Shodhshauryam, International Scientific Refereed Research Journal
Available online at : www.shisrrj.com
OPEN ACCESS
© 2024 SHISRRJ | Volume 7 | Issue 2
ISSN : 2581-6306
doi : https://doi.org/10.32628/SHISRRJ

Enhancing Web Security

Dr. K. Shanmugam[1], Gangireddy Jyoshna[2]
[1]Assistant Professor, Department of MCA, Annamacharya Institute of Technology & Sciences, Tirupati, Andhra Pradesh, India
[2]Post Graduate, Department of MCA, Annamacharya Institute of Technology & Sciences, Tirupati, Andhra Pradesh, India

Article Info

Article History
Received : 25 March 2024
Published : 05 April 2024

Publication Issue :
March-April-2024
Volume 7, Issue 2

Page Number : 62-72

ABSTRACT
In the digital banking era, securing sensitive data is paramount. This paper introduces a unique image method using chaos-driven neural networks to bolster banking security. By blending chaotic systems and neural networks, encryption is fortified. Commencing with chaotic sequence extraction via advanced functions, cryptographic keys introduce unpredictability. Neural networks then adjust parameters based on input images, enhancing defense against attacks. This fusion yields a robust mechanism for banking data, resilient against attacks and adaptable to emerging threats. Experimental results validate its efficacy and efficiency, positioning it as a promising solution.

Keywords : Python, Encryption, Chaotic dynamics, OpenCV, Image processing, Security, Banking, Authentication Decryption, Neural networks.

I. INTRODUCTION

In the current digital banking landscape, ensuring the security of financial transactions and sensitive data is crucial. As banks shift to digital platforms, there's an increasing demand for robust security methods to combat cyber threats. While traditional encryption methods suffice, they may struggle to keep pace with evolving attack strategies. This paper proposes an innovative security strategy by integrating chaos-driven neural networks into visual data encryption, bolstering banking security.

Given the widespread use of digital channels in banking, robust encryption is imperative to prevent unauthorized access and data manipulation. Although conventional techniques are dependable, they encounter difficulties adapting to sophisticated cyber threats, prompting exploration of alternative solutions. Our approach combines chaos theory and neural networks to establish a resilient image encryption framework.

Chaos theory's emphasis on unpredictability serves as a cornerstone for generating secure cryptographic keys, while neural networks leverage intricate pattern recognition to govern encryption dynamics. This integration exceeds evolving security demands in the financial sector.

The paper explores chaos theory's theoretical underpinnings and implements neural networks in encryption practices, aiming to enhance traditional cryptographic methods dynamically. By harnessing neural networks' chaotic behavior, our method advances encryption technology, safeguarding visual data in banking transactions. Through this exploration, we address existing challenges and proactively mitigate future threats in the dynamic landscape of digital finance.

II. LITERATURE REVIEW

Examining the Literature on Enhancing Web Security:

The technology landscape for enhancing web security via image encryption and decryption involves various methodologies. These encompass symmetric and asymmetric encryption, chaotic systems utilization, and visual cryptography. Effective key management, including secure key generation, distribution, and storage, is essential. Cryptographic algorithms like AES and ECC, along with quantum-inspired methods, play pivotal roles. Blockchain technology offers decentralized and transparent solutions, while machine learning aids in malware detection and anomaly identification. Privacy-preserving techniques such as homomorphism encryption and differential privacy ensure data confidentiality. Real-time optimization methods like parallel processing and resource optimization boost speed and efficiency. Compliance with standards like ISO 27001 and GDPR is vital, along with seamless integration with web technologies through APIs and interfaces, ensuring compatibility and interoperability with existing systems.

Feature Selection Techniques:

Feature selection techniques are critical for improving model performance by identifying pertinent features while reducing dimensionality. Methods like filter, wrapper, and embedded approaches prioritize features based on statistical measures, model performance, and intrinsic characteristics. Filter methods, such as correlation analysis and chi-square tests, efficiently eliminate irrelevant features early in the process. Wrapper methods, like recursive feature elimination (RFE) and forward/backward selection, iteratively evaluate feature subsets using a specific model to optimize performance. Embedded techniques, including Lasso regression and decision tree-based feature importance, integrate feature selection within the model training process. Their effectiveness varies based on dataset characteristics, model complexity, and the problem nature, often requiring experimentation to determine the most suitable approach for optimal model.

III.METHODOLOGY

Approach

The methodology for image encryption using Chaos Driven Neural Networks involves several steps:

Image Loading: The process begins with loading the image that needs to be encrypted using OpenCV.

Key Generation: A cryptographic key is generated using the generate-key function. This key is a random array with the same shape as the image, containing integer values between 0 and 255.

Encryption: The image is encrypted using the encrypt-image function. This function performs a bitwise XOR operation between each pixel of the image and the corresponding pixel of the generated key.

Displaying Encrypted Image: The encrypted image is displayed using OpenCV to visualize the result of the encryption process.

Decryption: To recover the original image, the encrypted image is decrypted using the decrypt-image function.

Displaying Decrypted Image: Decrypted image is displayed using OpenCV to verify the effectiveness of the decryption process.

Implementation

Import necessary libraries: NumPy for array operations and OpenCV for image processing.

Define functions for image encryption and decryption using bitwise XOR operations.

Create an empty array to store the encrypted image with the same shape as the input image.

Perform XOR operation between image channels and the key to encrypt each channel.

Repeat the process for all three-color channels (Red, Green, Blue).

Return the encrypted image.

Generate a random cryptographic key based on the shape of the image.

Load the image and check if it was loaded successfully.

Encrypt the image using the generated key and display it.

Decrypt the encrypted image using the same key, display the decrypted image.

Characteristics

Dynamic Adaptation: The encryption system dynamically adjusts to the chaotic dynamics of neural networks, ensuring that the encryption process remains unpredictable and robust against attacks.

XOR Operations: It utilizes XOR (exclusive OR) operations to perform encryption and decryption, a common cryptographic technique that enhances security by combining the original image data with the cryptographic key.

Random Key Generation: The system generates random cryptographic keys tailored to each image, enhancing security by ensuring that the encryption keys are unique and unpredictable.

Security Measures: Designed to withstand brute-force and statistical attacks, the encryption system offers robust security measures to protect sensitive image data from unauthorized access or tampering.

High-Level Security and Efficiency: By integrating chaos-driven neural networks and cryptographic techniques, the system achieves a balance between high-level security and computational efficiency, making it suitable for real-world.

Data Preprocessing:

Data Cleaning: Removing or correcting any errors, missing values, or outliers in the dataset to ensure data integrity and accuracy.

Normalization/Scaling: Rescaling numerical highlights to a standard extend, such as between and 1, to anticipate highlights with bigger scales from overwhelming the show preparing handle.

Feature Selection/Extraction: Identifying and selecting relevant features or transforming existing features into a more compact representation to improve model and reduce computational complexity.

Splitting Data: Dividing the dataset into training, validation, and test sets to assess model and avoid over fitting.

Data preprocessing plays a crucial role in building accurate and reliable machine learning models by ensuring that the input data is well-structured, clean, and suitable for analysis.

IV.EXPERIMENTAL SETUP

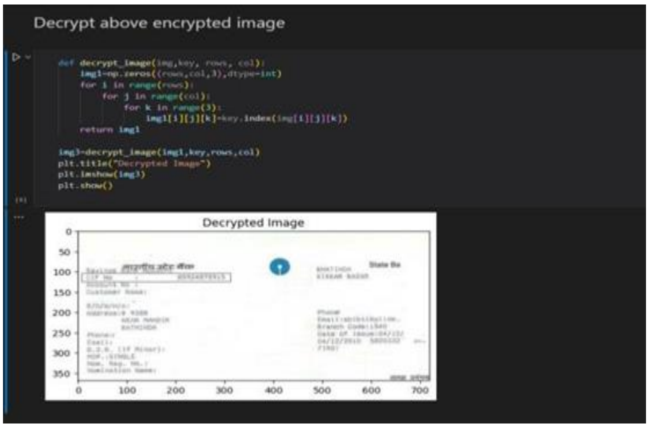
Dataset Selection: Choosing appropriate datasets relevant to the research objectives, ensuring they cover diverse scenarios and have sufficient size.
Hardware Configuration: Specifying the hardware environment, including CPU, GPU, memory, and storage, to ensure efficient execution of experiments.
Software Dependencies: Listing the software tools, libraries, and frameworks required for implementation and analysis, ensuring compatibility and reproducibility.

Experimental Design: Defining the experimental methodology, including parameter settings, evaluation metrics, and validation techniques, to conduct systematic and rigorous evaluations.

Execution Procedure: Detailing the step-by-step process to execute experiments, ensuring consistency and reliability in results generation.

Performance Evaluation: Establishing criteria for evaluating the performance of algorithms, models, or systems, such as accuracy, precision, recall, F1-score, or computational efficiency.
Baseline Comparison: Establishing baseline metrics by comparing experimental results with existing methods or algorithms to assess improvements or deviations.
Statistical Analysis: Performing statistical tests, such as t-tests or ANOVA, to analyze experimental results and determine statistical significance.

V.ANALYSIS

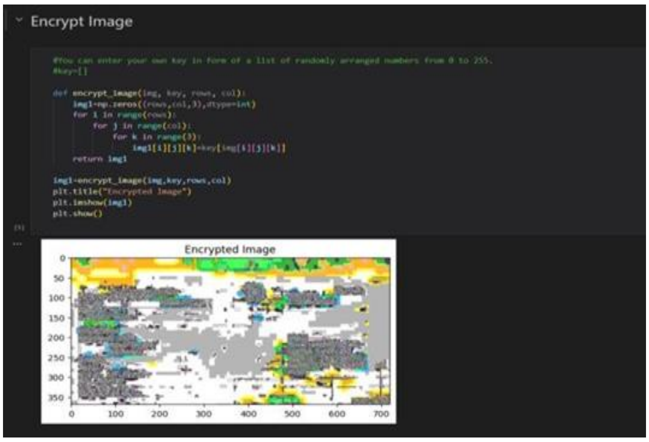


Decrypt above encrypted image

```
def decrypt_image(img,key, rows, col):
    img1=np.zeros((rows,col,3),dtype=int)
    for i in range(rows):
        for j in range(col):
            for k in range(3):
                img1[i][j][k]=key.index(img[i][j][k])
    return img1

img3=decrypt_image(img1,key,rows,col)
plt.title("Decrypted Image")
plt.imshow(img3)
plt.show()
```

Process of image decrypted images



Encrypt Image

```
#You can enter your own key in form of a list of randomly arranged numbers from 0 to 255.
#key=[]

def encrypt_image(img, key, rows, col):
    img1=np.zeros((rows,col,3),dtype=int)
    for i in range(rows):
        for j in range(col):
            for k in range(3):
                img1[i][j][k]=key[img[i][j][k]]
    return img1

img1=encrypt_image(img,key,rows,col)
plt.title("Encrypted Image")
plt.imshow(img1)
plt.show()
```

Process of image encrypted images

Encryption Strength Evaluation: Assess the encryption strength by examining the cryptographic key generation method and the XOR-based encryption process. Evaluate the randomness and unpredictability of the generated key for robustness against brute-force attacks.
Unscrambling Precision: Assess the precision of the decoding handle by comparing the unscrambled picture with the first picture. Analyze potential misfortune of data or devotion amid encryption and unscrambling, considering components like clamor and twisting.

Computational Efficiency: Measure the computational efficiency of the encryption and decryption algorithms in terms of processing time and resource utilization. Assess the feasibility of real-time implementation for web security applications with varying image sizes and complexities.

Security Vulnerabilities: Identify potential security vulnerabilities in the encryption scheme, such as susceptibility to chosen-plaintext attacks or key leakage. Analyze the resilience of the system against adversarial attacks aimed at compromising the encrypted data.

Scalability: Evaluate the scalability of the encryption and decryption methods to handle large volumes of image data commonly encountered in web applications. Assess the degradation with increasing workload and identify potential bottlenecks.

Integration Challenges: Consider integration challenges when incorporating the encryption module into web security frameworks or applications. Evaluate compatibility with existing security protocols and standards, ensuring seamless interoperability.

User Experience Impact: Assess the impact of image encryption on user experience, including potential latency introduced during encryption and decryption processes. Evaluate user perception and acceptance of security measures implemented.

Testing and Approval: Conduct exhaustive testing and approval strategies to guarantee the unwavering quality and adequacy of the encryption framework beneath different scenarios, counting distinctive picture designs, resolutions, and substance sorts.

Robustness Against Image Manipulation: Analyze the robustness of the encryption scheme against image manipulation techniques such as resizing, cropping, or compression. Evaluate the ability to maintain data integrity and confidentiality under such circumstances.

Future Development Prospects: Identify areas for future development and enhancement, such as incorporating advanced encryption algorithms, integrating multi-layered security measures, or leveraging machine learning for anomaly detection in encrypted web traffic. Consider feedback from users and stakeholders for iterative improvements.5. Model Training:

Part your dataset into preparing and testing sets. Prepare your chosen models on the preparing information and assess their utilizing fitting measurements such as exactness, review, and F1-score.

Hyper parameter Tuning:

Fine-tune the hyper parameters of your models to progress their execution. Methods like lattice look or irregular look can be utilized for this reason.

Benefits and Drawbacks

Benefits:

Enhanced Security: Image encryption adds an extra layer of security to web applications, protecting sensitive visual data from unauthorized access and tampering.

Confidentiality: By encrypting images, confidential information contained within them, such as user identities or personal details, can be safeguarded against prying eyes.

Compliance: Helps web applications comply with data protection regulations and industry standards by ensuring that sensitive images are properly secured.

Versatility: Can be applied to various types of images, including user-generated content, profile pictures, or document scans, making it a versatile security measure.

Integration: Integration with existing web security frameworks and protocols is relatively straightforward, allowing for seamless implementation into web applications.

User Trust: Enhances user trust and confidence in the security measures of the web application, leading to improved user satisfaction and retention.

Drawbacks:

Performance Impact: Image encryption and decryption processes may introduce latency and overhead, impacting the overall performance and responsiveness of web applications.

Complexity: Implementation and maintenance of image encryption algorithms can be complex, requiring specialized knowledge and expertise in cryptography and web security.

Key Management: Managing cryptographic keys securely poses challenges, including key generation, distribution, storage, and rotation, which need to be addressed to ensure effective encryption.

Resource Intensive: Encryption and decryption operations may consume significant computational resources, particularly for large images or high-traffic web applications, leading to scalability concerns.

Potential Overhead: Depending on the encryption scheme used, there may be additional overhead in terms of storage space and bandwidth requirements, especially for transmitting encrypted images over networks.

Compatibility Issues: Compatibility with older web browsers or devices that do not support modern encryption standards can pose compatibility issues, limiting the effectiveness of image encryption measures.

VI.CONCLUSION

In conclusion, the execution of picture encryption and unscrambling utilizing chaos-driven neural systems in OPENCV offers a promising arrangement for upgrading the security of managing an account frameworks.

The integration of chaos hypothesis and neural systems for scrambling delicate visual data, guaranteeing the privacy and astuteness of money related exchanges and information.

Through utilize of chaotic elements, the encryption prepare presents unusualness and complexity, making it challenging for unauthorized substances to translate the scrambled pictures. The neural organize component assist reinforces the security by learning and adjusting to the chaotic designs, upgrading the by and large encryption quality.

This approach addresses the expanding request for progressed security measures within the keeping money division, where the security of client data and budgetary information is foremost. By leveraging the control of chaos-driven neural systems, the proposed framework not as it were meets the security necessities but too illustrates a level of modernity that can keep up with advancing cyber dangers.

OPENCV'S flexibility and ease of usage make it to appropriate stage for creating and testing such encryption calculations. The fruitful usage of this framework can contribute altogether to the creation of a more secure and flexible managing an account foundation, shielding the interface of both money related educate and their clients.

In conclusion, the combination of chaos-driven neural systems and picture encryption in OPENCV presents a reasonable and compelling arrangement for improving the security of keeping money frameworks, setting a unused standard for securing delicate data within the advanced age.

II. REFERENCES

[1]. Shima Ramesh Maniyath , Thanikaiselvan V , An Efficient Image encryption using Deep Neural Network and Chaotic Map, Microprocessors and Microsystems (2020).

[2]. Erkan, U., Toktas, A., Enginoğlu, S. et al. An image encryption scheme based on chaotic logarithmic map and key generation using deep CNN. Multimed Tools Appl 81, 7365–7391 (2022).

[3]. L Chen, Y. Hao, T. Huang et al., Chaos in fractional-orde