**Effective Data Hiding Scheme using Machine Learning**

**INTRODUCTION**

In recent years, with the rapid development of the Internet and the rise of the Internet of Things, the transmission of information has become more frequent, and the issue of information security has also received more attention. One of the most effective ways to deal with the information security is to encrypt the relevant data [1], [2]. Generally speaking, images have a large amount of data and high redundancy, and their information value and accuracy are not as high as that of text (for example, an image in high-definition and not-so-highdefinition expresses almost the same amount of information) Therefore, ordinary image encryption generally requires less security than text encryption. The typical data encryption standard AES has high security, but this does not mean that it is universal for encryption in any situation. A good suggestion is to design a dedicated encryption algorithm according to the characteristics and application scenarios of the data. For example, it is reasonable to use lightweight encryption algorithms in restricted devices where security requirements are not too high. Likewise, encryption methods applicable to images are worth exploring. Chaos-based image encryption algorithms are considered to be very promising due to the sensitivity of chaotic systems to initial values, ergodicity and pseudo-randomness [3]. Researchers have studied chaos-based image encryption schemes from different perspectives [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. However, some cryptanalysts have demonstrated that there are obvious security problems with some of the proposed schemes [15], [16], [17], [18]. Most of the cracked algorithms are not resistant to plaintext-ciphertext pair attacks. Based on this, image encryption of key-associated plaintext is proposed, which is actually an image encryption scheme similar to the one-time pad strategy. The ideal one-time pad scheme is complete confidentiality [19]. In fact, the random key stream of arbitrary length is usually generated by the seed key through a pseudo-random generator, so this one-time pad-like encryption scheme cannot be completely kept secret. However, the security of this scheme is still very high, and we will call it a one-time pad scheme for the time being. According to Katz et al. [19], the security of one-time pad can be summarized into the following two principles: 1) Make sure that the seed key is not repeated every time it is encrypted. This is to prevent brute force attacks and plaintext-ciphertext pair attacks. 2) Make sure that the keystream generated by the seed key has sufficient randomness. The randomness of the key stream can ensure the randomness of the output ciphertext to prevent statistical attacks. Some researchers use the method of correlating plaintext to simulate the one-time pad scheme [1], [9], [10], [20], [21], [22], [23], [24], [25]. They map the seed key to the initial value of the chaotic system, and then generate unpredictable pseudorandom sequences for encryption by iterating the chaotic system. However, there are some problems with this. On the one hand, the process of mapping the seed key to the initial value of the chaotic system is not one-to-one correspondence. For instance, in formula (4) in [10], a large number of equivalent keys can be generated just by exchanging the positions of the hash values, so multiple hash values are mapped to the same initial value of the chaotic system, which greatly reduces the key space. In addition, for example [1], [10], [20] encrypts the same image with the same key, which does not conform to the idea of one-time pad (This doesn’t mean they are not secure, but since the key for each encryption needs to be secretly transmitted to the decrypting party, why not take a closer approach to the one-time pad?). On the other hand, although chaotic systems are pseudorandom, unpredictable. However, in finite precision devices, some literatures show that low-dimensional chaotic systems are prone to dynamic degradation [13], [26]. This makes the generated keystream predictable, which in turn leads to poor cryptographic statistical properties of the ciphertext. Some researchers solve this problem by using higher-dimensional, more complex chaotic systems [1], [10], however, this greatly increases the amount of computation and reduces the speed of encryption.

**1.1 Objective of the project:**

In recent years, chaotic image encryption algorithms with key and plaintext association have been developed, which are essentially similar to a one-time pad at a time because each encryption requires the transmission of the key. However, some existing schemes cannot uniquely map the seed key to the initial value of the chaotic system, which leads to the reduction of the key space of the encryption system. In addition, some schemes use the same key to encrypt the same image, which does not conform to the one-time pad strategy. This paper solves these problems from two aspects. On the one hand, random pixels are inserted into a plain image and then a hash value is generated using SHA-256. Different seed keys can be obtained even if the same image is encrypted. On the other hand, the Sequential Expansion Algorithm (SEA) and Feedback Iterative Piece-Wise Linear Chaotic Mapping (FI-PWLCM) are proposed to realize the one-to-one correspondence between the seed key and the encrypted key stream. SEA can quickly generate seed key sensitive and random sequences. FI-PWLCM achieves one-to-one correspondence with the seed key through feedback iteration with more control parameters. The mapping not only has the rapidity of PWLCM, but also can produce more complex chaotic sequences. Besides, this paper proposes a Segmented Coordinate Descent (SCD) method for histogram statistical optimization of images to improve the ability of cryptosystems against statistical attacks. Experiments and security analysis show that the algorithm can resist chosen-plaintext (chosen-ciphertext) attacks, brute force attacks, statistical attacks and so on. Compared with most current algorithms, it achieves the best performance in the statistical properties of histogram and entropy.

**2. LITERATURE SURVEY:**

**‘‘An image encryption algorithm based on a hidden attractor chaos system and the Knuth–Durstenfeld algorithm,’’**

Chaotic systems have been widely applied in digital image encryption due to their complex properties such as ergodicity, pseudo randomness and extreme sensitivity to their initial values and parameters. An image encryption algorithm based on a hidden attractor chaos system and Knuth–Durstenfeld algorithm is proposed. First, a hidden attractor chaos system is used to encrypt digital image. Compared to a self-excited attractor, the hidden attractor's attracting basin does not [intersect](https://www.sciencedirect.com/topics/mathematics/intersects) with any small neighbourhoods of the equilibria. It is difficult for attackers to reconstruct the attractor by finding equilibrium points. Therefore, the hidden attractor chaotic system is difficult to decrypt. Meanwhile, the hidden attractor chaos system is very sensitive to initial values and parameters. Second, the Knuth–Durstenfeld algorithm has good randomness. In addition, the Knuth–Durstenfeld algorithm can reduce the time complexity and the space complexity of the permutation while achieving good permutation effects. Thus, Knuth–Durstenfeld algorithm is used to permutate the digital image. Finally, DNA sequence operations are used to diffuse image pixels values. Some experimental analyses have been applied to measure the new scheme, and the experimental results illustrate the scheme possesses better encryption performances. This method can be applied in secure image communication fields.

**‘‘An image encryption algorithm based on chaotic system and compressive sensing,’’**

In this paper, we propose an [image encryption algorithm](https://www.sciencedirect.com/topics/computer-science/image-encryption-algorithm) based on the memristive chaotic system, elementary cellular automata (ECA) and [compressive sensing](https://www.sciencedirect.com/topics/computer-science/compressive-sensing) (CS). Firstly, the original image is transformed by discrete wavelet transform, and the sparse coefficient matrix is obtained. Next, a zigzag scrambling method and the ECA are adopted to scramble the sparse coefficient matrix successively, and this process may effectively improve the scrambling degree. And then, the measurement matrix produced by the memristive chaotic system is used to compress and perceive the scrambled image, and the final cipher image is obtained. In addition, SHA-512 hash function value of the original image is generated to calculate the parameters for zigzag confusion, the initial values of the chaotic system and the initial configurations of the ECA, which enhances the correlation between the algorithm and the plain image and makes the proposed encryption scheme resist the known-plaintext and chosen-plaintext attacks. Moreover, our algorithm can compress and encrypt the image simultaneously by use of CS, which may reduce the amount of data and storage space. Simulation results and performance analyses demonstrate the security and robustness of the proposed scheme.

**‘‘DNA chaos blend to secure medical privacy,’’**

In this technological era, it is highly essential to protect the digital medical data from the fraud and forgery as they are transmitted over the public channel. Also with the increased data traffic, it is hard to transmit the entire bulky medical data. New methods have come into the scene to reduce the traffic while maintaining the sufficient level of security. Partial encryption is one of the methods which selectively encrypt the bulky medical image. Meanwhile, if the same medical image is needed to be reused for another diagnosis, then it is recommended to protect the entire medical image. This paper proposes a hybrid encryption scheme based on deoxyribo nucleic acid and chaotic maps, which can be adaptable for both selective and full medical image encryption. The proposed algorithm uses multiple chaotic maps in single process to generate the highly random keys for encrypting the color digital imaging and communications in medicine image. The algorithm comprises three phases, namely, permutation, encoding, and diffusion. In all the phases, the selection of specific rule set depends on the key sequences produced from the combined chaotic system. Experimental results are carried out to validate the resistance of the developed algorithm toward statistical, differential, and brute force attacks.

**‘‘A symmetric image encryption scheme based on 3D chaotic cat maps,’’**

Encryption of images is different from that of texts due to some intrinsic features of images such as bulk data capacity and high redundancy, which are generally difficult to handle by traditional methods. Due to the exceptionally [desirable properties](https://www.sciencedirect.com/topics/computer-science/desirable-property) of mixing and sensitivity to initial conditions and parameters of [chaotic maps](https://www.sciencedirect.com/topics/computer-science/chaotic-map), chaos-based encryption has suggested a new and efficient way to deal with the intractable problem of fast and highly secure [image encryption](https://www.sciencedirect.com/topics/computer-science/image-encryption). In this paper, the two-dimensional chaotic cat map is generalized to 3D for designing a real-time secure symmetric [encryption scheme](https://www.sciencedirect.com/topics/computer-science/encryption-scheme). This new scheme employs the 3D cat map to shuffle the positions (and, if desired, grey values as well) of image pixels and uses another chaotic map to confuse the relationship between the cipher-image and the plain-image, thereby significantly increasing the resistance to statistical and differential attacks. Thorough experimental tests are carried out with detailed analysis, demonstrating the high security and fast encryption speed of the new scheme.

**‘‘Image encryption using chaotic logistic map,’’**

In recent years, the chaos based [cryptographic algorithms](https://www.sciencedirect.com/topics/computer-science/cryptographic-algorithm) have suggested some new and efficient ways to develop secure [image encryption](https://www.sciencedirect.com/topics/computer-science/image-encryption) techniques. In this communication, we propose a new approach for image encryption based on chaotic [logistic maps](https://www.sciencedirect.com/topics/mathematics/logistic-map) in order to meet the requirements of the secure image transfer. In the proposed image [encryption scheme](https://www.sciencedirect.com/topics/computer-science/encryption-scheme), an external secret key of 80-bit and two chaotic logistic maps are employed. The initial conditions for the both logistic maps are derived using the external secret key by providing different weightage to all its bits. Further, in the proposed [encryption process](https://www.sciencedirect.com/topics/computer-science/encryption-process), eight different types of operations are used to encrypt the pixels of an image and which one of them will be used for a particular pixel is decided by the outcome of the logistic map. To make the cipher more robust against any attack, the secret key is modified after encrypting each block of sixteen pixels of the image. The results of several experimental, statistical analysis and key sensitivity tests show that the proposed image encryption scheme provides an efficient and secure way for real-time image encryption and transmission.

**3. SYSTEM ANALYSIS**

**3.1 Existing System**

To secure data in network Chaotic Image Encryption is more in demand which will encrypt image based on a block cipher is to use a chaotic map to generate secret keys or chaotic sequences to substitute and diffuse the pixels or bits of the image to achieve a higher level of security. In the past many algorithms introduced which cannot uniquely map the seed key to the initial value of the chaotic system, which leads to the reduction of the key space of the encryption system.

**Disadvantages**

1. Less accuracy.

2. Time Taking processes

**3.2 PROPOSED SYSTEM**

To overcome from above issue author of this paper introducing Novel Chaotic image encryption scheme which is a combination of multiple algorithms such as SEA (Sequential Expansion Algorithm), FI-PWLCM (Feedback Iterative Piece-Wise Linear Chaotic Mapping) and SCD (Segmented Coordinate Descent).

Above scheme will execute all 3 algorithms step by step

1. First image will be taken as input and then convert image to 1 dimension vector
2. 10 random pixels will be added to end of the 1 dimension vector
3. SEA and FI-PWLCM will be applied. SEA is a simple and fast pseudo-random sequence generator. FI-PWLCM, which has more control parameters than PWLCM and can generate more complex chaotic sequences
4. SCD will be applied to convert high dimensional optimization problem to low dimensional optimization to reduce key space.

**Advantages**

1. High Accuracy
2. Less Time Taken

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

1. Feasibility Study
2. TEAM FORMATION
3. Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

  
When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behaviour of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analysing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development lifecycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* Business requirements describe in business terms what must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web-based user interface for audit workflow at NIC-CSD. Thus, it provides an easy access to. the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**SYSTEM REQUIREMENT:**

**HARDWARE REQUIREMENTS:**

# Processor - Intel i3(min)

* Speed - 1.1 GHz
* RAM - 4GB(min)
* Hard Disk - 500 GB

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows10(min)
* Programming Language - Python

**4. SYSTEM DESIGN**

**UML Diagram:**

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

* + **User Model View**
    1. This view represents the system from the users perspective.
    2. The analysis representation describes a usage scenario from the end-users perspective.
  + **Structural Model view**
    1. In this model the data and functionality are arrived from inside the system.
    2. This model view models the static structures.
* **Behavioural Model View**

It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

* **Implementation Model View**

In this the structural and behavioural as parts of the system are represented as they are to be built.

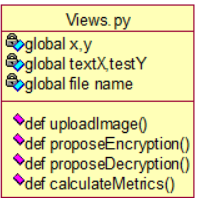
* **Environmental Model View**

In this the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

**Class Diagram:**

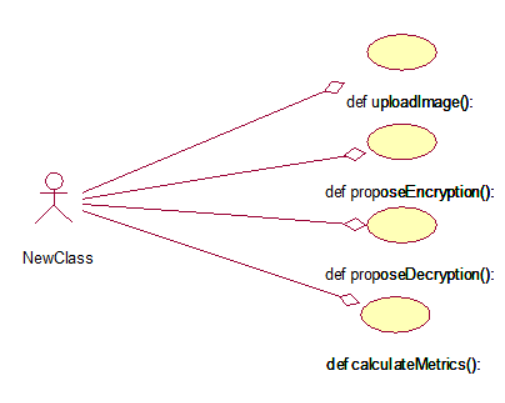
The class diagram is the main building block of object oriented modelling. It is used both for general conceptual modelling of the systematic of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modelling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake.



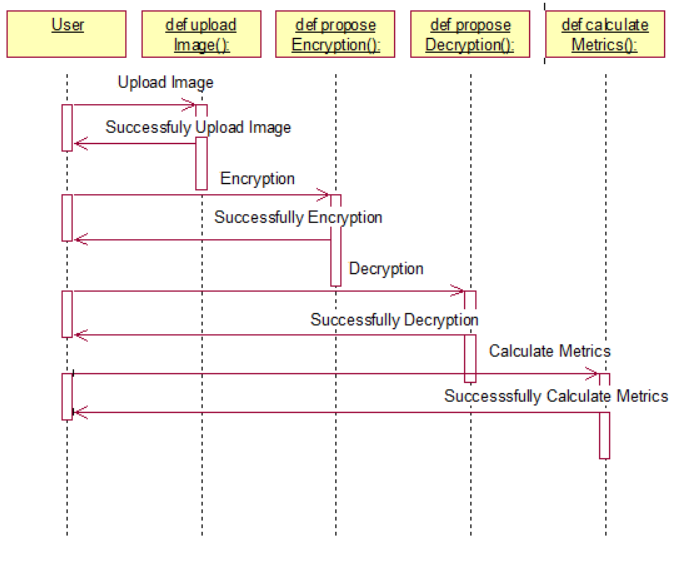
**Use case Diagram:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

****

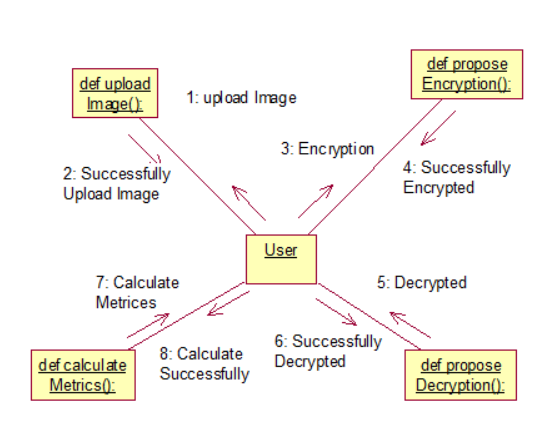
**Sequence diagram:**

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



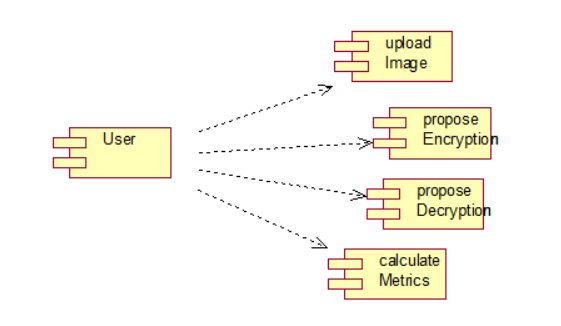
**Collaboration diagram:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behaviour of a system.



**Component Diagram:** In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

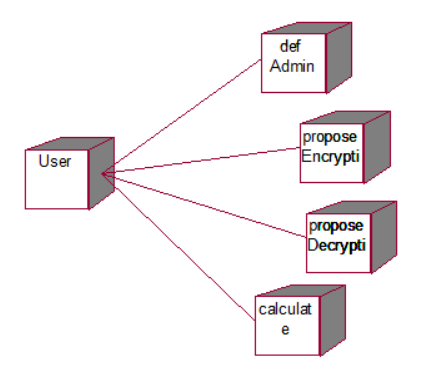
Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.

****

**Deployment Diagram:**

A **deployment diagram** in the Unified Modelling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.



**Activity Diagram:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another

activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent

**Upload Image**

**Propose Encryption**

**Propose Decryption**

**Calculate Metrics**

**Data Flow Diagram:**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.

User

1. upload Image 2. Successfully upload Image

3. propose Encryption 4.Successfully propose Encryption

5. propose Decryption 6. Successfully propose Decryption

7. Calculate Metrics 8. Successfully calculate Metrics

**5. IMPLEMETATION**

**5.1 PYTHON**

\* One of the most popular languages is Python. Guido van Rossum released this language in 1991. Python is available on the Mac, Windows, and Raspberry Pi operating systems. The syntax of Python is simple and identical to that of English. When compared to Python, it was seen that the other language requires a few extra lines.

\*It is an interpreter-based language because code may be run line by line after it has been written. This implies that rapid prototyping is possible across all platforms. Python is a big language with a free, binary-distributed interpreter standard library.

\* It is inferior to maintenance that is conducted and is straightforward to learn. It is an object-oriented, interpreted programming language. It supports several different programming paradigms in addition to object-oriented programming, including functional and procedural programming.

\* It supports several different programming paradigms in addition to object-oriented programming, including practical and procedural programming. Python is mighty while maintaining a relatively straightforward syntax. Classes, highly dynamic data types, modules, and exceptions are covered. Python can also be utilised by programmes that require programmable interfaces as an external language.

Here are some key features and characteristics of Python:

* Readability: Python emphasizes code readability with its clean and intuitive syntax. It uses indentation and whitespace to structure code blocks, making it easy to understand and maintain.
* Easy to Learn: Python's simplicity and readability make it an excellent choice for beginners. Its straightforward syntax and extensive documentation make it accessible for newcomers to programming.
* Interpreted Language: Python is an interpreted language, meaning that it doesn't need to be compiled before running. The Python interpreter reads and executes the code directly, making the development process faster and more interactive.
* Cross-platform Compatibility: Python is available for major operating systems like Windows, macOS, and Linux. This cross-platform compatibility allows developers to write code once and run it on different platforms without modifications.
* Large Standard Library: Python comes with a vast standard library that provides ready-to-use modules and functions for various tasks. It covers areas such as file I/O, networking, regular expressions, databases, and more, saving developers time and effort.
* Extensible and Modular: Python supports modular programming, enabling developers to organize code into reusable modules and packages. Additionally, Python allows integrating modules written in other languages, such as C or C++, providing flexibility and performance optimizations.
* Wide Range of Libraries and Frameworks: Python has a vibrant ecosystem with numerous third-party libraries and frameworks. These libraries, such as NumPy, pandas, TensorFlow, and Django, extend Python's capabilities for specific domains, making it a powerful tool for diverse applications.
* Object-Oriented: Python supports object-oriented programming (OOP) principles, allowing developers to create and work with classes and objects. OOP provides a structured approach to code organization, promoting code reuse and modularity.
* Dynamic Typing: Python is dynamically typed, meaning variable types are determined at runtime. Developers do not need to declare variable types explicitly, which enhances flexibility and simplifies code writing.

**5.2 Installation**

To install Python on your computer, follow these basic steps:

* Step 1: Visit the Python website Go to the official Python website at <https://www.python.org/>.
* Step 2: Select the operating system Choose the appropriate installer for your operating system. Python supports Windows, macOS, and various Linux distributions. Make sure to select the correct version that matches your operating system.
* Step 3: Check which version of Python is installed; if the 3.7.0 version is not there, uninstall it through the control panel and
* Step 4: Install Python 3.7.0 using Cmd.
* Step 5: Install the all libraries that required to run the project
* Step 6: Run

**5.3 Python Features:**

1. **Easy:** Because Python is a more accessible and straightforward language, Python programming is easier to learn.
2. **Interpreted language:** Python is an interpreted language, therefore it can be used to examine the code line by line and provide results.
3. **Open Source:** Python is a free online programming language since it is open-source.
4. **Portable:** Python is portable because the same code may be used on several computer standard
5. **libraries:** Python offers a sizable library that we may utilize to create applications quickly.
6. **GUI:** It stands for GUI (Graphical User Interface)
7. **Dynamical typed:** Python is a dynamically typed language, therefore the type of the value will be determined at runtime.

**5.4 Python GUI (Tkinter)**

* Python provides a wide range of options for GUI development (Graphical User Interfaces).
* Tkinter, the most widely used GUI technique, is used for all of them.
* The Tk GUI toolkit offered by Python is used with the conventional Python interface.
* Tkinter is the easiest and quickest way to write Python GUI programs.
* Using Tkinter, creating a GUI is simple.
* A part of Python's built-in library is Tkinter. The GUI programs were created.
* Python and Tkinter together give a straightforward and quick way. The Tk GUI toolkit's object-oriented user interface is called Tkinter.

Making a GUI application is easy using Tkinter. Following are the steps:

1) Install the Tkinter module in place.

2) The GUI applicatioMakeske the primary window

3) Include one or more of the widgets mentioned above in the GUI application.

4) Set up the main event loop such that it reacts to each user-initiated event.

Although Tkinter is the only GUI framework included in the Python standard library, Python includes a GUI framework. The default library for Python is called Tkinter. Tk is a scripting language often used in designing, testing, and developing GUIs. Tk is a free, open-source widget toolkit that may be used to build GUI applications in a wide range of computer languages.

**5.5 Python IDLE**

* Python IDLE offers a full-fledged file editor, which gives you the ability to write and execute Python programs from within this program. The built-in file editor also includes several features, like code completion and automatic indentation, that will speed up your coding workflow.
* Guido Van Rossum named Python after the British comedy group Monty Python while the name IDLE was chosen to pay tribute to Eric Idle, who was one of the Monty Python's founding members. IDLE comes bundled with the default implementation of the Python language since the 01.5. 2b1 release
* IDLE is used to execute statements similar to Python Shell. IDLE is used to create, modify, and execute Python code. IDLE provides a fully-featured text editor to write Python scripts and provides features like syntax highlighting, auto-completion, and smart indent.
* IDLE has two modes: interactive and script. We wrote our first program, “Hello, World!” in interactive mode. Interactive mode immediately returns the results of commands you enter into the shell. In script mode, you will write a script and then run it.
* The IDE Python IDLE is a good place to start as it helps you become familiar with the way Python works and understand its syntax. This IDE is good to start programming in Python due to its great debugger, but once you are fluent and start developing projects it is necessary to jump to another, more complete IDE.
* Python IDLE (Integrated Development and Learning Environment) is an interactive development environment included with the Python programming language. It provides a convenient way to write, execute, and debug Python code.

When you install Python, IDLE is typically installed along with it. To open IDLE, you can follow these steps:

* Open the command prompt (Windows) or terminal (macOS/Linux).
* Type "idle" and press Enter. Alternatively, you can specify the version with "idle3" or "idle2" for Python 3 or Python 2, respectively.
* Once IDLE is launched, you will see the Python shell, which is an interactive environment where you can type and execute Python code directly.

Here are some features and functionalities provided by Python IDLE:

* Editor: IDLE includes a text editor where you can write your Python code. It offers syntax highlighting, automatic indentation, and code completion to enhance your coding experience.
* Interactive Shell: The Python shell in IDLE allows you to execute Python code interactively. You can type commands, statements, or function calls directly in the shell, and Python will execute them immediately.
* Debugging: IDLE provides basic debugging capabilities to help you find and fix errors in your code. You can set breakpoints, step through code, inspect variables, and track the program's execution.
* Python Help: IDLE provides access to the Python documentation and built-in help. You can access the help menu to find information about Python modules, functions, classes, and more.
* Script Execution: In addition to the interactive shell, IDLE allows you to run Python scripts stored in files. You can write your code in the editor and execute it as a script to see the output or interact with the program.
* Customization: IDLE can be customized to suit your preferences. You can modify settings related to syntax highlighting, indentation, fonts, and more.
* Python IDLE serves as a beginner-friendly development environment and learning tool. It is suitable for writing small scripts, testing code snippets, experimenting with Python features, and learning the language's basics. However, for more advanced development projects, you may consider using other code editors or integrated development environments (IDEs) that provide additional features and better project management capabilities.

**5.6 Libraries**

In Python, libraries (also referred to as modules or packages) are collections of pre-written code that provide additional functionality and tools to extend the capabilities of the Python language. Libraries contain reusable code that developers can leverage to perform specific tasks without having to write everything from scratch.

Python libraries are designed to solve common problems, such as handling data, performing mathematical operations, interacting with databases, working with files, implementing networking protocols, creating graphical user interfaces (GUIs), and much more. They provide ready-to-use functions, classes, and methods that simplify complex operations and save development time.

**Libraries in Python offer various advantages:**

* Code Reusability:
* Efficiency:
* Collaboration
* Domain-Specific Functionality
* To use a Python library, you need to install it first.

There are some libraries following:

* **Pandas:**

Pandas are a Python computer language library for data analysis and manipulation. It offers a specific operation and data format for handling time series and numerical tables. It differs significantly from the release3-clause of the BSD license. It is a well-liked open-source of opinion that is utilized in machine learning and data analysis.

Pandas are a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Pandas are a Python library used for working with data sets.

* It has functions for analysing, cleaning, exploring, and manipulating data.
* The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.
* Pandas allow us to analyse big data and make conclusions based on statistical theories.
* Pandas can clean messy data sets, and make them readable and relevant.

Relevant data is very important in data science. Pandas are a Python library for data analysis. Started by Wes McKinney in 2008 out of a need for a powerful and flexible quantitative analysis tool, pandas have grown into one of the most popular Python libraries. It has an extremely active community of contributors. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself.

* **NumPy:**

The NumPy Python library for multi-dimensional, big-scale matrices adds a huge number of high-level mathematical functions. It is possible to modify NumPy by utilizing a Python library. Along with line, algebra, and the Fourier transform operations, it also contains several matrices-related functions.

NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

* NumPy is a Python library used for working with arrays.
* It also has functions for working in domain of linear algebra, Fourier transform, and matrices.
* NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
* NumPy stands for Numerical Python.
* In Python we have lists that serve the purpose of arrays, but they are slow to process.
* NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
* The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
* Arrays are very frequently used in data science, where speed and resources are very important.
* **Matplotlib:**

It is a multi-platform, array-based data visualization framework built to interact with the whole SciPy stack. MATLAB is proposed as an open-source alternative. Matplotlib is a Python extension and a cross-platform toolkit for graphical plotting and visualization.

Matplotlib is a popular Python library for creating static, animated, and interactive visualizations. It provides a flexible and comprehensive set of tools for generating plots, charts, histograms, scatter plots, and more. Matplotlib is widely used in various fields, including data analysis, scientific research, and data visualization.

Here are some key features and functionalities of the Matplotlib library:

* Plotting Functions
* Customization Options
* Multiple Interfaces
* Integration with NumPy and pandas
* Subplots and Figures:
* Saving and Exporting
* **Scikit-learn:**

The most stable and practical machine learning library for Python is scikit-learn. Regression, dimensionality reduction, classification, and clustering are just a few of the helpful tools it provides through the Python interface for statistical modeling and machine learning. It is an essential part of the Python machine learning toolbox used by JP Morgan. It is frequently used in various machine learning applications, including classification and predictive analysis.

Scikit-learn (also referred to as sklearn) is a widely used open-source machine learning library for Python. It provides a comprehensive set of tools and algorithms for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and pre-processing.

Here are some key features and functionalities of the Scikit-learn library:

* Easy-to-Use Interface:
* Broad Range of Algorithms:
* Data Pre-processing and Feature Engineering:
* Model Evaluation and Validation:
* Integration with NumPy and pandas:
* Robust Documentation and Community Support:
* **Keras:**

\* Google's Keras is a cutting-edge deep learning API for creating neural networks. It is created in Python and is designed to simplify the development of neural networks. Additionally, it enables the use of various neural networks for computation. Deep learning models are developed and tested using the free and open-source Python software known as Keras.

Keras is a high-level deep learning library for Python. It is designed to provide a user-friendly and intuitive interface for building and training deep learning models. Keras acts as a front-end API, allowing developers to define and configure neural networks while leveraging the computational backend engines, such as Tensor Flow or Theano.

Here are some key features and functionalities of the Keras library:

* User-Friendly API
* Multi-backend Support
* Wide Range of Neural Network Architectures
* Pre-trained Models and Transfer Learning:
* Easy Model Training and Evaluation:
* GPU Support:
* **h5py:**

\* The h5py Python module offers an interface for the binary HDF5 data format. Thanks to p5py, the top can quickly halt the vast amount of numerical data and alter it using the NumPy library. It employs common syntax for Python, NumPy, and dictionary arrays.

h5py is a Python library that provides a simple and efficient interface for working with datasets and files in the Hierarchical Data Format 5 (HDF5) format. HDF5 is a versatile data format commonly used for storing and managing large volumes of numerical data.

Here are some key features and functionalities of the h5py library:

* + HDF5 File Access
  + Dataset Handling:
  + Group Organization:
  + Attributes:
  + Compatibility with NumPy
  + Performance
* **Tensor flow**

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow is an end-to-end open source platform for machine learning. TensorFlow is a rich system for managing all aspects of a machine learning system; however, this class focuses on using a particular TensorFlow API to develop and train machine learning models.

TensorFlow is a popular open-source library for machine learning and deep learning. It provides a comprehensive set of tools, APIs, and computational resources for building and training various types of machine learning models, especially neural networks.

Here are some key features and functionalities of TensorFlow:

* Neural Network Framework:
* Computational Graphs
* Automatic Differentiation
* GPU and TPU Support
* Distributed Computing
* Deployment Capabilities
* **Tkinter**

Tkinter is an acronym for "Tk interface". Tk was developed as a GUI extension for the Tcl scripting language by John Ousterhout. The first release was in 1991. Tkinter is the de facto way in Python to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions. In fact, it's the only framework built into the Python standard library.

Tkinter is a standard Python library used for creating graphical user interfaces (GUIs). It provides a set of modules and classes that allow you to develop interactive and visually appealing desktop applications.

Here are some key features and functionalities of Tkinter:

* Cross-Platform Compatibility
* Simple and Easy-to-Use
* Widgets and Layout Management
* Event-Driven Programming
* Customization and Styling
* Integration with Other Libraries
* **NLTK**

NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc NLTK (Natural Language Toolkit) is the go-to API for NLP (Natural Language Processing) with Python. It is a really powerful tool to pre-process text data for further analysis like with ML models for instance. It helps convert text into numbers, which the model can then easily work with.

NLTK (Natural Language Toolkit) is a Python library widely used for working with human language data and implementing natural language processing (NLP) tasks. It provides a set of tools, corpora, and resources for tasks such as tokenization, stemming, tagging, parsing, sentiment analysis, and more.

Here are some key features and functionalities of NLTK:

* Text Processing
* Part-of-Speech Tagging
* Named Entity Recognition
* Chunking and Parsing
* Sentiment Analysis:
* WordNet Integration:
* **Scipy**

SciPy is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data.

SciPy is a powerful scientific computing library for Python that provides a wide range of mathematical algorithms and functions. It builds upon NumPy, another fundamental library for numerical computing, and extends its capabilities by adding additional tools for scientific and technical computing tasks.

Here are some key features and functionalities of SciPy:

* Numerical Integration:
* Optimization and Root Finding
* Linear Algebra
* Signal and Image Processing
* Statistics

**5.2 Sample Code:**

from tkinter import \*

import tkinter

from tkinter import filedialog

import numpy as np

from tkinter import simpledialog

from tkinter import ttk

from tkinter.filedialog import askopenfilename

import os

import matplotlib.pyplot as plt

import cv2

import timeit

import random

import hashlib

import math

import matplotlib.pyplot as plt

from SCD import \*

from math import log10, sqrt

from skimage.metrics import structural\_similarity as ssim

main = tkinter.Tk()

main.title("A Novel Chaotic Image Encryption Algorithm Based on Coordinate Descent and SHA-256") #designing main screen

main.geometry("1000x650")

global filename, w, h, channel

encryption\_done = 0

def uploadImage():

global filename

filename = filedialog.askopenfilename(initialdir="SampleImages")

text.delete('1.0', END)

text.insert(END,os.path.basename(filename)+" image loaded\n\n")

def getSCDOptimization():

x\_min = np.zeros(80, dtype=int) # All zeros

x\_max = np.full(80, 255, dtype=int) # All 255s

x\_initial\_scd = 0.5 #chi\_square\_stat # Given initial value which is chi square value

accuracy = 1 # Desired accuracy for stopping condition

m = 80 # Number of coordinates

max\_steps = 3 \* m # Maximum number of steps

# Run Segmented Coordinate Descent with floored values

x\_optimized, y\_optimized = SCD(x\_min, x\_max, x\_initial\_scd, accuracy, max\_steps)

return x\_optimized, y\_optimized

#generate FIPWLCM K one-to-one correspondence with the chaotic sequences

def getFIPWLCM(k, length\_of\_k):

start = 0

end = 1

c = 0

v1 = 0

v2 = 0

while start < length\_of\_k - 1:

v1 += k[start] ^ k[end]

start = end

end += 1

if end == 32:

end = 0

v2 += k[start] ^ k[end]

start += 1

end += 1

c = (v1 % 127) + (v2 % 128)

ki = []

for i in range(0, length\_of\_k):

ki.append((k[i] + c) % 256)

return ki

def proposeEncryption():

text.delete('1.0', END)

global filename, w, h, channel, encryption\_done

encryption\_done = 1

img = cv2.imread(filename) #read image

w, h, channel = img.shape

I0 = img.ravel() #3 dimensional image to one dimension

l = len(I0)#get length of one dimension vector

for i in range(0, 10): #add 10 random pixels to form NI0

random\_pixels = random.randint(np.min(I0), np.max(I0))

if i == 0:

NI0 = np.append(I0, random\_pixels)

else:

NI0 = np.append(NI0, random\_pixels)

x\_optimized, y\_optimized = getSCDOptimization() #get SCD optimization to convert high dimensional optimization to low dimension optimization

NI0 = NI0.tobytes()

#execute SEA and FI-PWLCM sequence generation and

k = hashlib.sha256(NI0).digest() #get 32 bytes hash value

temp = []

for i in range(len(k)):

temp.append(k[i])

k = np.asarray(temp)

k = np.minimum(k, x\_optimized)#convert K high dimension to SCD low dimension

length\_of\_k = len(k)

ki = getFIPWLCM(k, length\_of\_k)#get SEA and FIPWLCM sequences which can be used to generate K key value

K = []

i=0

start = 1

while i < l:

A = ki[1:start] #get A and B values

B = ki[start+1:length\_of\_k]

i += 1

C = 0

start += 1

if start == 32:

start = 1

for j in range(0, (min(len(A), len(B)))):#apply XOR or A and B values to generate C sequences

C += A[j] ^ B[j]

K.append(C) #add C sequences to K to form image encryption key

encrypted = []

for i in range(0, len(I0)):

encrypted.append(I0[i] ^ K[i])

encrypted = np.asarray(encrypted)

encrypted = np.reshape(encrypted, (w, h, channel))

np.save("EncryptedImages/"+os.path.basename(filename), encrypted)

keys = [K, encrypted, w, h, channel]

keys = np.asarray(keys)

np.save("keys/"+os.path.basename(filename), keys)

#encrypted = np.load("EncryptedImages/"+os.path.basename(filename)+".npy")

figure, axis = plt.subplots(nrows=1, ncols=2,figsize=(10,10))

axis[0].set\_title("Original Image")

axis[1].set\_title("Encrypted Image")

axis[0].imshow(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB))

axis[1].imshow(encrypted)

figure.tight\_layout()

plt.show()

#function to calculate SSIM between original and super predicted image

def getimageSSIM(original, decrypted):

original1 = cv2.cvtColor(original, cv2.COLOR\_BGR2GRAY)

decrypted1 = cv2.cvtColor(decrypted, cv2.COLOR\_BGR2GRAY)

ssim\_value = ssim(original1, decrypted1, data\_range = decrypted1.max() - decrypted1.min())

return ssim\_value

#function to calculate PSNR between original and super\_image

def getPSNR(original, decrypted):

mse = np.mean((original - decrypted) \*\* 2)

if(mse == 0):

return 0

max\_pixel = 255.0

psnr = 20 \* log10(max\_pixel / sqrt(mse))

return psnr

def proposeDecryption():

global encryption\_done, filename

text.delete('1.0', END)

encrypted\_filename = filedialog.askopenfilename(initialdir="EncryptedImages")

keys = np.load("keys/"+os.path.basename(encrypted\_filename), allow\_pickle=True)

K, encrypted, w, h, channel = keys[0], keys[1], keys[2], keys[3], keys[4]

encrypted\_image = np.load(encrypted\_filename)

I0 = encrypted.ravel()

decrypted = []

for i in range(0, len(I0)):

decrypted.append(I0[i] ^ K[i])

decrypted = np.reshape(decrypted, (w, h, channel))

cv2.imwrite("decrypt.png", decrypted)

if encryption\_done == 1:

original\_img = cv2.imread(filename)

decrypt\_img = cv2.imread('decrypt.png')

psnr = getPSNR(original\_img, decrypt\_img)

ssim = getimageSSIM(original\_img, decrypt\_img)

text.insert(END,'Propose PSNR : '+str(psnr)+"\n")

text.insert(END,'Propose SSIM : '+str(ssim)+"\n")

text.update\_idletasks()

encryption\_done = 0

figure, axis = plt.subplots(nrows=1, ncols=2,figsize=(10,10))

axis[0].set\_title("Encrypted Image")

axis[1].set\_title("Decrypted Image")

axis[0].imshow(encrypted\_image)

axis[1].imshow(cv2.cvtColor(cv2.imread('decrypt.png'), cv2.COLOR\_BGR2RGB))

figure.tight\_layout()

plt.show()

def calculateMetrics():

text.delete('1.0', END)

image1 = cv2.imread("experiment/encrypted.png")

image2 = cv2.imread("experiment/encrypted1.png")

npcr\_value = npcr(image1, image2)

uaci\_value = uaci(image1, image2)

text.insert(END,"Propose NPCR : "+str(npcr\_value)+"\n")

text.insert(END,"Propose UACI : "+str(uaci\_value)+"\n\n")

font = ('times', 16, 'bold')

title = Label(main, text='A Novel Chaotic Image Encryption Algorithm Based on Coordinate Descent and SHA-256', justify=LEFT)

title.config(bg='lavender blush', fg='DarkOrchid1')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=100,y=5)

title.pack()

font1 = ('times', 13, 'bold')

uploadButton = Button(main, text="Upload Sample Image", command=uploadImage)

uploadButton.place(x=10,y=100)

uploadButton.config(font=font1)

secretButton = Button(main, text="Run Propose Encryption", command=proposeEncryption)

secretButton.place(x=250,y=100)

secretButton.config(font=font1)

encryptButton = Button(main, text="Run Propose Decryption", command=proposeDecryption)

encryptButton.place(x=520,y=100)

encryptButton.config(font=font1)

exitButton = Button(main, text="Calculate NPCR & UACI Values", command=calculateMetrics)

exitButton.place(x=10,y=150)

exitButton.config(font=font1)

font1 = ('times', 12, 'bold')

text=Text(main,height=20,width=120)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=200)

text.config(font=font1)

main.config(bg='light coral')

main.mainloop()

**6. TESTING**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

## **Implementation**

## The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## **Testing**

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

### **System Testing**

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

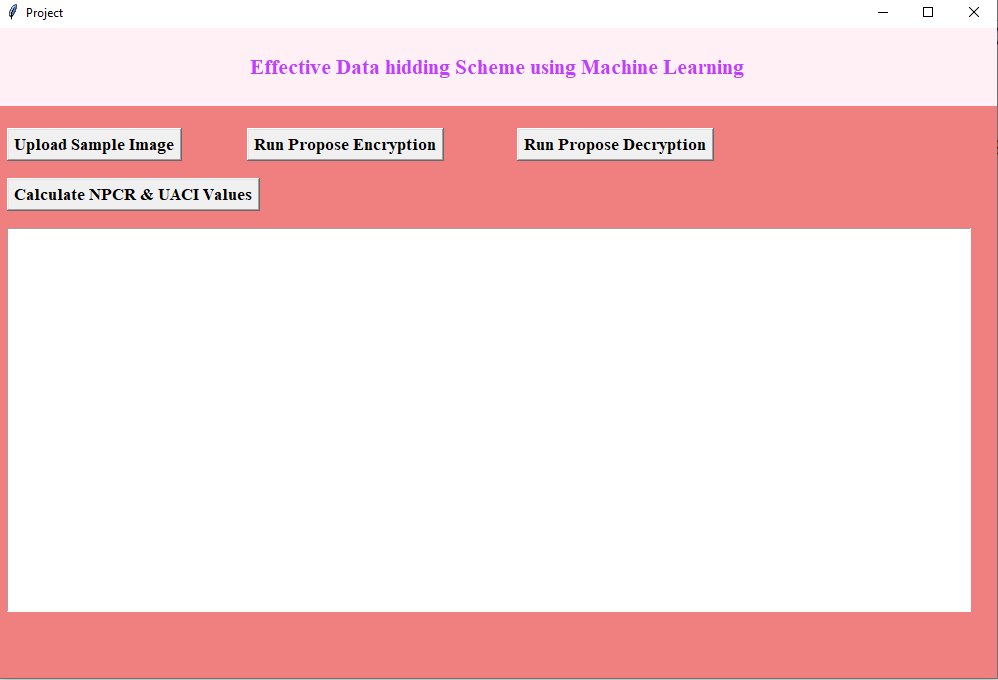
After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

**Acceptance Testing**

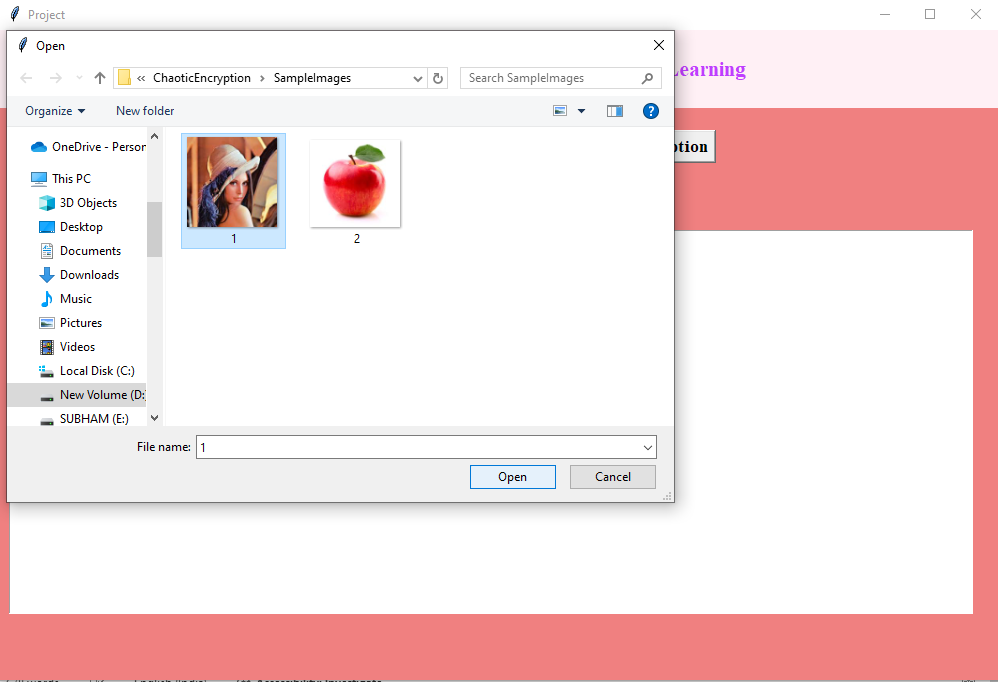
When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | **Actual** |
| 01 | Upload Image | Test whether Image upload or not upload into the system | If the image may not uploaded | We cannot do further operations | If image uploaded in we will do further operations | High | High |
| 02 | Propose Encryption | Test whether propose Encryption Successfully or not | If the propose Encryption may not Run Successfully | We cannot do further operations | we will do further  operations | High | High |
| 03 | Propose Decryption | Test whether propose Decryption Successfully or not | If the  Propose Decryption may not Run Successfully | We cannot do further operations | we will do further  operations | High | High |
| 04 | Calculate Metrics | Test whether calculate Metrics Successfully or not | If the  Calculate Metrics Successfully | We cannot do further operations | we will do further  operations | High | High |

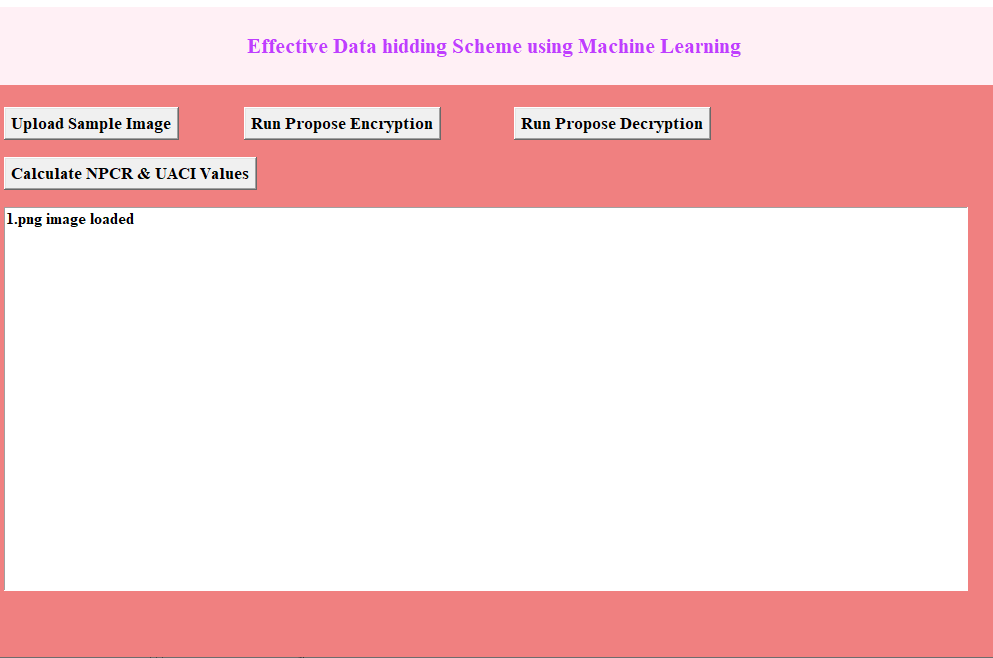
**7.SCREENSHOTS**:  
To run project double click on run.bat file to get below screen



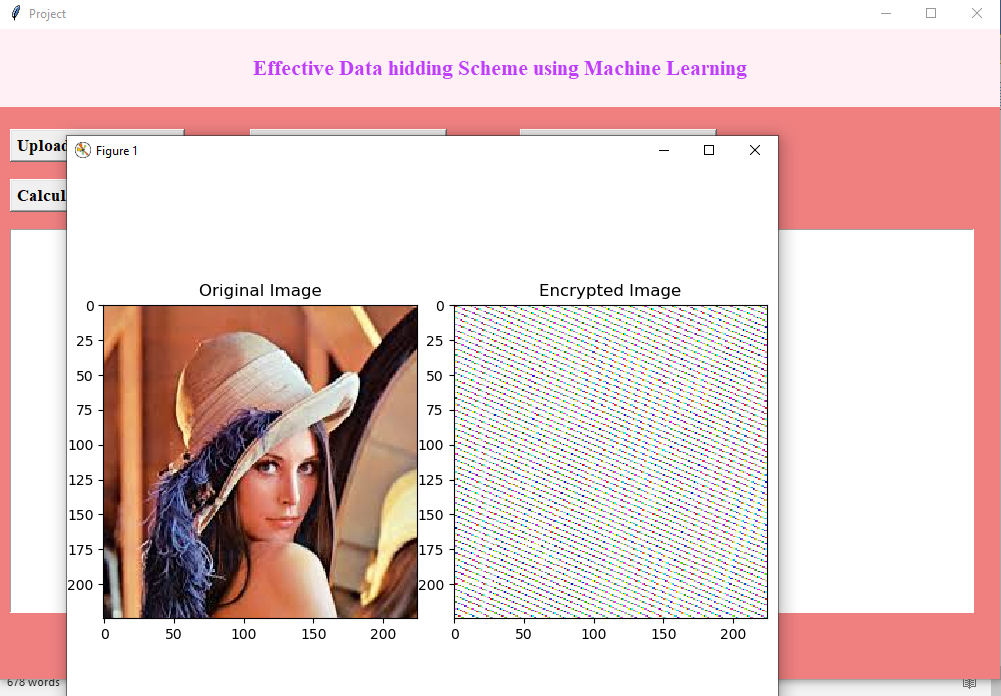
In above screen click on ‘Upload Sample Image’ button to upload image



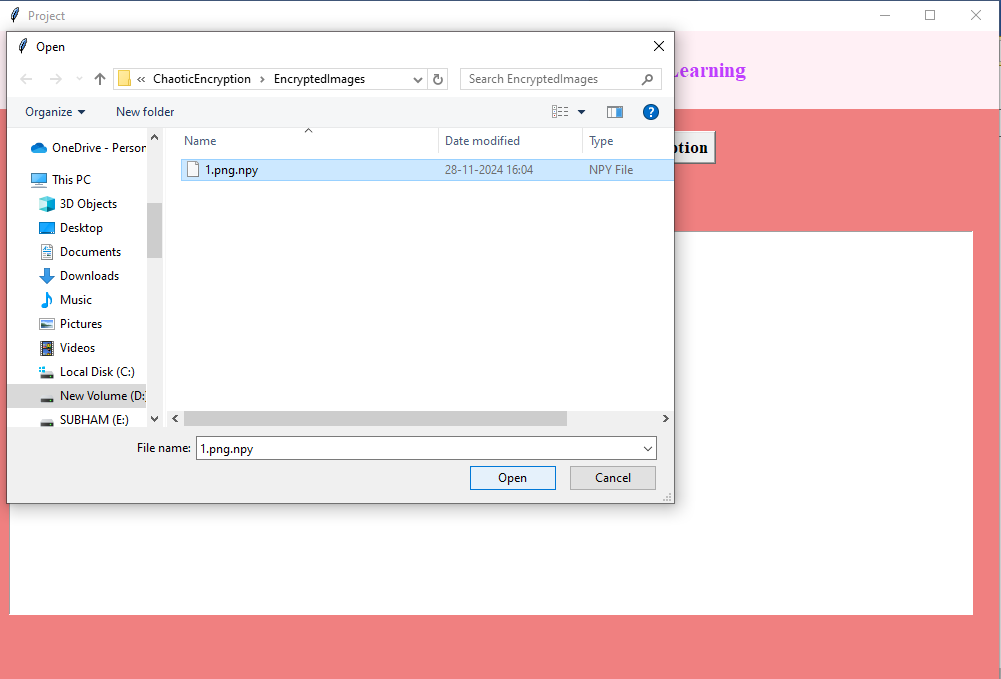
In above screen selecting and uploading sample image and then click on ‘Open’ button to load image and then will get below output



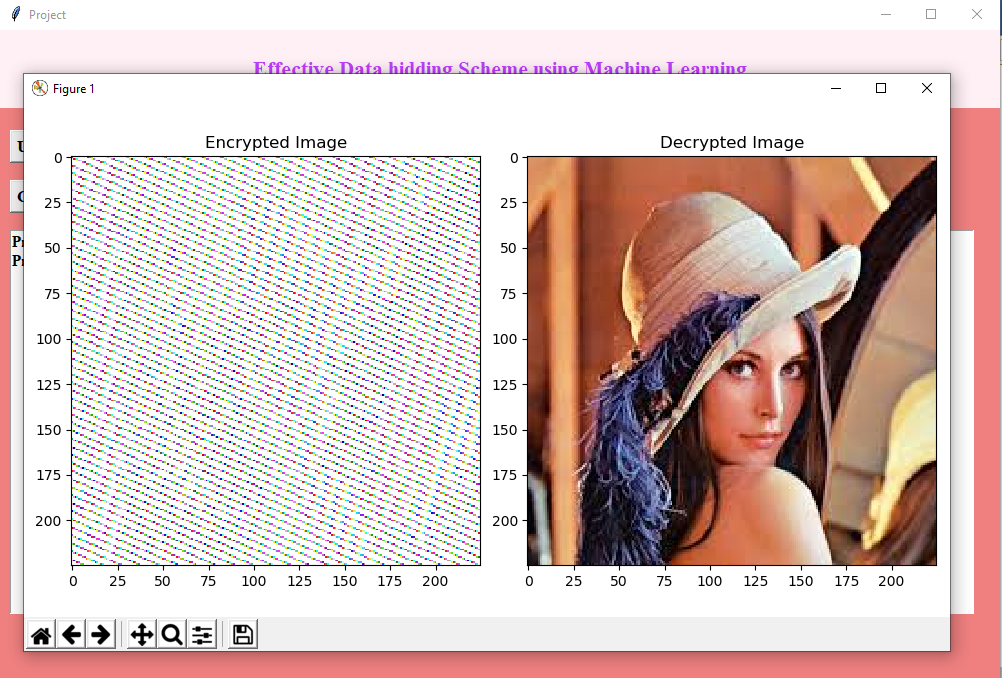
In above screen image loaded and now click on ‘Run Propose Encryption’ button to encrypt image and then will get below output



In above screen first image is the original image and second is the encrypted image and now click on ‘Run Propose Decryption’ button to upload encrypted image and then will get below output



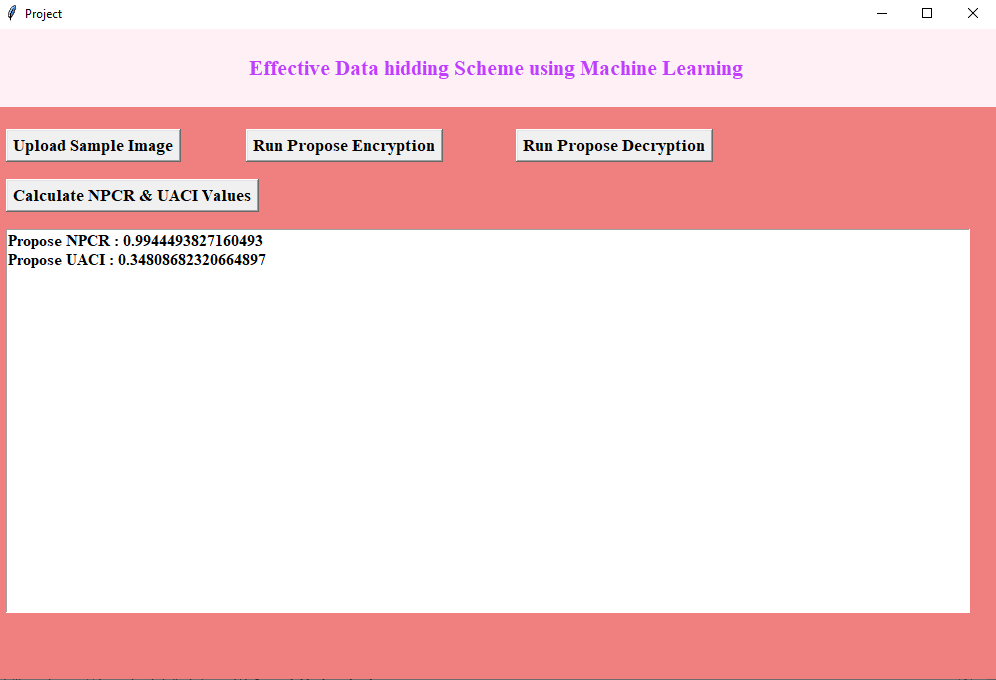
In above screen selecting and uploading encrypted image and then click on ‘Open’ button to get below output



In above screen first image is the encrypted image and second is the decrypted image and can see decrypted image perfectly recovered and now close above image to get below output



In above screen PSNR is 0 and SSIM is 100% which image decrypted is 100% similar to original image and now click on ‘Calculate NPCR and UACI’ button to get below output



In above screen can see NPCR and UACI values.

**8. CONCLUSION:**

In this paper, SEA and SCD methods are proposed, and PWLCM is improved to FI-PWLCM. This scheme works by randomly inserting pixel values and using SHA-256 to associate the key with the plain image. The proposed SEA and FI-PWLCM realize the one-to-one mapping between the seed key and the encryption key stream, which is very consistent with the one-time pad. The SCD method can effectively improve the histogram characteristics of the cipher image, make the distribution of pixels at all levels of the cipher image more uniform, and the information entropy is higher. This makes the scheme have better statistical characteristics of cryptography. In addition, this approach can obtain better histogram properties, entropy values and correlation in less time than existing meta-heuristic image encryption algorithms. Several experiments and security analysis show that the algorithm has a large enough key space, and can effectively resist selective text attack, brute force attack, statistical statistics, noise attack and clipping attack. This algorithm is less efficient for image encryption with large size, so in the future, we can reduce the amount of image data through compression, or speed up the encryption process through the technology of block and parallel computing.

**9. REFERENCES:**

[1] S. Wang, C. Wang, and C. Xu, ‘‘An image encryption algorithm based on a hidden attractor chaos system and the Knuth–Durstenfeld algorithm,’’ Opt. Lasers Eng., vol. 128, May 2020, Art. no. 105995.

[2] X. Chai, X. Zheng, Z. Gan, D. Han, and Y. Chen, ‘‘An image encryption algorithm based on chaotic system and compressive sensing,’’ Signal Process., vol. 148, pp. 124–144, Jul. 2018.

[3] D. Ravichandran, P. Praveenkumar, and J. B. B. Rayappan, ‘‘DNA chaos blend to secure medical privacy,’’ IEEE Trans. Nanobiosci., vol. 16, no. 8, pp. 850–858, Dec. 2017.

[4] G. Chen, Y. Mao, and C. K. Chui, ‘‘A symmetric image encryption scheme based on 3D chaotic cat maps,’’ Chaos, Solitons Fractals, vol. 21, pp. 749–761, Jul. 2004.

[5] N. K. Pareek, V. Patidar, and K. K. Sud, ‘‘Image encryption using chaotic logistic map,’’ Image Vis. Comput., vol. 24, no. 9, pp. 926–934, 2006.

[6] Y. Wang, K.-W. Wong, X. Liao, T. Xiang, and G. Chen, ‘‘A chaos-based image encryption algorithm with variable control parameters,’’ Chaos, Solitons Fractals, vol. 41, no. 4, pp. 1773–1783, 2009.

[7] X. Chai, Y. Chen, and L. Broyde, ‘‘A novel chaos-based image encryption algorithm using DNA sequence operations,’’ Opt. Lasers Eng., vol. 88, pp. 197–213, Jan. 2017.

[8] C. Li, G. Luo, K. Qin, and C. Li, ‘‘An image encryption scheme based on chaotic tent map,’’ Nonlinear Dyn., vol. 87, no. 1, pp. 127–133, 2017.

[9] Q. Xu, K. Sun, C. Cao, and C. Zhu, ‘‘A fast image encryption algorithm based on compressive sensing and hyperchaotic map,’’ Opt. Lasers Eng., vol. 121, pp. 203–214, Oct. 2019.

[10] M. Zhou and C. Wang, ‘‘A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks,’’ Signal Process., vol. 171, Jun. 2020, Art. no. 107484.

[11] S. M. Ismail, L. A. Said, A. G. Radwan, A. H. Madian, and M. F. Abu-ElYazeed, ‘‘A novel image encryption system merging fractional-order edge detection and generalized chaotic maps,’’ Signal Process., vol. 167, Feb. 2020, Art. no. 107280.

[12] P. Sneha, S. Sankar, and A. S. Kumar, ‘‘A chaotic colour image encryption scheme combining Walsh–Hadamard transform and maps,’’ J. Ambient Intell. Humanized Comput., vol. 11, no. 3, pp. 1289–1308, 2020.

[13] Z. Hua, Z. Zhu, S. Yi, Z. Zhang, and H. Huang, ‘‘Cross-plane colour image encryption using a two-dimensional logistic tent modular map,’’ Inf. Sci., vol. 546, pp. 1063–1083, Feb. 2021.

[14] Y. Xian and X. Wang, ‘‘Fractal sorting matrix and its application on chaotic image encryption,’’ Inf. Sci., vol. 547, pp. 1154–1169, Feb. 2021.

[15] C. Li, D. Lin, J. Lü, and F. Hao, ‘‘Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography,’’ IEEE MultiMedia, vol. 25, no. 4, pp. 46–56, Oct./Dec. 2018.

[16] S. Li and X. Zheng, ‘‘Cryptanalysis of a chaotic image encryption method,’’ in Proc. IEEE Int. Symp. Circuits Syst., vol. 2, May 2002, pp. II–II.

[17] E. Solak, C. Çokal, O. Yildiz, and T. Biyikoglu, ‘‘Cryptanalysis of ˇ Fridrich’s chaotic image encryption,’’ Int. J. Bifurcation Chaos, vol. 20, no. 5, pp. 1405–1413, 2010.

[18] J. Chen, L. Chen, and Y. Zhou, ‘‘Cryptanalysis of image ciphers with permutation-substitution network and chaos,’’ IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 6, pp. 2494–2508, Jun. 2021.

[19] J. Katz and Y. Lindell, Introduction to Modern Cryptography. Boca Raton, FL, USA: CRC Press, 2020.

[20] X. Wang, Y. Wang, X. Zhu, and C. Luo, ‘‘A novel chaotic algorithm for image encryption utilizing one-time pad based on pixel level and DNA level,’’ Opt. Lasers Eng., vol. 125, Feb. 2020, Art. no. 105851.