INTRODUCTION

1.1 Cryptography

Cryptography is the study of secure communications techniques that allow only the sender and intended recipient of a message to view its contents. The term is derived from the Greek word **kryptos**, which means hidden. It is closely associated to encryption, which is the act of scrambling ordinary text into what's known as ciphertext and then back again upon arrival. In addition, cryptography also covers the obfuscation of information in images using techniques such as microdots or merging. Ancient Egyptians were known to use these methods in complex hieroglyphics, and Roman Emperor Julius Caesar is credited with using one of the first modern ciphers.

When transmitting electronic data, the most common use of cryptography is to encrypt and decrypt email and other plain-text messages. The simplest method uses the symmetric or "secret key" system. Here, data is encrypted using a secret key, and then both the encoded message and secret key are sent to the recipient for decryption. The problem? If the message is intercepted, a third party has everything they need to decrypt and read the message. To address this issue, cryptologists devised the asymmetric or "public key" system. In this case, every user has two keys: one public and one private. Senders request the public key of their intended recipient, encrypt the message and send it along. When the message arrives, only the recipient's private key will decode it — meaning theft is of no use without the corresponding private key.

1.2 Cryptography In Cybersecurity

With an increasing number of users, devices and programs in the modern enterprise, combined with the increased deluge of data much of which is sensitive or confidential the importance of cybersecurity continues to grow. The growing volume and sophistication of cyber attackers and attack techniques compound the problem even further.

1.3 3-D Pythocrypt

There is a huge number of cryptographic systems which can be used to encrypt the message and transmit the ciphertext that is generated so that the confidential information is made readable only by the intended user.

Traditionally cryptographic systems are classified into two categories, Symmetric, and Asymmetric cryptographic systems. In Symmetric cryptographic systems, the same key is used for both encryption and decryption process, some of the popular symmetric techniques like the Caesar cipher model, play fair cipher model, and hill cipher model. The asymmetric cryptographic system uses different keys for encryption and decryption processes called public keys and private keys, the most popular and widely used asymmetric algorithm is RSA.

3-D Pythocrypt is a new and young technique that is used to encrypt and decrypt a given message using a known 3-D geometric shape [1]. This methodology uses the geometric attributes of shapes such as Area, Volume, Perimeter, etc., to generate an algorithm using which the plain text can be encoded into non-decipherable text. This new approach is unique compared to all other existing algorithms as it does not need a key for encryption but generates one for decryption [1].

1.4 Image Steganography

Steganography is derived from the two Greek words Stego and Graphia, Stego means covering and graphia which means writing, thus the translation is covered writing or the hiding the data [2]. Steganography, has gained acceptance in this context to hide the data that is not perceptible to human eyes [3]. However, the word Image-Steganography explains that it is the art of hiding the information or message inside the image. The advantage of Image-Steganography that attracts is one cannot predict the presence of information inside the Image. Any Cryptographic system used along with Image-Steganography provides better security against cryptanalyst.

There are many different techniques used to hide the information inside the images. In our project, we use Least Signification Bit substitution. The LSB Substitution is a method that is used to hide the information in the pixels of the image [3].

1.4.1 Least Significant Bit Technique

Least Significant Bit is a steganographic technique where the least significant bit of each pixel in the image is used to store the information. The information can be inserted in the LSB bit by converting it into binary format. The amount of data that can be encoded is determined by the size of the image or the number of pixels selected in the image. If we want to hide the information within the images LSB method is used. One bit of secret information is substituted in the 8th bit of each byte of file which is used as carrier [2].

1.5 Overview

In this project an attempt has been made to integrate the 3-D Pythocrypt cryptographic technique with Image Steganography and provide better security for message transmission.

3-D Pythocrypt is a new technique where we use the properties of 3-D geometric shapes to encrypt and decrypt plain text. Image-Steganography is a cryptographic technique used to hide the information within the pixel of the Image. The main ideology is that the message which needs to be transmitted is converted to unreadable form using the 3-D Pythocrypt algorithm and the ciphertext which is obtained is hidden in the pixels of the Image, the Image containing the ciphertext is transmitted to the receiver thereby maintaining the confidentiality and secrecy.

1.6 Problem Statement

Ensuring the security and privacy of data in computer network systems is a major challenge in the modern computer age. The most persistent issue found during data transmission is cryptanalysis attacks, which is interception of the message by un-authorized personnel. There are many methods to hide/encrypt data, but many cannot guarantee data leakage or avoid attackers from obtaining sensitive information. This project aims to improve the security and provide privacy for data transmission using multiple forms of data encryption.

LITERATURE SURVEY

2.1 A 3-D Advancement of Pythocrypt For Any File Type

In this paper the method of 3–d system with mapping between Cartesian and Polar coordinates that can be used for encryption and decryption of files of various types and sizes effectively will be discussed. Pythocrypt [1] was specifically developed for medical images whereas the present system proposed can be applied for any file type supported by any platform and hence it is termed file-type-independent. The algorithm proposed in this paper shall henceforth be addressed as "3-d Pythocrypt".

3-d Pythocrypt is a new concept which has been developed to encrypt and decrypt files of all types using a file-type independent algorithm proposed in Pythocrypt [1]. The algorithm "3-d Pythocrypt" uses the infinite divergence [1] of the 3-dimensional coordinate system definition of a sphere. The algorithm is unique compared to all the existing algorithms as it does not need a key for encryption, but generates one for the purpose of decryption. Also the entire algorithm is asymmetric, i.e. the encryption and decryption processes are dissimilar.

"3-d Pythocrypt" fares well against standard algorithms with 2 byte block size. However, it can outperform most algorithms as the cipher text will be too large in size and any sphere has infinite surface points with increase in block size. Further it also has infinite divergence for backtracking. A new radical approach of cryptanalysis titled "Multi agent pattern recognition" [1] is able to obtain parts of possible plain text after multiple iterations on a single file. This approach uses small pieces of codes called agents to try and obtain similar patterns in a packet in transit. They work independent of the source and obtain repeated patterns in the files. Since "3-d Pythocrypt" uses 16 bit blocks for this experiment, in English text, if there are 3 similar 2 letter blocks appearing consecutively, they would generate same cipher text [1].

2.2 Image Steganography Using LSB

This paper focuses on a process of hiding data to image by using the least significant bit(LSB) and the symmetric key between the sender and the receiver. Here we have to choose the bits that will get the minimum resolution between the original image and stego image. This paper further explains how the encryption and decryption processes are done.

This paper demonstrates two structures for Steganography: Initially it is the striking rationality which is also known as Least Significant Bit(LSB), and the second one is the latest system with LSB+KEY. The results executions have been looked up for the estimations of PSNR with individual checks. The calculation of LSB+KEY gives better demands concerning the PSNR regards. This is one of the investigated results in this work and still the work is in its head-way to improve the computations for still better code unconventionality and time complexity nature.

2.3 Image Steganography: A Review of The Recent Advances

The research work explores and discuss various deep learning methods available in image steganography field. Deep learning techniques used for image steganography can be broadly divided into three categories - traditional methods, Convolutional Neural Network-based and General Adversarial Network-based methods. Along with the methodology, an elaborate summary on the datasets used, experimental set-ups considered and the evaluation metrics commonly used are described in this paper. A table summarizing all the details are also provided for easy reference. This paper aims to help the fellow researchers by compiling the current trends, challenges and some future direction in this field.

This paper has elaborated on the techniques used in the recent times for image steganography, the current trends. Along with it, details on the datasets and evaluation metrics are detailed. Challenges faced, some discussions on the gaps and the scopes for future direction is also evaluated in this paper. It can be concluded that deep learning has tremendous potential in the image steganography field taking into consideration that all the challenges and gaps are filled.

DEVELOPING ENIGMA

Enigma is a tool or a system application which is a platform independent. It doesn't require any databases as the result will not be stored for later uses. There is no need for Internet connection for running the tool. The whole application is built using Python Programming language.

3.1 Why Python?

Python scripts and APIs can be tailor made into effective network monitoring and forensics tools. Their versatility makes them ideal in assorted applications including cyber security, data mining, Internet of Things, cloud simulation, grid implementation, etc.

The software development companies prefer Python language because of its versatile features and fewer programming codes. Nearly 14% of the programmers use it on the operating systems like UNIX, Linux, Windows and Mac OS. The programmers of big companies use Python as it has created a mark for itself in the software development with characteristic features like-

- Interactive
- Interpreted
- Modular
- Dynamic
- Object-oriented
- Portable
- High level
- Extensible in C++ & C

Advantages or Benefits of Python

The Python language has diversified application in the software development companies such as in gaming, web frameworks and applications, language development, prototyping, graphic design applications, etc. This provides the language a higher plethora over other programming languages used in the industry. Some of its advantages are-

- **Extensive Support Libraries**

It provides large standard libraries that include the areas like string operations, Internet, web service tools, operating system interfaces and protocols. Most of the highly used programming tasks are already scripted into it that limits the length of the codes to be written in Python.

- **Integration Feature**

Python integrates the Enterprise Application Integration that makes it easy to develop Web services by invoking COM or COBRA components. It has powerful control capabilities as it calls directly through C, C++ or Java via Jython. Python also processes XML and other markup languages as it can run on all modern operating systems through same byte code.

- **Improved Programmer's Productivity**

The language has extensive support libraries and clean object-oriented designs that increase two to ten fold of programmer's productivity while using the languages like Java, VB, Perl, C, C++ and C#.

- **Productivity**

With its strong process integration features, unit testing framework and enhanced control capabilities contribute towards the increased speed for most applications and productivity of applications. It is a great option for building scalable multi-protocol network applications.

3.2 3-D Pythocrypt

3-D Pythocrypt technique is built in Python Programming language, only a limited number of libraries were used to develop the algorithm. The datatype that are used extensively in developing 3-D Pythocrypt algorithm are decimal, lists, Strings, tuples.

3.2.1 Decimal Module In Python

The Python decimal module provides support for fast correctly-rounded decimal floating point arithmetic. By default, Python interprets any number that includes a decimal point as a double precision floating point number. The Decimal is a floating decimal point type which

more precision and a smaller range than the float. It is appropriate for financial and monetary calculations. It is also closer to the way how humans work with numbers.

Unlike hardware based binary floating point, the decimal module has a user alterable precision which can be as large as needed for a given problem. The default precision is 28 places.

Some values cannot be exactly represented in a float data type. For instance, storing the 0.1 value in float (which is a binary floating point value) variable we get only an approximation of the value. Similarly, the 1/3 value cannot be represented exactly in decimal floating point type. Neither of the types is perfect; generally, decimal types are better suited for financial and monetary calculations, while the double/float types for scientific calculations.

3.3 Image Steganography

Image Steganography requires modules that allow us to work with images. Python provides library called PIL to work with images and its properties.

3.3.1 PIL (Python Imaging Library) And Pillow

Some of the most common image processing libraries are: OpenCV, Python Imaging Library (PIL), Scikit-image, Pillow. However, in this tutorial, we are only focusing on **Pillow module** and will try to explore various capabilities of this module.

Pillow is built on top of PIL (Python Image Library). PIL is one of the important modules for image processing in Python. However, the PIL module is not supported since 2011 and doesn't support python 3.

Pillow module gives more functionalities, runs on all major operating system and support for python 3. It supports wide variety of images such as "jpeg", "png", "bmp", "gif", "ppm", "tiff". You can do almost anything on digital images using pillow module. Apart from basic image processing functionality, including point operations, filtering images using built-in convolution kernels, and color space conversions.

The Pillow library contains all the basic image processing functionality. You can do image resizing, rotation and transformation.Pillow module allows you to pull some statistics data out of image using histogram method, which later can be used for statistical analysis and automatic contrast enhancement.

SOFTWARE REQUIREMENTS AND SPECIFICATIONS

4.1 Software Requirements and Specification

The goal of software requirements and specification is to describe what the proposed system should do. It is the medium through which the user needs are accurately specified. It provides reference for the validation of the final product.

- System execution requirements
  - System : Workstation class / Pentium Dual Core, Make/Model Dell T5400
  - Hard Disk : 256 GB
  - Input Devices : Keyboard, Mouse
  - Ram : 4 GB
- Build environment
  - Operating system : Windows 7+
  - Programming Language : Python 3+

DESIGN

In this chapter, the architectural design is presented first and then the functional architecture overview, followed by a detailed description of ENIGMA.

5.1 Architecture

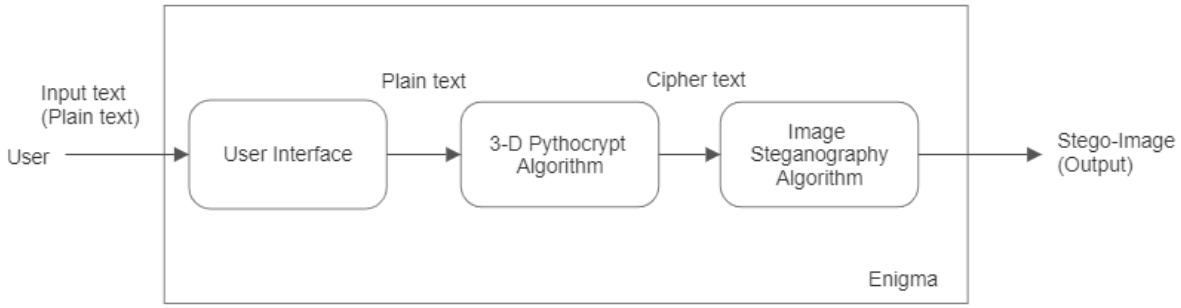

Fig. 5.1 Enigma design during encryption phase.

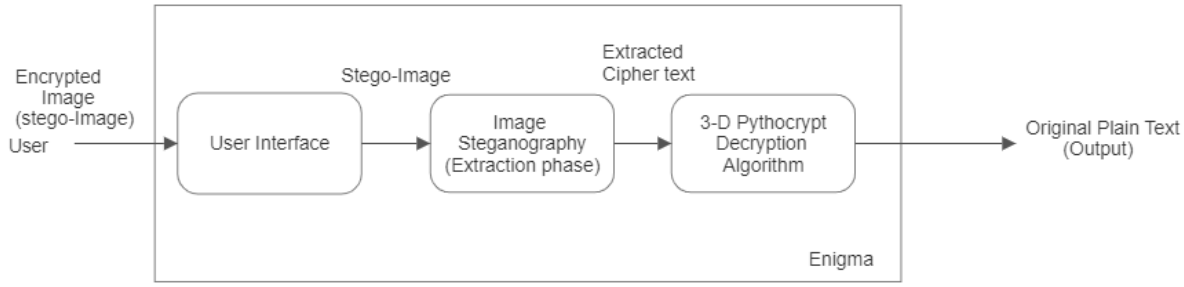

Fig. 5.2 Enigma design during decryption phase.

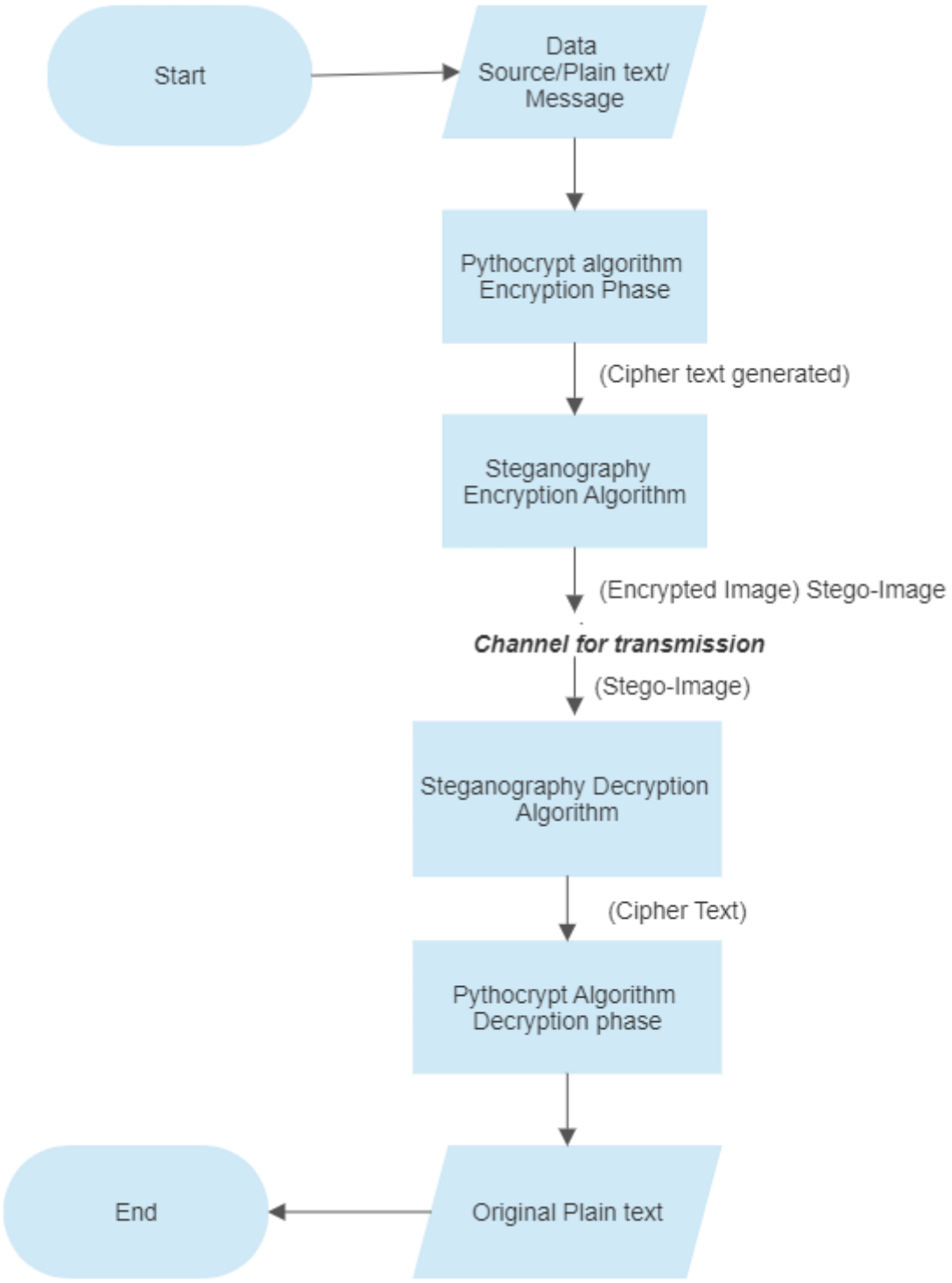

Fig. 5.3 Process flow of Enigma

IMPLEMENTATION

Project implementation (or project execution) is the phase where visions and plans become reality. This is the logical conclusion, after evaluating, deciding, visioning, planning, applying for funds and finding the financial resources of a project. Technical implementation is one part of executing a project.

6.1 3-D Pythocrypt

6.1.1 3-D Pythocrypt Encryption Phase

The 3-D Pythocrypt algorithm is based on the properties of 3-D geometric shapes. This algorithm takes the properties like Area, Volume, etc. of the shapes. The message which consists of English alphabets and/or symbols is converted to its ASCII equivalent values which are in decimal format. These numbers are halved and considered as inputs to the formulae to calculate the ciphertext. In our research work, we have used different 3-D shapes to make the algorithm more secure. Whenever a user inputs a plain text, its corresponding ASCII value is generated. The entire ASCII value is halved. There are two variables in the formula namely ‘a’ and ‘h’. The halved ASCII values are the inputs for these variables. For a particular plain text, a shape is chosen and the encryption operation is performed. The result is the cipher text obtained from the 3-D Pythocrypt encryption phase. Along with the cipher text, any one value of the variable ‘a’, ‘a2’, or ‘h’ can be used as the key. Table 1. Contains the different 3-D geometric shapes with their respective volumetric formula.

Whenever the user inputs a new plain text, a particular shape and formula from Table 6.1. is chosen and is used for the encryption process. Id column in Table 6.1. Contains the id’s assigned to the shapes which will help us to communicate the shape used at encryption to the receiver.

Id	Shape	Volumetric Formula
1	Octahedron	$v = (2 * a^2 * h)/3$
2	Hexagonal Prism	$v = (3 * \sqrt{3} * a^2 * h) / 3$
3	Pentagonal Prism	$v = (\sqrt{(5(5+2\sqrt{5}))} * a^2 * h) / 4$
4	Octagonal Prism	$v = 2 * (1 + \sqrt{2}) * a^2 * h$
5	Pentagonal Pyramid	$(5 * \tan(54^\circ) * h * a^2)/12$

Table. 6.1 Different 3D shapes and their corresponding volumetric formula

6.1.2 3-D Pythocrypt Decryption Phase

3-D Pythocrypt Decryption: All the information that is required to decrypt the cipher text (geometric shape used, key) is embedded in the cipher text. The key is extracted from the cipher text using the delimiter. Here the delimiter is '.'. As mentioned earlier, any one value of variables 'a', 'a2', or 'h' can be used as the key. The decryption formula changes accordingly. If we use 'a' or 'a2' as the key then, we need to find 'h'. However, if we use 'h' as the key then, we need to find the value of 'a'. Table 2. Describes the different equations that need to be used to calculate the unknown value.

Id	Shape	Decryption Formula		
		if key = a	If key = a^2	If key = h
1	Octahedron	$a^2 = (a * a)$ $h = (3 * v)/(2 * a^2)$	$h = (3 * v)/(2 * a^2)$	$a^2 = (3 * v)/(2 * h)$ $a = \sqrt{a^2}$
2	Hexagonal Prism	$a^2 = (a * a)$ $h = (2 * v)/(3 * \sqrt{3} * a^2)$	$h = (2 * v)/(3 * \sqrt{3} * a^2)$	$a^2 = (2 * v)/(3 * \sqrt{3} * h)$ $a = \sqrt{a^2}$
3	Pentagonal Prism	$a^2 = (a * a)$ $h = (4 * v)/\sqrt{(5(5+2\sqrt{5}))} * a^2$	$h = (4 * v)/\sqrt{(5(5+2\sqrt{5}))} * a^2$	$a^2 = (4 * v)/\sqrt{(5(5+2\sqrt{5}))} * h$ $a = \sqrt{a^2}$
4	Octagonal Prism	$a^2 = (a * a)$ $h = v /((2 * (1 + \sqrt{2}) * a^2)$	$h = v /((2 * (1 + \sqrt{2}) * a^2)$	$a^2 = v /((2 * (1 + \sqrt{2}) * h)$ $a = \sqrt{a^2}$
5	Pentagonal Pyramid	$a^2 = (a * a)$ $h = (12 * v)/(5 * \tan(54^\circ) * a^2)$	$h = (12 * v)/(5 * \tan(54^\circ) * a^2)$	$a^2 = (12 * v)/(5 * \tan(54^\circ) * h)$ $a = \sqrt{a^2}$

Table. 6.2 3-D Pythocrypt decryption formulae list

Once 'a' and 'h' values are successfully retrieved, these two are combined to form the ASCII values of the original plain text. Eventually, the original plain text is extracted from its ASCII values.

6.2 Image Steganography Using LSB

6.2.1 Image Steganography Encryption Phase

In image steganography the information is hidden exclusively in images. The most common and popular method of modern-day steganography is to make use of LSB (least significant bit) of picture's pixel information. This technique works best when the file is longer than the message file and if image is grayscale. When applying LSB techniques to each byte of a 24-bit image, three bits can be encoded into each pixel.

Pixels:
(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)
A: 01000001
Result:
(00100110 11101001 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)

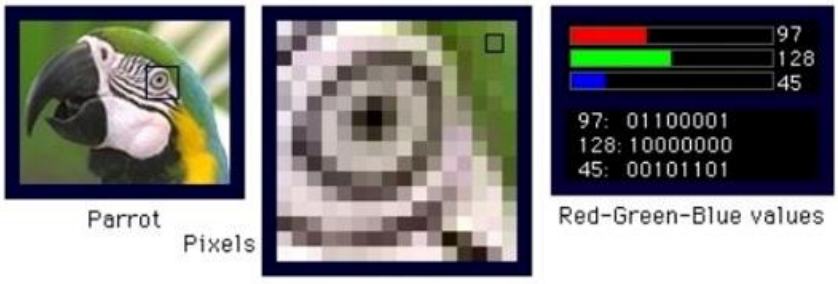

Fig. 6.1 LSB Encryption phase.

The three underlined bits are the only three bits that were actually altered. LSB insertion is easy to implement, at the same time, easily attacked. Today steganography is being incorporated into digital technology. The techniques have been used to create the watermarks that are in our nation's currency, as well as encode music information in the ever-popular mp3 music file. Copyrights can be included in files, and fingerprints can be used to identify the people who break copyright agreements. While Slight modifications in the color palette and simple image manipulations will destroy the entire hidden message.

6.2.2 Image Steganography Decryption Phase

We need to first calculate how many pixels is the text stored in. Each character is represented in 8 bits. So, the number of pixels in which the text is stored will be 13 * 8 = 104. Now after knowing this, we need to traverse through the image, one pixel at a time. We store the Least Significant Bit (LSB) of each pixel in an array extracted_bits. After extracting the LSBs of the required pixels, we need to take every 8 bits from extracted_bits and convert it to the corresponding character. In this way, the text stored in the stego image can be extracted.

SOFTWARE TESTING

Software testing is the process of evaluation a software item to detect differences between given input and expected output, also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way the user wants it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

7.1 Basics of Software Testing

There are two basics of software testing: blackbox testing and whitebox testing.

7.1.1 Black Box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system as shown in figure 7.1. It is also called functional testing. Black box testing is often used for validation.

7.1.2 White Box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system as shown in figure 7.2. It is also called structural testing and glass box testing, often used for verification.

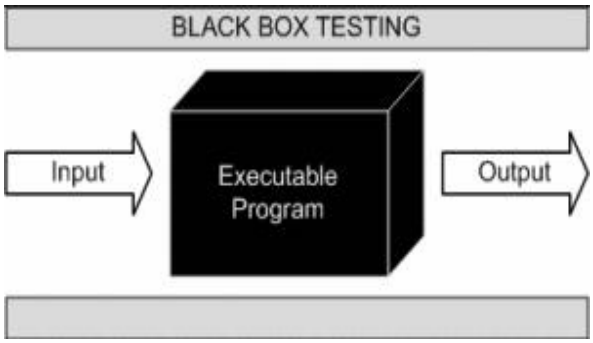

Fig 7.1 Black box testing

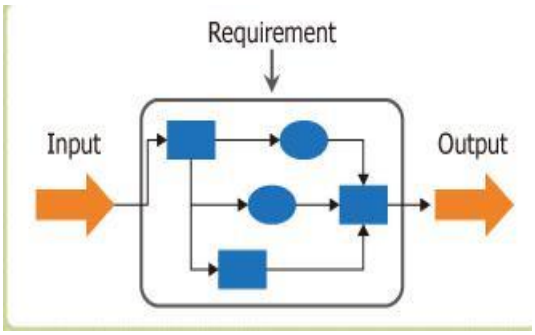

Fig 7.2 White box testing

7.2 A Perspective on Testing

Some of the terms associated are:

- **Error:** People make errors. A good synonym is mistake. When People make mistake while coding, we call these mistakes bugs. A requirements error may be magnified during design and amplified still more during coding.
- **Fault:** A fault is the result of an error. A fault is a representation of an error.
- **Failure:** Failure occurs when fault executes.
- **Incident:** An incident is the symptom associated with the failure that alerts the user to the occurrence of failures.
- **Test:** Testing is obviously concerned with errors, failures and incidents. A test has two distinct goals: to find the failures and to demonstrate correct execution.
- **Test Case:** Test Case has and identity and is associated with the program behaviour.

Top-down testing can proceed in a depth-first or breadth-first manner. For depth- first integration testing each module is tested in increasing detail, replacing more and more levels of detail with the actual code rather than stubs. Alternatively, breadth-first would process by refining all the modules at the same level of control throughout the application. In practice, a combination of two techniques would be used. At the initial stage, all the modules might be only partly functional, possibly being implemented only to deal with non-erroneous data. These would be tested in breadth first search, but over a period of time each would be replaced with successive refinements which were closer to the full functionality. This allows depth-first testing of a module to be performed simultaneously with breadth-first testing of all modules.

The other major category of integration testing in Bottom up integration testing where an individual module is tested from a test harness. Once a set of individual modules have been tested they are then combined into a collection of modules known as builds which are then tested by a second test harness. This process can continue until the build consists of the entire application.

In practise a combination of top down and bottom up testing would be used. In a large software project being developed by a number of sub-teams, or a smaller project where different modules were built by individuals. The sub-teams or individuals would conduct bottom-up testing of the modules which they were constructing before releasing them to an integration team which would assemble them together for top-down testing.

Unit testing deals with testing a unit as a whole. This would test the interaction of many functions but confine the test within one unit. The exact scope of a unit is left to interpretation. Supporting test code, sometimes called Scaffolding may be necessary to support an individual test. This type of testing is driven by the architecture and implementation teams. This focus is called black-box testing because only the details of the interface are visible to the tester. Limits that are global to a unit are tested here.

In the construction industry, scaffolding is a temporary, easy to assemble and disassemble frame placed around a building to facilitate the construction of the building. The construction workers first build the scaffold and then the building. Later the Scaffold is removed, exposing the completed building.

Similarly, in software testing one particular test may need some supporting software. This software can establish a correct evaluation of the test to take place. The scaffolding software may establish state and values for data structures as well as providing dummy external functions for the test. Different scaffolding software may be needed from one test to another test. Scaffolding software rarely is considered part of the system.

Sometimes the scaffolding software becomes larger than the system software being tested. Usually the scaffolding software is not of the same quality as the system software and frequently is quite fragile. A small change is test may lead to much larger changes in the scaffolding.

Internal and unit testing can be automated with the help of coverage tools, analyses the source code and generates a test that will execute every alternative thread of execution. Typically, the coverage tool is used in a slightly different way. First the coverage tool is used to augment the source by placing information prints after each line of code. Then the testing suite is executed, generating an audit trial. This audit trial is analysed and reports the percent of the total system code executed during the test suite. If the coverage is high and untested source lines are of low impact to the systems quality, then no more additional tests are required.

Specific tests which can be performed in either or both stages include the following.

- **Regression Testing:** Where this version of software is tested with the automated test harness used with the previous versions to ensure that the required features of the previous version are still working in new version.

- **Recovery Testing:** Where the software is deliberately interrupted in a number of ways to ensure that the appropriate techniques for restoring any lost data will function.

- **Security Testing:** Where unauthorized attempts to operate the software or parts of it, might also include attempts to obtain access the data or harm the software installation or even the system software. As with all types of security determined will be able to obtain unauthorized access and the best that can be achieved is to make this process as difficult as possible.

- **Stress Testing:** where abnormal demands are made upon the software by increasing the rate at which it is asked to accept or the rate at which it is asked to produce information. More complex tests may attempt to create very large sets or cause the software's to make excessive demands on the operating systems.

- **Performance Testing:** Where the performance requirements, if any are checked. These may include the size of the software when installed, amount of main memory and/or secondary memory it requires and the demands made by the operating system when running with normal limits or the response time

- **Alpha and Beta Testing:** This is where the software is released to the actual end user. An initial release, the alpha release might be made available to selected users only who are expected to report bugs and other detailed observations back to the production team. Once the application changes necessary by the alpha phase can be made to larger more representative set users, before the final release is made to all users. The final process should be a software audit where the complete software project is checked to ensure that it meets production management requirements. This ensures that all required documentation has been produced is in correct format and is of acceptable quality. The purpose of this review is: firstly to assure the quality of the production process and by implication of construction phase commences. A formal hand over from the development team at the end of the audit will mark the transition between the two phases.

- **Integration Testing:** Integration testing can proceed in a number of different ways, which can be broadly characterised as top down or bottom up. In top down integration testing the high-level control routines are tested first, possibly with the middle level control structures present only as stubs. Sub-program stubs are incomplete sub-programs which are only present to allow the higher-level control routines to be tested.

7.3 Testing Model

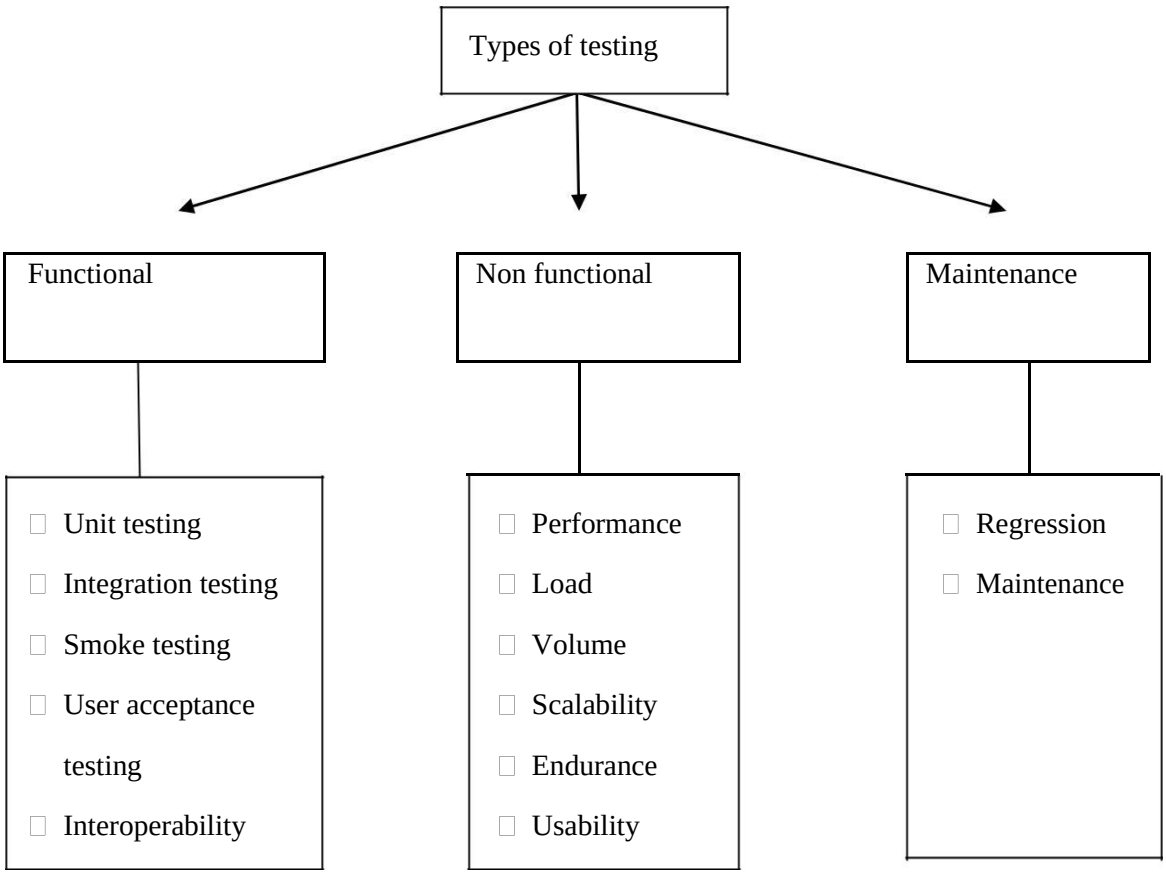

Fig 7.3 Types of testing

7.4 Levels of Testing

7.4.1 Unit Testing

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In object-oriented programming a unit is often an entire interface, such as a class, but could be an individual method.

For unit testing first adopt the code testing strategy, which examined the logic of program. During the development process itself all the syntax errors etc. got rooted out. For this developed test case that result in executing every instruction in the program or module i.e. every path through program was tested. Test cases are data chosen at random to check every possible branch after all the loops.

7.4.2 Error Handling

In this system we have tried to handle all the errors that occurred while running the application. For Testing we used Top-Down design a decomposition process which focuses as the flow of control, at latter strategies concern itself with code production.

The first step is to study the overall aspects of the tasks at hand and break it into a number of independent modules. The second step is to break one of these modules further into independent sub modules. One of the important features is that each level the details at lower levels are hidden. So unit testing was performed first and then system testing.

7.4.3 Integration Testing

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined may not produce the desired functions. Integrated testing is the systematic testing to uncover the errors with an interface. This testing is done with simple data and developed system has run successfully with this simple data. The need for integrated system is to find the overall system performance.

Steps to perform integration testing

Step 1: Create a Test Plan

Step 2: Create Test Cases and Test Data

Step 3: Once the components have been integrated execute the test cases

Step 4: Fix the bugs if any and re test the code

Step 5: Repeat the test cycle until the components have been successfully integrated

Name of the Test	Integration testing
Test plan	To check whether the system works properly when all the modules are integrated.
Test Data	Sample data models and sample data for testing all the modules

Table 7.1 Test cases for integration testing

7.4.4 System Testing

Ultimately, software is included with other system components and the set of system validation and integration tests are performed. System testing is a series of different tests whose main aim is to fully exercise the computer-based system. Although each test has a different role all work should verify that all system elements are properly integrated and formed allocated functions. Test cases for input output is as shown in table 7.3

Name of the Test	**System Testing**
Item being tested	Over all functioning of GUI with all functions properly linked.
Sample Input	Sample data models and sample data for testing all the modules.
Expected Output	All the modules like data as requested should be displayed.
Actual Output	Requested data is presented to the user vis the GUI.
Remarks	Successful

Table 7.2 Test cases for Input-Output

7.4.5 Validation Testing

Validation testing is a concern which overlaps with integration testing. Ensuring that the application fulfils its specification is a major criterion for the construction of an integration testing. Validation testing also overlaps to a large extent with system testing, where the application is tested with respect to its typical working environment.

Consequently, for many processes no clear division between validation and system testing can be made. At the culmination of black box testing, software is completely assembled is as a package. Interfacing errors have been uncovered and the correct and final series of tests, i.e., validation tests begins. Validation test is defined with a simple definition that validation succeeds when the software function in a manner that can be reasonably accepted by the customer.

7.4.6 Output Testing

After performing validation testing, the next step is output testing of the proposed system. Since the system cannot be useful if it does not produce the required output. Asking the user about the format in which the system is required tests the output displayed or generated by the system is required tests the output displayed or generated by the system under consideration.

The output format is considered in two ways, one is on screen format and the other is printed format. The output format on the screen is found to be corrected as the format was designated in the system has according to the user needs. As for the hard copy the output comes according to the specification requested by the user on the GUI. The output testing does not result in any correction in the system.

7.4.6 Test Data and Output

Taking various kind soft data plays a vital role in system testing. After preparing the test data system under study is tested using the test data. While testing, errors are again uncovered and corrected by using the above steps and corrections are also noted for future use.

RESULTS AND SNAPSHOTS

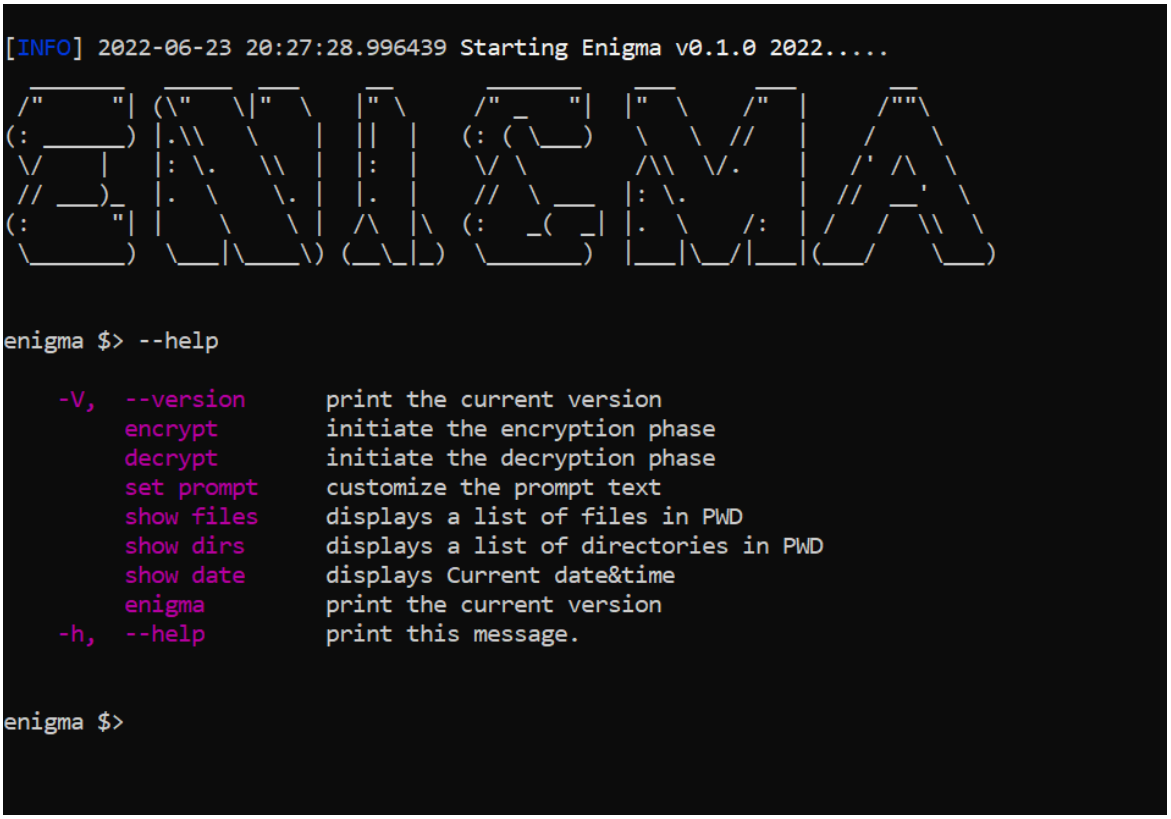

Fig 8.1 List of commands available in Enigma

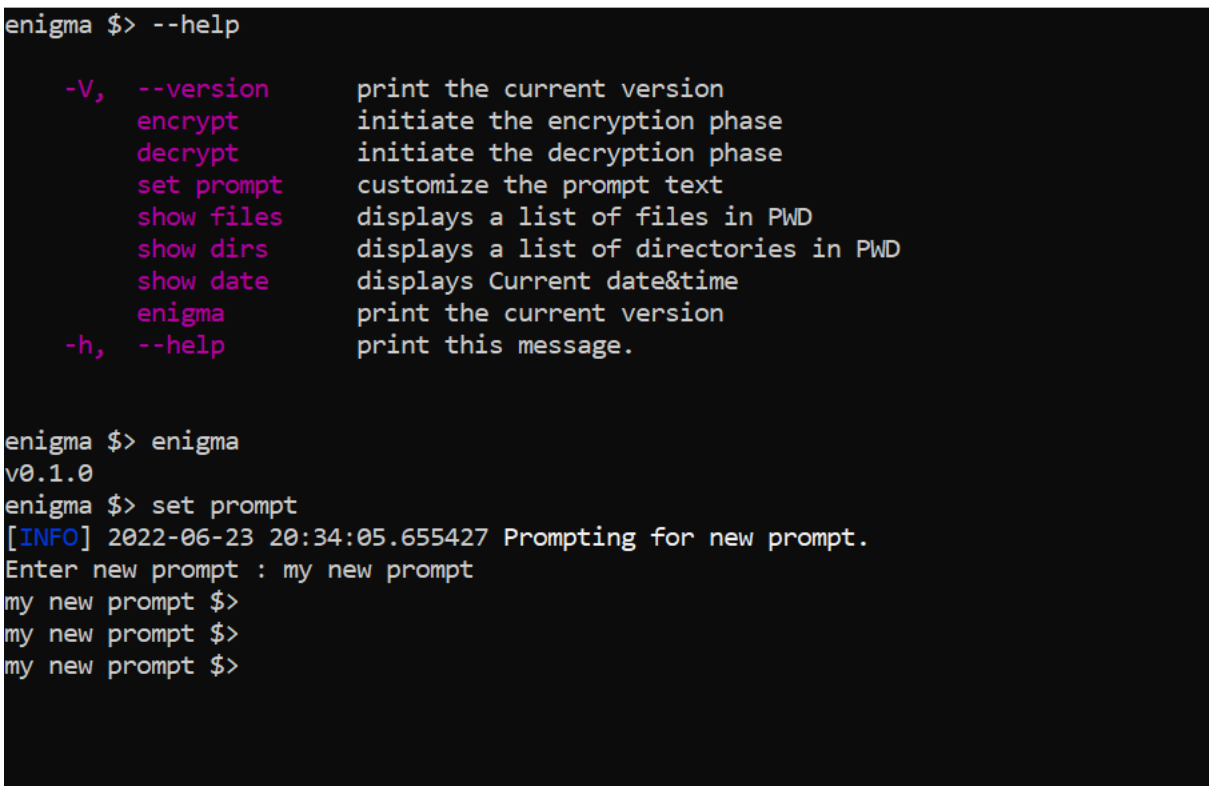

Fig 8.2 New Prompt command output

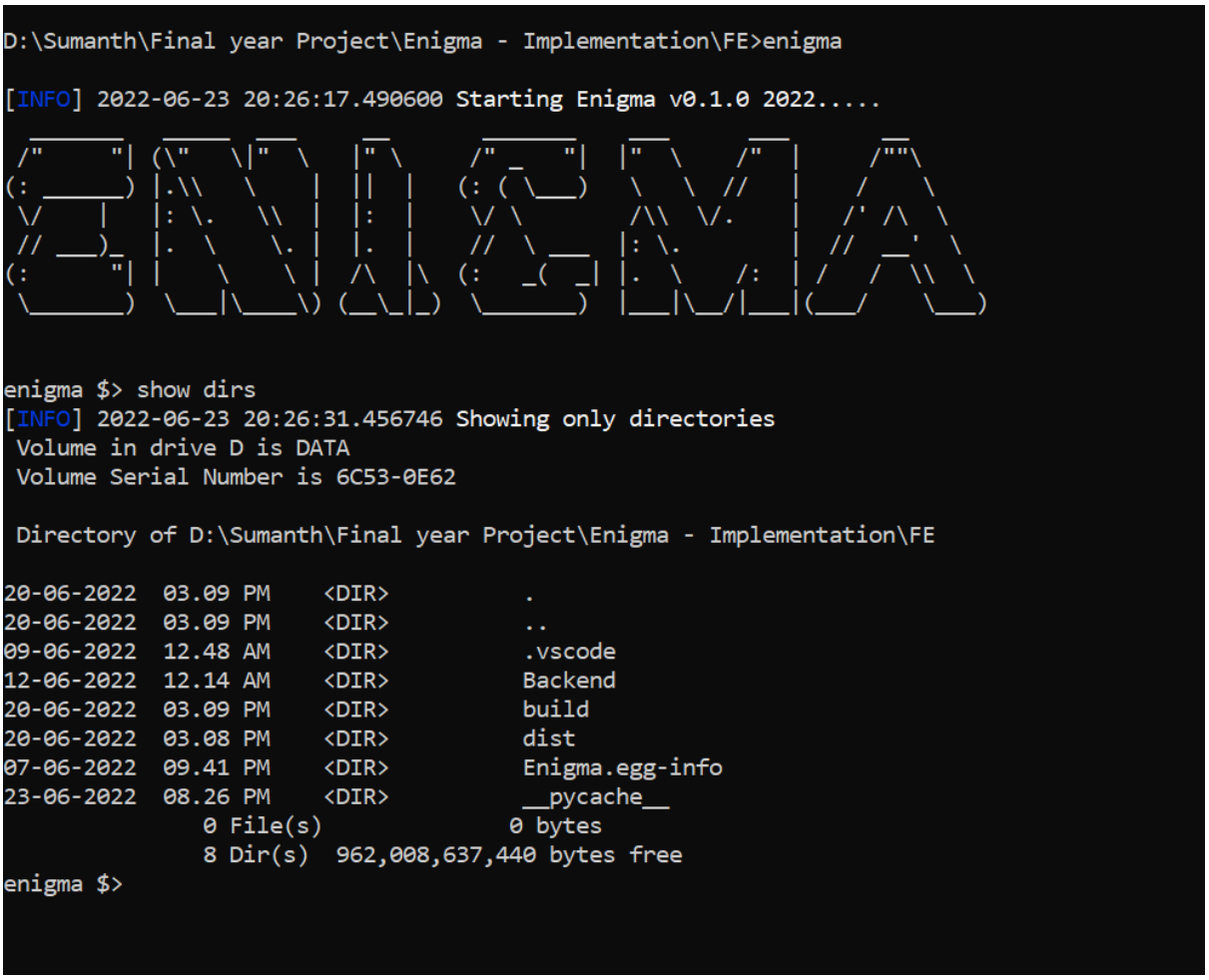

Fig. 8.3 Show dirs command output

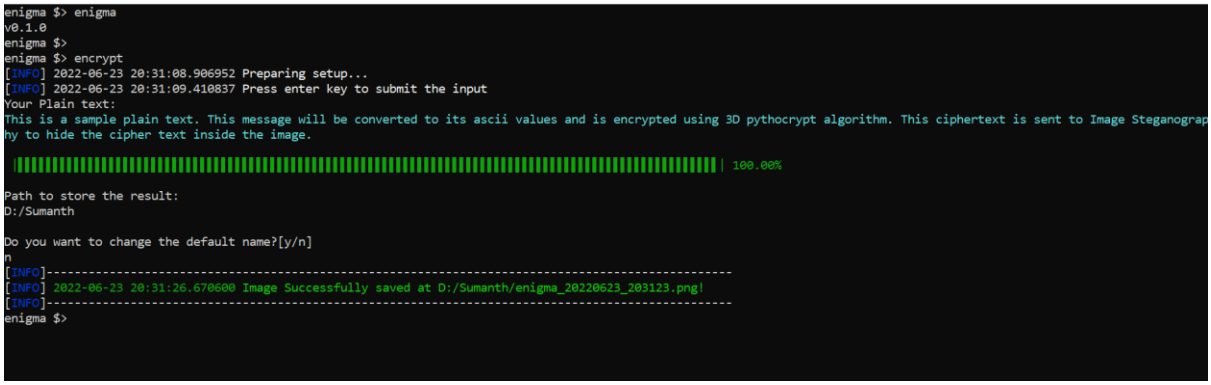

Fig. 8.4 Version specification and Encrypt command output

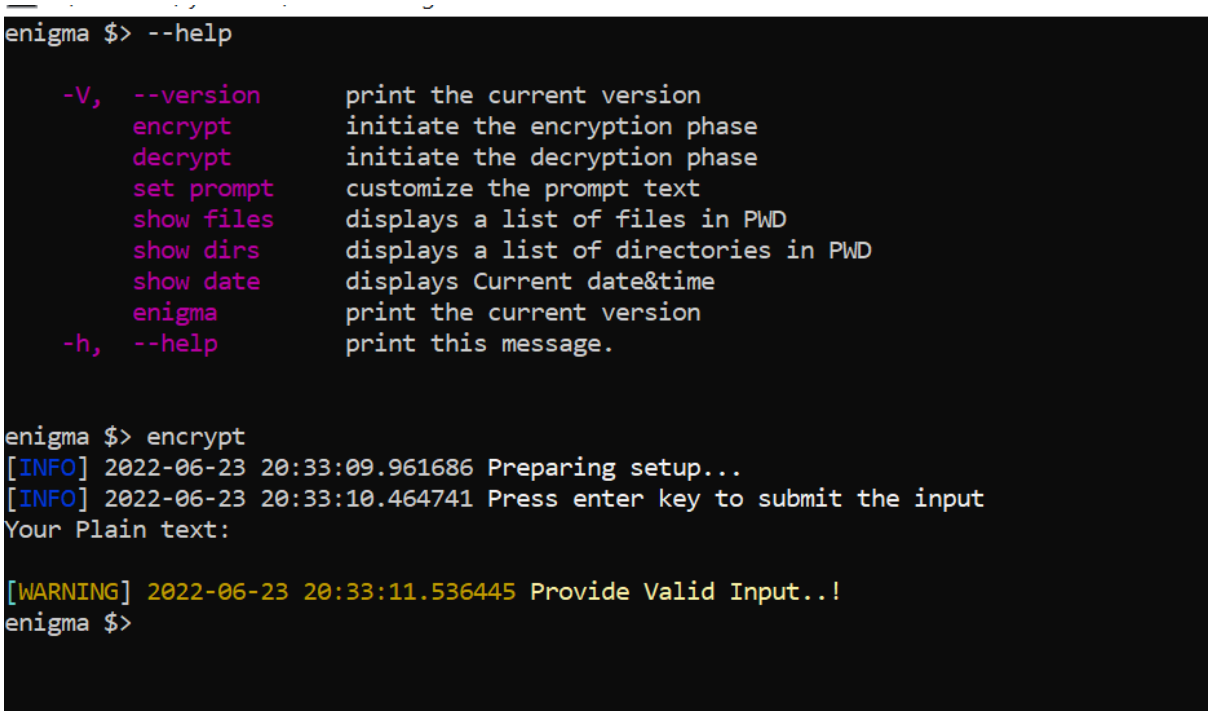

Fig. 8.5 Input Validation

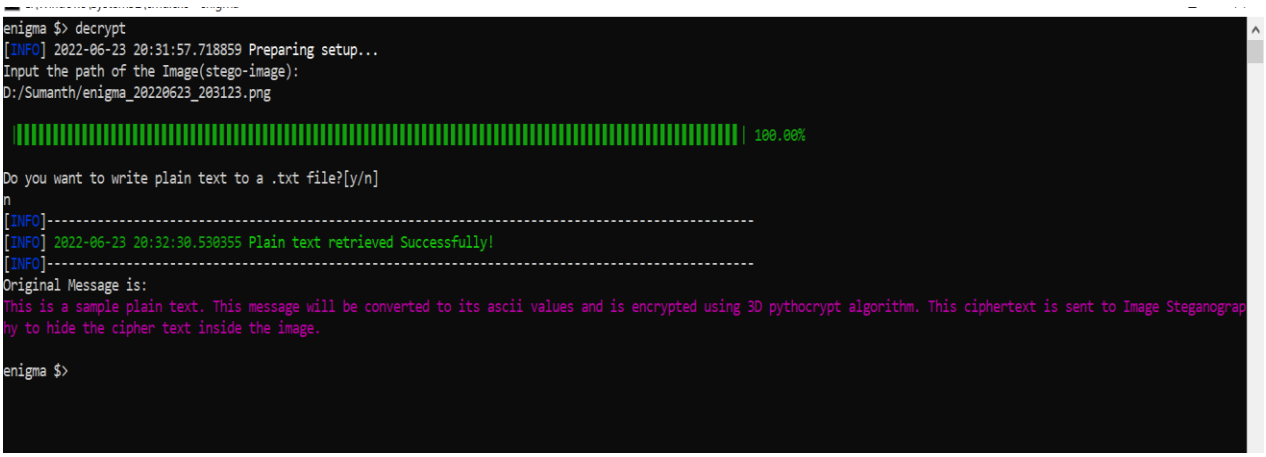

Fig. 8.6 Decrypt command output

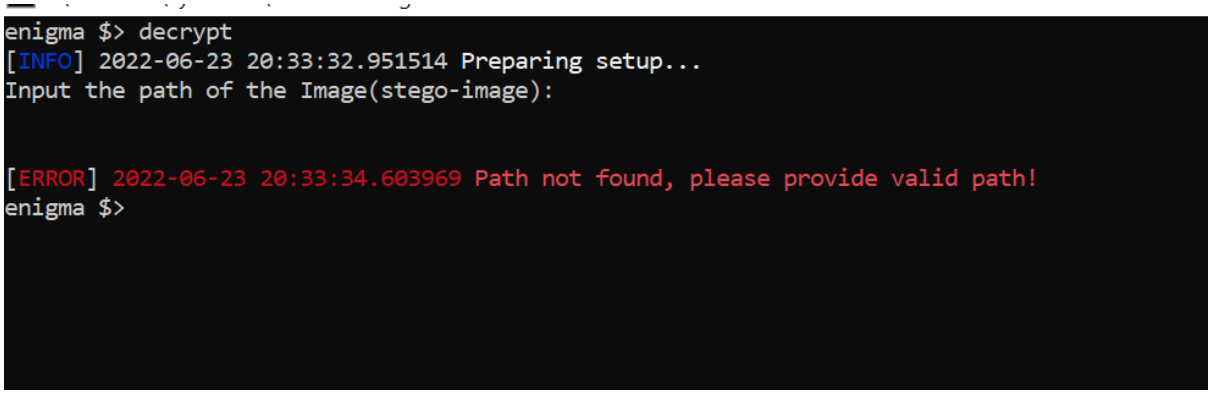

Fig. 8.7 Output file path validation

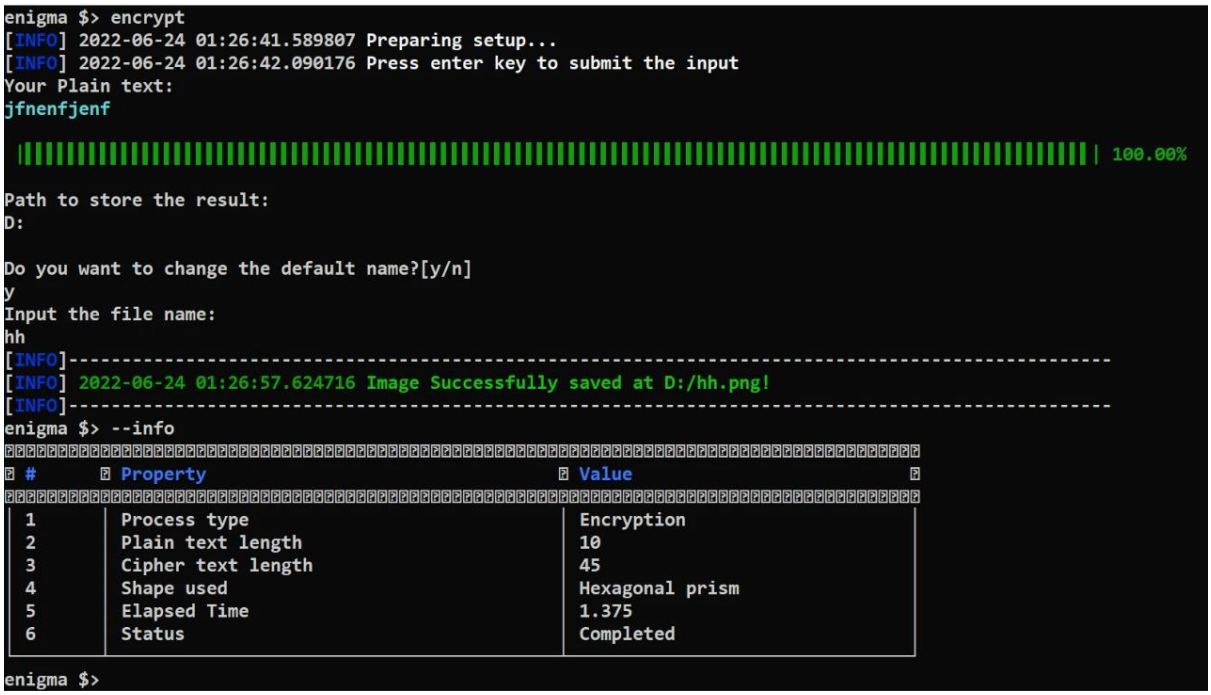

Fig. 8.8 Performance stats

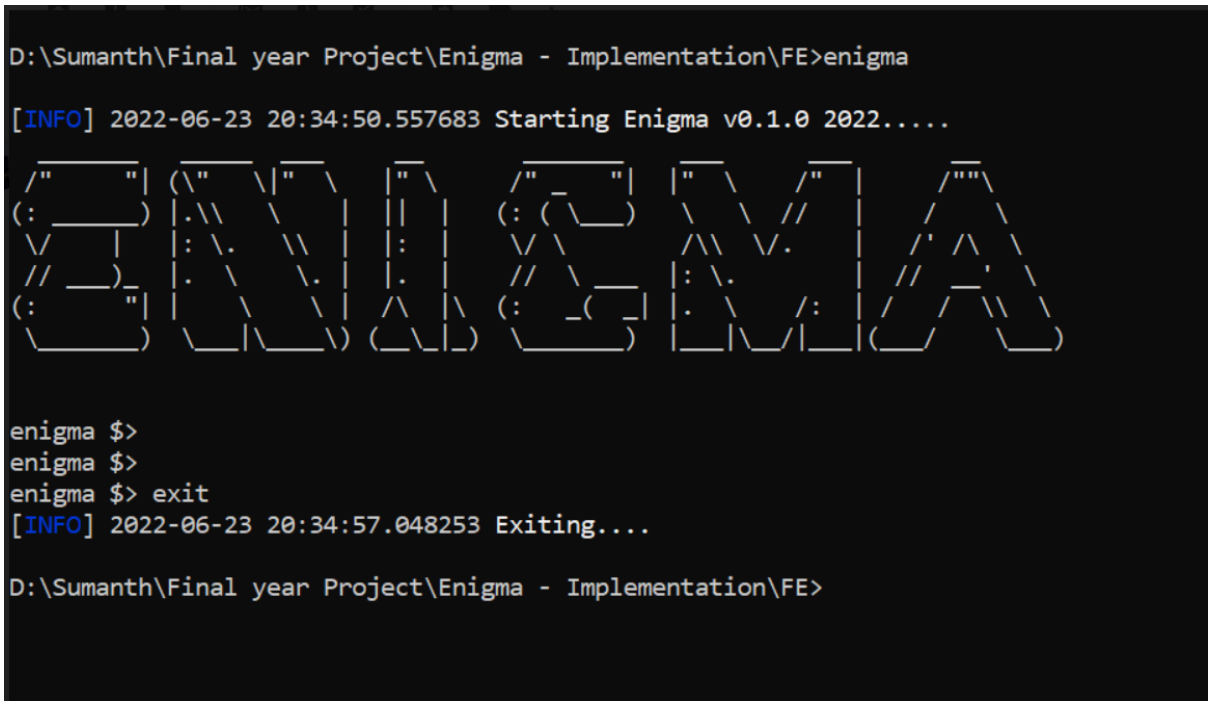

Fig. 8.9 Exit Command

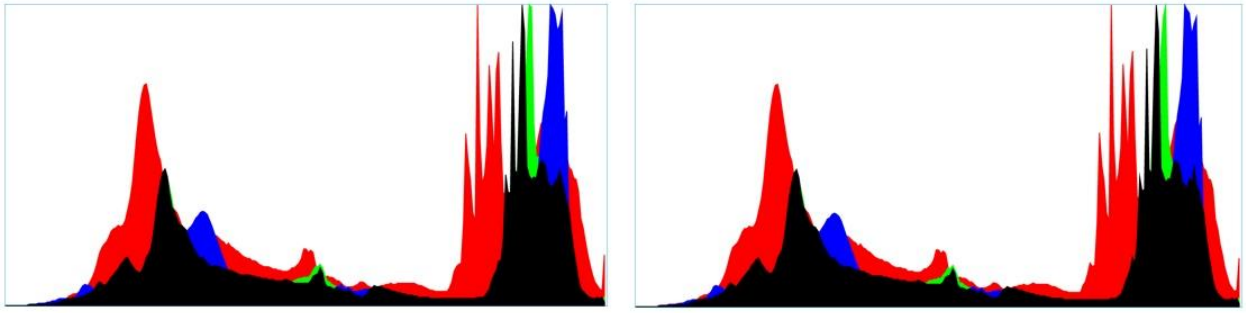

Fig. 8.10 Histogram comparison source image vs output image

SCOPE AND FUTURE WORK

9.1 Scope

Project scope is the part of project planning that involves determining and documenting a list of specific project goals, deliverables, features, functions, tasks, deadlines, and ultimately costs. In other words, it is what needs to be achieved and the work that must be done to deliver a project**.**

The main scope of the project is the encryption and decryption of transmitting message. Whenever a message needs to be transmitted in a communication channel securely, we can use either complete project or only the 3D Pythocrypt segment. Other than transmission, this project can be used

- As a hash function to store the data.
- As an encryption method to store data in cloud.
- Graphical password Authentication.

9.2 Future Work

This part includes the improvisations which can be done on Enigma. Several features can be added to this application in order to enhance it.

- Version 0.2.0 – Implementation of custom user selectable Images.
- Version 1.1.0 – Implementation of text file input for plain text.
- Version 2.1.0 – Implementation of other file formats for plain text inputs like, .pdf, .docx, .xlsx etc.

REFERENCES

[1] Jois, H. S., Bhaskar, N., & Shesha Prakash, M. N. (2015). A 3-d advancement of PythoCrypt for any file type. Journal of Open Innovation: Technology, Market, and Complexity, 1(2). https://doi.org/10.1186/s40852-015-0022-8

[2] Gede, I., Putra Dewangga, A., Purboyo, T. W., & Nugrahaeni, R. A. (2017). A New Approach of Data Hiding in BMP Image Using LSB Steganography and Caesar Vigenere Cipher Cryptography. In International Journal of Applied Engineering Research (Vol. 12). http://www.ripublication.com

[3] Subramanian, N., Elharrouss, O., Al-Maadeed, S., & Bouridane, A. (2021). Image Steganography: A Review of the Recent Advances. IEEE Access, 9, 23409–23423. https://doi.org/10.1109/ACCESS.2021.3053998

[4] https://www.youtube.com/watch?v=P6a7Xw2ZLgE&t=6s

[5] https://realpython.com/python-typer-cli/#demo

[6] https://codeburst.io/building-beautiful-command-line-interfaces-with-python-26c7e1bb54df

[7] https://www.youtube.com/watch?v=BKvj4FH67H8

[8] https://www.youtube.com/watch?v=x1eaT88vJUA

[9] https://www.youtube.com/watch?v=ywiFraNBTv4&list=WL&index=27

[10] https://www.youtube.com/watch?v=AntTxtOWyAI&list=WL&index=29&t=483s

[11] https://www.youtube.com/watch?v=GnSKhetBa48&list=LL&index=10

[12] https://pypi.org/project/pyclick/

[13] https://pypi.org/project/PyDecimal/

[14] https://www.geeksforgeeks.org/python-ascii-art-using-pyfiglet-module/#:~:text=pyfiglet%20takes%20ASCII%20text%20and,pip%20install%20pyfiglet

PROJECT ACHIEVEMENT

We have presented our project at International Conference on Information Processing and Computing (ICIPC) on 27th June, 2022. It is one of the leading international conferences for presenting novel and fundamental advances in the fields of Information Processing and Computing aims to provide a platform for researchers, engineers, academicians as well as industrial professionals from all over the world to present their research results and development activities in Information Processing and Computing and related fields.

International Conference on Information Processing and Computing

ICIPC 2022

www.icipc.in

OAKLAND UNIVERSITY

Jointly organized by

Department of Information Technology, QIS College of Engineering and Technology, Ongole – 523 272, AP, India

And

Oakland University, Rochester, Michigan, United States of America

CERTIFICATE OF EXCELLENCE

This certificate is awarded to

Shrinidhi Holla

From "Jyothi Institute of Technology "for the academic presentation entitled "Enhancement of 3-D Pythocrypt using AES Technique by generating the key using Chitra Kavya "presented at the International Conference on Information Processing and Computing 2022 Virtual Conference, June 27, 2022

Dr. Vijayan Sugumaran
Conference Chair
Oakland University, Rochester, USA

Dr. S. Jafar Ali Ibrahim
Conference Co – Chair
QIS College of Engineering and Technology, Ongole, AP, India

International Conference on Information Processing and Computing

ICIPC 2022
www.icipc.in

OAKLAND UNIVERSITY

Jointly organized by

Department of Information Technology, QIS College of Engineering and Technology, Ongole – 523 272, AP, India

And

Oakland University, Rochester, Michigan, United States of America

CERTIFICATE OF EXCELLENCE

This certificate is awarded to

Sumanth N

From "Jyothi Institute of Technology "for the academic presentation entitled "Enhancement of 3-D Pythocrypt using AES Technique by generating the key using Chitra Kavya "presented at the International Conference on Information Processing and Computing 2022 Virtual Conference, June 27, 2022

Dr. Vijayan Sugumaran
Conference Chair
Oakland University, Rochester, USA

Dr. S. Jafar Ali Ibrahim
Conference Co – Chair
QIS College of Engineering and Technology, Ongole, AP, India

International Conference on Information Processing and Computing

ICIPC 2022
www.icipc.in

OAKLAND UNIVERSITY

Jointly organized by

Department of Information Technology, QIS College of Engineering and Technology, Ongole – 523 272, AP, India

And

Oakland University, Rochester, Michigan, United States of America

CERTIFICATE OF EXCELLENCE

This certificate is awarded to

Prajwal Vs

From "Jyothi Institute of Technology "for the academic presentation entitled "Enhancement of 3-D Pythocrypt using AES Technique by generating the key using Chitra Kavya "presented at the International Conference on Information Processing and Computing 2022 Virtual Conference, June 27, 2022

Dr. Vijayan Sugumaran
Conference Chair
Oakland University, Rochester, USA

Dr. S. Jafar Ali Ibrahim
Conference Co – Chair
QIS College of Engineering and Technology, Ongole, AP, India