Secure Image Encryption: A Hex-Masked Approach with HMAC and Diffie-Hellman Key Exchange for Enhanced Security

Shivam, shivam.2021c@vitstudent.ac.in

Aditi Jain, aditi.jain2021@vitstudent.ac.in

Abstract: This research explores an advanced approach for securing image data through a comprehensive encryption scheme. The proposed method employs a Hex-Masked encryption technique, integrating the robust security measures of HMAC (Hash-based Message Authentication Code) and the Diffie-Hellman Key Exchange protocol. The Hex-Masked approach involves the utilization of hexadecimal masks to enhance the encryption process, thereby fortifying the confidentiality of sensitive image information. The incorporation of HMAC ensures data integrity and authenticity, guarding against potential tampering and unauthorized access. Furthermore, the Diffie-Hellman Key Exchange protocol facilitates secure key generation and exchange between communicating entities, elevating the overall security of the image encryption process. Through a systematic evaluation, this research demonstrates the effectiveness of the proposed approach in providing enhanced security for image data, offering a resilient solution for safeguarding sensitive visual information in various applications. The originality lies in the innovative algorithm tailored specifically for secure internet transfer. Overall, the study advances the field of image encryption for online transmission.

Keywords: Image Encryption; Hexadecimal; HMAC; Diffie-Hellman; Masking

1. Introduction: In the contemporary era of rapid technological advancements and ubiquitous digital communication, safeguarding the confidentiality and integrity of visual data, such as images, is of paramount importance. The proliferation of data-sharing platforms, coupled with the growing threat landscape, necessitates the development of sophisticated encryption techniques to protect sensitive visual information from unauthorized access, manipulation, and interception. This paper introduces a cutting-edge approach to image encryption, employing a Hex-Masked methodology seamlessly integrated with two robust security mechanisms: Hash-based Message Authentication Code (HMAC) and the Diffie-Hellman Key Exchange protocol. Figure 1 illustrates the yearly occurrences of data breaches and the number of individuals affected by them in the United States from 2005 to 2022. [1].

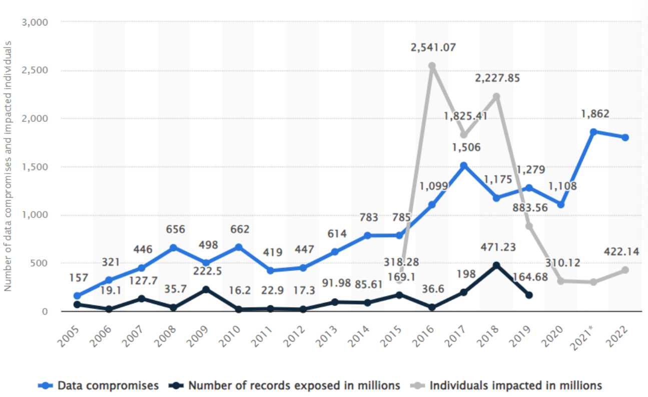


Fig. 1: Annual number of data compromises

Historical Background and Present Scenario of Image Encryption -

Encryption methods have been employed throughout history to secure sensitive information. In the context of images, historical encryption primarily focused on traditional cryptographic ciphers, including substitution and transposition methods. Before the digital era, analog image encryption involved techniques such as visual cryptography, where an image was divided into multiple shares, and the original could only be revealed when the shares were combined. With the advent of digital imaging and widespread internet usage, the need for secure image transmission and storage became more pronounced. Digital encryption methods began to evolve to address the unique challenges digital data poses.

Modern image encryption relies on advanced cryptographic algorithms, including symmetric and asymmetric encryption, to ensure robust protection against unauthorized access. Symmetric key algorithms like AES (Advanced Encryption Standard) are widely used for efficient bulk data encryption, while asymmetric key algorithms such as RSA (Rivest–Shamir–Adleman) facilitate secure key exchange and digital signatures. The adoption of Public Key Infrastructure (PKI) has streamlined the management of digital certificates and public-private key pairs, enhancing the security of image encryption. Cryptographic hash functions, like SHA-256, are integral for ensuring data integrity. Hash-based Message Authentication Code (HMAC) is often employed to verify the integrity of images during transmission. Hybrid encryption schemes, combining symmetric and asymmetric encryption, are commonly used to leverage the efficiency of symmetric encryption and the secure key exchange provided by asymmetric encryption.

In the present scenario, image encryption continues to evolve, balancing the need for robust security with the demands of an interconnected and data-centric world. The integration of advanced cryptographic techniques and the exploration of innovative approaches ensure that image encryption remains a dynamic and critical aspect of information security.

There are various advantages of Image Encryption. Image encryption offers several advantages in the realm of secure online transfer:

1) Confidentiality and Unauthorized Access Prevention: Image encryption serves as a robust safeguard, allowing only authorized users with the requisite decryption keys to access sensitive

visual data. This fundamental feature ensures the confidentiality of the images and provides a critical defense against unauthorized access.

2) Secure Transmission and Communication: Encrypted images can be transmitted securely over networks, reducing the risk of interception by unauthorized entities. This feature is essential in scenarios where secure communication and transmission of visual data are paramount, such as in video conferencing or messaging applications.

3) Integrity Verification and Insider Threat Prevention: Encryption methods include mechanisms for verifying the integrity of images, enabling the detection of any unauthorized modifications or tampering.

4) Compliance and Privacy Protection: Image encryption plays a crucial role in helping organizations adhere to privacy regulations and data protection laws. By securing visual data, it minimizes the risk of privacy breaches and information leakage, ensuring compliance with various legal frameworks.

5) Adaptability and Integration: The flexibility of image encryption allows for adaptation to various applications, providing customized security measures based on unique requirements. Moreover, image encryption seamlessly integrates with access control mechanisms, enhancing overall data protection by restricting access to authorized individuals or systems.

The Hex-Masked approach represents a novel paradigm in encryption, utilizing hexadecimal masks to fortify the confidentiality of image data. This innovative technique adds layer of complexity to the encryption process, enhancing resistance against potential cryptographic attacks. Complementing this, the integration of HMAC ensures the maintenance of data integrity, serving as a cryptographic checksum to detect and prevent tampering. Furthermore, the Diffie-Hellman Key Exchange protocol is incorporated to facilitate secure and efficient key generation and exchange between communicating entities, adding an extra dimension to the overall security of the image encryption process.

This paper aims to contribute significantly to the domain of image encryption by offering a comprehensive solution that addresses the multifaceted aspects of security. By amalgamating the strengths of the Hex-Masked approach, HMAC, and the Diffie-Hellman Key Exchange protocol, we seek to provide an advanced framework capable of fortifying image data security across diverse applications. However, as with any cryptographic methodology, it is imperative to scrutinize potential shortcomings and challenges to ensure a well-rounded understanding of the proposed approach. Through a detailed exploration of its strengths and weaknesses, this research seeks to contribute to the ongoing discourse on fortified cryptographic methods for protecting visual data in our interconnected and data-driven world.

To address the challenges and limitations of existing techniques, we propose a novel image encryption algorithm specifically designed for secure internet transfer. Our algorithm incorporates a 48-bit private key generated using a robust cryptographic algorithm.

The process begins with the conversion of the original image into hexadecimal format. Subsequently, string matching techniques are employed to identify matching patterns within the image. The identified patterns are then replaced with the symbol '#', resulting in an encrypted image file. To ensure secure transmission, both the encrypted image file and the corresponding public key are sent to the recipient. The recipient uses the private key to decrypt the encrypted image, and the hexadecimal code is converted back into the original image.

Image encryption techniques are essential for securing sensitive information during online transfer. This paper explored the historical background and present scenario of image encryption, highlighting the advantages and disadvantages associated with this approach. By evaluating existing encryption techniques and proposing a novel algorithm, we aim to contribute to the advancement of image encryption for secure internet transfer. As digital communication continues to evolve, ensuring the confidentiality and integrity of images remains a critical concern, making ongoing research in this field vital for the protection of digital assets.

2. Related Work: T. Naga Lakshmi and S. Jyothi [2] discussed the PolyGram substitution cipher in their paper. PolyGram substitution ciphers are algorithms that group and substitute blocks of bits or characters by encrypting with a key. These algorithms make it harder for cryptanalysts to destroy single character frequencies, which are preserved under simple substitution ciphers. The technique involves replacing a block of alphabets with another alphabet, resulting in redundant information being difficult to identify. This method is performed block-wise instead of character-wise basis. The PolyGram substitution cipher finds applications in image encryption as well, where three varying keys are selected and the image is maintained in matrix S. The process continues until all sub-matrices get encrypted, and the original image is communicated over a network. Furthermore, Seyed Mohammad Seyedzade et al. [3] presented a paper on an Image Encryption algorithm based on hash function. The core idea of the algorithm involves using one portion of the image's data to reciprocally encrypt the other half. High security, high sensitivity, and fast speed are distinctive features of the technique that can be used for colour and grayscale picture encryption. The algorithm comprises two main components: In the initial phase, the image is preprocessed by dividing it into two equal halves. The second creates a random number mask using a hash function. The remaining portion of the image that will be encrypted is then XORed with the mask. The aim of this study is to enhance the entropy of images.

In a study by Seema Kharod et al. [4], they introduced an enhanced password security scheme based on hashing, incorporating salting and differential masking techniques. The authors suggest a brand-new method that includes hashing, salting, and the development of a crash list using a differential masking procedure. The password file contains the crash list for every user. A security breach involving a password file could lead to hackers attempting to log in using one of the passwords from the compromised list. When a user tries to access using one of these phrases, the application detects it and can block that user or address.

A dynamic hybrid cryptosystem for encrypting photos during transmission and storage is presented in the paper by Flores-Carapia et al. [5]. For increased security, it combines symmetric and asymmetric encryption techniques. In every encryption cycle, the symmetric algorithm generates different boxes and permutations by using Lorenz equations. ElGamal's Diffie–Hellman protocol is used to distribute the secret key, which is created by concatenating two private numbers. In order to generate a seed and construct 128 strings that are connected to the secret key value in a blockchain representation, the proposal makes use of SHA-512. Tests show the cryptosystem's resilience to several types of assaults, and other indicators confirm its high cipher quality. Additionally, simulated noise attacks on encrypted images are used to assess the system, and a 5 x 5 filter is used to test and enhance sharpness loss. Michelle S Henriques and Prof. Nagaraj K. Vernekar [6] stated that in order to protect the communication among devices within an IoT system, a design combining symmetric and asymmetric cryptography is defined in this work. Instead of just utilizing an asymmetric cryptographic algorithm, combining both symmetric and asymmetric cryptography speeds up encryption. The challenge associated with distributing session keys is resolved and the method of symmetric encryption is strengthened by using random keys each time. The AES encryption technique is quick and has the benefit of having a high level of attack resistance. [7][8] hence it is used in image encryption. The paper by Qi Zhang and Quinding [9] introduces an approach that employs the AES algorithm while implementing key control to encrypt images. The approach combines various features and boasts a straightforward design. Given MATLAB's proficient numerical calculation capabilities, particularly in the context of arrays and matrix operations, and the AES algorithm's incorporation of matrices as fundamental units, executing image encryption using the AES algorithm in the MATLAB environment becomes straightforward. Through experimental results, histogram analysis, and key sensitivity analysis, it is evident that this method achieves excellent image encryption outcomes. Additionally, the decryption process mirrors the encryption structure, facilitating easy restoration of the original image. Considering the ease of implementing the AES algorithm in both software and hardware, it lays a strong foundation for future image encryption in both software and hardware-based transmission encryption. Consequently, it is reasonable to anticipate a promising future for image encryption using this approach.

Komal D Patel and Sonal Belani [10] discussed various image encryption techniques in their paper. The two main cryptography techniques they discussed are Secret key cryptography and public key cryptography. Secret key cryptography, also known as symmetric key cryptography, involves the use of a shared secret code known to both the sender as well as receiver, known as the key. The sender encrypts messages using this key, while the receiver decrypts them using the same key.

Further, public key cryptography, commonly referred to as asymmetric key cryptography, employs a pair of keys for encryption and decryption. This method utilizes a matched set of public and private keys.

Cryptography techniques are utilized when transmitting confidential messages between parties over a communication line. These techniques rely on specific algorithms for data encryption. Ensuring the security of data and images is a critical aspect within the vast and continuously expanding realm of digital transfers. The paper by Ahmad Ashraf et al. [11] discusses the critical need for medical information that protects patient privacy, with a particular emphasis on medical imaging, which are sensitive and require strong security measures. The suggested hybrid crypto-algorithm, MID-Crypt, attempts to ensure high security requirements and effectively conceal features in medical images that are shared between laboratories and physicians. MID-Crypt employs Advanced Encryption Standard (AES) with modifiable keys for encryption and Elliptic-curve Diffie-Hellman (ECDH) for image masking. Integrated are key management, digital signatures from patients for authenticity, and Merkle trees for integrity. Peak signal-to-noise ratio (PSNR), correlation, entropy, timing, histogram analysis, and other performance evaluation metrics show how outstanding MID-Crypt is in these areas. Its strong defense against frequent assaults such side-channel, differential, man-in-the-middle, and algebraic attacks is highlighted by comparison with other studies.

The research paper by Morteza Saberi Kamarposhti, Amirabbas Ghorbani, and Mehdi Yadollahi [12] on image encryption provides a comprehensive analysis aimed at understanding and evaluating various encryption techniques, assessment metrics, challenges, and future directions. It categorizes encryption methods into several categories, including symmetric, asymmetric, hybrid, and emerging techniques like chaos-based algorithms and neural networks. Each category undergoes thorough examination, emphasizing their strengths, weaknesses, and practical applications. Their survey emphasizes the importance of selecting appropriate evaluation metrics to accurately gauge the performance and security of image encryption algorithms. Challenges such as noise robustness, lossy compression, and metric selection are identified and discussed extensively to provide insights for addressing limitations. Real-world applications across diverse sectors, including healthcare, banking, defense, and multimedia, underscore the critical role of image encryption in maintaining data security and privacy. By summarizing key findings and identifying potential areas for future research and improvement, the survey serves as a valuable resource for researchers, practitioners, and policymakers in understanding the current landscape and guiding future advancements in image encryption. Overall, the survey offers comprehensive insights into the complexities and opportunities within the realm of image encryption.

In a paper by E. S. I. Harba [13], there is combination of below specified methods to achieve encryption. Symmetric AES algorithm: Used to encrypt files, providing robust encryption for the data being transferred.

1. Asymmetric RSA: Employed to encrypt the AES password, enhancing the security of password transmission by utilizing a public-private key pair.
2. HMAC (Hash-based Message Authentication Code): Utilized to encrypt symmetric passwords and/or data, ensuring secure transmission and making it challenging for attackers to compromise the communication between server-client or client-client.
3. By employing this combination of encryption techniques, the proposed method aims to minimize the risk of exposing unencrypted credentials, enhance the security of transmitted authentication data, and create a robust defense against common attack methods.

N. Thein et al. [14] stated that due to the enormous size of the images, image encryption and decryption take a long time. Given that accuracy and time consumption are the two factors that affect image security systems the most, the effective algorithm should be used depending on the application's desired level of security. Six distinct picture encryption methods—DES, AES, Blowfish, RSA, El-Gamal, and chaos methods—are evaluated in this work. Through the results of simulations, the value and efficacy of each method are demonstrated. The paper by Manish

Gupta et al. [15] introduces a novel image encryption method combining watermarking and cryptographic techniques to ensure secure communication among IoT devices. Employing a two-level security approach, it utilizes a discrete wavelet transform (DWT)--based watermarking scheme and a hybrid encryption technique incorporating logistic chaotic maps and crossover operations. This hybrid approach enhances encryption effectiveness compared to traditional chaotic algorithms. The proposed method undergoes rigorous evaluation against cryptographic attacks, demonstrating its robustness with high values in metrics like NPCR and information entropy. With NPCR reaching 99.63 and information entropy achieving 7.9973, the scheme proves its efficacy in strengthening security for image transmission in IoT environments, making it a promising solution for securing sensitive data in interconnected systems. In their work, Francois et al. [16] introduced an image encryption algorithm based on symmetric key cryptography, focusing on achieving a substantial key space. It is also feared that many techniques, including Advanced Encryption Standard (AES), may be vulnerable to cryptanalysis by analyzing histograms [17, 18].

3. Problem Statement: With the widespread use of images in digital communication, keeping them safe from prying eyes is a challenge. Existing methods for encrypting images have limitations, and there's a need for a better solution. This project aims to create a more secure way of encrypting images using hex masking, HMAC, and Diffie-Hellman key exchange. The goal is to make sure your pictures stay private and secure when you share or store them digitally. The project aims to bring a fresh perspective to image encryption for a safer and more private digital experience.

4. Proposed Model: The proposed methodology for secure image encryption integrates a Hex-Masked approach with HMAC and Diffie-Hellman key exchange to ensure robust confidentiality, integrity, and authenticity of transmitted visual data as shown in Figure 2.

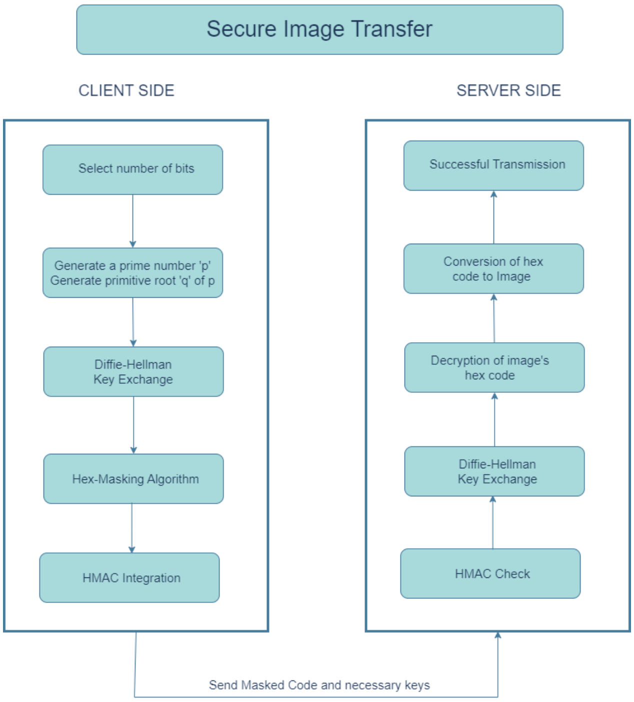

Secure Image Transfer
CLIENT SIDE
SERVER SIDE
Select number of bits
Generate a prime number 'p'
Generate primitive root 'q' of p
Diffie-Hellman Key Exchange
Hex-Masking Algorithm
HMAC Integration
Successful Transmission
Conversion of hex code to Image
Decryption of image's hex code
Diffie-Hellman Key Exchange
HMAC Check
Send Masked Code and necessary keys

Fig. 2: Block diagram of Secure Image Transfer

Initially, on the sender side, a shared secret key is generated using the Diffie-Hellman key exchange protocol. This key, typically 32 bytes in length, is then divided into two separate keys, each consisting of 16 bytes. The first 16-byte key is designated for Hex-Masked encryption, while the second key is used for HMAC verification. The flowchart depicting the Diffie-Hellman key exchange is presented in Figure 3.

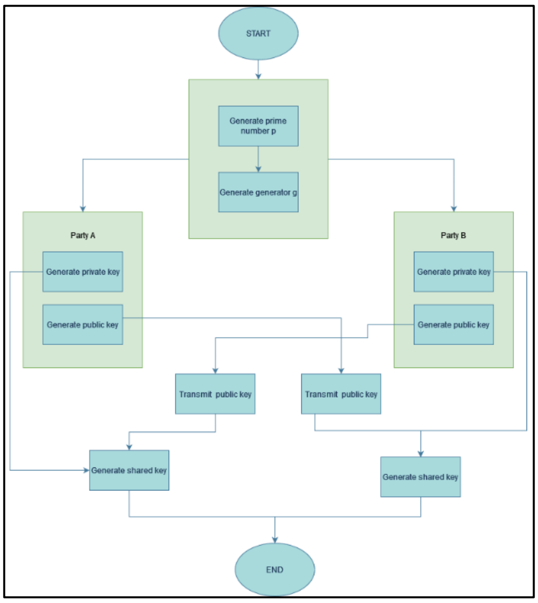


Fig. 3: Flowchart for Diffie-Hellman key Exchange

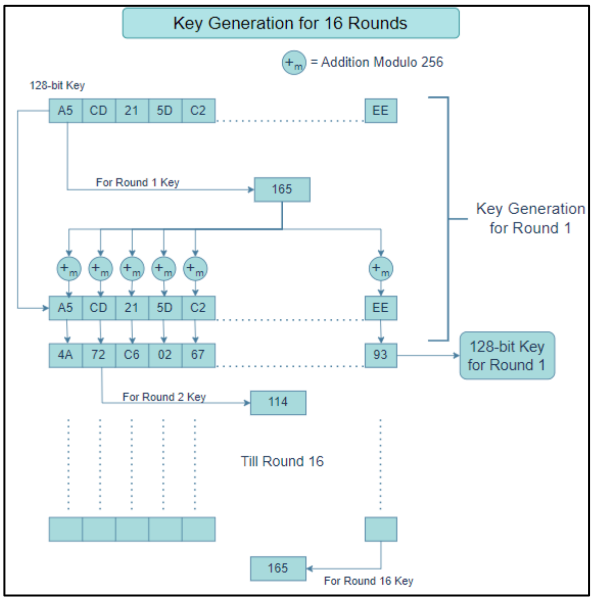


Fig. 4. Flowchart for Key generation (16 Rounds)

Encryption is done in 16 rounds. The key generation process (Fig. 4) for the 16 rounds utilizes the modulo operator. Initially, a 128-bit key is generated from the Diffie-Hellman key exchange. For the first round, the first 8 bits of this key are selected and added to the key for that round. This addition operation is performed modulo 256 to produce a new 128-bit key. Subsequently, for each successive round, the next 8 bits of the 128-bit key generated at the end of the previous round are used for the addition operation. This iterative process continues for a total of 16 rounds, ensuring the generation of unique and secure keys for each encryption round.

During encryption, the Hex-Masked technique is applied to the image data using the first key, employing a hexadecimal mask to modify pixel values and enhance confidentiality. Hex-Masked encryption technique is employed as a fundamental step in securing the image data. This process involves converting the image into hexadecimal code representation. Simultaneously, a mask, generated using the 16-byte key obtained through the Diffie-Hellman key exchange, is created. The length of this mask corresponds to the length of the hexadecimal representation of the image. Subsequently, the hexadecimal representation of the image is XORed with the generated mask. The resultant XORed data undergoes further processing, where it is subjected to HMAC (Hash-based Message Authentication Code) for additional security measures. The same is visualized in Figure 5.

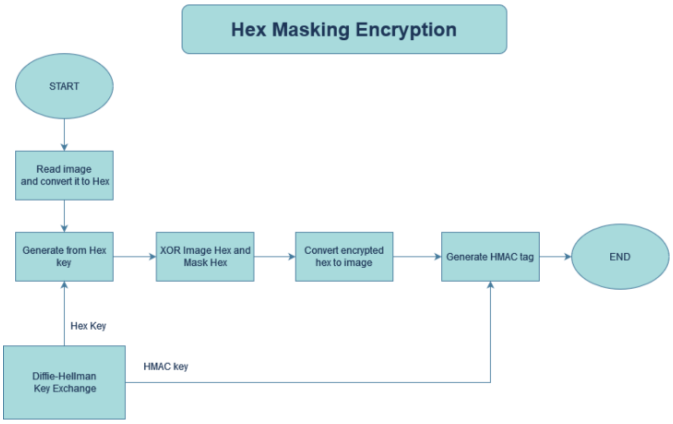

Hex Masking Encryption
START
Read image and convert it to Hex
Generate from Hex key
XOR Image Hex and Mask Hex
Convert encrypted hex to image
Generate HMAC tag
END
Hex Key
Diffie-Hellman Key Exchange
HMAC key

Fig. 5: Flowchart of Encryption using Hex-Masking technique

Simultaneously, the HMAC of the encrypted image data is calculated using the second key to ensure data integrity and authentication. The encrypted image hex, along with the HMAC tag, is then transmitted to the receiver side over a secure channel.

At the receiver side, the HMAC is verified using the second key to ensure the integrity and authenticity of the received data. The shared secret key is reassembled by combining the two 16-byte keys obtained from the Diffie-Hellman key exchange process. The image data is decrypted using the reassembled key through the Hex-Masked decryption process, reversing the hexadecimal masking to retrieve the original pixel values. Verification of the decrypted image data is performed by comparing the recalculated HMAC with the received HMAC. Successful HMAC verification indicates the integrity of the decrypted image data, which can then be outputted for further processing or display. Throughout the encryption and decryption process, end-to-end security measures are implemented to mitigate the risk of unauthorized access or tampering, ensuring the confidentiality and integrity of the image data at all stages of transmission and storage. The decryption using the Hex Masking technique is depicted in Figure 6.

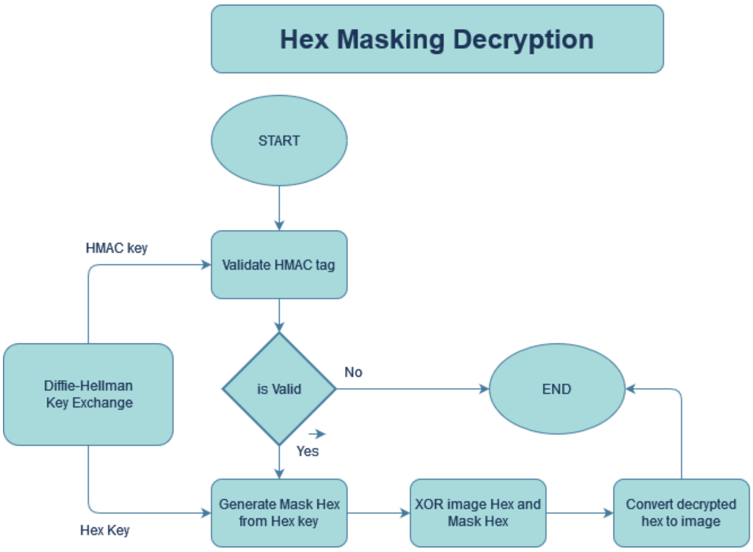

Hex Masking Decryption
START
HMAC key
Validate HMAC tag
Diffie-Hellman Key Exchange
is Valid
No
END
Yes
Generate Mask Hex from Hex key
XOR image Hex and Mask Hex
Convert decrypted hex to image
Hex Key

Fig. 6: Flowchart of Decryption using Hex-Masking technique

5. Result Analysis: The visual evaluation of the encrypted images resulting from the implementation of the proposed algorithm is presented below. Figure 7 shows the original image that is to be transmitted. Figure 8 is the image obtained after encryption. The reconstructed image after decryption is shown in Figure 9.

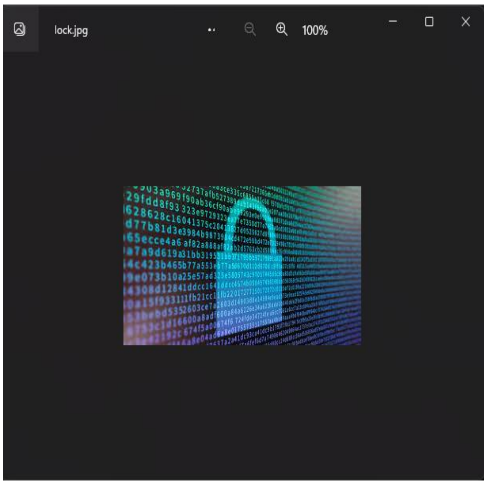
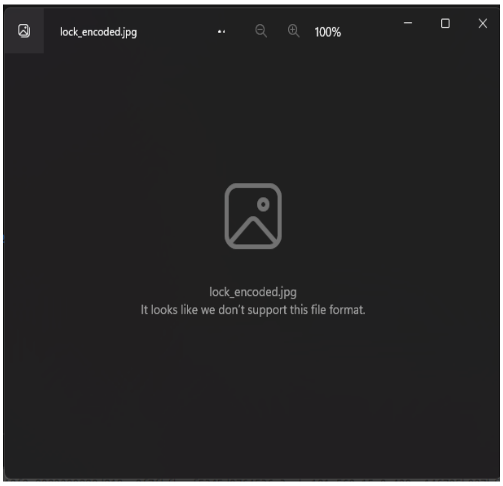


Fig. 7: Original Image to be encrypted and transmitted



lock_encoded.jpg
It looks like we don't support this file format.

Fig. 8: Encrypted image

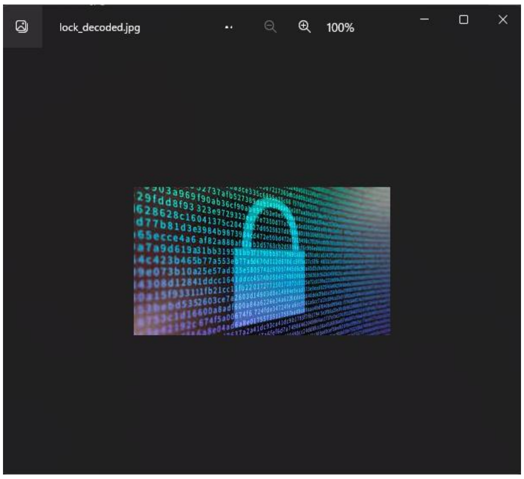


Fig. 9: Received image after decryption

The original file size and the encrypted file sizes are the same, hence no extra space or memory is utilized. Below are the graphical representations of the same.

Figure 10 presents the relationship between the file size, measured in kilobytes (KB), and the encryption time, measured in seconds, for different encryption algorithms, namely DES, RSA, AES, and HexMasking (our algorithm). The purpose of the graph is to compare the performance of these algorithms in terms of encryption time for files of varying sizes.

The x-axis represents the file size, showing increasing values as we move from left to right. The y-axis represents the encryption time, showing increasing values as we move from bottom to top. Each algorithm is represented by a separate line on the graph.

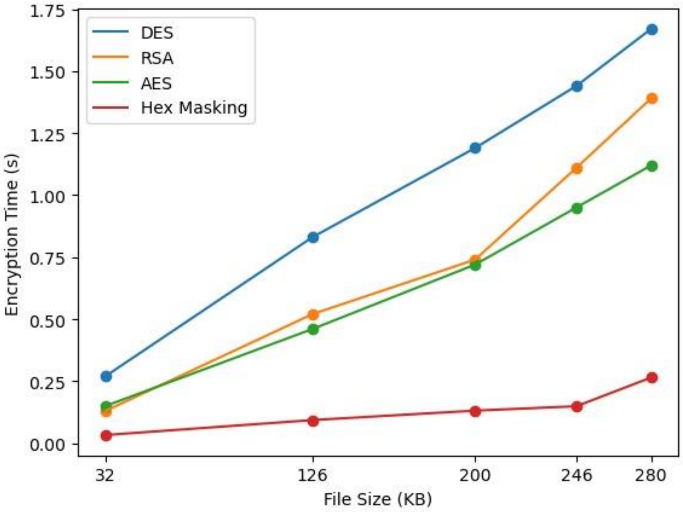
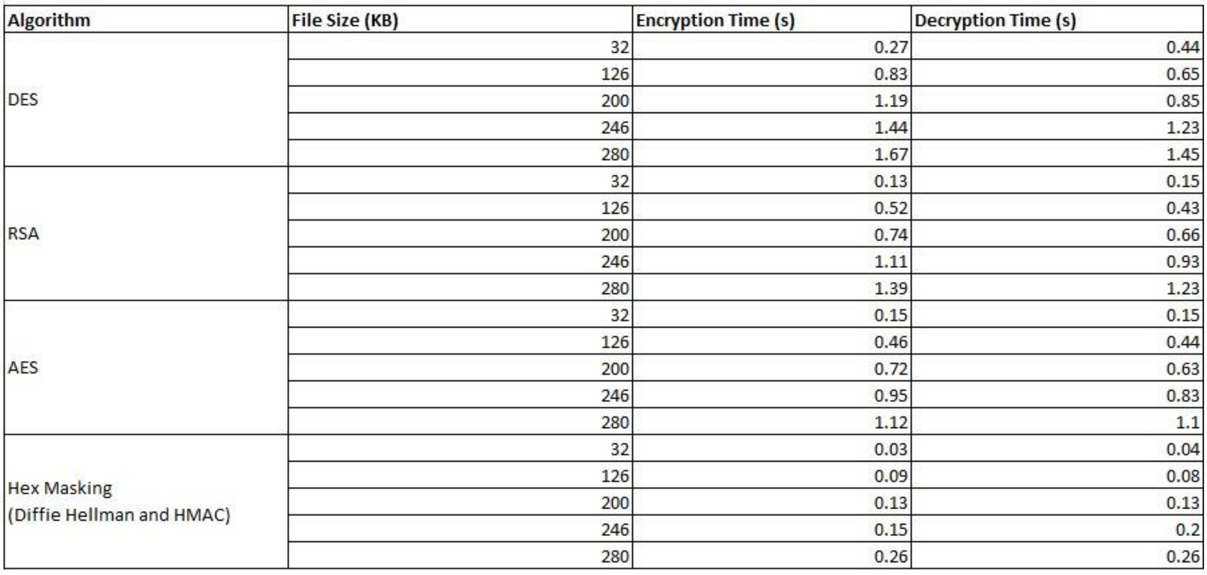


Fig 10: File size vs Encryption time for various algorithms

Table 1 shows the File size, encryption time, and decryption size comparisons between various algorithms [19] and the proposed algorithm. Encryption time for the Hex-Masking algorithm includes key generation and 16 rounds of encryption time. We can infer that the proposed algorithm is performing at par and even better in some cases with the already existing algorithms on the respective basis.

Table 1: Comparisons of file size, encryption, and decryption time between various cryptographic algorithms



Algorithm	File Size (KB)	Encryption Time (s)	Decryption Time (s)
DES	32	0.27	0.44
	126	0.83	0.65
	200	1.19	0.85
	246	1.44	1.23
	280	1.67	1.45
RSA	32	0.13	0.15
	126	0.52	0.43
	200	0.74	0.66
	246	1.11	0.93
	280	1.39	1.23
AES	32	0.15	0.15
	126	0.46	0.44
	200	0.72	0.63
	246	0.95	0.83
	280	1.12	1.1
Hex Masking (Diffie Hellman and HMAC)	32	0.03	0.04
	126	0.09	0.08
	200	0.13	0.13
	246	0.15	0.2
	280	0.26	0.26

Figure 11 presents the relationship between the file size, measured in kilobytes (KB), and the decryption time, measured in seconds, for different decryption algorithms, namely DES, RSA, AES, and HexMasking (our algorithm). The purpose of the graph is to compare the performance of these algorithms in terms of decryption time for files of varying sizes.

The x-axis represents the file size, showing increasing values as we move from left to right. The y-axis represents the decryption time, showing increasing values as we move from bottom to top. Each algorithm is represented by a separate line on the graph.

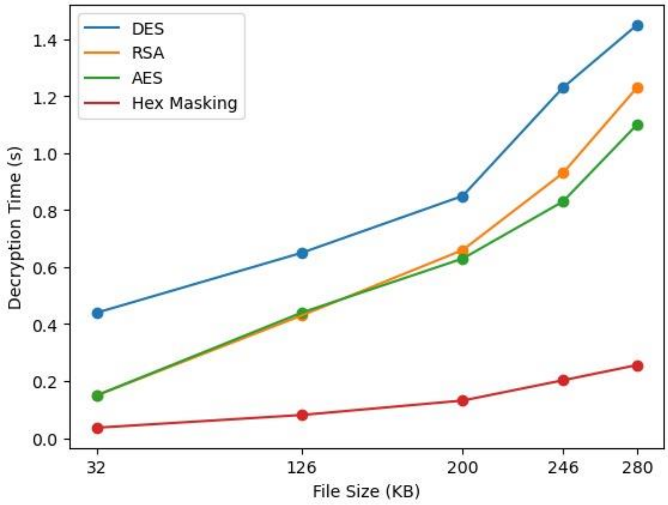


Fig 11: File size vs Decryption time for various algorithms

Figure 12 illustrates the relationship between the file size, measured in kilobytes (KB), and the execution time, measured in seconds, for the Hex Masking Algorithm. The purpose of the graph is to evaluate the Hex Masking Algorithm's performance concerning both encryptions along decryption time for files of varying sizes.

The x-axis represents the file size, with increasing values as we move from left to right. The y-axis represents the execution time, showing increasing values as we move from bottom to top. The graph consists of two lines or curves representing encryption time and decryption time for the Hex Masking Algorithm.

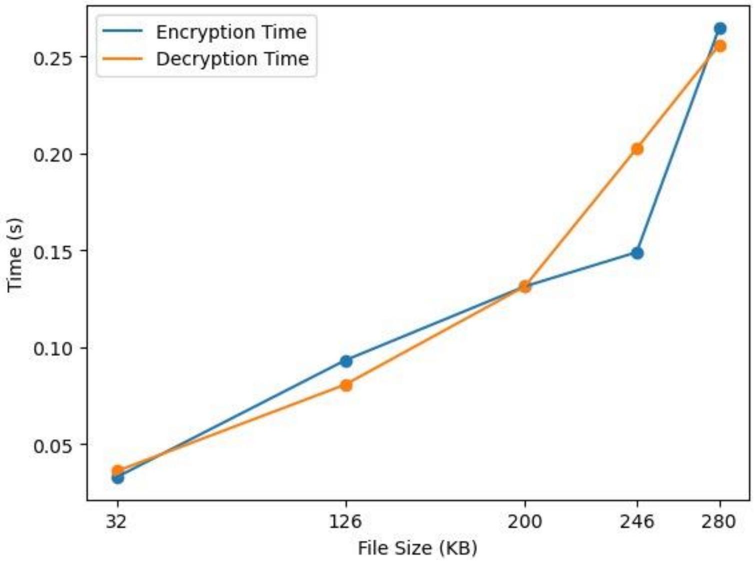


Fig 12: Encryption time vs Decryption time for HexMasking algorithm

6. Conclusion: In conclusion, the Hex-Masked encryption algorithm, incorporating the principles of Hex-Masked encryption, Diffie-Hellman key exchange, and HMAC, offers a robust and efficient method for securing image data. By converting the image into hexadecimal code and generating a mask using a 16-byte key derived from the Diffie-Hellman key exchange, the algorithm introduces an additional layer of security. The XOR operation between the image hex code and the mask ensures enhanced confidentiality, while the subsequent HMAC process further strengthens data integrity. The iterative key generation process, utilizing the modulo operator and successive 8-bit segments of the Diffie-Hellman key, ensures the creation of unique keys for each encryption round, enhancing the algorithm's resilience against cryptographic attacks. The comprehensive integration of these techniques in the Hex-Masked algorithm results in a balanced approach that prioritizes both security and efficiency.

The computation time of our proposed model is significantly shorter compared to other algorithms. This makes it suitable for applications where speed and rapid operation are prioritized over stringent security requirements. It can be effectively employed in scenarios where extremely robust security measures are not essential, striking a balance between efficiency and security for various practical applications. Furthermore, the algorithm's adaptability and scalability make it suitable for various applications requiring secure image transmission and storage, particularly in IoT environments, healthcare, finance, and multimedia sectors. While the algorithm demonstrates promising results in terms of security, future research, and optimizations could further enhance its performance and applicability across diverse scenarios. Overall, the Hex-Masked encryption algorithm presents a promising solution for safeguarding image data in today's increasingly interconnected and data-driven landscape.

References:

[1] Annual number of data compromises. https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/

[2] Singaraju, J., & Lakshmi, T. N. (2019). Enhanced Substitution Cipher Technique for Image Encryption. In Proceedings of International Conference on Recent Trends in Computing, Communication & Networking Technologies (ICRTCCNT).

[3] Seyedzade, S.M., Mirzakuchaki, S., & Atani, R.E. (2010). A novel image encryption algorithm based on hash function. In 2010 6th Iranian Conference on Machine Vision and Image Processing (pp. 1-6). IEEE.

[4] Kharod, S., Sharma, N., & Sharma, A. (2015). An improved hashing based password security scheme using salting and differential masking. In 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions) (pp. 1-5). IEEE.

[5] Flores-Carapia, Rolando, Víctor Manuel Silva-García, and Manuel Alejandro Cardona-López. 2023. "A Dynamic Hybrid Cryptosystem Using Chaos and Diffie–Hellman Protocol: An Image Encryption Application" Applied Sciences 13, no. 12: 7168.

[6] Henriques, M.S., & Vernekar, N.K. (2017). Using symmetric and asymmetric cryptography to secure communication between devices in IoT. In 2017 International Conference on IoT and Application (ICIOT) (pp. 1-4). IEEE.

[7] Zhang, M.R., Shao, G.C., & Yi, K.C. (2004). T-matrix and its applications in image processing. Electronics Letters, 40(25), 1.

[8] Fan, L., Luo, J., Liu, H., & Geng, X. (2014). Data security concurrent with homogeneous by AES algorithm in SSD controller. IEICE Electronics Express, 11(13), 20140535

[9] Zhang, Q., & Ding, Q. (2015). Digital Image Encryption Based on Advanced Encryption Standard (AES). Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC) (pp. 1218-1221).

[10]. Patel, K.D., & Belani, S. (2011). Image encryption using different techniques: A review. International Journal of Emerging Technology and Advanced Engineering, 1(1), 30-34.

[11] Ahmad, Ashraf, Yousef AbuHour, Remah Younisse, Yasmeen Alslman, Eman Alnagi, and Qasem Abu Al-Haija. 2022. "MID-Crypt: A Cryptographic Algorithm for Advanced Medical Images Protection" Journal of Sensor and Actuator Networks 11, no. 2: 24.

[12] SaberiKamarposhti, M., Ghorbani, A., & Yadollahi, M. (2024). A comprehensive survey on image encryption: Taxonomy, challenges, and future directions. Chaos, Solitons & Fractals, 178, 114361.

[13] E.S.I. Harba (2017). Secure Data Encryption Through a Combination of AES, RSA and HMAC. Computer Unit and Internet, College of Arts, University of Baghdad, Baghdad, Iraq · Volume: 7, Issue: 4, Pages: 1781-1785.

[14] Thein, N., Nugroho, H.A., Adji, T.B., & Mustika, I.W. (2017). Comparative Performance Study on Ordinary and Chaos Image Encryption Schemes. International Conference on Advanced Computing and Applications (ACOMP), 122-126.

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

[16] François, M., Grosges, T., Barchiesi, D., & Erra, R. (2012). A new image encryption scheme based on a chaotic function. Signal Processing: Image Communication, 27(3), 249-259.

[17] Anuradha, K., & Naik, P.P.S. (2015). Medical image cryptanalysis using histogram matching bitplane and adjoin mapping algorithms. International Journal of Magnetic Engineering, Technology and Management Research, 2, 100-105

[18] Karuvandan, V., Chellamuthu, S., & Periyasamy, S.S. (2016). Cryptanalysis of AES-128 and AES-256 block ciphers using Lorenz information measure. International Arab Journal of Information Technology, 13(6B), 1054-1060.

[19] Setiadi, D.R.I.M., Rachmawanto, E.H., Sari, C.A., Susanto, A., & Doheir, M. (2018). A Comparative Study of Image Cryptographic Method. In 2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), 336-341.