**A New Intensity Transform Based on Bit Plane Slicing.**

**Abstract**

In this project we will aim to layer several different image encryption techniques that have been proposed in the literature around bit-plane slicing, such as the rotation method or the XOR method, to create an algorithm that is even less susceptible to brute force techniques.

One layer of the transforms will feature an intensity transform inspired by bit plane slicing technique to further increase the complexity of the encryption.

**Introduction**

With the worlds’ rapid movement towards digitalisation of communication, security and privacy of information are becoming more and more important, leading to large amounts of research going into the study of data encryption techniques in this domain. Optical encryption techniques gather a lot of interest as they provide the ability to process large images very quickly to produce results that are very difficult to brute force as they allow hiding of information along different transforms.

Steganography is the umbrella term used for techniques that use transforms on images to hide the original data. Steganography is several applications in various fields such as image copyright protection, fraud detection, copy control etc.

Steganography is also used on medical records and aadhar cards to conceal private information to protect the identities of individuals.

The term “cover image” describes the medium in which secret information is embedded. Modern image data hiding uses two basic principles for concealing data, namely the embedding and extraction algorithms. The essential components of the embedding algorithm are a cover image, secret data, and a secret key. Stego-image is obtained by concealing the secret data into the cover image using the embedding algorithm. The stego image is then used as an input object to extract the secret data.

Image steganography uses image processing techniques for embedding.

The embedding process in data hiding uses a few basic schemes and plays an important role in any stego-image. The commonly used techniques in steganography are based on the spatial domain, but for this project we will be combining transforms on the intensity domain as well.

**Scope of the study**

This study will focus on using several encryption techniques based on bit plane slicing to create an encryption algorithm that is extremely difficult to brute force. The primary focus will be see how several different bit plane slicing algorithms can be applied on top of one another to encrypt an image, and to see the increase in encryption efficiency when using an intensity transform that uses bit plane slicing as its base.

**Objective**

to use bit-plane slicing as the base and then add additional special transforms on top that use randomly generated cat maps to encrypt the image. This will result in an encryption algorithm that uses the same base and therefore is able to create very strong encryptions for a very small additional overhead each time.

**Methodology**

For this project, we will start by selecting a greyscale image for simplicity, but this process can very easily be carried out on a 3 or 4 channel image by applying the same transforms across each channel individually.

The procedure for the project will be the following.

1. Break the image into its individual bit planes (8 planes for an 8-bit image).
2. Apply various transforms on the spatial domain on each bit plane separately. For this project we will apply the same transforms on all the bit planes, but different transforms can be applied on different planes.
3. Apply different transforms on the intensity domain on each bit plane separately. For this project we will apply the same transforms on all the bit planes, but different transforms can be applied on different planes.
4. Apply bit-plane reversal randomisation (Novel)
5. Apply all steps in reverse order to obtain the original image from the encrypted image (decryption).

**Process**

This section will showcase a basic application built using 2 different transforms. The first one is the bit intensity transform[1], followed by a special transform based on the chaos randomised encryption technique[2] with the application of filters.

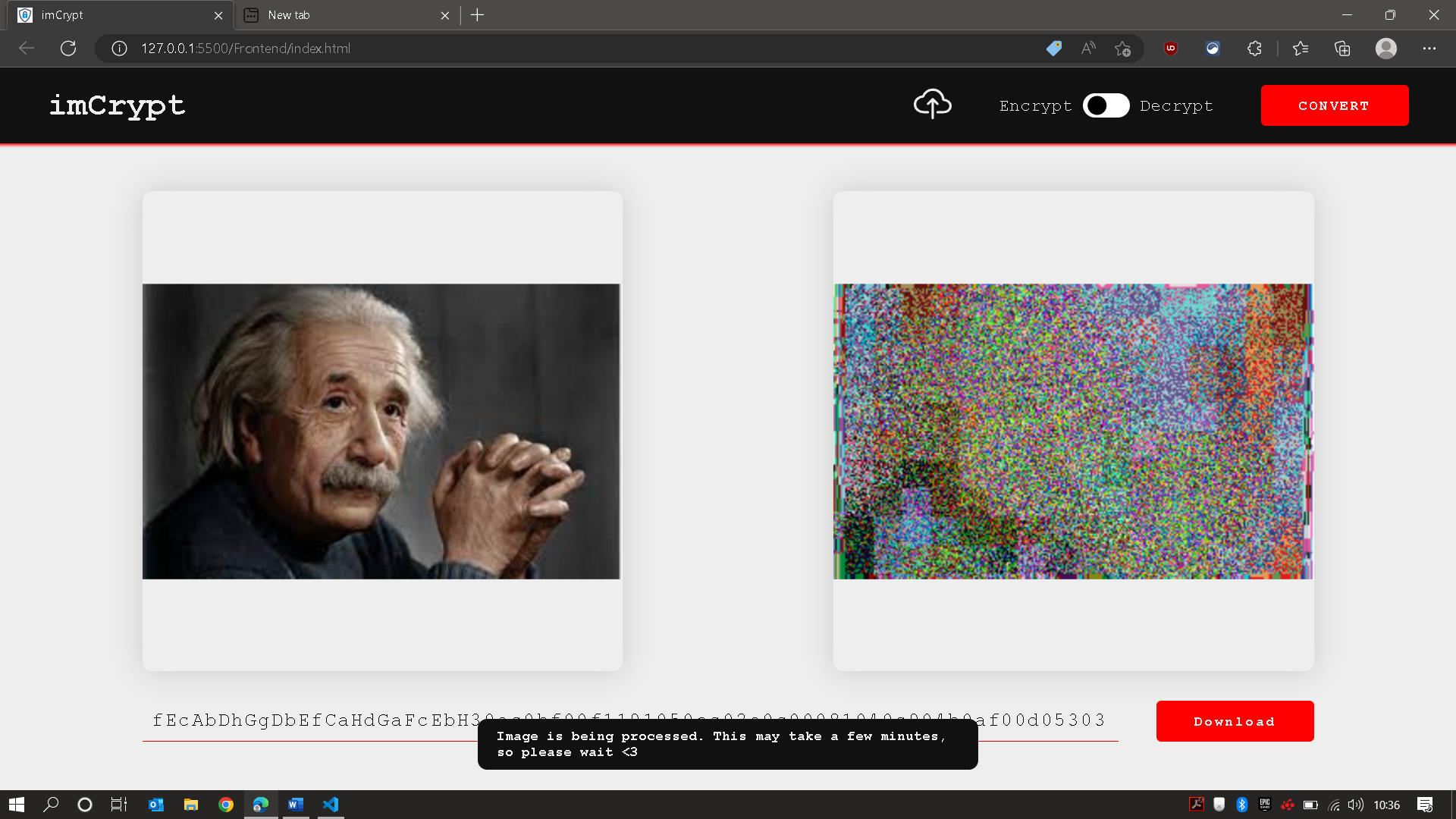
**Encryption Steps**

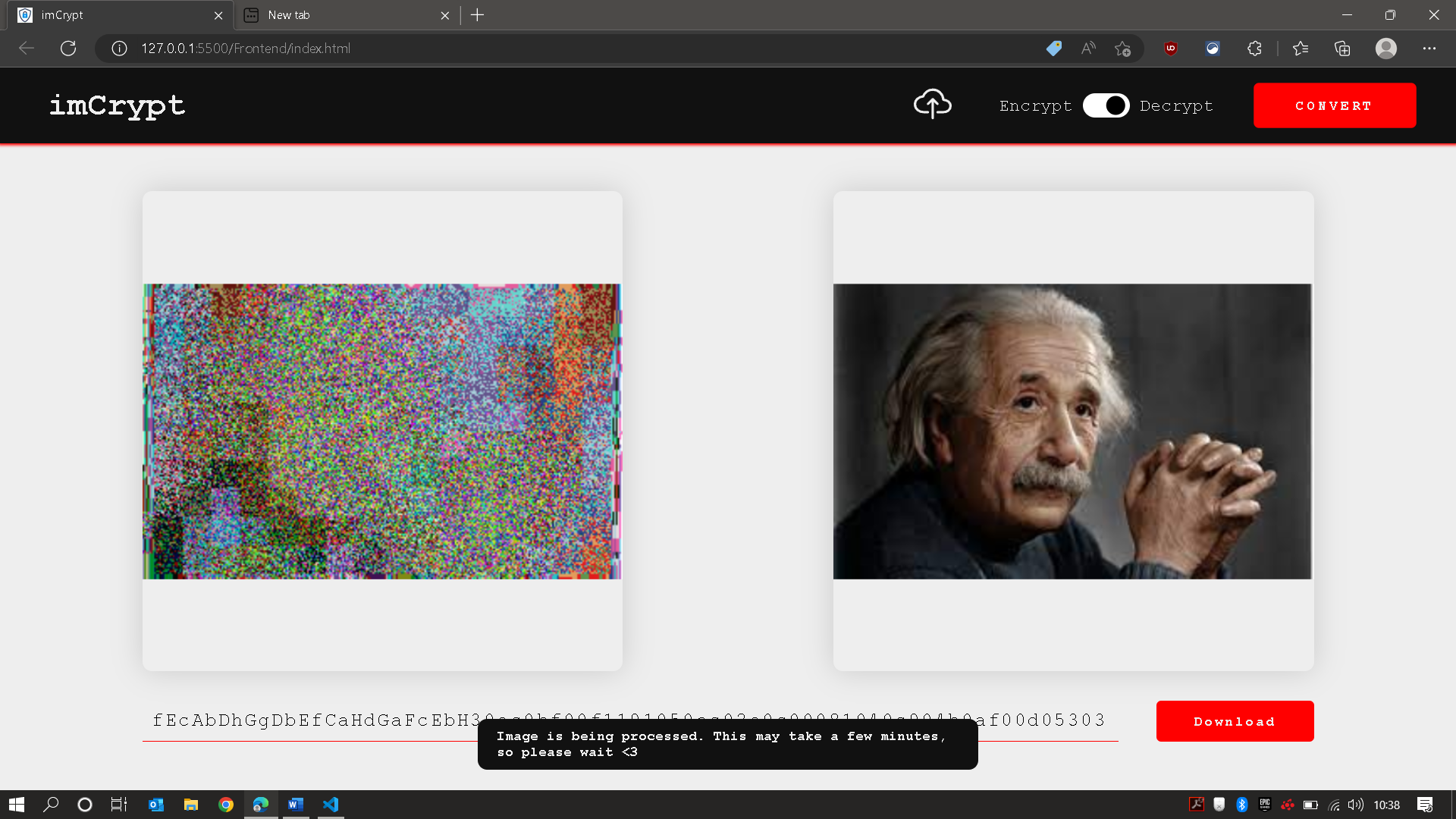
1. Load an image in RGB. This is done using the PIL module and numpy.
2. Separate the 3 channels into 3 different grayscale images, each representing a channel of the original image.
3. Randomly generate an order to reorder the bit planes for each channel. After re-ordering the bit planes, hash the generated order to hide the information.
4. After performing a bit intensity transform on all 3 channels, perform the spacial transform on all 3 channels.
5. The special transform involves generating a filter of size NxN, with values randomly filled between 0 and N^2. For this application, N = min(width, height)/10.
6. Perform a window traversal over each channel with the filter as the window, and re-arrange each value using the formula: F(u, v) = f(filter(u,v)/N, filter(u,v)%N), where F(u,v) is the value in the transformed channel at u,v, filter(u,v) is the value at the filter at u,v, and f(u,v) is the value of the original image at u,v.
7. Hash the filter to hide the information.
8. After having performed the bit intensity transform and the spacial transform on all 3 channels, join the 3 channels back to get an encrypted image in RGB.
9. Join all the generated hashes in some order to create one final hash that will be used to decrypt the image.

**Encryption Steps**

1. Load the encrypted image in RGB and separate it into 3 channels.
2. Retrieve all required information (3 orders and 1 filter) by breaking the hash into its subsequent pieces and unhashing them.
3. Apply the transforms in the reverse of the order they were applied in.
4. Merge the 3 channels back into an RGB image to retrieve the original image.

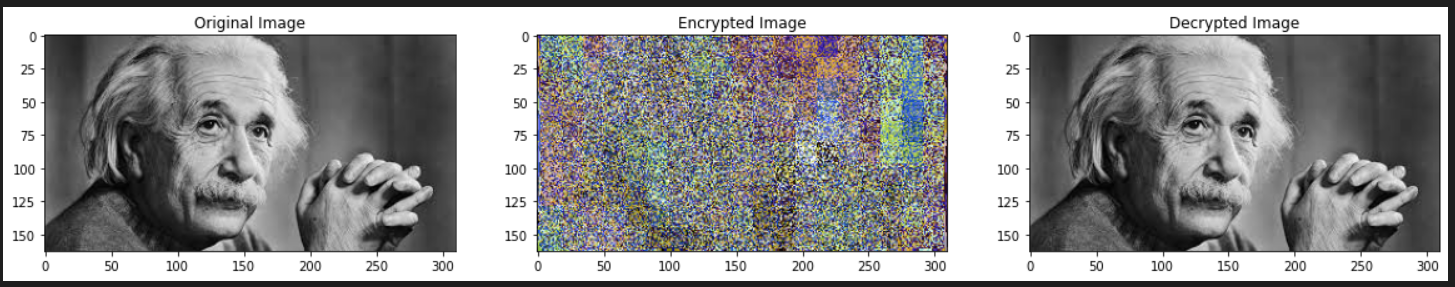
**Application**





**Results**

Since the bit intensity transform technique used here is an independent process, it does not affect the outcome of any other step. This results in the addition of an extra layer of security that has possibly 8! Or 40,320 combinations, resulting in an encrypted image that is a lot harder to brute force. Since bit-plane slicing has already been carried out in order to perform transformations on different planes as per literature, this additional step has very little overhead for a very high increase in efficiency.

 A picture containing graphical user interface

Description automatically generated

**Conclusion**

Although this technique may not be the single most promising technique, no technique can claim to fit this position. What matters is that the combinations of such small techniques provide several different layers of encryption, which makes the image share dependencies across several different secret keys instead of just 1, making it close to impossible to decrypt without all the original keys.

Furthermore, this novel method definitely provides a very high increase in security for very low overhead and computational cost, making it an extremely easy to implement step for a much secure encryption.

**Benefit To Society**

As right to intellectual property, security and privacy grows more important by the day, encryption algorithms become more and more important. Even techniques that offer very little performance are important as they add another layer of dependency, making it that much harder to brute-force when used in combination with other existing techniques.

This technique provides a very high boost in performance and is therefore an extremely powerful technique to boost the security of encryption architectures that use bit-plane slicing as their base.

**References**

[1]

[2]A new chaos-based image encryption system

https://www.sciencedirect.com/science/article/abs/pii/S0923596515001691