Journal of Applied Engineering and Technological Science
Vol 5(1) 2023: 268-278

A MODIFIED KLEIN ENCRYPTION-BASED KNIGHT TOUR FOR IMAGE ENCRYPTION

Emaan Oudha Oraby[1*], Rafeef Mohammed Hamza[2]
General Directorate of Education in Al-Qadissiyah Government, Ministry of Education, Iraq[1]
Department of Information Systems, College of Computer Science and Information Technology, Al-Qadisiyah University, Iraq[2]
eman.iraq2017@gmail.com[1], Rafeef.hamza@qu.edu.iq[2]

Received: 19 September 2023, Revised: 31 October 2023, Accepted: 04 November 2023
**Corresponding Author*

ABSTRACT
The security considerations should be balanced with the specific use case and potential risks associated with using lightweight encryption. The security offered by lighter encryption algorithms could not be as high as that offered by heavier encryption techniques. In this paper, a Modified KLEIN Algorithm is proposed for image encryption based on the Knight Tour movement in Chessboard. The required key generation is represented by inputting an image as a key image and then applying a specific operation based on knight tour movement to produce a key scheduled in the proposed encryption algorithm. The movement of Knight Tour applied in modifying the proposed algorithm for increasing security. The experimental results explain the efficiency of a modified algorithm when comparing the histogram of the input image with the encrypted image also the correlation is tested before and after encryption in three horizontal, vertical, and diagonal which explains there are very low values of them in all directions. The similarity test also explains the high differences between the plain and encrypted images. The chessboard movement might be useful when used with another encryption algorithm which increases the confidentiality of transferring data. The contribution of this work is the use of an image as a key for encryption with a specific planning method which helps in key management.
Keywords: *KLEIN Encryption, Knight Tour, Image Encryption Lightweight Encryption.*

1. Introduction

A lightweight encryption algorithm is a cryptographic algorithm designed to provide security while minimizing computational and resource requirements (Singh et al., 2017) (Raj et al., 2021). These algorithms are often used in situations with limited resources, including embedded systems, Internet of Things (IoT), and low-power devices, where memory, computing capacity, and energy usage are restricted. There are many methods developed to be used in IoT systems as a lightweight method such as SIMON, SPECK, PRESENT, Chaskey, Grain, HIGHT, RECTANGLE …etc. (Kushwaha et al., 2014; Mahmood et al., 2016). these methods should balance between security and complexity. the using of encryption methods in applications depends on specific requirements for the goal of implementations (Gong et al., 2011; L. Liu et al., 2018). One of the lightweight block ciphers is KLEIN uses Rijndael's rounds method. It is implemented on low-end hardware specifications (Chatterjee & Chakraborty, 2020). To find its efficiency a differential fault analysis (DFA) for the S-box to determine the input key for encryption (Hoomod et al., 2020; Younus & Younus, 2019). The KLEIN method in several IoT devices, therefore the weak point should be found to process them (Hoomod et al., 2020; S. Singh et al., 2017b). many papers have studied the security and resistance of the KLEIN. In the proposed method the knight tour issue is applied to a key image to generate a secret key to increase the secrecy of KLEIN encryption (Rahman et al., 2018). The method used to perform individual encryption such as key-dependent operations such as substitution and permutation improves security (Boussif et al., 2020).

- **KLEIN Encryption Algorithm**

The KLEIN Encryption Algorithm is a block cipher method (64-bit) with key sizes of 64, 80, or 96 bits and iterations 12, 16, or 20 (Gong et al., 2011). the Klein block cipher has configurable key sizes of 64, 80, and 96 bits with 12, 16, and 20 iterations, respectively (Gong et al., 2011). Each cycle of the Klein block cypher uses four algorithms. SubNibble, MixNibble, RotateNibble, and AddRoundKey. To be more exact, AddRoundKey is used to first or the state before it enters a round. The lead is then split into 16 4-bit nibbles, which are all converted by the identical involute 4 4 Sbox shown in Table 1. To save the costs of implementation and memory requirements, the KLEIN designers chose this Sbox as opposed to a byte-wise one (Lallemand & Naya-Plasencia, n.d.) [14].

Table 1 - KLEIN Sbox.

x	0	1	2	3	4	5	6	7	8	9	a	B	C	d	E	F
S[x]	7	4	A	9	1	F	B	0	C	3	2	6	8	e	D	5

The state is then rotated two bytes to the leftRotateNibles, and each half of the state is then transformed by MixNibbles using Rijndael MixColumn. Let's not forget the fact that the MixColumn phase uses 4 bytes interpreted as components of $GF(2^8) = GF(2)/X^8 + X^4 + X^3 + X + 1$. The output, which is made up of 4 bytes, is produced by multiplying the following:

$$\text{matrix:}\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

It may be helpful to note that multiplying X by 02 is similar to realizing a l of X. This is true in two situations: $(x << 1)$ if the “most significant bit” (MSB) of, and $(X << 1) \oplus 1b$ otherwise. Be aware that the previous steps may be thought of as nibble-wise, whereas the last step is byte-wise. In contrast to Rijndael, the final round does not omit the step MixNibbles. The encryption algorithm needs one more key than the number of rounds due to the addition of a final whitening key after the procedure.

The round keys are computed from the master key with the key schedule algorithm. In the following, Note that the $is\ k^r$ the round -key round r. Key scheduling takes place as follows: i^{th} sub key is divided in two parts x, y $(k_i = x.y)$ then cycling left shift one bit in each part of key x, y $(k_i = x \ll .y<<)$ and swap (x, y) such as (x, y) =(y, x circled $\oplus\ x, y$) (Kushwaha et al., 2014).

- **Knight Tour in a Chessboard**

Finding the right series of movements that a knight to make such that they visit each square on the chessboard precisely once is known as "The Knight's Tour," a classic chess conundrum. The knight makes an L-shaped movement, travelling a pair of squares in one direction (either horizontally or vertically), followed by one square in the other direction (Mahmood et al., 2016). Finding a Knight's Tour on an 8x8 chessboard is a well-studied problem with known solutions. One standard algorithm to solve this problem is Wendorff’s Rule, which suggests always choosing the next move to the square with the fewest remaining valid moves. The starting point is very important to fix the tour visit blocks (in the 2D matrix) and the larger chessboards to be more complicated Knight's Tour (Mahmood et al., 2016a). the knight tour is explained in Figure 1.

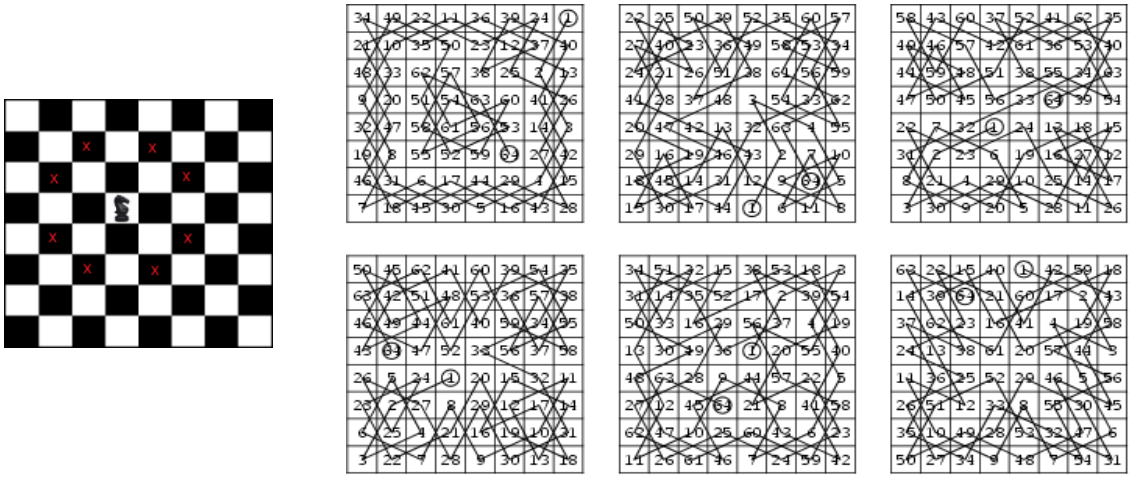

Fig 1. Chess Game Different Knight Moves (Liping & Xi, 2020)

2. Literature Review

Several methods of lightweight encryption are used in IoT devices to provide a level of security related to the low specification of these devices. One of these methods is KLEIN lightweight encryption. There are several methods related to the proposed method are illustrated as follows:

A technique to produce numbers based mostly on the motions of the knight on a chessboard connected to digital photographs was suggested by Hasoon, J. N., et al. (Hasoon et al., 2023). Photos from many sources, including websites, social media, and mobile photos, were used to create the keys. Blocks of the chosen picture are created, and each block has a predetermined principal route that prohibits spots from being revisited. To boost unpredictability, more unused chessboard locations are added. Each movement is preceded by an exclusive OR procedure across a block and its transposition. The technique effectively produces pseudo-random numbers with strong security attributes, enhanced linear complexity, and favorable statistical traits. The key quality is assessed via NIST tests, with successful results. B. S. Shashikiran, et al.(Shashikiran et al., 2021) When it comes to picture encryption, the focus is on using the short knight's tour technique and steganography to hide the secret image within another image. This strategy makes use of an open tour knight's moves on a 5x5 block, which is a minimum knight's tour. Notably, knight's trips smaller than 5x5 blocks do not visit every place in the block. Additionally, there is no chance of a closed knight's tour for odd-sized blocks.

Khan, M. F., et. al., this research introduced a fresh approach to crafting secure S-boxes, resilient against algebraic and chaos-based cryptanalysis, thereby enhancing cryptographic primitives' security. Achieving true randomness is crucial for cipher design due to its unpredictability, irreversibility, and irreproducibility. Generating true random numbers and integrating them into S-box construction is a central challenge. The proposed technique extracts genuine random bits from underwater acoustic waves, dynamically generating S-boxes through knight's tours.

Singh, M.et al. (M. Singh et al., 2015) proposed a visual cryptography method based on Knight's Tour Problem that elevates image encryption. The key size is not fixed and the number of shared images is variable (maybe one image only).

Bisht, K., and Deshmukh, M. (Bisht & Deshmukh, 2020) proposed a method for image encryption based on a creation matrix based on Knight Tour with a specific size. These matrices are split into sub-matrices based on input size. An additional encryption model is used for neighbors to find the total encrypted image.

3. Research Methods

A method for image encryption is proposed using a modified lightweight encryption algorithm called the Modified-KLEIN encryption algorithm. The key generation is represented by using a key image for key generation using Knight tour in chess board as a secret key of

appropriate length for the KLEIN encryption algorithm. The key should be sufficiently random and kept confidential. Image Preparation that converts the image into a suitable format for encryption. This typically involves converting the image into a pixel array or matrix representation and dividing the matrix into fixed-size blocks. Each block will be independently encrypted using the KLEIN algorithm. The block size depends on the specific requirements and constraints of the encryption scheme. Then apply the Modified KLEIN encryption algorithm to each block of the image. The KLEIN algorithm typically involves a series of rounds with key-dependent operations, such as substitution, permutation, and bitwise operations. The exact details of the algorithm would be beyond the scope of this response. The final step is image reconstruction by combining the encrypted blocks' original positions. The total steps of the proposed method are explained in Figure 2.

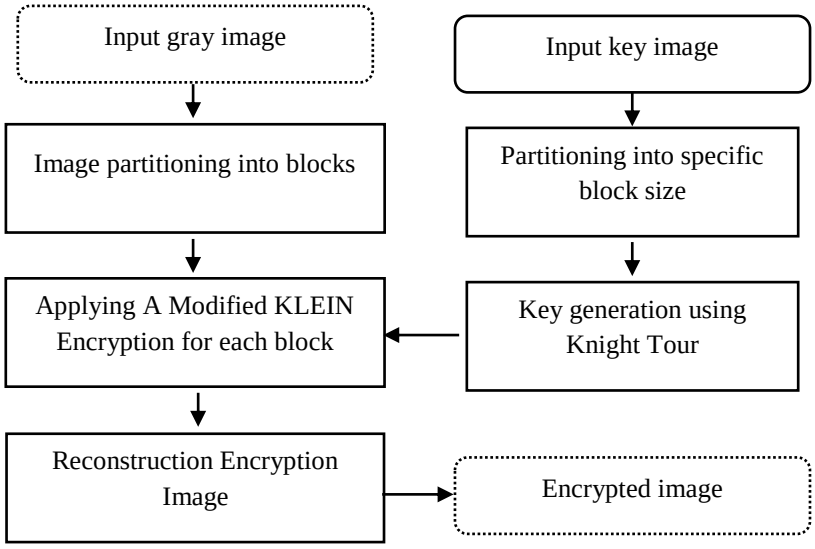

Fig. 2. The suggested method's block diagram.

- **Image partitioning**

Image partitioning divides the input image into smaller, manageable units for encryption. It allows the encryption algorithms on individual blocks of the image instead of the entire image simultaneously. This approach provides several benefits, including parallel processing and efficient memory usage, as explained in Figure 3. The image is segmented into N × N pixel square fixed-size chunks with equal dimensions. This approach is straightforward and allows for easy implementation.

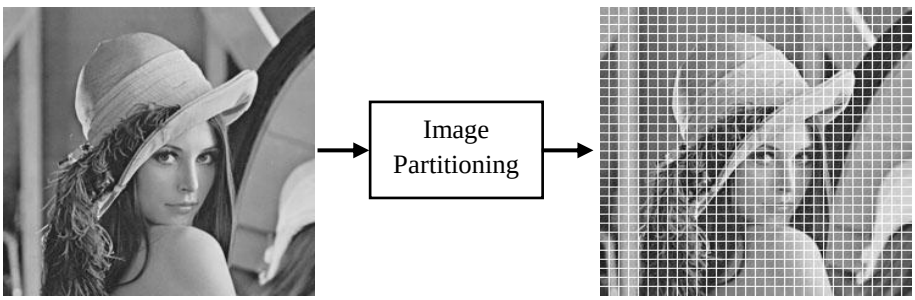

Fig. 3. Image Partitioning.

- **Key generation using Knight Tour**

Because the initial and second bits of an image file might change during compression, encoding, or depending on the kind of image file, the first step entails zeroing those two bits, which are also the least important. Then, values are substituted for the zeroes. Whenever the bits

that remain are primarily ones and replaced with values, these values are changed to zero values. The picture pixels are translated into a binary vector, and via it, the keys are created, if the majority of the other bits are zero. This increases the unpredictability of the bytes utilized and aims to approach the total amount of zeros and units.

The produced bits are divided into smaller pieces, after which the transposition matrix is computed for each piece in a two-dimensional matrix that is equivalent to the chessboard. The beginning position and beginning angle (clockwise or anticlockwise) are specified, and all pathways are allocated employing all starting chessboards and all movements possible for the knight tour. Every beginning point has a unique route that also includes a requirement that utilizes a chessboard one piece at a time to prevent movement loops. To get the bit corresponding to the route, the value of the vector produced by the preceding operation is split into equal blocks of the same size as the chessboard. The partitioned block is an 8*8 square matrix that is XORed with its transpose matrix.

According to each place, the chosen route uses the associated bits. To prevent knight tour movements from looping and returning to the same place on the chessboard, each location is given a value of zero. By doing this, the same location won't be used again. Keys are input for the beginning positions for various lengths. The beginning position of each block is then discovered using an equation. To boost the unpredictability, the reminder bit was combined with the vector of bits at the end of the vectors that the quantized matrix produced. The chosen bits are added to the end of the chain, lengthening it, using all of the bits, and attempting to generate unpredictability even while replaying the same movie. A new sequence is then taken. The resultant bits are merged into a group of bits that will be utilized later and are tested for randomness to see whether the required level of unpredictability was obtained.

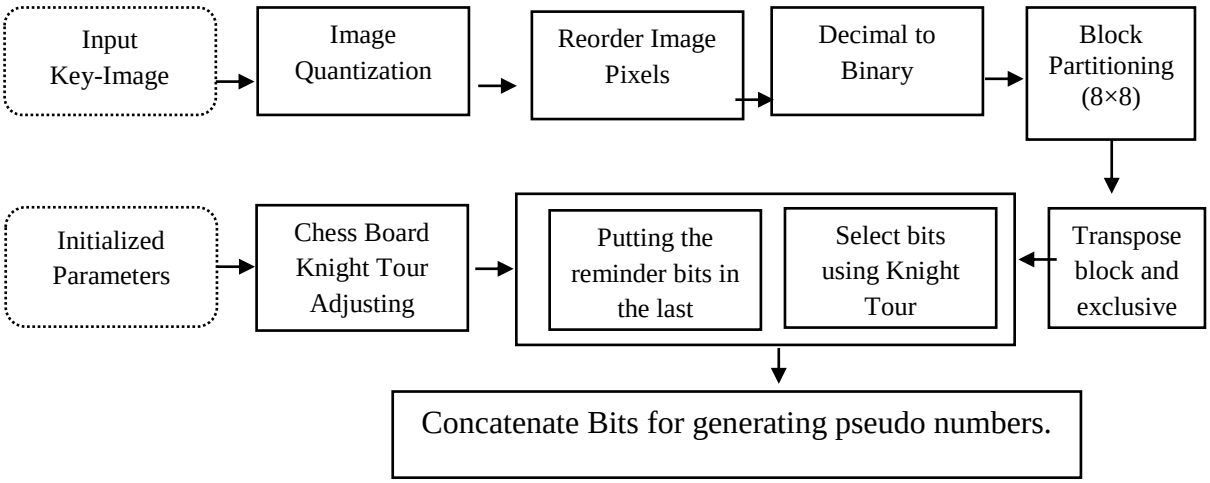

Fig. 3. A Knight Tour In Chessboard For Key Generation

- **Applying A Modified KLEIN Encryption**

The encryption algorithm is applied on each block of a specific size such as (8×8) by splitting the pixel by Knight Tour on a chess bord starting by any previously specified number and moving in a specific direction (clockwise or counterclockwise) without return to used block to avoid the infinite loop, so all pixels are visited one time in total. The new vector is split into 8-byte length vectors that are input to the Modified-KLEIN Encryption algorithm. KLEIN operates on a 64-bit block size and supports key lengths of 64, 80, 96, 112, or 128 bits. For a 64-bit key, there are 12 cycles, and for keys longer than 64 bits, there are 16 rounds. The cipher is based on a combination of a Feistel network structure, and a key schedule derived from the key extension technique. The Feistel network in KLEIN alternates between two branches with a 32-bit round function. The round function utilizes simple operations such as bitwise XOR, bitwise AND, and bit rotations. The key schedule expands the original key to generate subkeys for each round by

overlapping the generated key with previous sequences and applying it. The Knight Tour movement in the chess bord is used for planning of selecting the next chosen number in the key to obtain sub key that differs from the previous key. The KLEIN encryption process:

- **Key Expansion**

The original key is expanded to produce a set of round keys. The size of the key determines the number of rounds: Applying the Knight Tour chess board movement to the created key and overlapping the fresh one with the old one to obtain a new sub-key in each round results in 12 rounds for a 64-bit key. The XOR operation is applied with the input pixel (block) with the first-round key.

- **Main Rounds**

KLEIN consists of several rounds where the main operations are Substitution-Permutation Networks (SPNs). In each main round, the following steps are performed:

- **Add-Round-Key**

The initial set of information encryption uses an XOR technique to combine a block of pixels (a set of pixels) and the subkey.

- **Sub-Nibble**

The bitwise operation transforms the linear data array employing the non-linear 4-bit S-box.

- **Rotate-Nibble**

In the Rotate Nibble algorithm, the state is rotated four nibbles, (two bytes) to the left per each round.

- **Mix-Nibble**

The columns of the data array are multiplied by a modular polynomial equation via the Mix-Nibble technique.

- **Merge Encryption Blocks**

The encryption state of each KLEAN encryption process is merged into a block that is specified before reconstructing the encrypted blocks. All blocks are put in the new image to obtain the encrypted image.

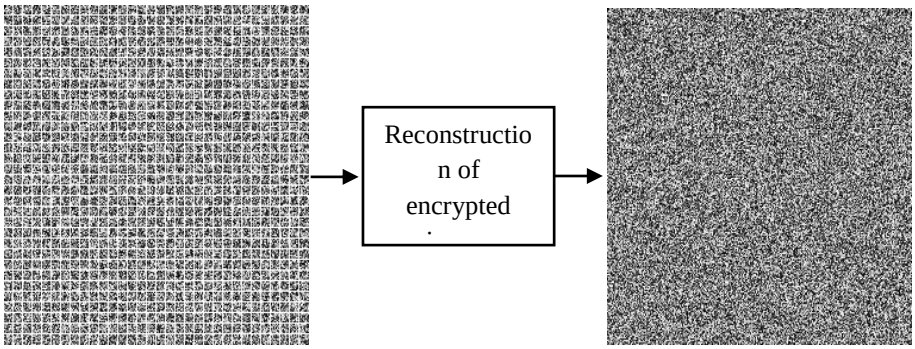

Fig. 4. Cipher Block Reconstruction In The Image.

4. Results and Discussions

The experiment results of the implementation of the proposed method are represented by evaluating it on a set of standard images, woman, Baboon, Pepper, Lenna, car, and house images respectively. The first test is the histogram test which is applied to plain and encrypted images as shown in Figure 5.

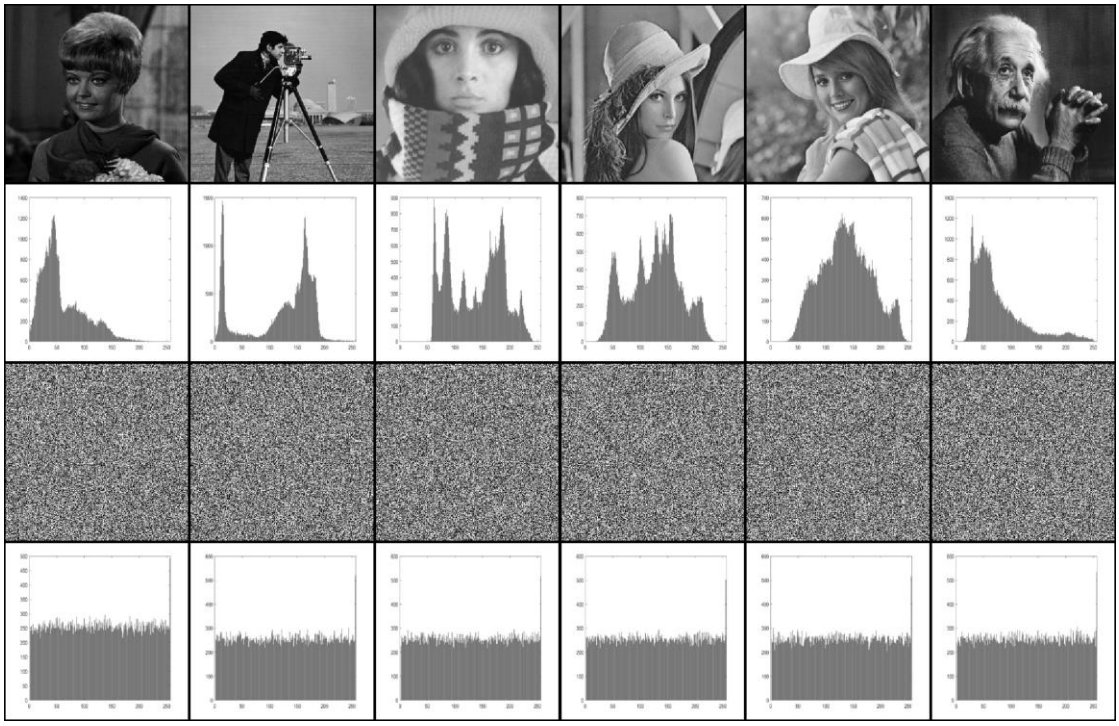

Fig. 5. The Plain Images And Encrypted Images With Their Histograms

- **Histogram Test**

The histograms of unencrypted photos and the corresponding encrypted (output) pictures follow the application of the encryption method (X. Liu et al., 2022; Muthalagu et al., 2020). The histogram test is applied to the plain image and finds the distribution of color intensity then applied the same test on the encrypted image. The histogram of the encryption image should be uniform and no curves are there which means no clear distribution to the results. the uniform distribution makes the guessing of any image features. The attacker could not collect any statistical information about the original image (plain image) and the information appeared as random values as explained in Figure (5).

- **Correlation Test**

The correlation test is applied to find the relation between pixels in an image with its neighbors in three directions horizontal, vertical, and diagonal (Yakout et al., n.d.; Younus & Younus, 2019). This test is applied to the color band of the image before and after encryption to ensure that the encryption breaks the correlations. High correlation values indicate that there may be a recognizable pattern or structure in the encrypted image that might be used to weaken the encryption's security. A strong encryption technique should be resistant to various attacks that rely on patterns and correlations by having low correlations in both directions, which implies that there should be little to no similarity between pixel values in the original and encrypted photos. Furthermore, the correlation test is only one of several tests and metrics used to assess the strength and security of an encryption scheme, just as with any cryptographic research. Extensive security evaluations need resistance to several cryptographic attacks in addition to a detailed analysis of numerous components. As seen in Figure 6, the correlation is applied diagonally to test the pixel with the neighbor pixel diagonally, vertically to test the pixel with a neighbor in a column, and horizontally to test the pixel with a neighbor in a row.

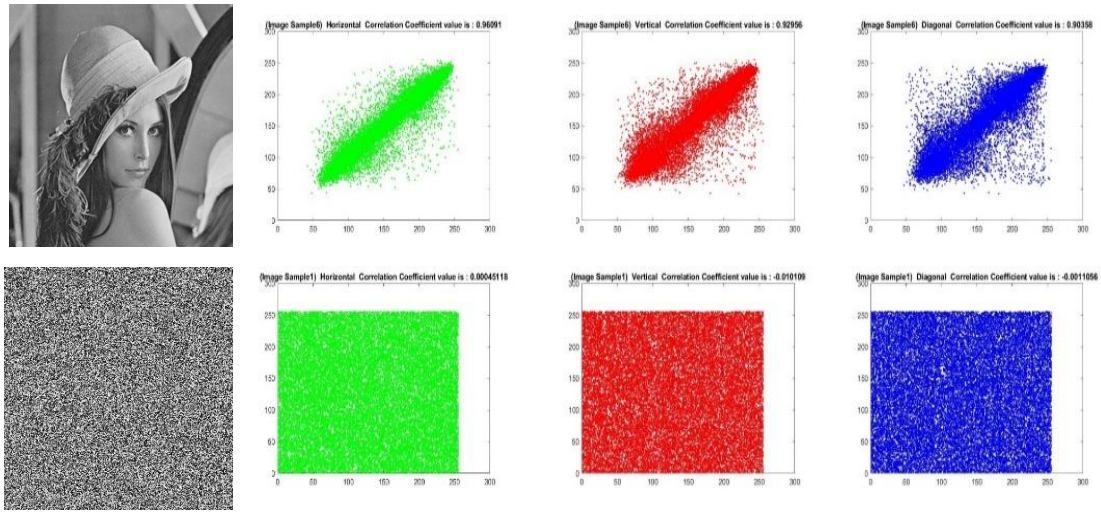

Fig. 6. Correlation Test For Lenna Image

The time required for applying an encryption algorithm can vary based on several factors, including the algorithm's complexity, the hardware used for encryption/decryption, and the size of the input data (the image size). To measure the encryption and decryption times for different image sizes (128x128, 256x256, 512x512). The time measurements can then be recorded for each image size to assess the algorithm's performance. The time consumed for applying the proposed algorithm is measured and explained in Table 2 for the encryption of three sizes of image 128×128, 256×256, and 512×512 in encryption and decryption time.

Table 2 - Time-Consuming For The First Hybrid Encryption/Decryption Algorithm

	Image size					
	Time of size 128x128 in msec		Time of size 256x256 in msec		Time of size 512x512 in msec	
Image #	"Encryption"	"Decryption"	"Encryption"	"Decryption"	"Encryption"	"Decryption"
1	0.024883	0.026074	0.243135	0.206362	0.945512	0.85976
2	0.001348	0.091306	0.066572	0.438198	0.23978	1.446784
3	0.092757	0.050347	0.47579	0.289418	1.675457	1.103044
4	0.038415	0.008419	0.28222	0.12408	1.03268	0.431439
5	0.073559	0.076336	0.295392	0.30051	0.993852	1.076188
6	0.09585	0.004835	0.476845	0.178041	1.624739	0.574728
Average	0.054469	0.042886	0.306659	0.256102	1.085337	0.915324

- **Peak Signal to Noise Ratio Test**

Several objective tests are used to assess the image quality and dissimilarity between original and encrypted images, these tests include:
Equation (1)(Jangir & Pandey, 2021; Sarosh et al., 2022) may be used to determine the peak signal-to-noise ratio (PSNR), which is employed when comparing the quality of an encrypted picture to an unencrypted image. Max is the maximum pixel intensity.

$$PSNR = 10\, log_{10} \frac{max^2}{MSE}$$

- **Mean Squared Error Test**

while MSE's goal is to determine the average square error for two pictures, pixel by pixel, which may be calculated using equation (2) (Liping & Xi, 2020; Mahmood et al., 2016), and max denotes the highest estimate of pixels in the frame. M*N: size of the image, I (x, y) and R (x, y) pixels of the original and restored image.

$$MSE = \frac{1}{M * N} \sum_{y=1}^{N} [I(x,y) - R(x,y)]^2$$

- **Entropy Test**

Shannon entropy was first postulated by Claude Shannon in 1948 (Shannon, 1948). The Shannon entropy has since found widespread use throughout information science. It calculates the randomness of the data in a message and its expected value in bits. S, a random variable, has a Shannon entropy that may be calculated using the equation (3)(Wu et al., 2018).

$$H(S) = H(P_1, \dots, P_n) = -\sum_{i=1}^{n} P_{(si)} \log_2 p_{(si)}$$

- **Structural Similarity Index Metric Test**

Reveals the degree to which the decrypted and original images are comparable. This degree is the quality assessment and estimation that was obtained from multiple windows of the image of the same size. The following Equation (4) can be used to calculate the similarity metric(Kukreja et al., 2021):

$$SSIM(I_1, I_2) = \frac{(2\mu_{I_1}\ \mu_{I_2}\ + C_1)(2\sigma_{I_1\ I_2}\ +\ C_2)}{(\mu_{I_1}^2\ +\ \mu_{I_2}^2 +\ C_1)(\sigma_{I_1}^2 +\ \sigma_{I_2}^2 +\ C_2)}$$

where I_1 and I_2 are the two images , μI1 and μI2 are the average value of I_1 and I_2 , $\sigma_{I_1}^2$ and $\sigma_{I_2}^2$ variance of I_1 and I_2 , . $\sigma_{I_1\ I_2}$ is Covariance of I_1 and I_2 , C_1 and C_2 variables used to stabilize the division where C_1 =$(K_1 L)^2$, L max value of pixel , $K_1 = 0.01$, $K_2 = 0.03$.

Table 3 - Image Quality Test For Data Set 1 Using A Proposed Encryption Algorithm.

#	MSE	PSNR	SNR	SIM	Entropy
1	7899.899	0.004091	1.823994	106.1292	7.218364
2	7300.797	0.004655	1.623212	94.54877	7.006679
3	8969.757	0.00403	1.404365	136.2962	7.219362
4	8437.564	0.004219	1.700803	103.0271	6.993448
5	7466.443	0.004379	1.699944	116.2079	7.015174
6	8683.699	0.004405	1.453238	97.92014	6.887846
Av	8126.36	0.004297	1.617593	109.0216	7.056812

5. Conclusion

In conclusion, the significance of lightweight encryption algorithms within the delicate equilibrium of security and efficiency is paramount. The proposed innovation, involving the adaptation of the KLEIN image encoding method to incorporate the unique mobility of a chessboard's knight, holds the potential to amplify security measures. This inventive approach, manifesting in the creation of a key image through the orchestrated movement of a Knight tour, presents a compelling framework for enhancing encryption. Empirical evidence, as demonstrated by disparities in graphs, correlation tests, and similarity assessments between original and encrypted images, attests to the heightened security achieved. Moreover, the prospect of fortifying data transmission confidentiality gains traction through the proposal to amalgamate the chessboard movement with an additional encryption algorithm. As we navigate forward, the call for sustained research and experimentation resonates, guiding the exploration of this fortified encryption paradigm's full potential and applicability across the intricate landscape of real-world scenarios.

References

Bisht, K., & Deshmukh, M. (2020). Encryption algorithm based on knight's tour and n-neighbourhood addition. *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, 31–36. https://doi.org/10.1109/SPIN48934.2020.9071013

Boussif, M., Aloui, N., & Cherif, A. (2020). Securing DICOM images by a new encryption algorithm using Arnold transform and Vigenère cipher. *IET Image Processing*, *14*(6). https://doi.org/10.1049/iet-ipr.2019.0042

Chatterjee, R., & Chakraborty, R. (2020). A Modified Lightweight PRESENT Cipher For IoT Security. In *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pp. 1-6. https://doi.org/10.1109/ICCSEA49143.2020.9132950

Gong, Z., Nikova, S., & Law, Y. W. (2011). KLEIN: a new family of lightweight block ciphers. *International Workshop on Radio Frequency Identification: Security and Privacy Issues*, 1–18. https://doi.org/10.1007/978-3-642-25286-0_1

Hasoon, J. N., Fadel, A. H., Hameed, R. S., & Khalaf, B. A. (2023). Pseudo number generation based on the knight tour in chess board. *AIP Conference Proceedings*, *2475*(1). https://doi.org/10.1063/5.0102803

Hoomod, H. K., Naif, J. R., & Ahmed, I. S. (2020). A new intelligent hybrid encryption algorithm for IoT data based on modified PRESENT-Speck and novel 5D chaotic system. *Periodicals of Engineering and Natural Sciences*, *8*(4), 2333-2345. http://dx.doi.org/10.21533/pen.v8i4.1738

Jāmiʿah al-Mustanṣirīyah, Institute of Electrical and Electronics Engineers. Iraq Section, & Institute of Electrical and Electronics Engineers. (n.d.). *1st International Conference of Computer and Applied Sciences (1st - CAS 2019): December 18-19, 2019*.

Jangir, A., & Pandey, J. G. (2021). GIFT cipher usage in image data security: hardware implementations, performance and statistical analyses. *Journal of Real-Time Image Processing*, *18*(6), 2551–2567. https://doi.org/10.1007/s11554-021-01146-3

Kukreja, S., Kasana, G., & Kasana, S. S. (2021). Copyright protection scheme for color images using extended visual cryptography. *Computers and Electrical Engineering*, *91*(February), 106931. https://doi.org/10.1016/j.compeleceng.2020.106931

Kushwaha, P. K., Singh, M. P., & Kumar, P. (2014). A survey on lightweight block ciphers. *International Journal of Computer Applications*, *96*(17).

Lallemand, V., & Naya-Plasencia, M. (2014, March). Cryptanalysis of KLEIN. In *International Workshop on Fast Software Encryption* (pp. 451-470). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-46706-0_23

Liping, X., & Xi, W. (2020, August). Improved Encryption Algorithm for Digital Image of Knight Tour Based on Cell Block. In *2020 Chinese Control and Decision Conference (CCDC)* (pp. 1146-1151). IEEE. https://doi.org/10.1109/CCDC49329.2020.9164366

Liu, L., Hao, S., Lin, J., Wang, Z., Hu, X., & Miao, S. (2018). Image block encryption algorithm based on chaotic maps. *IET Signal Processing*, *12*(1), 22–30. https://doi.org/10.1049/iet-spr.2016.0584

Liu, X., Deng, J., Sun, Q., Xue, C., Li, S., Zhou, Q., Huang, X., Liu, H., & Zhou, J. (2022). Differentiation of intracranial solitary fibrous tumor/hemangiopericytoma from atypical meningioma using apparent diffusion coefficient histogram analysis. *Neurosurgical Review*, *45*(3), 2449–2456. https://doi.org/10.1007/s10143-022-01771-x

Mahmood, A. S., Rahim, M. S. M., & Othman, N. Z. S. (2016). Implementation of the binary random number generator using the knight tour problem. *Modern Applied Science*, *10*(4), 35. http://dx.doi.org/10.5539/mas.v10n4p35

Muthalagu, R., Bolimera, A., & Kalaichelvi, V. (2020). Lane detection technique based on perspective transformation and histogram analysis for self-driving cars. *Computers & Electrical Engineering*, *85*, 106653.

Rahman, A. U., Sultan, K., Musleh, D., Aldhafferi, N., Alqahtani, A., & Mahmud, M. (2018). Robust and Fragile Medical Image Watermarking: A Joint Venture of Coding and Chaos Theories. *Journal of Healthcare Engineering*, *2018*. https://doi.org/10.1155/2018/8137436

Raj, V., Janakiraman, S., & Amirtharajan, R. (2021). Optimal concurrency on FPGA for lightweight medical image encryption. *Journal of Intelligent and Fuzzy Systems*, *40*(6), 10385–10400. https://doi.org/10.3233/JIFS-200203

Sarosh, P., Parah, S. A., & Bhat, G. M. (2022). An efficient image encryption scheme for healthcare applications. *Multimedia Tools and Applications*, *81*(5), 7253–7270. https://doi.org/10.1007/s11042-021-11812-0

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Shashikiran, B. S., Shaila, K., & Venugopal, K. R. (2021). Minimal block knight's tour and edge with lsb pixel replacement based encrypted image steganography. *SN Computer Science*, *2*(3), 139. https://doi.org/10.1007/s42979-021-00542-7

Singh, M., Kakkar, A., & Singh, M. (2015). Image encryption scheme based on Knight's tour problem. *Procedia Computer Science*, *70*, 245–250. https://doi.org/10.1016/j.procs.2015.10.081

Singh, S., Sharma, P. K., Moon, S. Y., & Park, J. H. (2017). Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing*, 1–18. https://doi.org/10.1007/s12652-017-0494-4

Wasit University. College of Engineering, & Institute of Electrical and Electronics Engineers. (n.d.). *International Conference on Advances in Sustainable Engineering and Applications : 14-15 March 2018, Wasit, Iraq : ICASEA 2018 conference proceedings*.

Wu, Y., Noonan, J. P., & Agaian, S. (2011). Shannon entropy based randomness measurement and test for image encryption. *arXiv preprint arXiv:1103.5520*. https://doi.org/10.48550/arXiv.1103.5520

Yakout, A., Husseina, S. A., Bakryb, S., & Thabita, M. (n.d.). *BIOCHEMICAL AND HEMATOLOGICAL ALTERATIONS ASSOCIATED WITH DOXORUBICIN INDUCED TOXICITY IN RAT S*. 11-16.

Younus, Z. S., & Younus, G. T. (2019). Video steganography using knight tour algorithm and LSB method for encrypted data. *Journal of Intelligent Systems*, *29*(1), 1216–1225. https://doi.org/10.1515/jisys-2018-0225