Volume (2) Issue (1)

20 Mar 2026

Rawafed Journal of Scientific Studies and Research

Encryption and Decryption of Digital Images Using the XOR Algorithm

Zwha Abdulhmid Mohamed Albeerish - Saedah Mohammed Omar Albeerish

(Faculty member, Department of Computer Science, Faculty of Science, University of Benghazi, Libya)

https://doi.org/10.65723/RMSP2680

Abstract:

This research demonstrates the performance of the XOR algorithm in symmetric encryption and decryption of digital images, which is important for the safe transfer of data by internet. The method uses the inherent properties of XOR, such as the bitwise exclusivity (where A⊕0=A and A⊕A=0) and commutative nature to generate a pseudo-random keystream from which a private-key is applied on. This keystream is then used to perform pixel-wise XOR operations on picture data formats like BMP, JPEG and PNG. This approach transforms the starting pixel values within color channels into a ciphered image without changing the dimensions or memory requirements, thereby preserving confidentiality, integrity, and rapid recovery by employing the same keystream. MATLAB was used to implement XOR algorithm by translated picture matrices in to integers. Analysis of the original images' histograms against the encrypted ones indicated equality in terms of the metrics involved, including means, which stood at 129.57 and 127.50, variance at 5445.09, and entropy at 7.25, thereby indicating uniformity of the pixels in the histograms of the encrypted images, thus improving their resistance towards cryptanalytic attacks. The paper indicates how the use of symmetric XOR encryption proves much more efficient in computing terms compared to the asymmetric approach such as RSA, which involves high computing costs and may prove difficult when it comes to encrypting very large images. It shows how XOR encryption minimizes computational time and is able to handle issues of key reuse efficiently, making it suitable for application in the context of live streams and signals.

Keywords: Encryption, Decryption, Cryptography, Grayscale Images, Histogram Analysis

Introduction

Mathematical transformations have always been a source of inspiration, study and research for human beings. Noting that such a transformation plays a vital role in varied applications makes it all the more fascinating. Any mathematical function f, which is periodic with a defined cycle, i.e., $f(x + c) = f(x)$, where x is a variable and c is a constant, has properties that enable it to process signals, images and other real-time data effectively (Anosova & Kurlin, 2023).

The XOR function also exhibits such a periodic behavior and hence can be explored further for encryption and decryption of real-time data like images and audio. XOR algorithm is a digital signal processing technique used for encryption and decryption of digital images, audio and other forms of data. Its use is illustrated here for security in transmission over the internet (Sabeti & Amerehei, 2022). XOR is employed for the processes of encryption and decryption of images transmitted over the internet. This encryption and decryption take place at the pixel level based on a particular key generated. The size of

the hidden message can vary between 1-64 bits and it has been observed through various tests that at a lower bit rate the hidden message is exposed soon enough (Masood et al., 2022). Also, any signal containing frequencies lower than the lowest frequency contained in the input data can be effectively used. Failing this a random signal generating device is used which can be substituted easily to show the same efficiency for different bit rates, inputs and frequencies (Abugharsa, A. B., et al., 2012).

Encryption:

The Encryption process relies on the XOR algorithm for encryption. XOR algorithm is a symmetric primitive that performs encryption and decryption operations on images using the same Key or Keystream. When a Key is used for encryption, the same Key is needed for decryption. The process generates a corresponding Keystream that is a random byte generated from a Key and is used to perform pixel by pixel XOR operation to produce cipher image (Ihsan & Doğan, 2023).

The Keystream generation can be done by many methods like pseudo-randomly generating a random number, using the image as a seed to generate a random Number, etc. After performing the XOR operation on the original image it creates the cipher image and if the generated Keystream is available at the decryption end the original image can be recovered. The desirable features of image encryption are confidentiality, key management and performance (Mfungo et al., 2023).

When transferring the encrypted information using symmetric encryption of an image often the image is needed on the receiver end. Hence the image is decryption is often used. In hybrid asymmetric encryption the main encryption is performed using asymmetric techniques and the generated symmetric key of the hybrid scheme is used to symmetric cipher the image (Gour et al., 2024).

The public key of asymmetric is freely distributed but the private key needs to be secured. Encryption key exchange can also be used. As intermediate solution the encrypted image can be sent over. Other possibility is encrypting the meta-data or encrypting only the smaller block of the image. All these hybrids' techniques used generally need a large computation resource compared to symmetric encryption. Hence the hybrid encryption is avoided for image encrypting (Halak et al., 2022).

2.1. **Symmetric Encryption**

The Extended XOR Algorithm is a symmetric encryption and decryption method requiring a secret key to generate a pseudo-random keystream, which is XORed with image data. This keystream is produced using a cryptographically strong pseudo-random number generator (PRNG), implementing a stream-cipher model for images (Sabeti & Amerehei, 2022).

XORing happens on every pixel of the source image or the bitmap version of the image. This makes sure that the transformation happens on the pixels and their places. The encryption takes into consideration the color channels of the image and processes them separately. It does not take into account the transparency channel (Amrullah et al., 2025).

This results in an encrypted image that preserves integrity and storage capacity. Confidentiality is ensured, preventing unauthorized access, with additional methods such as hiding the image or reformatting it. Integrity guarantees the image is delivered accurately, especially during transmission. An integrity keystream can be derived from the primary keystream without needing an extra key (Ananth and Ramaiah., 2024).

Key security is critical in any cryptosystem. Various key management strategies have been devised according to system needs, as a compromised key jeopardizes the encryption, while a lost key renders images unrecoverable. Adequate entropy in key generation and independent key creation across devices are crucial. Performance remains significant in ensuring security. An overview of the work-flow illustrates this process (Pillai and Polimetla., 2024).

2.2. Asymmetric encryption

Hybrid approaches combine asymmetric schemes like RSA with symmetric encryption for images. Asymmetric cryptosystems enable secure key exchange through public/private key pairs, allowing users to send images encrypted with a symmetric key. After establishing trust, these systems simplify key

distribution and avoid the complexities of symmetric methods. When applied to image files, RSA facilitates asymmetric encryption of metadata while ensuring compatibility. While RSA introduces some overhead during key exchange, it enhances the efficacy of the encryption process, benefiting the overall performance (Abugharsa, A. B., et al., 2012)

Decryption

XOR-based schemes can encrypt images in formats such as BMP, JPEG, PNG, GIF, and TIFF (Abugharsa, A. B., et al., 2012). Pixel values of color images are processed independently; participants can employ a stream-cipher approach to secure images in a second way. Encryption of an image is performed using pixel values and a secret keystream according to the symmetric encryption principle. Decryption entails using the same keystream on those pixel values to reproduce the original image, maintaining the end-to-end transmission latency (Andono, 2022). The original image can thus be recovered from the ciphertext as long as the keystream remains constant and side information does not leak the keystream information. Security can be compromised significantly if multiple images are encrypted using the same keystream (Guan et al., 2024).

3.1. Symmetric Decryption

Digital images can be encrypted using various techniques, and decoding them is critical for information security. Techniques utilizing symmetric, asymmetric, and pseudorandom methods have been developed. (Hosny et al., 2023) XOR is a simple image encryption technique that protects and embeds information in images for transmission, as it minimally affects neighboring pixels. Decryption retrieves the original image, relying on the same keystream used for encoding. The effectiveness of XOR depends on the correct keystream; using an incorrect one yields a distorted image resembling the original (Masood et al., 2022). The decoded image's quality hinges on the difference between the applied and original keystreams. Noise can make corruption less noticeable, ensuring a level of robustness. Original data from the correct keystream can still be recovered, allowing for reconstruction of the original image without additional information, even if some pixels are compromised (Feng et al., 2025).

3.2. Asymmetric Decryption

Asymmetric encryption does not fit well with digital images because they usually contain a lot of information. In a hybrid approach, which combines asymmetric and symmetric encryption, the asymmetric primitive generates a public/private key pair and personal key for each image. As a workaround to negotiate the personal key, one can either encrypt an additional file that contains it or use other means of exchange that can be forwarded (Babatunde, A. N., et al., 2019).

If a watermark or fingerprinting is embedded in the image file, the public/private key can be derived from it. Even when the public key is exposed, the image remains confidential, but it considerably slows down the process. In the simplest hybrid design, only extra administrative data is encrypted through the asymmetric mechanism, leaving the body intact. Another option considered is symmetric key exchange using an asymmetric scheme to handle larger data sets (Abugharsa, A. B., et al., 2012).

Key exchange during XOR encryption does not allow using the same key for different files, which makes deciphering impossible. The key can be transmitted in one image or even separately. Even though the encryption of keystream affects security negatively, it might be good enough for unimportant files (Shriram et al., 2024). The usage of direct asymmetric encryption requires lots of computing resources, whose amount depends on the used platform. Asymmetric encryption is too costly in terms of resources required for large, unimportant files. User benchmarks on an average PC show that asymmetric encryption methods are about 1,000 times slower than symmetric ones (Kapoor & Thakur, 2022).

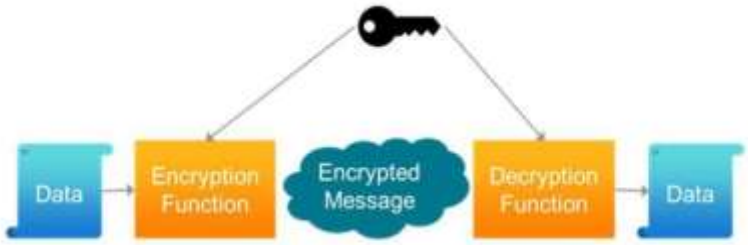
The XOR-based image encryption systems present certain decryption possibilities including maintaining the keystream, determining the keystream using the key, and repairing transmission errors. The three have different complexities, vulnerabilities to ciphertext-only attacks, and correctness. Maintaining the same keystream for images cannot be possible since the pixel-wise XOR alters the plain text, meaning that there is need for generation of another keystream by use of the encryption key (Nujumudeen et al.,

2025). Some stream ciphering techniques used in image encryption solve problems that are encountered in others, but still there is need to generate a new keystream. In such cases, it becomes easy to restore the keystream since there will be no difference. For the case of key-dependent keystreams, consistency with the methods used to encrypt images is essential since the same inputs produce the same outputs. Most conventional image-encryption algorithms facilitate recovery of keystreams because the control values are not known.

Cryptography.

XOR, which stands for logical exclusive-or, is an operator that works on pairs of bits or binary strings. When XOR operates on two bits, the output will be 1 if the two bits are not equal, and 0 otherwise. XOR can be used to operate on any strings using the following rule. Streams of binary data, such as image files, can be considered as sequences of bits, making XOR suitable for processing. Even though image formats like JPEG and PNG introduce a high degree of redundancy, there remains a significant amount of structural and spatial information. Therefore, image encryption requires a careful mathematical study of the cryptographic properties of XOR and an understanding of the associated risks (Abugharsa, A. B., et al., 2012).

The properties of the XOR operation allow any sent data, including images, to be retrieved by simply repeating the same parameter, which means that if a key is reused, the information becomes easily exposed even after only sending an image once. Additionally, security is also compromised if the plaintext information is already known or if the original data is acquired as the image format does not change; the data can easily be reconstructed from the known image (Babatunde, A. N., et al., 2019).



General Framework of Image Encryption Using XOR Algorithm

Need of Security

Obtaining, storing, and modifying images have become easier within the digital world. Consequently, images play a significant role in many fields and applications and have introduced new security issues. Due to extensive use, image information confidentiality, integrity, and authenticity must be protected. (Zheng2023) Protection of private images from any type of digital thieves, such as unwanted disclosure and image tampering, controls the number of criminals stored on the data server and can preserve security of many other data structures included in the database. Prevention against illegitimate use is a major issue that prevents leakage of important images and the accompanying image information (Razak et al., 2025). The activity of hybrid, social and video networks widely promote image stealing. Drawing and altering an image by other people without permission or criminal intention is a big threat for private and uploaded images. Therefore, to provide safety and to escape unwanted forgery, image content security must be ensured through cryptographic processing. Hence, image encryption is performed to maintain safety of private data and important information (A. Sathishkumar et al., 2011).

METHODOLOGY

This study explores the application of the XOR algorithm in the processes of data encryption and decryption. It is an essential procedure that can be used to execute encryption techniques in symmetric

keys. It aims to protect the data from being accessed by other parties. XOR algorithm functions as a cipher, depicting the principles of this encryption technique. It plays a vital role in cryptography because it shows the importance of the XOR algorithm in encrypting data by converting it to a code. A key is essential for conducting a bitwise XOR operation on each character in the data. The original information can be restored through decryption by utilizing the same key in conjunction with the XOR method.
A^ 0 = A, A ^ A = 0,
A ^B = B^A,
(A ^B) ^ C = A ^ (B^ C), (B ^A) ^ A = B^ 0 = B
It can be decrypted once more by employing the identical key and the XOR technique. This compromises the encryption. The text Wiki can be encoded and decoded utilizing the recurring key 11110011, for example.

Keystream Generation Process in XOR Encryption System

Before understanding how XOR encryption and decryption work, it is important to know the basic rules of cryptography that make safe communication possible. Encryption is the best way to make sure that two people can keep their messages private. To send a message from person A to person B, both people must use a specific key. This makes sure that their exchange is safe and private.This can be broken down into a few easy-to-understand parts.
1- Plaintext is the message that A wants to send. A desire to send B an unencrypted message.
2- Cryptography changes readable text into an encoded format with a key, which keeps the information safe from people who shouldn't be able to see it.
3- B will use the same key to decode the message when they get it. This puts the message back to how it was before, which helps B understand what A meant.
These stages show how encryption and decryption work, which are two important parts of cryptography.

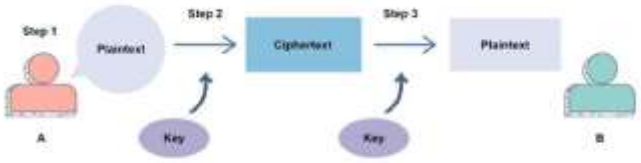


Illustration of the Encryption and Decryption Process

Utilize MATLAB software to implement the XOR algorithm mechanism.

XOR ENCRYPTION AND DECRYPTION IN MATLAB

Encryption software employs a 128-bit key in conjunction with designated data to function efficiently. The procedure for data encryption commences with the concatenation of both the data and the key via an

XOR operation. By executing the same XOR operation with the identical key, the original data can be restored. This XOR methodology can also be effectively utilized for the encryption and decryption of image files. In this specific context, the image data is transformed into integers to facilitate the XOR process. This conversion ensures that the data remains inaccessible without the corresponding key. Access to the image necessitates this key, which is exclusively possessed by both the sender and the recipient post-encryption. The sender selects the key to enhance the security of the data transmission. The identical XOR technique is employed during the decryption process; it is crucial that the keys utilized for both encryption and decryption are congruent to guarantee accurate data recovery.

RESULTS

Original Image Before Encryption

Encrypted Image Using XOR Algorithm

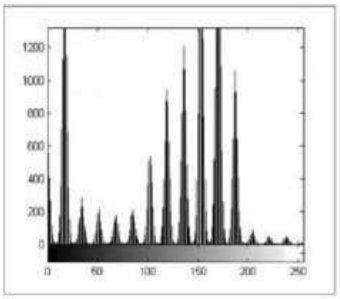
Decrypted Image After Applying XOR Algorithm

Upon comparing the Mean, Variance, and Entropy values of both Input and Output image files, we receive the results detailed below:

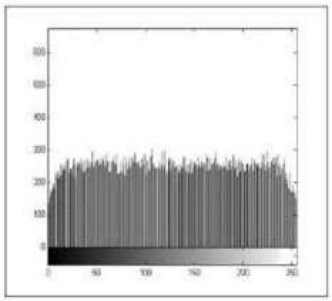
Property	Value for Input Image	Value for Output Image
histogram mean	129.57	127.50
histogram variance	5445.09	5445.09
histogram entropy	7.25	7.25

Histogram Analysis

The histogram shows how the pixel values are spread out in both the original and encrypted images, as shown in the figure. A histogram of a picture shows how its pixel values are arranged. A study of different histograms shows that encrypted images have a more even spread of pixel values than their original or decrypted versions. This uniformity makes it harder to decrypt an encrypted image, which makes it less predictable. The histograms also show that using the XOR function to scramble images makes them more secure.



Histogram of the Original Image Pixel Distribution



Histogram of the Encrypted Image Pixel Distribution

CONCLUSION

Understanding and implementing cryptographic techniques is critical for protecting digital data during the transmission and storage process. Because of such emerging needs, an analysis was conducted on an encrypting and decrypting algorithm of digital images based on Exclusive-OR. With the successful analysis of a particular technique, it was confirmed that the encryption process requires minimal time and that data integrity and confidentiality are intact. The process helps to secure various sizes and types of digital images while simultaneously allowing secure transmission and an effective method of image recovery. Thus, the employment of an image encryption algorithm is a necessity for further protecting sensitive digital data and copyrighted digital images.

References

1. Abugharsa, A. B., Basari, A. S. B. H., & Almangush, H. (2012). A Novel Image Encryption using an Integration Technique of Blocks Rotation based on the Magic cube and the AES Algorithm. arXiv preprint arXiv:1209.4777.
2. Amrullah, A., Ernawan, F., Raffei, A. F. M., & Chuin, L. S. (2025). TDSF: Two-phase tamper detection in semi-fragile watermarking using two-level integer wavelet transform. Engineering Science and Technology, an International Journal, 61, 101909.
3. Ananth, R., & Ramaiah, N. S. (2024). An exhaustive review of the stream ciphers and their performance analysis. Int. J. Reconfigurable Embedded Syst. (IJRES), 13(2), 4864.
4. Andono, P. N. (2022). Improved pixel and bit confusion-diffusion based on mixed chaos and hash operation for image encryption. Ieee Access, 10, 115143-115156.
5. Anosova, O., & Kurlin, V. (2023). Density functions of periodic sequences of continuous events. Journal of Mathematical Imaging and Vision, 65(5), 689-701.
6. Babatunde, A. N., Jimoh, E. R., Oshodi, O., & Alabi, O. A. (2019). Performance analysis of gray code number system in image security. Jurnal Teknologi dan Sistem Komputer, 7(4), 141-146.
7. Beighton, M., Bartlett, H., Simpson, L., & Wong, K. K. H. (2023, June). Key recovery attacks on grain-like keystream generators with key injection. In Australasian Conference on Information Security and Privacy (pp. 89-108). Cham: Springer Nature Switzerland.
8. Feng, W., Tang, Z., Zhao, X., Qin, Z., Chen, Y., Cai, B., ... & Wen, H. (2025). Two-Dimensional Coupling-Enhanced Cubic Hyperchaotic Map with Exponential Parameters: Construction, Analysis, and Application in Hierarchical Significance-Aware Multi-Image Encryption. Axioms, 14(12), 901.
9. Gour, A., Malhi, S. S., Singh, G., & Kaur, G. (2024). Hybrid cryptographic approach: for secure data communication using block cipher techniques. In E3S Web of Conferences (Vol. 556, p. 01048). EDP Sciences.
10. Guan, Q., Deng, H., Liang, W., Zhong, X., & Ma, M. (2024). Multi-images encryption and watermarking with small number of keys via computational ghost imaging. Optics & Laser Technology, 168, 109957.
11. Halak, B., Yilmaz, Y., & Shiu, D. (2022). Comparative analysis of energy costs of asymmetric vs symmetric encryption-based security applications. Ieee Access, 10, 76707-76719.
12. Hosny, K. M., Zaki, M. A., Lashin, N. A., Fouda, M. M., & Hamza, H. M. (2023). Multimedia security using encryption: A survey. IEEE Access, 11, 63027-63056.
13. Ihsan, A., & Doğan, N. (2023). Improved affine encryption algorithm for color images using LFSR and XOR encryption. Multimedia Tools and Applications, 82(5), 7621-7637.
14. Kapoor, J., & Thakur, D. (2022). Analysis of symmetric and asymmetric key algorithms. In ICT analysis and applications (pp. 133-143). Singapore: Springer Nature Singapore.
15. Masood, F., Driss, M., Boulila, W., Ahmad, J., Rehman, S. U., Jan, S. U., ... & Buchanan, W. J. (2022). A lightweight chaos-based medical image encryption scheme using random shuffling and XOR operations. Wireless personal communications, 127(2), 1405-1432.

16. Mfungo, D. E., Fu, X., Wang, X., & Xian, Y. (2023). Enhancing image encryption with the kronecker xor product, the hill cipher, and the sigmoid logistic map. Applied Sciences, 13(6), 4034.
17. Nujumudeen, F., Mubarak, D. M. N., & Hussain, T. (2025). Lightweight XOR-based visual cryptography using random shares for secure colour image sharing with minimal shares. Scientific Reports, 15(1), 42868.
18. Pillai, S. E. V. S., & Polimetla, K. (2024, February). Analyzing the impact of quantum cryptography on network security. In 2024 international conference on integrated circuits and communication systems (ICICACS) (pp. 1-6). IEEE.
19. Razak, M. I. F. B. M., Majid, M. A., Ajra, H., Islam, M. S., & Abdullah, A. (2025, April). An Evaluation of Security Features for Preventing Illegal Distribution of Digital Content Using Quantitative Methods. In 2025 4th International Conference on Computing and Information Technology (ICCIT) (pp. 84-91). IEEE.
20. Sabeti, V., & Amerehei, M. (2022). Secure and Imperceptible Image Steganography in Discrete Wavelet Transform Using the XOR Logical Function and Genetic Algorithm. ISeCure, 14(2).
21. Sathishkumar, G. A., & Sriraam, D. N. (2011). Image encryption based on diffusion and multiple chaotic maps. arXiv preprint arXiv:1103.3792.
22. Shriram, P., Navghare, N., & Bhalerao, A. Y. (2024, January). File Encryption Using AES and XOR Algorithm for Data Security. In 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT) (pp. 407-413). IEEE.
23. Zheng, W. (2023). Current technologies and applications of digital image processing. Journal of Biomedical and Sustainable Healthcare Applications, 3(1), 013-023.