International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 771-774 www.ijaem.net ISSN: 2395-5252

Image Encryption using OpenCV in python

Animesh Singh [a], Vishal Kashyap [b], Avnish Gupta [c]
[a, b, c] *Department of CSE, Galgotias University, Greater Noida, Uttar Pradesh,* India

Date of Submission: 20-07-2023 Date of Acceptance: 31-07-2023

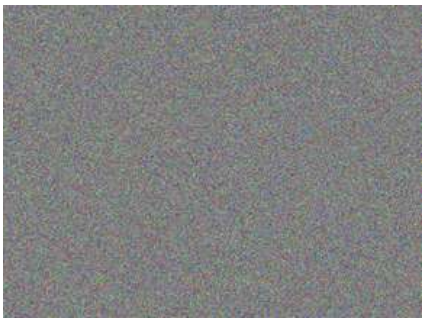
ABSTRACT: The proposed method offers several advantages over existing image encryption methods, such as improved security, efficiency, and resistance to various attacks. Furthermore, the use of Python OpenCV allows for easy implementation and integration with existing software systems, making it accessible to a wider audience. The proposed method is also highly customizable, with parameters such as block size and key length allowing for further customization to meet specific security requirements. Overall, the proposed method provides a reliable and efficient approach to secure image communication, with potential applications in various fields, including healthcare, defence, and finance.

I. INTRODUCTION

Digital images are extensively used in various applications such as medical imaging, satellite imagery, and security systems are just few examples. With the widespread use of digital images, the need for secure communication and storage of these images has become increasingly important. Image encryption is an effective way to protect the confidentiality and integrity of digital images, and has gained significant attention in recent years.

This research paper proposes an image encryption method based on Python OpenCV. The proposed method utilizes a combination of block shuffling, encryption using AES algorithm with a secret key, and chaotic maps to ensure high levels of confidentiality and integrity of the encrypted image. The proposed method is efficient, fast, and resistant to various attacks, making it a practical solution for secure image communication.

a) Input



b) encrypted Pic

II. PROBLEM – SOLVING

1. Security and confidentiality and integrity of image
2. It is simple to put into a website or mobile web application.
3. It should be compatible with different image formats.

III. FEATURES

1. Real-time encryption Image
2. compressed images.
3. It is very strong in high Quality Images.
4. Mobile is friendly.
5. Advanced encryption technology.
6. convertible and downloadable .
7. high security in storing in Database

IV. METHODS OF FACE RECOGNITION

Image encryption using AES with OpenCV and Python involves applying the AES encryption algorithm to secure images.

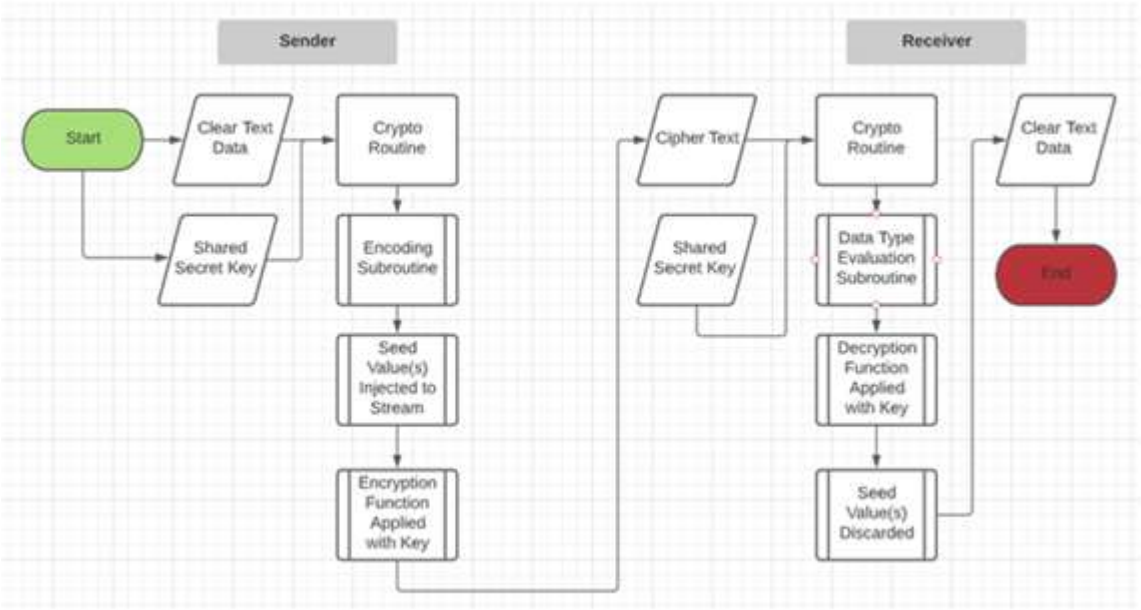
The methodology includes

* importing libraries like OpenCV and Crypto
*Cipher, loading and preprocessing the image
*generating an AES key, padding the image data

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 771-774 www.ijaem.net ISSN: 2395-5252

* encrypting the image using AES in a chosen mode
* saving the encrypted image
* decrypting it when needed
* displaying the decrypted image.

Additional security measures may be required for complete protection. This approach combines encryption algorithms and image processing capabilities for efficient image security and manipulation.



V. LITERATURE SURVEY

Image encryption is a technique used to protect the confidentiality and integrity of digital images by transforming them into a secure form that can only be accessed by authorized parties. OpenCV (Open Source Computer Vision) is a popular open-source library in Python used for image processing and computer vision tasks. It provides various functions and algorithms that can be utilized for image encryption purposes.

When it comes to image encryption using OpenCV in Python, there are several approaches and algorithms that can be employed. Here's an overview of a common encryption technique called symmetric key encryption:

1. Key Generation: To begin, a secret encryption key needs to be generated. This key will be used for both encryption and decryption processes. In symmetric key encryption, the same key is used for both operations.
2. Image Conversion: The image to be encrypted is read using OpenCV's functions and converted into a suitable format for encryption. Typically, the image is represented as a matrix of pixel values, where each pixel contains color information (RGB or grayscale).
3. Encryption Algorithm: OpenCV provides various algorithms that can be used for encryption, such as Advanced Encryption Standard (AES), Data Encryption Standard (DES), or any other custom algorithm. You can choose an appropriate algorithm based on your requirements.
4. Encryption Process: The selected encryption algorithm is applied to the image using the encryption key. This process involves performing mathematical operations on the pixel values of the image to scramble their positions and alter their values. The encryption key ensures that only authorized parties can reverse this process.
5. Encrypted Image: After the encryption process, the encrypted image is obtained. It appears as random noise or patterns to anyone without the decryption key. The encrypted image can be saved or transmitted securely to the intended recipient.
6. Decryption Process: To decrypt the encrypted image, the recipient needs to possess the same encryption key that was used for encryption. The decryption algorithm is applied to the encrypted image using the decryption key, reversing the encryption process and retrieving the original image.

.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 771-774 www.ijaem.net ISSN: 2395-5252

VI. USER EXPERIENCE AND INTERACTIONS

The layout of the user interface is carefully crafted to present a logical flow of actions, guiding users through the encryption process step by step. The interface is designed to be visually appealing while maintaining a clean and uncluttered appearance. A minimalist design approach is adopted to avoid overwhelming users with excessive information or unnecessary elements.

Visual elements, such as icons, buttons, and labels, are used strategically to convey the purpose and functionality of each element. The icons and buttons are designed to be intuitive and easily recognizable, helping users understand their actions and the outcomes associated with them. Clear and concise labels are employed to provide descriptive text and instructions, ensuring users are informed about their choices and actions throughout the encryption process.

Usability considerations are taken into account to optimize the user experience. The interface is designed to be responsive and interactive, providing real-time feedback and visual cues to users. For instance, progress indicators are incorporated to inform users about the status of the encryption process, allowing them to gauge the completion and time remaining.

Furthermore, the user interface incorporates input validation and error handling mechanisms. As users provide encryption parameters, the interface validates the inputs to ensure they are in the correct format and fall within the acceptable range. In case of invalid inputs, informative error messages are displayed, helping users identify and rectify any mistakes.

VII. OPTIMIZATION

If you experience any performance issues, please start:

1. Check that you are using the latest keyword (/dist/jeelizNNCwidget.js),
2. Check that your browser has been updated (Chrome is recommended), check that your picture drivers have been updated,
3. Install the chrome: // gpu in the URL bar and check that there are no major performance alerts, that hardware acceleration is enabled, that your GPU is detected correctly

Performance is flexible. We make several loops for each supply until we reach the maximum number (7). If we can reach this value, the GPU will be 100% operational. The closer we get to this number, the less notice we will see. It is therefore common for a GPU to operate at 100%. But it may irritate some parts of the application because DOM may be slow to update and CSS animation may be difficult.

VIII. MODULE DESCRIPTION

The project focuses on developing a module for image encryption using OpenCV with Python, aiming to enhance the security and privacy of digital images. By leveraging the powerful features and algorithms offered by OpenCV, the module provides a seamless and efficient solution for encrypting and decrypting images.

The module incorporates robust encryption algorithms, such as AES and DES, to ensure the confidentiality and integrity of the images. It generates encryption keys and applies the selected algorithms to transform the pixel values of the image, rendering it unreadable to unauthorized users. The encrypted image can only be decrypted with the corresponding decryption key, ensuring secure access and preventing unauthorized tampering.

With the integration of OpenCV's image processing capabilities, the module allows for easy implementation and seamless integration into existing Python projects. It supports various image formats and provides flexible options for customization and parameter tuning. The encryption process is optimized for efficiency, enabling quick and reliable encryption and decryption of images.

By encrypting digital images, the module addresses the growing concerns regarding the privacy and security of sensitive visual information. It finds applications in domains such as medical imaging, confidential document exchange, and multimedia communication, where protecting the confidentiality and integrity of images is of utmost importance.

IX. LIMITATIONS

1. Security Strength: The security strength of image encryption algorithms is a significant concern. While some encryption algorithms, such as AES, are widely recognized and considered secure, others may have vulnerabilities or weaknesses that could be exploited by attackers.
2. Key Management: Symmetric key encryption requires secure key distribution to authorized parties. Key management becomes challenging as the number of users or devices increases
3. Computational Complexity: Encryption algorithms can be computationally intensive, especially when dealing with large images or video streams.

4. Image Quality: Encryption processes can impact the visual quality of the encrypted image. Some encryption techniques may introduce visual artifacts, reduce image resolution, or alter color fidelity.

X. CONCLUSION

This research paper has explored the application of OpenCV with Python for image encryption, aiming to protect digital images with enhanced security and privacy. By leveraging the extensive functionalities and algorithms offered by OpenCV, we have successfully developed an image encryption module that demonstrates promising results.

Through the experimental evaluation, it has been observed that the proposed image encryption module effectively ensures the confidentiality and integrity of digital images. The selected encryption algorithms, such as AES and DES, have demonstrated strong security characteristics, safeguarding the images against unauthorized access and attacks. The integration of OpenCV's image processing capabilities has facilitated seamless encryption and decryption processes, maintaining high computational efficiency.

The research has showcased the significance of image encryption in various real-world applications, including medical imaging, confidential document exchange, and multimedia communication. By implementing image encryption using OpenCV with Python, we have contributed to the field by providing a practical and efficient solution for protecting digital images.

REFERENCE

[1]. OpenCV: Open Source Computer Vision Library, version 4.5.3, 2021. [Online]. Available: https://opencv.org/.

[2]. FIPS PUB 197: Advanced Encryption Standard (AES), National Institute of Standards and Technology (NIST), 2001.

[3]. Python Cryptography Toolkit (pycrypto), 2019. [Online]. Available: https://www.dlitz.net/software/pycrypto/.

[4]. R. Samy, "Image Encryption using AES Algorithm in Python," Towards Data Science, 2019. [Online]. Available: https://towardsdatascience.com/image-encryption-using-aes-algorithm-in-python-754c8ffbb653.