**Encryption using Cryptography and Steganography**

**A Project Work**

*Submitted in the partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN COMPUTER SCIENCE (INFO. SECURITY)**

**Submitted by:**

**YASH SAMDANI**

**17BCS3645**

**Under the Supervision of:**

**MISS SEEMA BARDA**

****

**APEX INSITUTE OF TECHNOLOGY**

**CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,**

**PUNJAB**

**MARCH, 2019**

**DECLARATION**

I, **Yash Samdani**, student of **‘Bachelor of Engineering in Computer Science with Info Sec ’**, **session: 2018 - 2019**, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled ‘**Encryption Using Cryptography and Steganography’** is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

**(Yash Samdani)**

**Candidate UID: 17BCS3645**

**Date:**

**Place:**

**CERTIFICATE**

This is to certify that the work embodies in this dissertation entitled ***‘Encryption using Cryptography and Steganography’*** being submitted by **Yash Samdani Roll No. – 17BCS3645** for partial fulfillment of the requirement for the award of **Bachelor of Engineering** in ***Computer Science with Info Sec*** discipline to Apex Institute of Technology, Chandigarh University, Punjab during the academic year 2018 - 2019 is a record of bonafide piece of work, undertaken by him/her the supervision of the undersigned.

**APPROVED AND SUPERVISED BY**

**(MISS SEEMA BARDA)**

**ACKNOWLEDGEMENT**

I have given my full and sincere efforts in this project. However, it would not have been possible without the kind support and help of my team members and various other individuals. I would like to extend my sincere thanks to all of them.

I am highly indebted to Chandigarh University for their guidance and constant supervision as well as for providing necessary information regarding the project and also, for their support in completing the project. I would like to express my gratitude towards my parents and members of Chandigarh University for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to industry persons for giving me such attention and time. My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

**TABLE OF CONTENTS**

* **Abstract**
* **Introduction**
* **Problem Statement**
* **Purpose**
* **Project Scope**
* **Project Overview**
* **System Analysis and Design**
* **Algorithm**
* **Literature Survey**
* **Tools and Software used**
* **Results**
* **References**

**Abstract**

The internet generates very large amount of data on a daily basis. While some of the information are trivial others are sensitive. As a matter of fact, the security of some information traversing the internet is critical to the survival of the user. So now more than ever we need to protect our data from various threats like viruses, hackers or unauthorized users, this can be achieved by encryption. We have mainly two different types of encryption methods, via **Cryptography** and through **Steganography.**

**Cryptography** or **cryptology** is the practice and study of techniques for secure communication in the presence of third parties called adversaries. More generally, cryptography is about constructing and analyzing protocols that prevent third parties or the public from reading private messages; various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation are central to modern cryptography

**Steganography** is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the internet. For hiding secret information in images, there exists a large variety of steganography techniques some are more complex than others and all of them have respective strong and weak points. Different applications may require absolute invisibility of the secret information, while others require a large secret message to be hidden. This project report intends to give an overview of image steganography, its uses and techniques. It also attempts to identify the requirements of a good steganography algorithm and briefly reflects on which stenographic techniques are more suitable for which applications.

**INTRODUCTION**

In today’s world as we head towards the world of digitalization and internet, we are now more than ever are more prone to various attacks of information leak, data loss, viruses, hackers etc. So, while keeping ourselves wide open on internet we need to protect our data from all of them, protecting our data can be done by using encryption, antivirus, firewall etc.

**Encryption** is one of the most important methods for providing data security, especially for end-to-end protection of data transmitted across networks. Plain text is **encrypted** using an **encryption** algorithm and an **encryption** key. This generates an unreadable text which is called as cipher text (**encrypted** data).

This project is an Encryption software which can be used to protect our data by two ways, either by Cryptography or by Steganography. This is the basic project that allows us to encrypt the data present in any text file or we can say takes text file as input and can be used to encrypt the data within the file in any other file like image file. The project is simple GUI based.

**PROBLEM STATEMENT**

Encryption has long been used by militaries and governments to facilitate secret communication. It is now commonly used in protecting information within many kinds of civilian systems. Now a day, there are various security services which are used for secure communication over Internet and other networks are:

## Authentication

## Although messages may often include information about the entity sending a message, that information may not be accurate. Digital signatures can be used to authenticate the source of messages. When ownership of a digital signature secret key is bound to a specific user, a valid signature shows that the message was sent by that user. The importance of high confidence in sender authenticity which is provided in [26] especially obvious in a financial context.

## Integrity

In many scenarios, the sender and receiver of a message may have a need for confidence that the message has not been altered during transmission. Although encryption hides the contents of a message, it may be possible to change an encrypted message without understanding it.

## Non-repudiation

Non-repudiation or more specifically non-repudiation of origin, is an important aspect of digital signatures. By this property, an entity that has signed some information cannot at a later time deny having signed it. Similarly, access to the public key only does not enable a fraudulent party to fake a valid signature.

**PURPOSE**

The purpose of this project is to create an encryption software which can further be improvised to run on various different platforms like android and further can be integrated to various applications like WhatsApp, Slack etc.

**Project Scope**

This project is developed for encryption of data taking input from text file and encrypting the data within the file. This project is developed for hiding information in any image file. The scope of the project is implementation of steganography tools for hiding information includes any type of information file and image files and the path where the user wants to save Image and extruded file.

This is just a basic project but can be used in various fields like cyber Security etc. This project can be used by any two users to share information with each other preventing it from various attacks like man in the middle attacks. Easy and convenient for encryption. The program works for variable size of grid.

**Project Overview**

**Methodology**

User needs to run the application. The user has two-tab options – encrypt and decrypt. If user select encrypt, application give the screen to select image file, information file and option to save the image file. If user select decrypt, application gives the screen to select only image file and ask path where user want to save the secrete file.  
This project has two methods – Encrypt and Decrypt.  
In encryption the secret information is hiding in with any type of image file.  
Decryption is getting the secret information from image file.

**System Analysis & Design**  
  
Steganography system requires any type of image file and the information or message that is to be hidden. It has two modules encrypt and decrypt. Microsoft .Net framework prepares a huge amount of tool and options for programmers that they simples programming. One of .Net tools for pictures and images is auto-converting most types of pictures to BMP format. I used this tool in this software called “Steganography” that is written in C#.Net language and you can use this software to hide your information in any type of pictures without any converting its format to BMP (software converts inside it).   
The algorithm used for Encryption and Decryption in this application provides using several layers lieu of using only LSB layer of image. Writing data starts from last layer (8st or LSB layer); because significant of this layer is least and every upper layer has doubled significant from its down layer. So every step we go to upper layer image quality decreases and image retouching transpires.  
  
The **encrypt module** is used to hide information into the image; no one can see that information or file. This module requires any type of image and message and gives the only one image file in destination.  
  
The **decrypt module** is used to get the hidden information in an image file. It take the image file as an output, and give two file at destination folder, one is the same image file and another is the message file that is hidden it that.

**System Specification:**

**Hardware Specification:**

➢ Processer: Pentium – III or more (i5 Processor Recommended)

➢ RAM: >512 MB

➢ Hard Disk: >2GB

➢ Internet Connection

**Software Specification:**

➢ Windows 10 Operating System

➢ PyCharm / Python 3.6

## Algorithm

**For Steganography🡪**

Encryption:

* NumPy and PIL are imported.
* Tow variable are defined one stores the data which has to be stored in image and one stores the length of data.
* The image file is opened and the pixel data is stored in array which is converted and stored into a list.
* The file is opened in read mode and convert every entity of data into 8 bit binary format and store it in string format.
* We have to check the data which has to be encrypted should not be larger then the image pixel data which we had stored in list.
* Then

1.converted a single value of the pixel to binary format

2.changed the last two digits of binary value with the data

3.Converted to integer Again

4.If i+2==l then final 2 digits got appended and text ended

We have to check after end of every pixel as l%2 can be any

* We save the image in .png format not in .jpeg because there is loss of data but not in .png.

Decryption:

* NumPy and PIL are imported.
* The image is opened and pixels are stored in array with the help of NumPy.
* With the help of loop, we take out the binary format of color at every pixel.
* The binary format should not be greater the 8 bits.
* We take the last two least significant bits which we had replaced in encryption that is the data.
* We store the data in string format.
* We store that string formatted data inside the file which is the real data.

**For cryptography**

* Tkinter is imported as tk
* Import askopenfilename from tkinter.filedialog

Encryption:

* Open the file you want to encrypt
* Read the data from the file and store it in msg
* Open the key file and store its value
* Use the following code to get the encrypted data:

key\_c = ord(key[i % len(key)])

msg\_c = ord(c)

encryped.append(chr((msg\_c + key\_c) % 127))

* Store the encrypted data in a file
* Close the file.

Decryption:

* Open the encrypted file and store its data
* Open the key file and store its data
* Decrypt the data using the following code:

key\_c = ord(key[i % len(key)])

enc\_c = ord(c)

msg.append(chr((enc\_c - key\_c) % 127))

* Store the decrypted data in a file
* Close the file.

Note: Later on, both cryptography and steganography are combined in a single program and are connected with Graphical User Interface.

**LITERATURE SURVEY**

**Introduction to Encryption and Decryption**

Data encryption is the conversion of data into a form, called a ciphertext, that cannot be easily understood by unauthorized people. Decryption is the process of converting encrypted data back into its original form, so it is easily understood. Encryption is a mechanism for hiding information by turning readable text into a stream of gibberish in such a way that someone with the proper key can make it readable again.

About 1900 BC An Egyptian scribe used non-standard hieroglyphs in an inscription. Kahn lists this as the first documented example of encryption (written cryptography).

Encryption helps to you protect the privacy of your messages, documents and sensitive files. In its earliest form, people have been attempting to conceal certain information that they wanted to keep to their own possession by substituting parts of the information with symbols, numbers and pictures, this paper highlights in chronology the history of Cryptography throughout centuries. For different reason humans have been interested in protecting their messages.

**Types of Encryption Algorithms:**

**Symmetric Key Algorithms**

Symmetric key encryption algorithms use a single secret key to encrypt and decrypt data. You must secure the key from access by unauthorized agents because any party that has the key can use it to decrypt data. Secret-key encryption is also referred to as symmetric encryption because the same key is used for encryption and decryption. Secret-key encryption algorithms are extremely fast (compared to public-key algorithms) and are well suited for performing cryptographic transformations on large streams of data.

Brief definitions of the most common encryption techniques are given as follows:

DES (Data Encryption Standard), was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology). DES is (64 bits key size with 64 bits block size).  DES also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt. The key ostensibly consists of 64 bits; however, only 56 of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56 bits, and it is always quoted as such. Every 8th bit of the selected key is discarded, that is, positions 8, 16, 24, 32, 40, 48, 56, 64 are removed from the 64 bit key leaving behind only the 56 bit key. Since that time, many attacks and methods recorded the weaknesses of DES, which made it an insecure block cipher.

3DES is an enhancement of DES; it is 64-bit block size with 192 bits key size. In this standard the encryption method is similar to the one in the original DES but applied 3 times to increase the encryption level and the average safe time. It is a known fact that 3DES is slower than other block cipher methods.

RC2 is a block cipher with a 64-bits block cipher with a variable key size that range from 8 to128 bits. RC2 is vulnerable to a related-key attack using 234 chosen plaintexts. RC2 is a 64-bit block cipher with a variable size key. Its 18 rounds are arranged as a source-heavy Feistel network, with 16 rounds of one type (MIXING) punctuated by two rounds of another type (MASHING).

Blowfish is block cipher 64-bit block - can be used as a replacement for the DES algorithm. It takes a variable length key, ranging from 32 bits to 448 bits; default 128 bits. Blowfish is unpatented, license-free, and is available free for all uses. Blowfish has variants of 14 rounds or less. Blowfish is successor to Twofish.

AES is a block cipher. It has variable key length of 128, 192, or 256 bits; default 256. it encrypts data blocks of 128 bits in 10, 12 and 14 rounds depending on the key size. AES encryption is fast and flexible; it can be implemented on various platforms especially in small devices. Also, AES has been carefully tested for many security applications. AES is based on a design principle known as a substitution-permutation network, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits.

**Asymmetric Key Algorithms**

Asymmetric Key Algorithm has been said to be the most significant new development in cryptography in the last 300-400 years. Modern PKC (Public Key Cryptography) was first described publicly by Stanford University professor Martin Hellman and graduate student Whitfield Diffie in 1976.

Asymmetric key algorithm is also called Public-key Algorithm. Public-key cryptography is a fundamental and widely used technology around the world, and enables secure transmission of information on the internet and other communication systems; this concept was proposed in. It is also known as asymmetric cryptography because the key used to encrypt a message differs from the used to decrypt it. In public-key cryptography, a user has a pair of cryptographic keys – a public-key and a private-key. The private-key is kept secret, while the public-key may be widely distributed and known for any user. Messages are encrypted with the recipient’s public key and can only be decrypted with the corresponding private key.

Generic PKC employs two keys that are mathematically related although knowledge of one key does not allow someone to easily determine the other key. One key is used to encrypt the plaintext and the other key is used to decrypt the ciphertext. The important point here is that it does not matter which key is applied first, but that both keys are required for the process to work. Because a pair of keys is required, this approach is also called asymmetric cryptography. In PKC, one of the keys is designated the public key and may be advertised as widely as the owner wants. The other key is designated the private key and is never revealed to another party. It is straight forward to send messages under this scheme.

**Tools and Software used**

**PyCharm**

PyCharm is the most popular IDE for Python, and includes great features such as excellent code completion and inspection with advanced debugger and support for web programming and various frameworks. PyCharm is created by Czech company, Jet brains which focusses on creating integrated development environment for various web development languages like JavaScript and PHP.

**Python 3.6**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

## History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

**Python Libraries Used**

**NumPy**

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

**Numeric**, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.

## Operations using NumPy

Using NumPy, a developer can perform the following operations −

* Mathematical and logical operations on arrays.
* Fourier transforms and routines for shape manipulation.
* Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

**PIL**

Pillow is a Python Imaging Library (PIL), which adds support for opening, manipulating, and saving images. The current version identifies and reads a large number of formats. Write support is intentionally restricted to the most commonly used interchange and presentation formats.

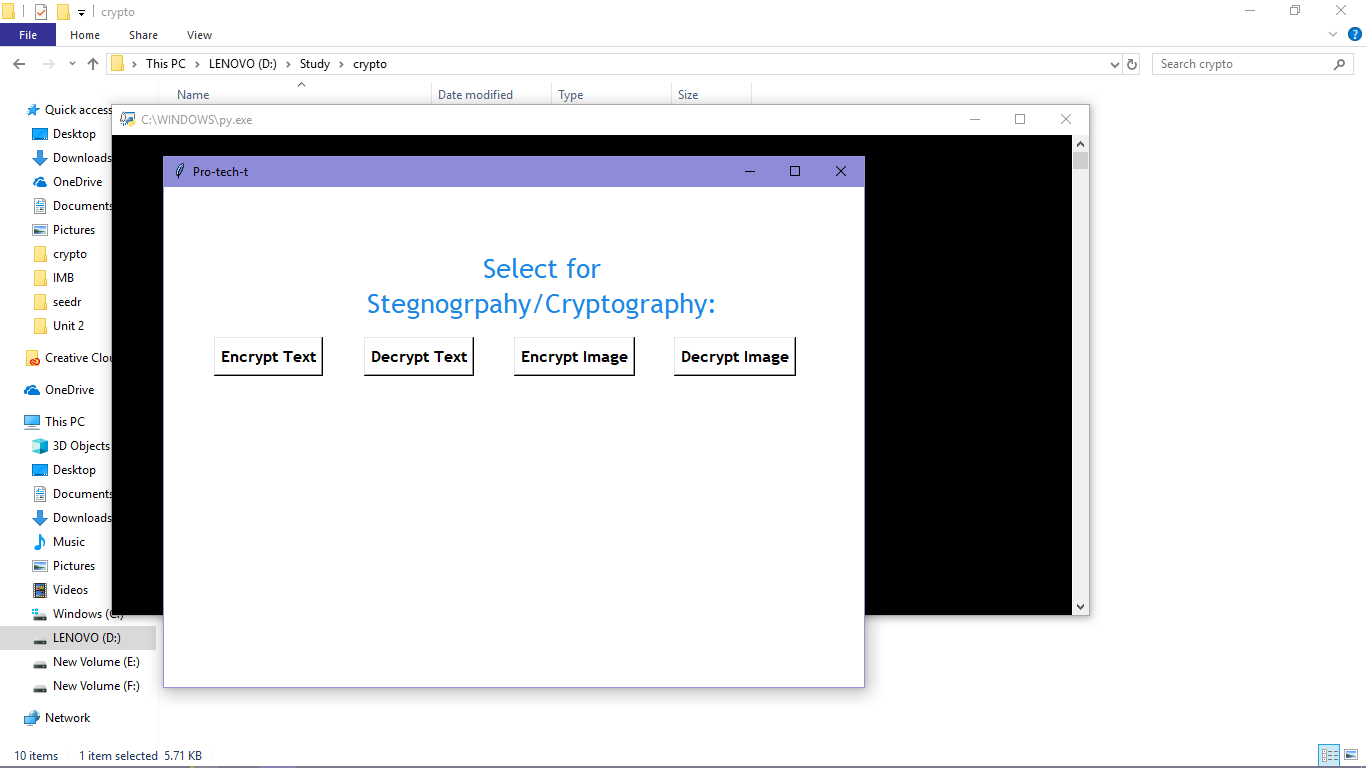
**Tkinter**

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

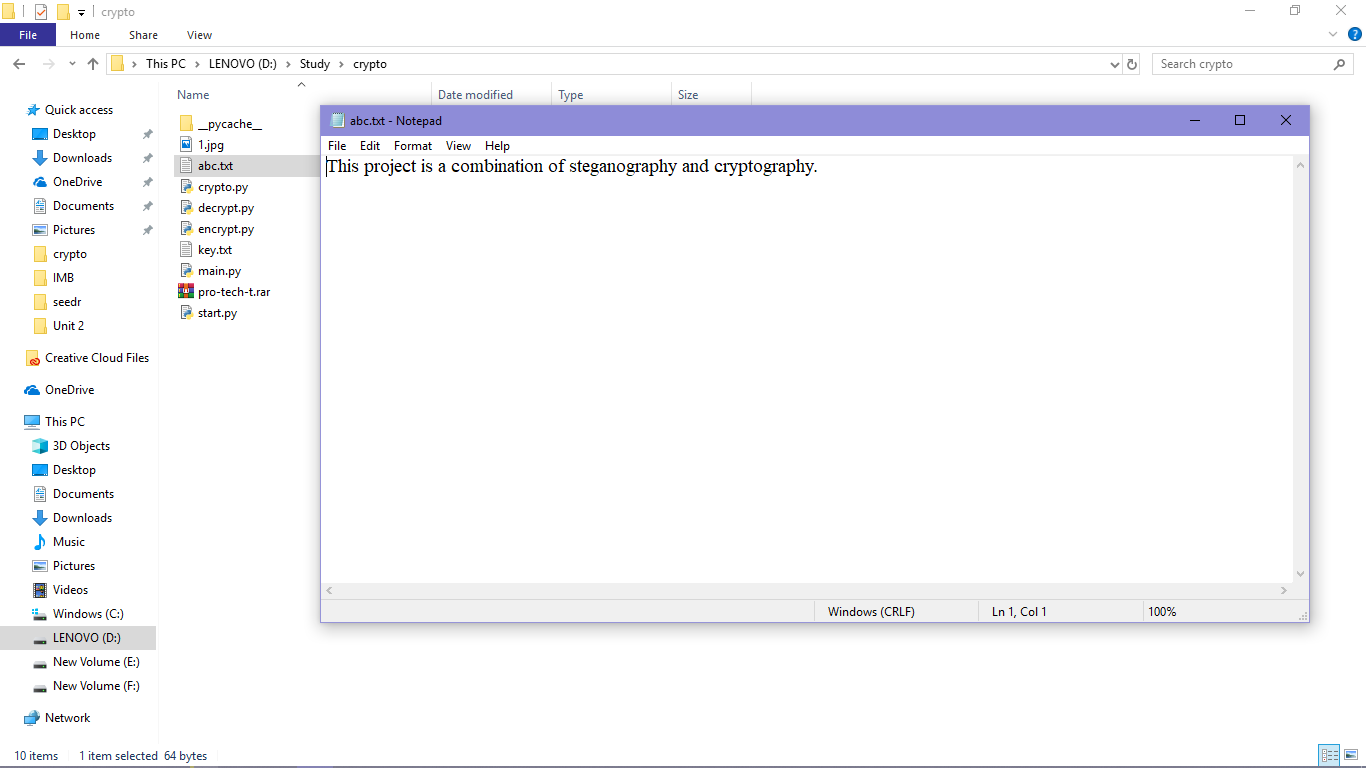
Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

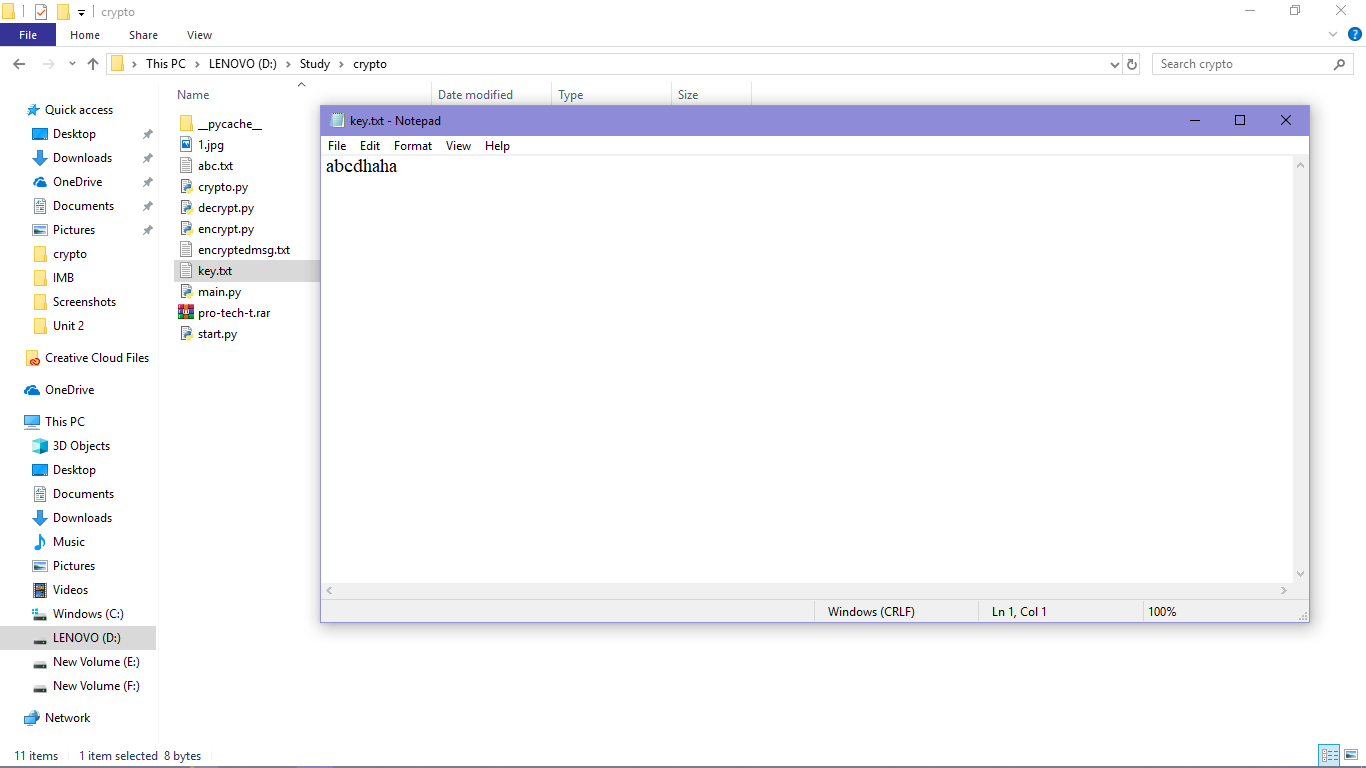
* Import the *Tkinter* module.
* Create the GUI application main window.
* Add one or more of the above-mentioned widgets to the GUI application.
* Enter the main event loop to take action against each event triggered by the user.

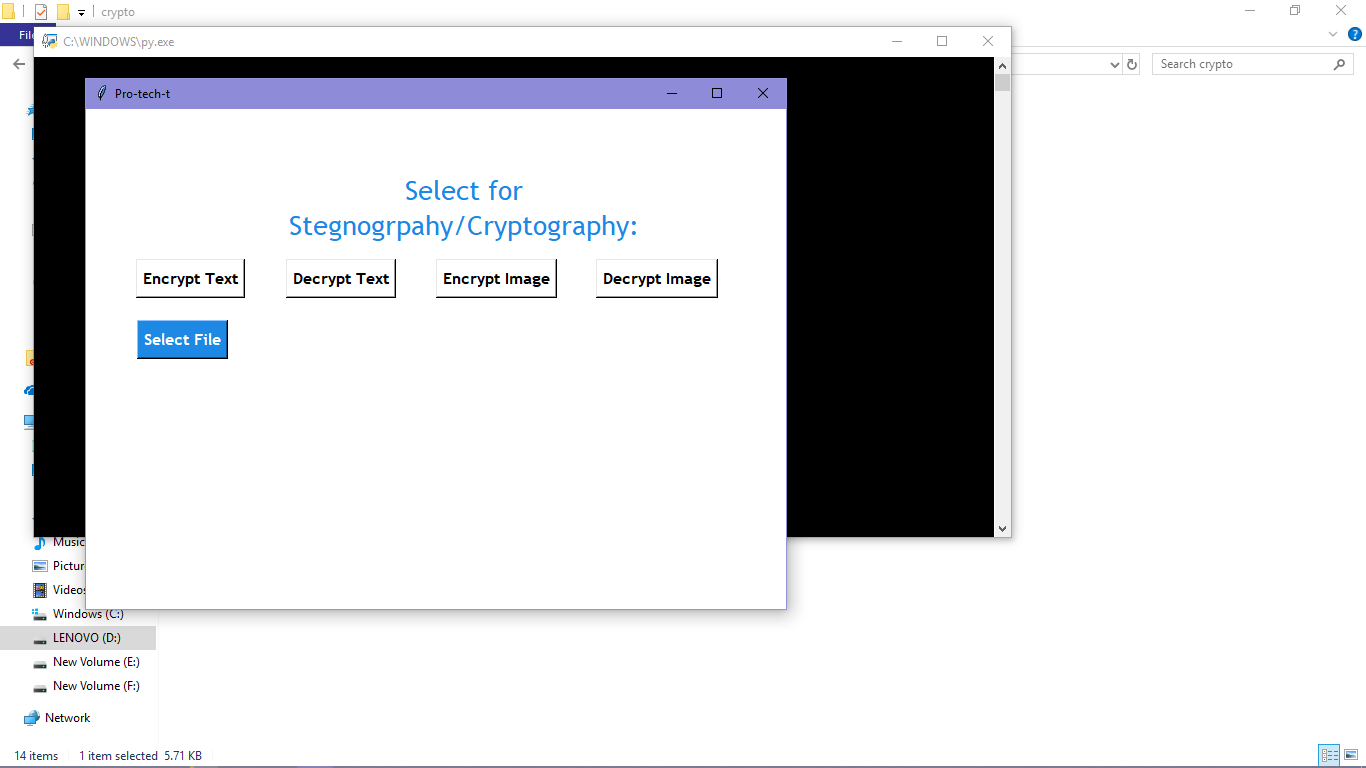
**Results**



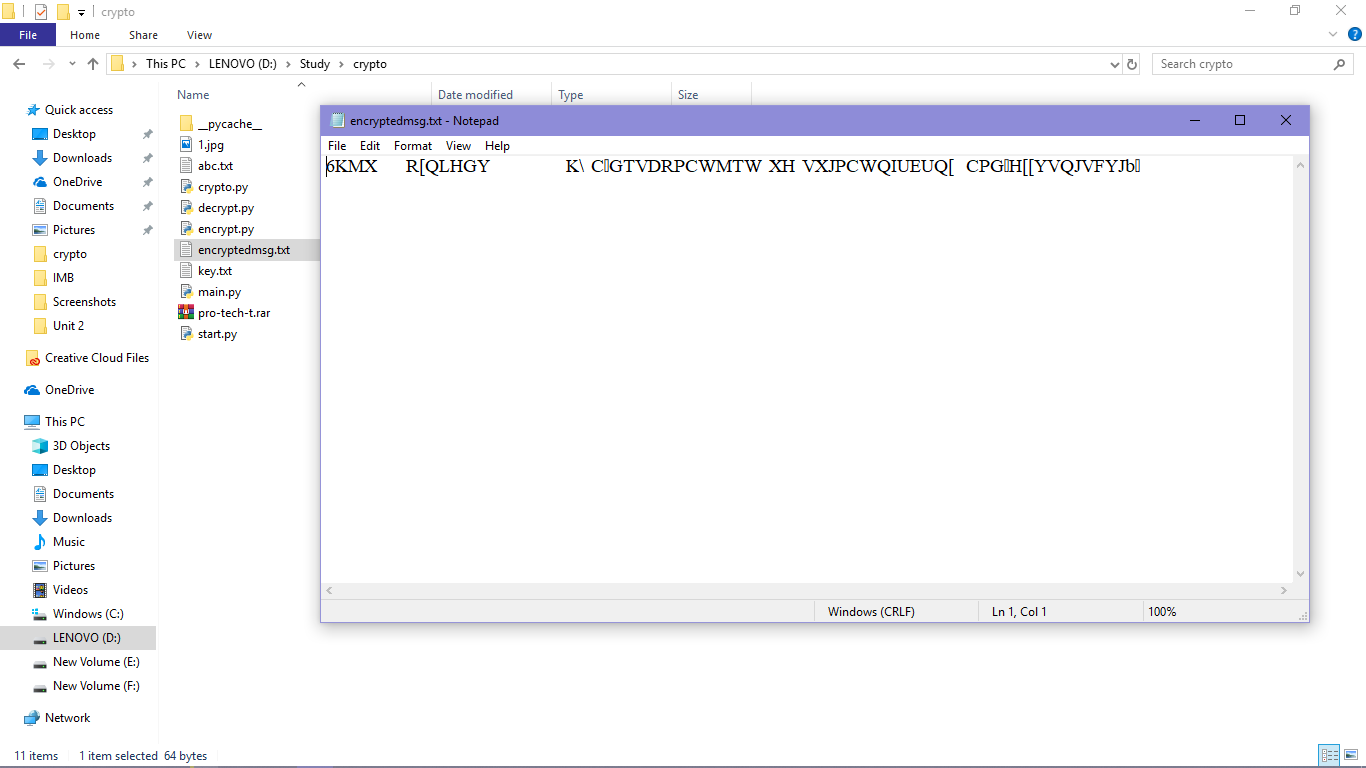
* First, we encrypt the text file “abc.txt”



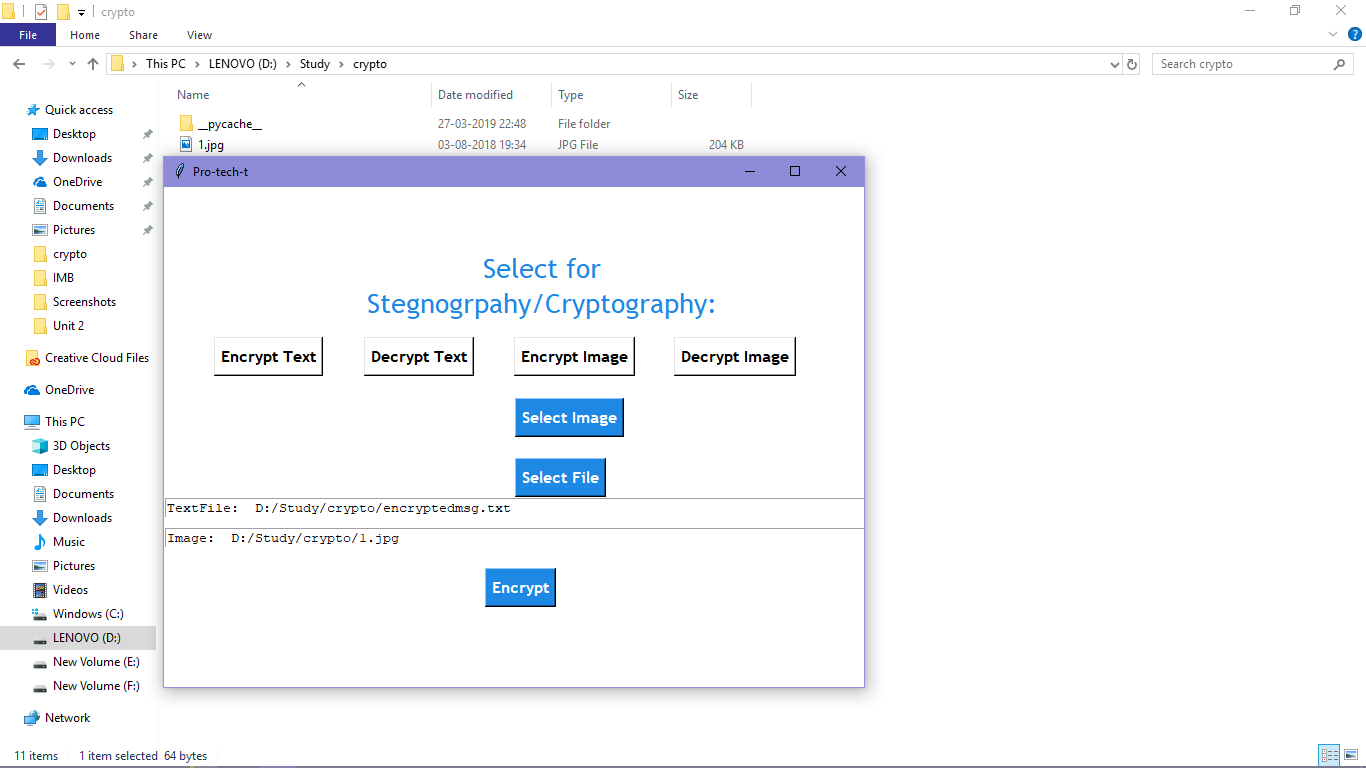




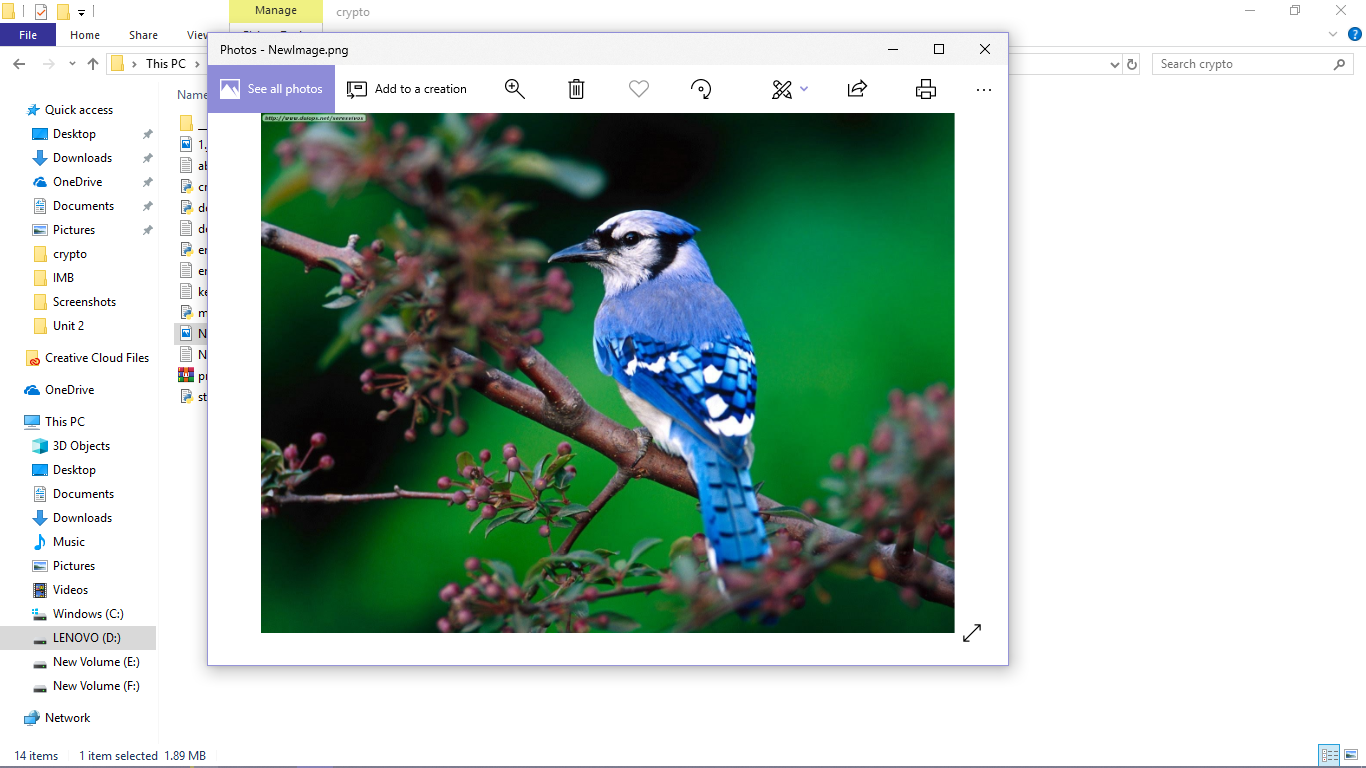
Encrypted data:



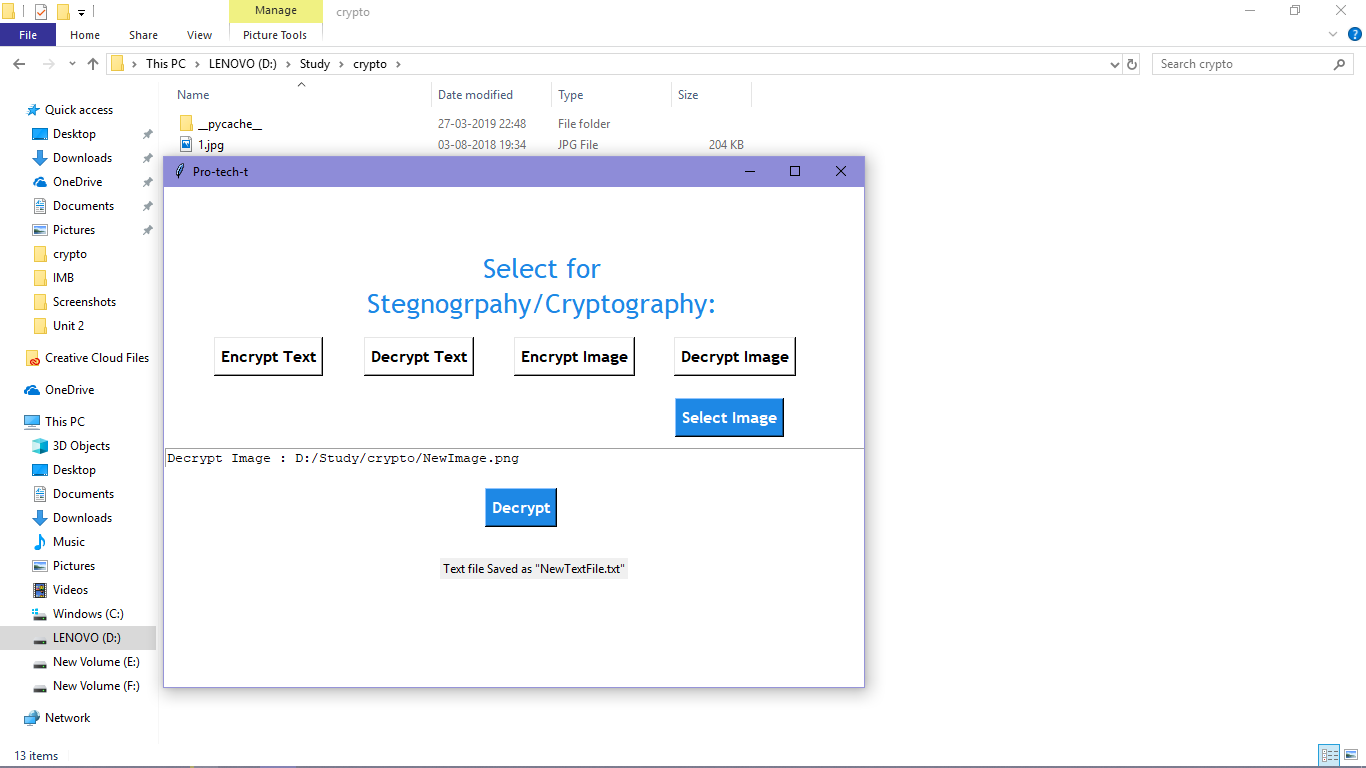
* Then we embed the encrypted text with an image file



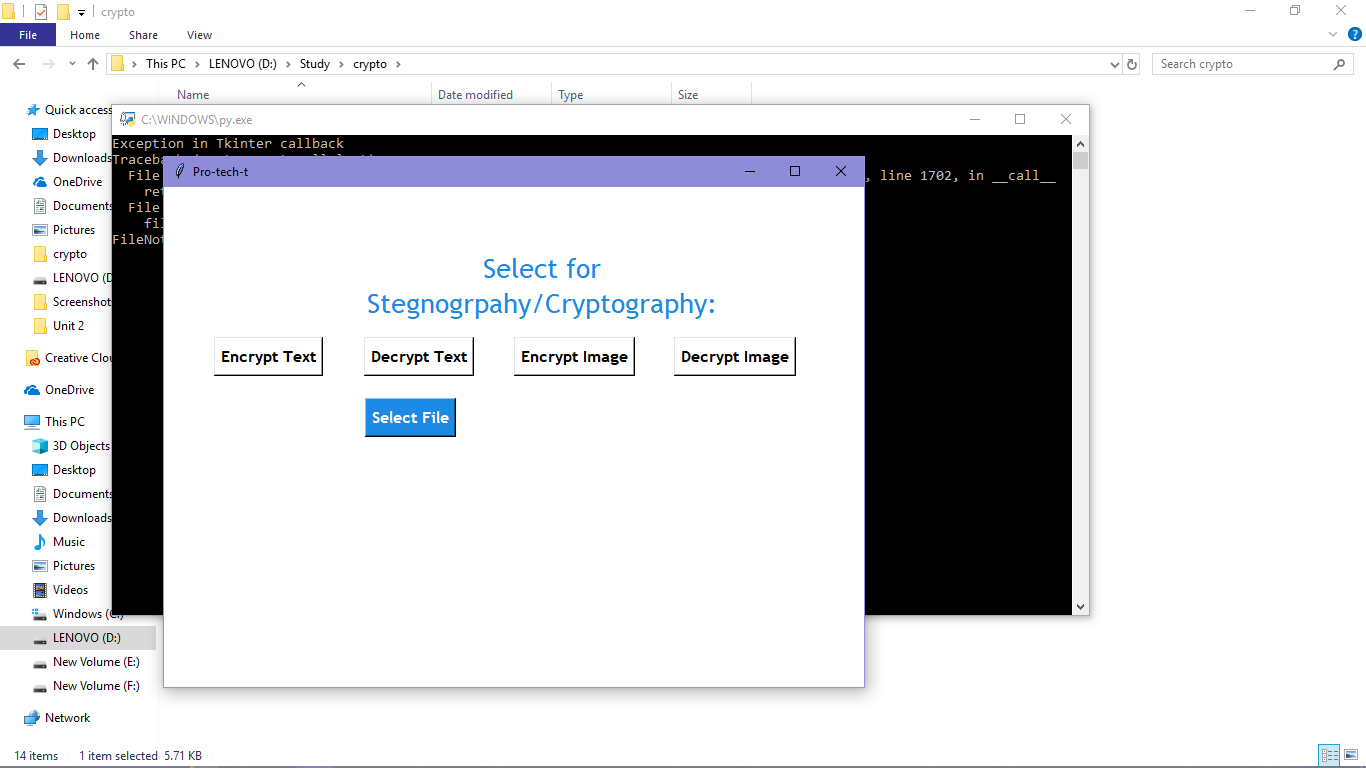
Steganographic image:



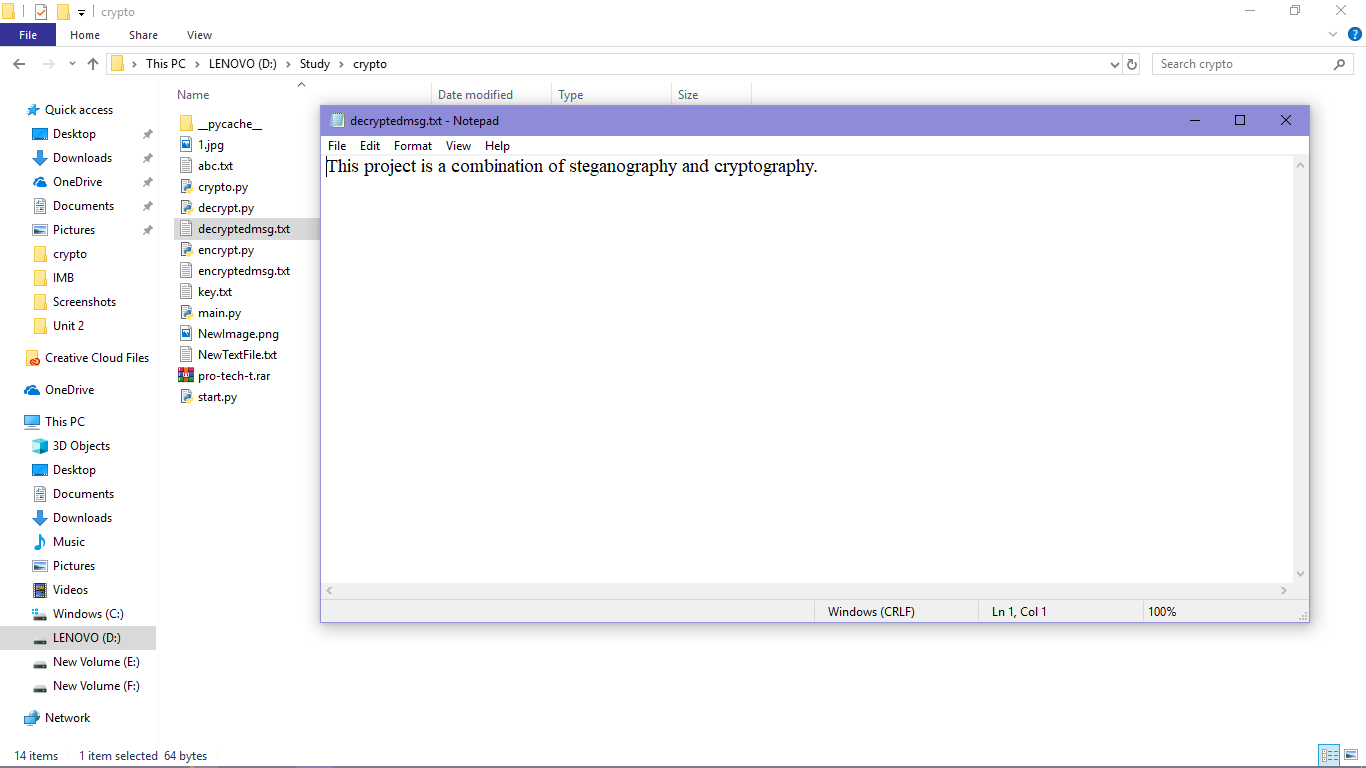
* For decoding the data, we extract the encrypted data from the steganographic image file



* Then we decrypt the encrypted data



Decrypted data:



* Hence, data is once again achieved.

**References**

* <https://www.tutorialpoint.com>
* <https://www.python.org/>
* <https://www.numpy.org/>
* <https://www.geeksforgeeks.org/>
* [www.tkinter.org/](http://www.tkinter.org/)