SOFTWARE REQUIREMENT SPECIFICATION

For

Chaotically Encrypted Pixels

Submitted By

Specialization	SAP ID	Name
BTech CSE(CSF)	500086715	Vishal Odhar
BTech CSE(CSF)	500083113	Vijay Malik
BTech CSE(CSF)	500087942	Mayank Sonwane
BTech CSE(CSF)	500086709	Tanushpreet Kaur

Department of Systemics

School Of Computer Science

UNIVERSITY OF PETROLEUM & ENERGY STUDIES,

DEHRADUN, UTTARAKHAND- 248007

Dr. Virender Kadyan

Project Guide **Cluster Head**

Table of Contents

Revision History

Date	Change	Reason for Changes	Mentor Signature
02/09/2023	Title change		
12/09/2023	Synopsis	Literature Review	
13/09/2023	Synopsis	Methodology	

1. INTRODUCTION

In today's digital age, the secure transmission and storage of sensitive visual information, such as transport images, have become important in ensuring the integrity and confidentiality of critical data [1]. The management of traffic system is very complex so to manage it An Intelligent Transport System (ITS) is used which uses smart devices to capture the traffic data in the form of images. This system uses smart devices to perform real-time traffic management, public data collection, and vehicles speed checking [5]. These smart devices capture and transmit traffic data in the form of images to server via public channel, which can be vulnerable to theft and misuse during communication. To address these challenges, a novel image encryption approach has been developed by Atul Kumar and Mohit Dua.

An image encryption method that combines the power of Gated Recurrent Units (GRU) and chaos theory principles [1]. GRU is a type of recurrent neural network architecture, exhibits its use in sequential data processing tasks, while chaos theory generates unpredictable and complex cryptographic keys. The fusion of these two elements results in a robust image encryption scheme which provides a solution to the specific demands of securing transport-related visual data.

The encryption algorithm that uses the Gated Recurrent Unit (GRU) and Sine-Cosine chaotic map to encrypt transport images is divided into three phases. In the first phase, two intermediate keys and the seed value required for creating chaotic sequence are generated using unique combinations of 128-bit share key and 128-bit initial vector. In the second phase, permutation is performed using one of the intermediate keys and the chaotic sequence generated by the novel Sine-Cosine chaotic map. The final phase performs the diffusion process using the other intermediate key and GRU approach that uses the chaotic sequence generated by the Sine-Cosine map [1].

1.1 Purpose of the Project

The purpose of the project is to increase the security and privacy of images, especially in the area of transport or communication. The goal is to protect the confidentiality of sensitive visual information during transmission or storage.

1.2 Target Beneficiary

The target beneficiary of the project is the transport industry, to be more specific images related to the transportation industry, such as images related to vehicles, traffic tracking or other visual data related to transportation. The project may have practical applications in transportation systems, such as secure image transmission in traffic management, vehicle recognition, or surveillance systems.

1.3 Project Scope

A. Algorithm Development:

- Design and implement a novel encryption algorithm that uses GRU for key generation.
- Incorporate chaos-based methods to further strengthen encryption.
- Explore the use of advanced chaos algorithms (e.g., logistic map, Lorenz system) for key generation and diffusion.

B. Dataset Selection:

- Choose diverse transport image datasets to evaluate the algorithm's effectiveness.
- The datasets should include images related to various modes of transportation, such as road, rail, air, and sea transport.

C. Security Analysis:

- Conduct a comprehensive security analysis of the encryption algorithm.
- Evaluate the algorithm's resistance against common cryptographic attacks (e.g., brute force, differential, and statistical attacks).
- Assess the key sensitivity and the impact of chaos-based methods on security.

D. Performance Evaluation:

- Measure the encryption and decryption time for different image sizes.
- Analyze the algorithm's performance in terms of computational efficiency.
- Ensure minimal impact on image quality during encryption and decryption.

E. Implementation:

- Develop software or a programming framework to implement the proposed encryption method.
- Create a user-friendly interface for encrypting and decrypting transport images.

F. Documentation:

- Produce comprehensive documentation, including technical reports, user manuals, and code documentation.

1.4 References

[1] Atul Kumar, Mohit Dua (2022) A GRU and chaos-based novel image encryption approach for transport images. https://doi.org/10.1007/s11042-022-13902-z

[2] Chai X, Chen Y, Broyde L (2017) A novel chaos-based image encryption algorithm using DNA sequence operations. Opt Lasers Eng 88:197–213. https://doi.org/10.1016/j.optlaseng.2016.08.009

[3] Chen J, Zhu Z, Fu C, Yu H (2013) An improved permutation-diffusion type image cipher with a chaotic orbit perturbing mechanism. Opt Express 21(23):27873. https://doi.org/10.1364/oe.21.027873

[4] Dua M, Suthar A, Garg A, Garg V (2020) An ILM-cosine transform-based improved approach to image encryption. Complex Intell Syst 7:1–17. https://doi.org/10.1007/s40747-020-00201-z

[5] Thoms GRW, Muresan R, Al-Dweik A (2019) Chaotic encryption algorithm with key controlled neural networks for intelligent transportation systems. IEEE Access 7:158697–158709

2. PROJECT DESCRIPTION

2.1 Data/ Data structure

The data structures and the flow of data within the code are:

1. img: This is the input image loaded using OpenCV's cv2.imread. It is a 2D NumPy array representing the grayscale image.

2. ChaoticSequence: This is a list of lists generated by the chaoticSequ_Sine_Cosine_Map function. It represents the key and the pixel index shuffling sequence. Each list inside ChaoticSequence corresponds to a row of the image, and each element in the list corresponds to the order in which pixels are shuffled.

3. PermutatedImage: This is the image after shuffling the pixels row-wise using the permutatedImage function. It is another 2D NumPy array, representing the shuffled image.

4. hiddenUnit: This is the number of hidden units used in the GRU operation, and it seems to be a scalar value.

5. GRUVal: This is the result of the GRU operation performed on ChaoticSequence with hiddenUnit as a scalar value. It's a 2D NumPy array representing the result of the GRU operation.

6. key2: This is a scalar value used for encryption and decryption.

7. output: This is the result of multiplying GRUVal by key2. It's used in the encryption and decryption processes.

8. Encryptimage: This is the encrypted image obtained by performing XOR operations on PermutatedImage, output, and (ChaoticSequence * key2).

9. decryptimage: This is the decrypted image obtained by performing XOR operations on Encryptimage, output, and (ChaoticSequence * key2)

10. original_row: This is the unshuffled image obtained by using the unshuffleimg function on decryptimage. It's a 2D NumPy array representing the original image.

Data structures:

- NumPy arrays for images

- Lists for sequences and keys

2.2 SWOT Analysis

STRENGTHS	WEAKNESS
Innovative approach to encryption: The project presents a new encryption method that combines chaotic GRU maps and sine cosine, which could be a unique and effective approach to secure transport images. Tailored for transport images: The project focuses on the specific needs of transport image encryption, making it potentially more efficient and relevant to the intended application. Performance Evaluation: The inclusion of comprehensive performance analysis demonstrates a commitment to rigorous testing and validation of the proposed encryption method.	Complexity: The use of advanced techniques such as GRU and chaos-based maps can introduce complexity into the encryption process, which could be a potential weakness in terms of implementation and understanding. Resource intensive: These advanced encryption methods can be resource intensive in terms of computing power and memory, which may limit their practicality for certain systems.
OPPORTUNITIES	**THREATS**
Security Enhancement: With the growing need for secure image transmission in the field of transportation and surveillance, the project offers an opportunity to significantly increase security and privacy. Potential Applications: The encryption method developed in this project may have wider applications beyond transport images and opens the door to other areas where secure image encryption is required.	Competing Approaches: The field of image encryption is highly competitive and there may be other encryption methods that are equally or more effective, making it challenging to gain widespread adoption. Security Threats: Any encryption method is vulnerable to attack, and the project may face threats from adversaries attempting to break the encryption and compromise the security of transmission images. Technological Advances: Rapid advances in encryption technologies may render the proposed method obsolete or less effective over time.

2.3 Project Features

1. GRU and Chaos-Based Encryption Model:

   Develop a novel encryption model that combines Gated Recurrent Units (GRU) and chaos-based encryption techniques to protect transport images.

2. Chaos Generator:

   Implement a chaos generator to provide the chaotic sequences required for the encryption process.

3. Encryption Algorithm:

   Design and implement the encryption algorithm that integrates the GRU and chaos-based methods to encrypt the transport images.

4. Key Management:

Develop a secure key management system for generating and distributing encryption keys.

5. Image Preprocessing:

   Implement image preprocessing techniques such as resizing, normalization, and data augmentation as needed.

2.4 Design and implementation constraints

1. Key Management:

   The security of this encryption system heavily relies on the key used for encryption and decryption. Key management and distribution are critical aspects, and a secure method for key exchange or storage should be implemented.

2. Key Sensitivity:

   The sensitivity of the key values must be well-understood. Even a small change in the key can lead to a completely different decryption result. Proper mechanisms for key protection and recovery are essential.

3. Performance:

The algorithm may not be suitable for real-time applications or scenarios where high-speed encryption and decryption are required.The computational complexity of the algorithm should be considered, especially for large images.

4. Key Space:

   The size of the key space is important for security. A small key space makes the algorithm vulnerable to brute-force attacks. The key parameters and the number of iterations should be chosen carefully to ensure a sufficiently large key space.

5. Initialization Parameters:

   The initial values (such as 'x' and 'r' in your chaoticSequ_Sine_Cosine_Map function) may impact the security of the algorithm. Proper choices for these parameters are crucial for achieving good chaos properties.

6. Error Handling:

   The algorithm does not seem to include error detection or correction mechanisms. In practical applications, it's important to handle errors gracefully, especially when transmitting or storing encrypted data.

7. Robustness:

   The algorithm assumes that the input image is of a specific format (grayscale) and size. Adapting it to handle different image formats, sizes, and types can be challenging.

8. Algorithm Security:

   The security of this algorithm depends on the underlying chaos-based operations. The cryptographic strength and resistance to various attacks need to be thoroughly evaluated.

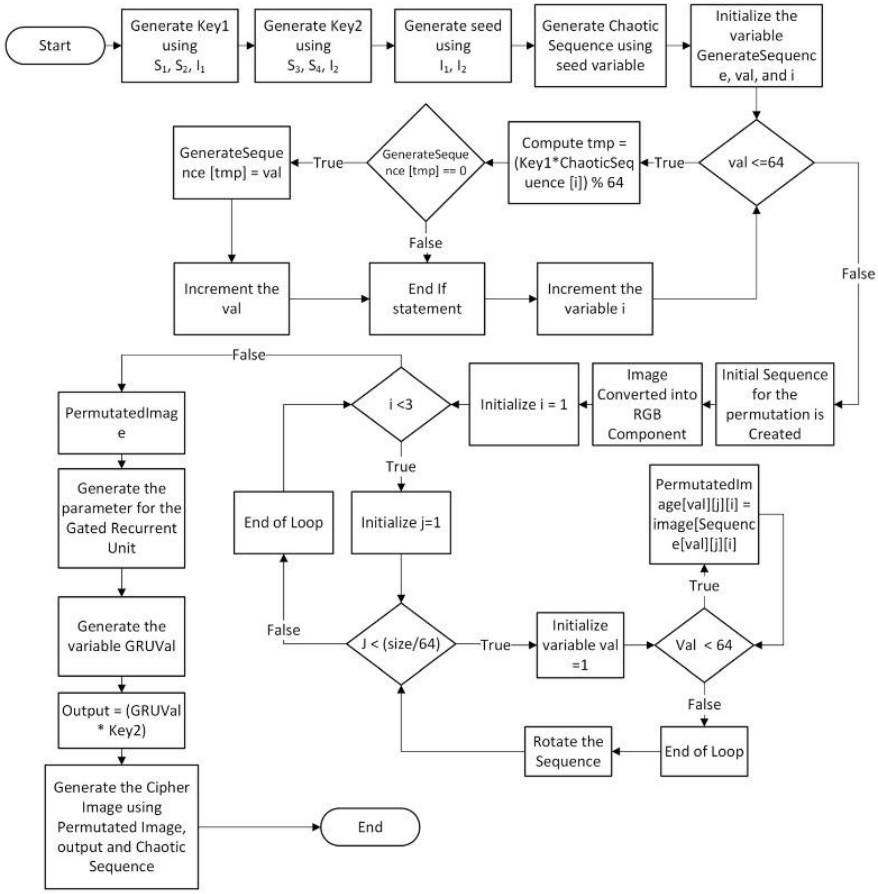
9. Testing and Validation:

   Rigorous testing and validation processes should be in place to ensure that the algorithm works as intended and that it provides the expected level of security.

10. Documentation:

    Proper documentation of the algorithm, including how it works, its parameters, and its limitations, is important for future maintenance and understanding.

2.4 Design diagram:



3. SYSTEM REQUIREMENTS

3.1 User Interface

The user interface for the project on "A GRU and chaos-based novel image encryption approach for transport images" should provide an intuitive and user-friendly experience for both encryption and decryption processes. It should include options for users to upload their image files, select encryption parameters, and initiate the encryption process. Additionally, it should offer the functionality to decrypt the encrypted images with the same parameters, ensuring a seamless user experience.

3.2 Protocols

The project aims to develop a novel image encryption approach for securing transport images using a combination of a Gated Recurrent Unit (GRU) and chaos-based techniques. This approach leverages the GRU to enhance the encryption process, introducing dynamic key

generation and stronger security measures. Chaos-based methods will be integrated to introduce unpredictability and randomness, enhancing the robustness of the encryption. The combined approach seeks to provide a highly secure and efficient solution for safeguarding transport images, ensuring the confidentiality and integrity of the data during transmission, making it suitable for applications in transportation systems, surveillance, and other sensitive domains.

4. NON-FUNCTIONAL REQUIREMENTS

4.1 Performance requirements

The following are the five basic requirements needed:

- Security
- Encryption Speed
- Encryption Efficiency
- Key Management
- Compatibility

4.2 Security requirements

Here are some of the security measures we intend to take for our project:

1. **Confidentiality:**

   Ensure that the encryption method preserves the confidentiality of transport images and prevents unauthorized access to sensitive operational data.

2. **Integrity:**

   The encryption method should guarantee the integrity of the encrypted images, meaning that the data remains unchanged during transfer or storage.

3. **Authentication:**

   Implement authentication mechanisms to verify the identity of users and devices involved in the encryption and decryption processes.

4. **Key management:**

   Establish robust key management practices to securely generate, distribute, store, and manipulate encryption keys that are critical to the security of encrypted images.

5. **Protection against attacks:**

   The encryption method should be designed to withstand various types of attacks such as brute force attacks, cryptanalysis and other known cryptographic attacks.

6. **Secure communication:**

   Ensure that data transfer between devices or systems involved in the image encryption process is secure to prevent interception and eavesdropping.

7. **User access control:**

   Implement access control mechanisms to restrict access to encrypted images to only authorized users or systems.

4.3 Software Quality Attributes

1. **Security:**

   The software should exhibit strong security attributes to ensure the confidentiality and integrity of the encrypted images, protecting against unauthorized access and attacks.

2. **Performance:**

   The software should perform encryption and decryption operations efficiently, with minimal impact on system resources. Performance attributes may include encryption speed and resource utilization.

3. **Reliability**:

   The software should be reliable, ensuring that encrypted images can be consistently decrypted without errors or data loss. It should also be robust against unexpected failures.

4. **Usability**:

   While the primary focus is on security, the software should still have a user-friendly interface for those involved in configuring and using the encryption method.

5. **Scalability**:

   The software should be designed to scale as needed, accommodating varying workloads and increased data volume without a significant decrease in performance.

APPENDIX A: GLOSSARY

KEY	A secret or parameter used in encryption and decryption processes. It is essential for converting data between its encrypted and decrypted states.
ALGORITHM PARAMETERS	The specific values or settings used in an encryption algorithm, such as key length, initialization vectors, and other configuration options.
ENTROPY	A measure of randomness or disorder in data. In encryption, higher entropy is often desirable as it indicates greater security.Pixel: The smallest unit of an image, often represented as a dot or point on a grid. Each pixel contains color or grayscale information.
CHAOTIC SYSTEM	A dynamic system that exhibits chaotic behavior, which is characterized by sensitivity to initial conditions, aperiodic and unpredictable behavior, and high complexity.GRU (Gated Recurrent Unit): A type of recurrent neural network (RNN) architecture that is used for modeling sequential data. It has gating mechanisms that help control the flow of information in and out of the network.
CRYPTANALYSIS	The process of analyzing and deciphering encrypted data without access to the encryption key, typically done by attackers trying to break the encryption.
CHAOS-BASED ENCRYPTION	An encryption approach that uses chaotic systems or chaotic processes to generate keys for encrypting and decrypting data. Chaos-based encryption is often used for its unpredictability and complexity.