Introduction

Cryptography **is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient.**
While cryptography is the science of securing data, *cryptanalysis* is the science of analyzing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck. Cryptanalysts are also called *attackers*.

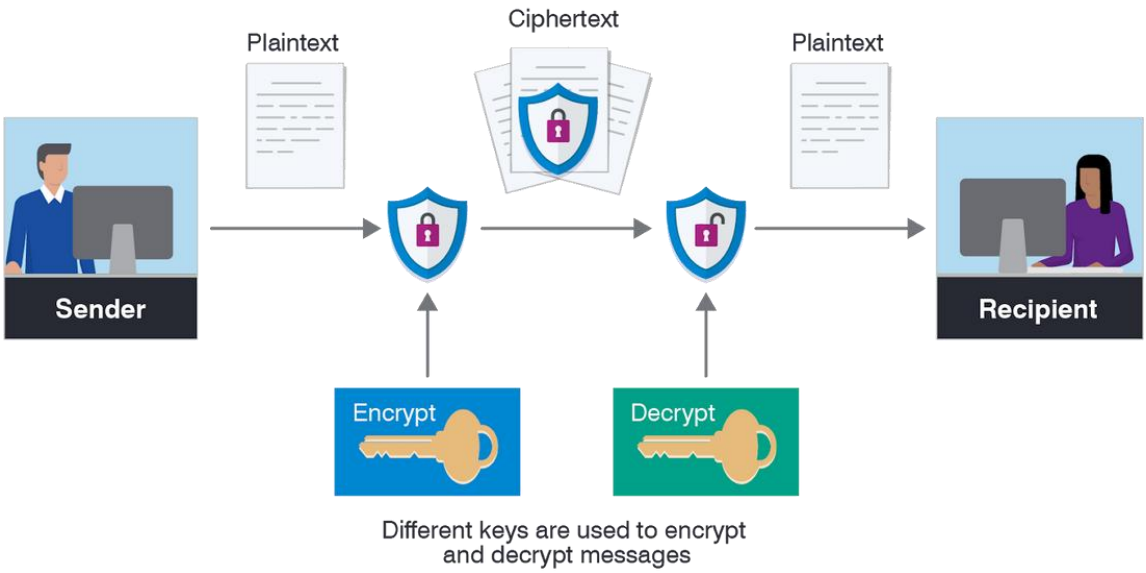
NEED OF CRYPTOGRAPHY

Cryptography in digital world offers three core area that protect you and your data from attempt theft, theft or an unauthorize use of your data and possible fraud. Cryptography covers this essential area; authentication, integrity, and confidentiality

Authentication: Authentication is a process in Cryptography that offers certificates as a solution, which are called "digital IDs," coz they can be used to verify the identity of someone you don't know. Hence it is up to you to decide whether someone is authentic or not.

Integrity: Integrity is about how you protect your data, corresponding to that certificate it can be used in another technique that's "digital signatures", to ensure that nobody can impersonate you. One can easily forge email, but it's very hard to forge a digitally signed email message and so on it's hard for someone to modify or manipulate a message that you have digitally signed.

Confidentiality: By using Cipher you can keep your information secret especially when you send sensitive data over a network. How can you be sure that nobody finds out about your financial transactions, or your personal records, or your other secret information? It can give you solution through "cipher". A cipher is intelligent system that know how to encrypt and decrypt data. Before you send sensitive data over a network, or store it on a disk, you can encrypt it, which turns it unreadable. If you need the data again, you can use the cipher to decrypt the data. Now you are the only person that can be able to decrypt the data. If you're sending data to someone, you can ensure that only that person is able to decrypt the message. Also, it is important to learn about key data management, public and private key encryption, and how to includes a secure talk application that encrypts all data sent over the network.



Excysting Method of Encryption

There are different encryption methods based on the type of keys used, key length, and size of data blocks encrypted. Here we discuss some of the common encryption methods.

1. Advanced Encryption Standard (AES)

Advanced Encryption Standard is a symmetric encryption algorithm that encrypts fixed blocks of data (of 128 bits) at a time. The keys used to decipher the text can be 128-, 192-, or 256-bit long. The 256-bit key encrypts the data in 14 rounds, the 192-bit key in 12 rounds, and the 128-bit key in 10 rounds. Each round consists of several steps of substitution, transposition, mixing of plaintext, and more. AES encryption standards are the most commonly used encryption methods today, both for data at rest and data in transit.

2. Rivest-Shamir-Adleman (RSA)

Rivest-Shamir-Adleman is an asymmetric encryption algorithm that is based on the factorization of the product of two large prime numbers. Only someone with the knowledge of these numbers will be able to decode the message successfully. RSA is often used in digital signatures but works slower when large volumes of data need to be encrypted.

3. Triple Data Encryption Standard (TripleDES)

Triple Data Encryption Standard is a symmetric encryption and an advanced form of the DES method that encrypts blocks of data using a 56-bit key. TripleDES applies the DES cipher algorithm three times to each data block. TripleDES is commonly used to encrypt atm and Unix passwords.

4. Twofish

Twofish is a license-free encryption method that ciphers data blocks of 128 bits. It's considered the successor to the **Blowfish** encryption method that ciphered message blocks of 64 bits. Twofish always encrypts data in 16 rounds regardless of the key size. Though it works slower than AES, the Twofish encryption method continues to be used by many files and folder encryption software solutions.

Proposed Method with Architecture

3DES encryption

3DES is an encryption cipher that was derived from the original Data Encryption Standard (DES). It became prominent in the late nineties, but has since fallen out of favor due to the rise of more secure algorithms.

3DES algorithm uses the Data Encryption Standard (DES) cipher three times to encrypt its data.

3DES algorithm uses the Data Encryption Standard (DES) cipher three times to encrypt its data.

As the security weaknesses of DES became more apparent, 3DES was proposed as a way of extending its key size without having to build an entirely new algorithm. Rather than using a single key as in DES, 3DES runs the DES algorithm three times, with three 56-bit keys:

- Key one is used to **encrypt** the plaintext.
- Key two is used to **decrypt** the text that had been encrypted by key one.
- Key three is used to **encrypt** the text that was decrypted by key two.

In each stage, the complete DES process is followed as outlined above.

Now, you may be wondering “How can applying decryption in the second step enhance security?”

The answer is that it uses a separate key. If the first key was also used to decrypt the data in the second step, then the data would be right back where it started.

However, since it uses a different key, the decryption process doesn’t actually serve to decrypt the data. It may seem logically perverse, but decrypting with a separate key only serves to jumble up the data even further.

Once the second key has “decrypted” the data, the third key is applied to encrypt it again. The result is the 3DES ciphertext.

3DES is structured this way because it allows implementations to be compatible with single key DES, two key DES and three key DES (these are covered in the following section). This would not work if encryption was used in all three steps.

Implementation:

3DES is believed to still be secure because it requires 2^112 brute-force operations which is not achievable with foreseeable technology.

We will be required some python libraries to make our code quick responsive and secure. This are some standard libraries which are written using higher mathematical encryption methodology so to make it more secure toward brute force and any other attack.

This are the libraries that I have used here:-

1. **Crypto. Cipher: This package contains algorithms for protecting the confidentiality of data.**
2. **Crypto. Hash: Cryptographic hash functions take arbitrary binary strings as input, and produce a random-like fixed-length output (called *digest* or *hash value*).**
3. **Getpass: Many programs that interact with the user via the terminal need to ask the user for password values without showing what the user types on the screen. The getpass module provides a portable way to handle such password prompts securely.**
4. **Crypto.Protocol.KDF (Key Derivation Function): This module contains a collection of standard key derivation functions. A key derivation function derives one or more secondary secret keys from one primary secret (a master key or a pass phrase). This is typically done to insulate the secondary keys from each other, to avoid that leakage of a secondary key compromises the security of the master key, or to thwart attacks on pass phrases (e.g., via rainbow tables).**

Now we all have what we are using is this program. so now let's understand how we implemented the algorithm using python.

1. **Starting from the main function at the very beginning we ask user whether he wants to encrypt or decrypt then according to his/her input the program will be assigned to a specific function. If any invalid or wrong input is fed then next after the program will report user to check and correct the input he selected.**

2. **If the input indicates toward the encryption, then the Code will be direct to the encryption function. Before that we have to input the name of the media image that we want to encrypt and make sure that image lies in the same directory as where your program.**

3. **We had passed the image to function which is then passed to the read function which convert and read the image in the form of bits. After that program ask for the key to the use which should be minimum 8-character log so that it should match the triple DES criteria.**

4. **Then both the key and the image bit string is passed to the encryption function which encrypt the message in the following sequence: 1. Encrypt with key one, 2. Decrypt with key second, and then again encrypt with third key. It is to take note that here key 1,2,3 is not at all same. in fact, it will be meaningless if we keep all three keys same. And our libraries take care of that part. Using their mathematical algorithms.**

5. **After this the output is stored in the file with the same extension in the same directory but this time it is encrypted and if you try to open it you will encounter an error.**

6. **Here the process of encryption over. Now let's look at the decryption.**

7. **Here input is encrypted media file which we want to decrypt and have it in its original form and use the same key which is used at the encryption time.**

8. **Decryption takes place in the following steps: 1. Decryption, 2. Encryption,3. Decryption**

9. **And then the decrypted file stored back in its original form in the same folder where program is stored.**

Conclusion:

We have achieved our expected result that is to encrypt our image so that we can transfer it to anywhere without bothering about its security.