)) \tag{4.3}$$

$$R_p q = \frac{COV(p,q)}{\sqrt{D(p)}\sqrt{D(q)}} \tag{4.4}$$

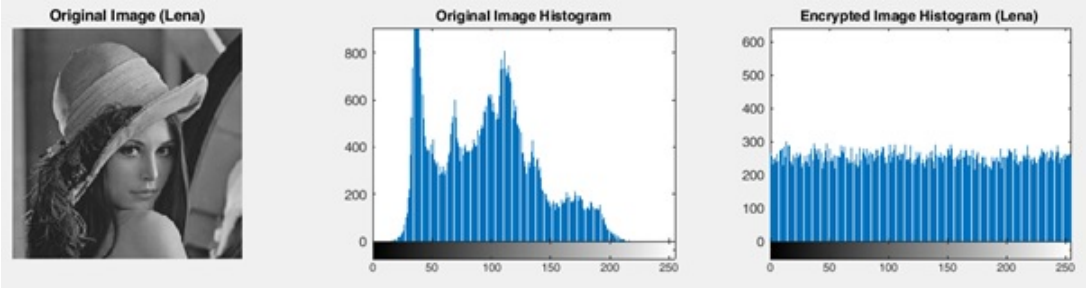

Fig. 4.5: Histogram analysis between encrypted and original lenna image

Table 4.1: Comparison of NPCR on Lena Image

Algorithms	NPCR in percentage
Hu, Y. et. al. [27]	99.6124
Nkandeu, Y. P. K. et. al. [28]	99.6300
Shah, T. et. al. [29]	99.6206
Proposed system	99.64

Table 4.2: Comparison of Information Entropy on Lena Image

Algorithms	Entropy
Gupta, M. et. al. [10]	7.9965(Avg.)
Gupta, M. et. al. [11]	7.9971(Avg.)
Proposed system	99.64

The next three mathematical computations use the values of two neighboring pixels in the image, p and n. The vertical, horizontal, and diagonal correlations of the original and encrypted LENNA image are shown in Figure 4.6.

$$E(p) = \frac{1}{t}\sum_{i=1}^{t} p_i \tag{4.5}$$

$$D(p) = \frac{1}{t}\sum_{i=1}^{t} (p_i - E(p))(q - E(q)) \tag{4.6}$$

$$COV(p,q) = \frac{1}{t}\sum_{i=1}^{t} (p_i - E(p))(q_i - E(q)) \tag{4.7}$$

4.1.7. Comparative Analysis between proposed scheme and existing schemes. This section compares the suggested system to current cryptographic techniques based on the NPCR and information entropy assessment parameters. Tables 4.1, 4.2, 4.3, and 4.4 demonstrate how the suggested system outperforms the current approaches.

5. Hardware Implementation. The results of comparing the hardware implementation of the suggested scheme with other existing methods are displayed in table 5.1. Since every algorithm is verified on hardware to ensure its stability and effective operation, a few IoT units were employed in this case for the proposed work's testing.

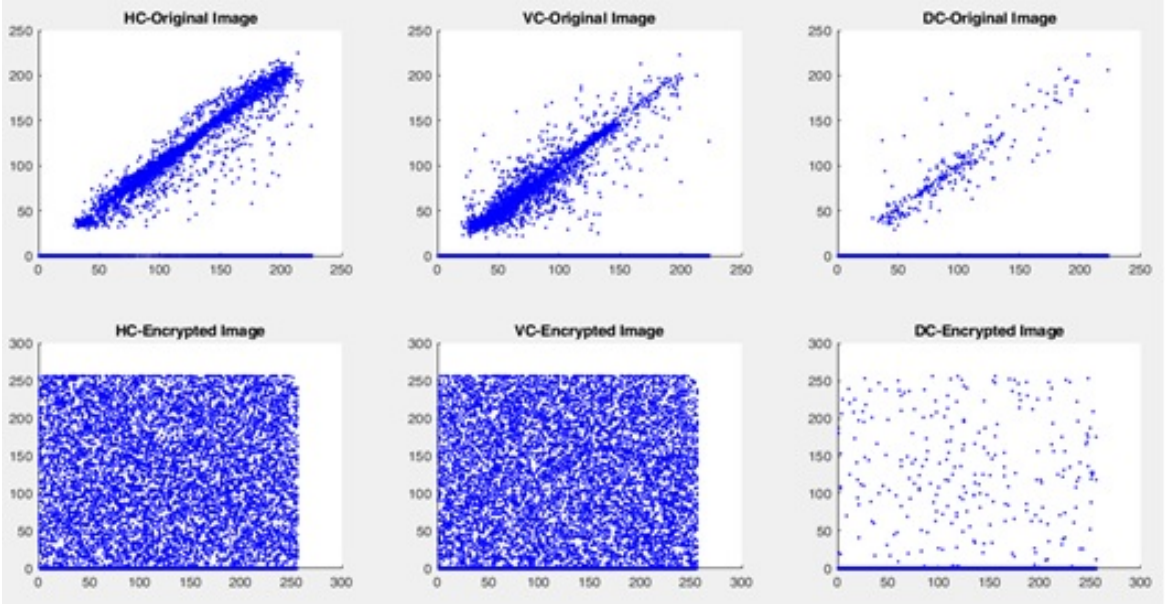

Fig. 4.6: All correlations between encrypted and original Lenna images

Table 4.3: Comparison of Information Entropy and NPCR on Panda Image

Algorithms	Entropy	NPCR (in %)
Gupta, M. et. al. [10]	7.9938(Avg.)	99.58(Avg.)
Gupta, M. et. al. [11]	7.9964(Avg.)	99.59(Avg.)
Proposed system	7.9964	99.61

Table 4.4: Comparison of Information Entropy and NPCR on Onion Image

Algorithms	Entropy	NPCR (in %)
Gupta, M. et. al. [10]	7.9964(Avg.)	99.61(Avg.)
Gupta, M. et. al. [11]	7.9963(Avg.)	99.60(Avg.)
Proposed system	7.9963	99.62

Table 5.1: Comparative analysis between the existing and presented scheme.

Algorithms	Device	Block Size	Key Size	Code Size
AES [30]	AVR	64	128	1570
IDEA [31]	AVR	64	80	596
SIT [32]	Atmega 328	64	128	826
Proposed System	Atmega 328	64,80, and 96	128	738

6. Conclusion and Future Scope. An efficient, portable, and safe encryption method is required because the majority of interactions conducted with IoT devices, such as smart gadgets, use images. This article presented a unique encryption strategy to secure Internet of Things terminals by utilising a hybrid pseudo-random number generator and an optimized watermarking scheme. This study uses a hybrid pseudo-random number generator for the encryption keys. The recommended algorithm is therefore safer. Before every transfer,

the conventional process calls for the exchange of a secret key. Here, key-wrapping techniques share session keys amongst themselves. The suggested study performs better than previous research on many evaluation metrics and on certain IoT hardware components for evaluating the robustness and effective operation of the suggested approach. The future work is to implement the Subsequent research on additional IoT terminals physically and evaluate the outcomes using additional metrics.

REFERENCES

[1] Hosny, K. M., Kamal, S. T., and Darwish, M. M., *A color image encryption technique using block scrambling and chaos*, Multimedia Tools and Applications, 2002, pp. 1–21.

[2] Alexan, W., Elkandoz, M., Mashaly, M., Azab, E., and Aboshousha, A., *Color image encryption through chaos and kaa map*, IEEE Access, 2002, 11, pp. 11541–11554.

[3] Araghi, T. K., and Megías, D., *Analysis and effectiveness of deeper levels of SVD on performance of hybrid DWT and SVD watermarking*, Multimedia Tools and Applications, 2023, pp. 1–22.

[4] Arora, S. M., and Kadian, P. , *Enhanced image security through hybrid approach: protect your copyright over digital images*, Wireless Communication Security, 2022, pp. 35–57.

[5] Khanam, T., Dhar, P. K., Kowsar, S., and Kim, J. M., *SVD-based image watermarking using the fast Walsh-Hadamard transform, key mapping, and coefficient ordering for ownership protection*, Symmetry, 2022, 12(1), pp. 52.

[6] Koohpayeh Araghi, T., Abd Manaf, A., Alarood, A., and Zainol, A. B., *Host feasibility investigation to improve robustness in hybrid DWT+ SVD based image watermarking schemes*, Advances in Multimedia, 2018.

[7] Mohammed, A. A., Salih, D. A., Saeed, A. M., and Kheder, M. Q., *An imperceptible semi-blind image watermarking scheme in DWT-SVD domain using a zigzag embedding technique*, Multimedia Tools and Applications, 2020,79(43-44), pp. 32095–32118.

[8] Wan, W., Wang, J., Zhang, Y., Li, J., Yu, H., and Sun, J., *A comprehensive survey on robust image watermarking*, Neurocomputing, 2022,488, pp. 226–247.

[9] Liu, J., Huang, J., Luo, Y., Cao, L., Yang, S., Wei, D., and Zhou, R., *An optimized image watermarking method based on HD and SVD in DWT domain*, IEEE Access, 2019,7, pp. 80849–80860.

[10] Gupta, M., Gupta, K. K., and Shukla, P. K., *Session key based fast, secure and lightweight image encryption algorithm*, Multimedia Tools and Applications, 2021,80(7), pp. 10391–10416.

[11] Gupta, M., Gupta, K. K., and Shukla, P. K., *Session key based novel lightweight image encryption algorithm using a hybrid of Chebyshev chaotic map and crossover*, Multimedia Tools and Applications, 2021,80(25), pp. 33843–33863.

[12] Gupta, M., Gupta, K. K., Khosravi, M. R., Shukla, P. K., Kautish, S., and Shankar, A., *An intelligent session key-based hybrid lightweight image encryption algorithm using logistic-tent map and crossover operator for internet of multimedia things*, Wireless Personal Communications, 2021,121(3), pp. 1857–1878.

[13] Gupta, M., Singh, V. P., Gupta, K. K., and Shukla, P. K., *An efficient image encryption technique based on two-level security for internet of things*, Multimedia Tools and Applications, 2023,82(4), pp. 5091–5111.

[14] Rostami MJ, Shahba A, Saryazdi S and Nezamabadi-pour H, *A novel parallel image encryption with chaotic windows based on logistic map*, Comput Electr Eng , 2017,62, pp. 384-–400.

[15] Liu, Y., Zhang, J.,Han, D., Wu, P., Sun, Y. and Moon, Y.S., *A multidimensional chaotic image encryption algorithm based on the region of interest*, Multimedia Tools and Applications, 2020.

[16] Li, R., *Fingerprint-related chaotic image encryption scheme based on blockchain framework*, Multimedia Tools and Applications, 2020.

[17] Dagadu, J.C., Li, J., Aboagye, E., *Medical Image Encryption Based on Hybrid Chaotic DNA Diffusion*, Wireless Personal Communications, 2019.

[18] Liu, H., Zhao, B., Huang, L., *A novel quantum image encryption algorithm based on crossover operation and mutation operation*, Multimedia Tools and Applications, 2019.

[19] Zhu H, Zhao C, Zhang X, *A novel image encryption-compression scheme using hyper-chaos and Chinese remainder theorem*, Signal Process, 2013,28(6), pp. 670–-680.

[20] Zhang, X., Zhou, H., Zhou, Z., Wang, L. and Li, C., *An Image Encryption Algorithm Based on Hyper-chaotic System and Genetic Algorithm*, Qiao J. et al. (eds) Bio-inspired Computing: Theories and Applications. BIC-TA. Communications in Computer and Information Science, 2018,952.

[21] Samhita, P., Prasad, P., Patro, K. and Acharya, B., *A Secure Chaos-based Image Encryption and Decryption Using Crossover and Mutation Operator*, IJCTA, 2016,9(34), pp. 17–28.

[22] Vidwans, A., and Ramaiya, M., *Session Key Based an Efficient Cryptographic Scheme of Images for Securing Internet of Things*, SN Computer Science, 2023,4(5), 527.

[23] Kricha, Z., Kricha, A., and Sakly, A., *A robust watermarking scheme based on the mean modulation of DWT coefficients*, Security and Communication Networks, 2018, pp. 1–-16.

[24] El Abbadi, N. K., Mohamad, A., and Abdul-Hameed, M., *Image Encryption based on singular value decomposition*, Journal of Computer Science, 2014, 10(7), 1222.

[25] Su, Q., and Chen, B., *A novel blind color image watermarking using upper Hessenberg matrix*, AEU-International Journal of Electronics and Communications, 2017, 78, pp. 64–-71.

[26] Ahmed, T; and Rahman, Md. M., *The Hybrid Pseudo Random Number Generator*, International Journal of Hybrid Information Technology, 2016, 9(7), pp. 299–-312.

[27] Hu, Y., Yu, S., and Zhang, Z., *On the cryptanalysis of a bit-level image chaotic encryption algorithm*, Mathematical Problems in Engineering, 2020, pp. 1–-15.
[28] Nkandeu, Y. P. K., Mboupda Pone, J. R., and Tiedeu, A., *Image encryption algorithm based on synchronized parallel diffusion and new combinations of 1D discrete maps*, Sensing and Imaging, 2020, 21, pp. 1–-36.
[29] Shah, T., Haq, T. U., and Farooq, G., *Improved SERPENT algorithm: design to RGB image encryption implementation*, IEEE Access, 2020, 8, pp. 52609–52621.
[30] Poettering B. Rijndaelfurious , *aes-128 implementation for avr devices, 2007*, 2013.
[31] Eisenbarth T., Gong Z., Guneysu T. and Heyse S., *Compact implementation and performance evaluation of block ciphers in attiny devices*, International Conference on Cryptology in Africa, 2012, pp. 172–187.
[32] Usman M., Ahmed I., Aslam M. I., Khan S., Shah U.A., *SIT: A Lightweight Encryption Algorithm for Secure Internet of Things*, (IJACSA) International Journal of Advanced Computer Science and Applications, 2017, 8(1).
[33] Storn, R., and Price, K., *Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces*, Journal of global optimization, 1997, 11, pp. 341–359.

Edited by: Manish Gupta
Special issue on: Recent Advancements in Machine Intelligence and Smart Systems
Received: Feb 22, 2024
Accepted: May 21, 2024