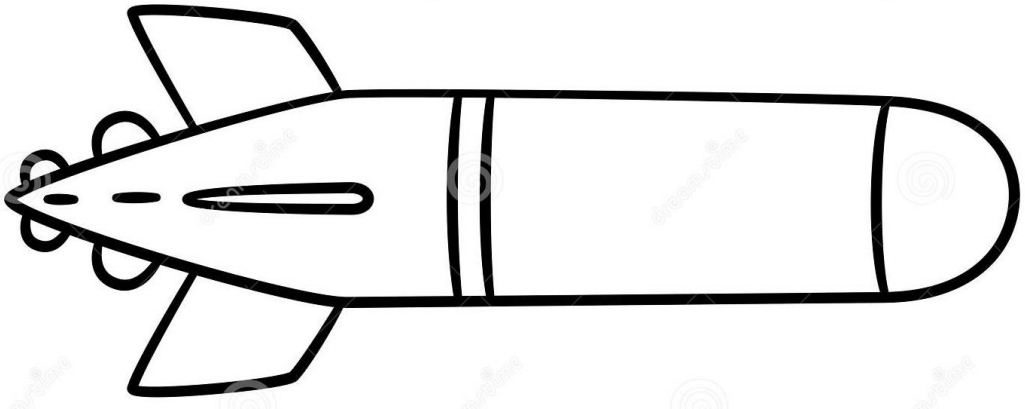
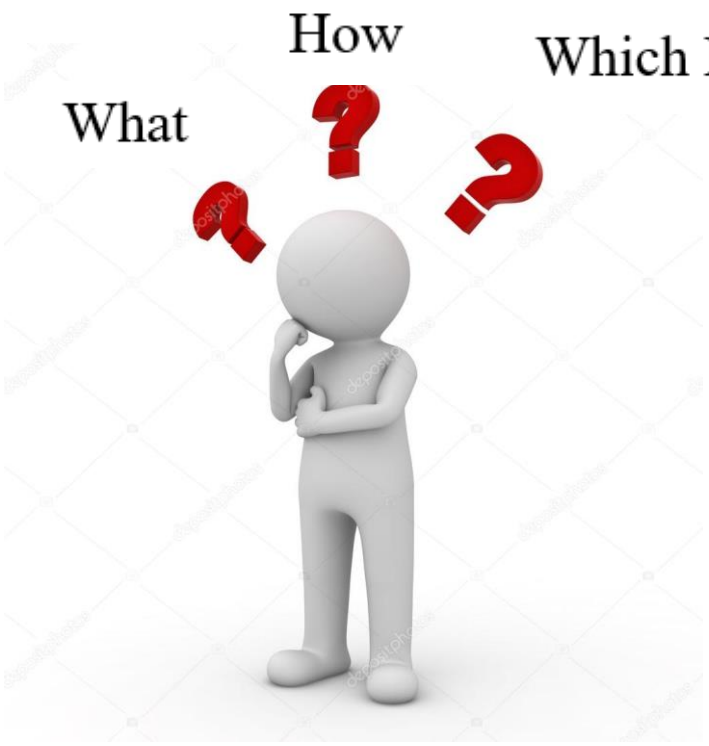
ENIGMA

A TOOL FOR CYBERSECURITY





Current Trend

Useful

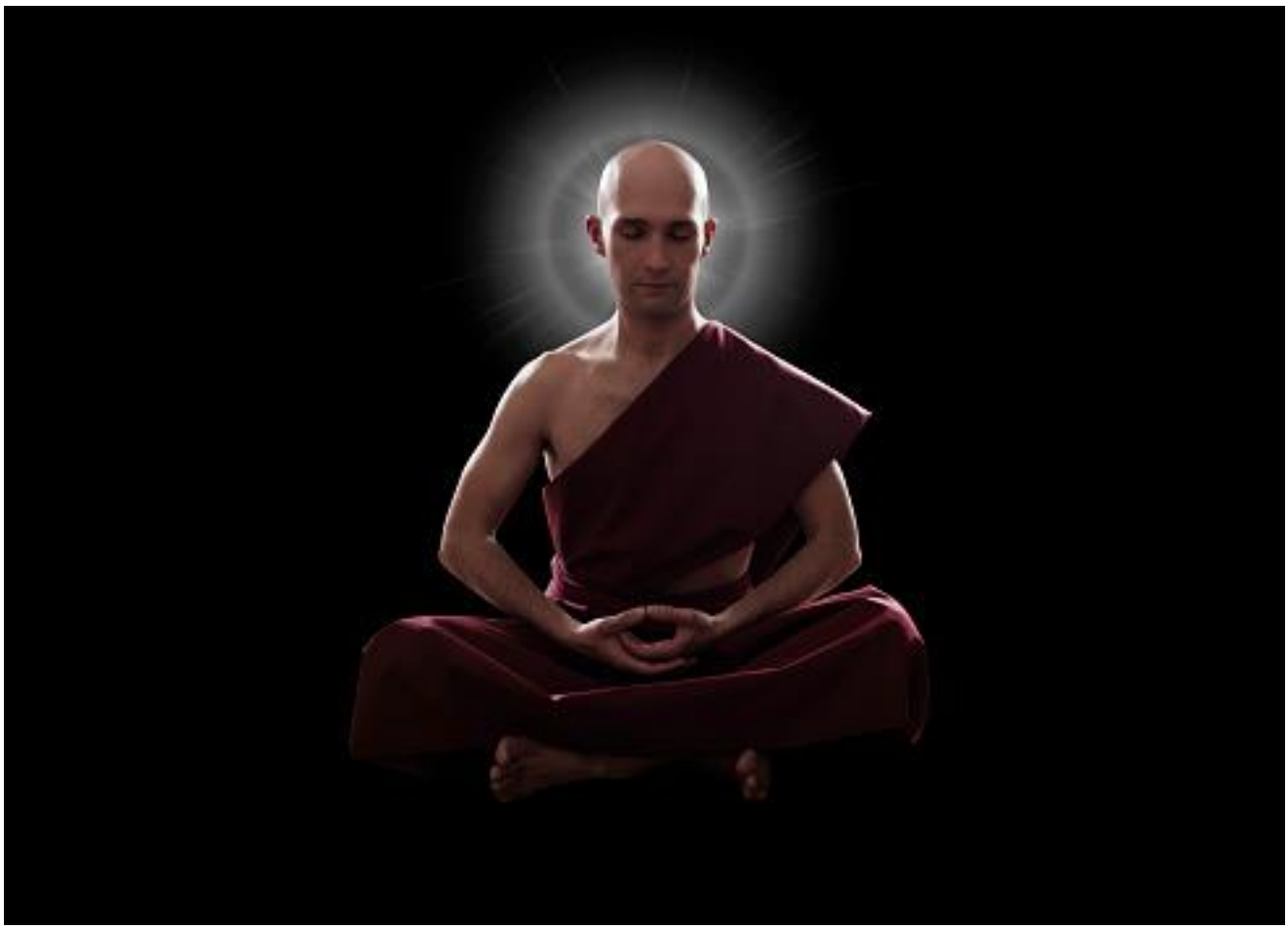
Unique

Simple

Easy to build

Future Scope

Cost Effective 😊

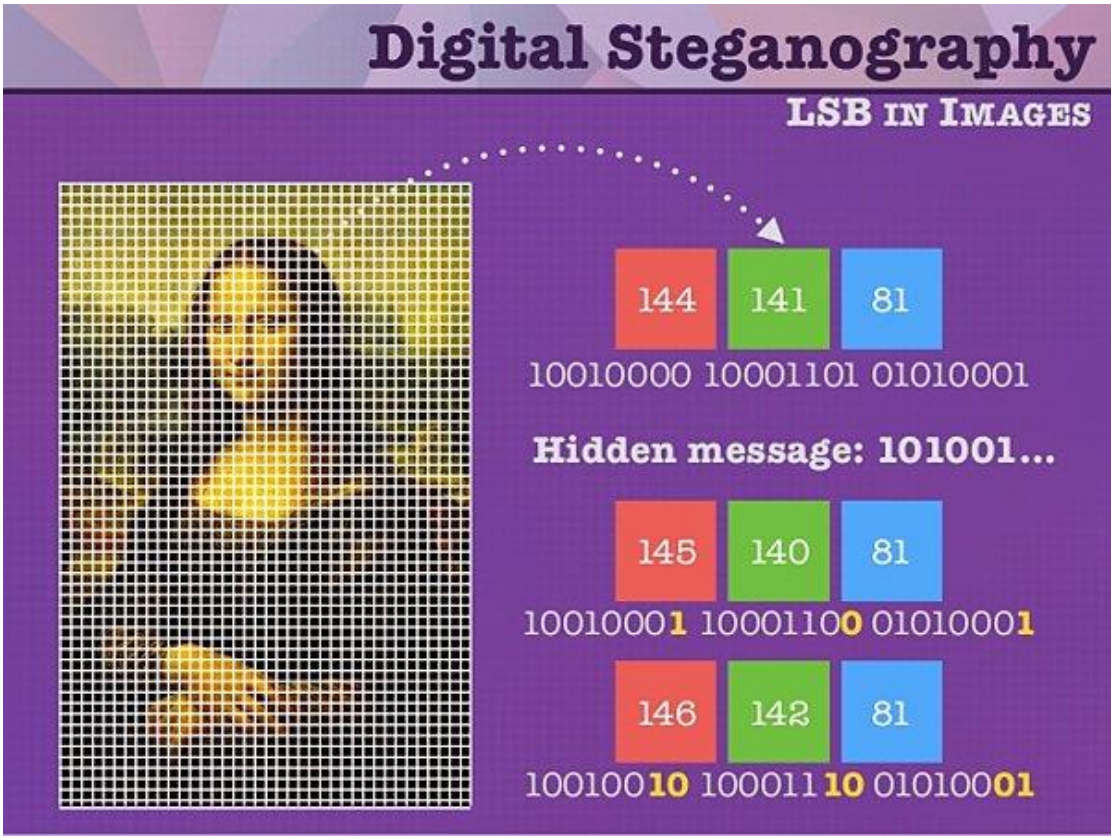


Research | Open Access | Published: 18 December 2015

A 3-d advancement of PythoCrypt for any file type

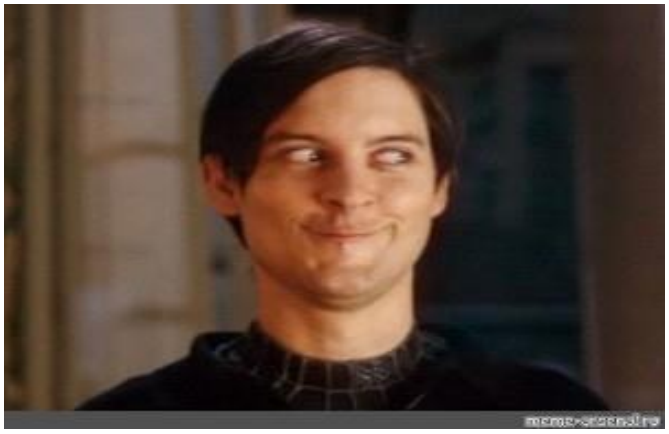
Harsha S. Jois, N. Bhaskar & M. N. Shesha Prakash

Journal of Open Innovation: Technology, Market, and Complexity **1**, Article number: 19 (2015) | Cite this article



What??

We can hide text inside the images?



I have a plan 😉

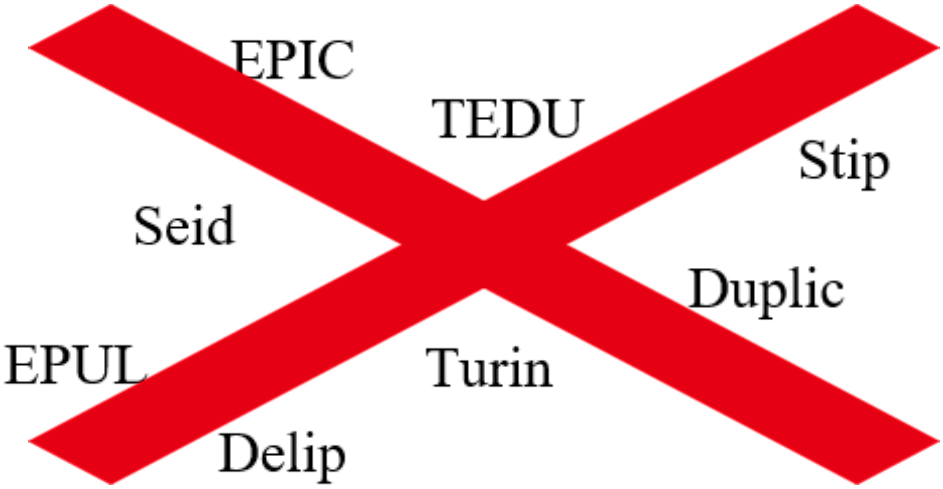
We will combine 3D Pythocrypt and Image Steganography.

How?

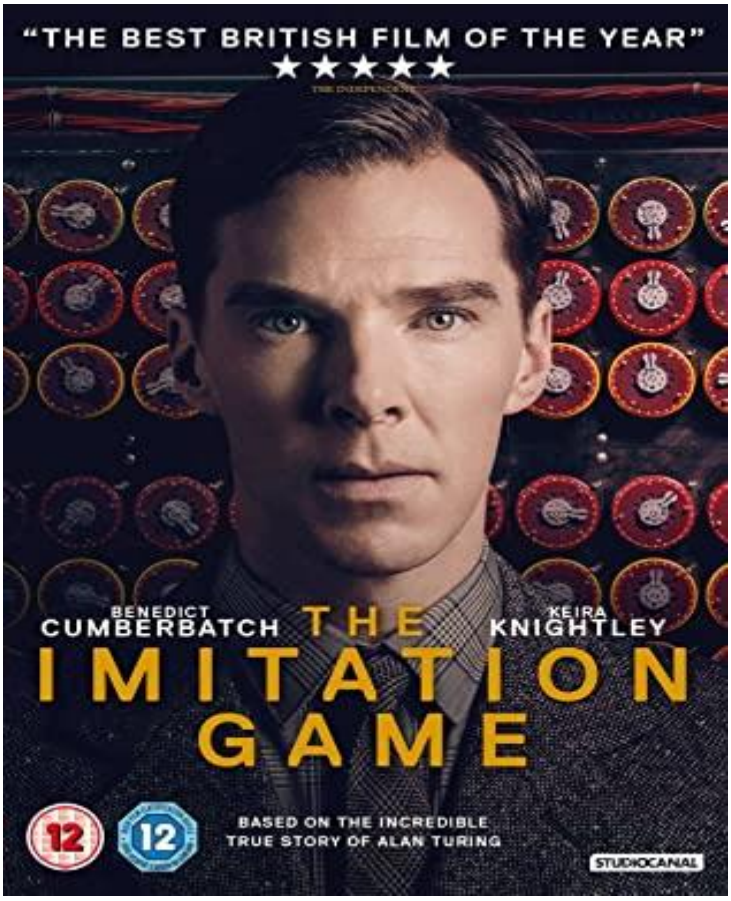
First we will encrypt the plain text using 3D Pythocrypt algorithm and generate the cipher text. This cipher text is passed to the Image Steganography algorithm where it will be hidden inside the image. This stego-image will be passed to the receiver side. Hence a secure communication is achieved.

We will create a tool that integrates both these techniques.

BUT WHAT IT SHOULD BE NAMED?



Then

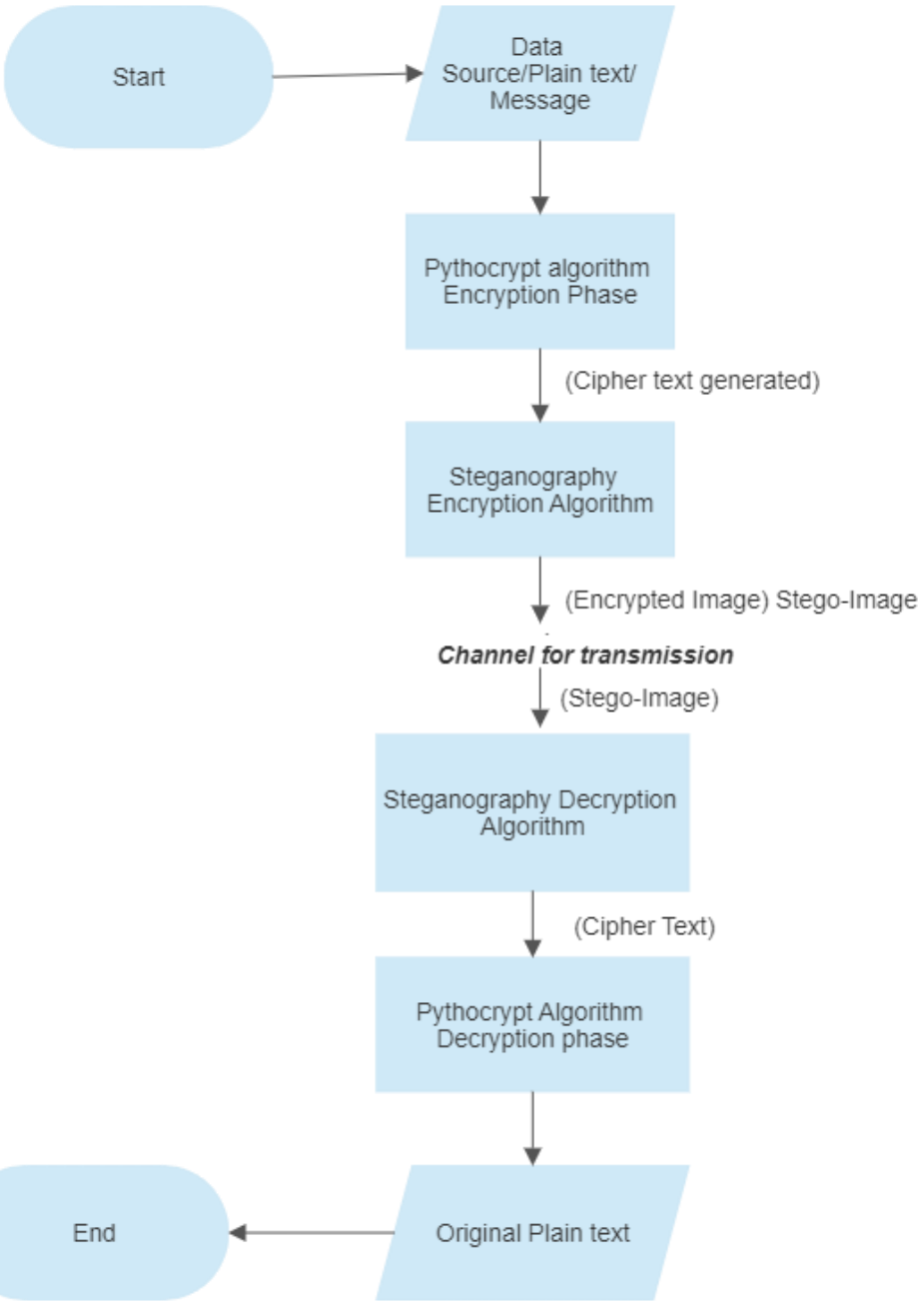


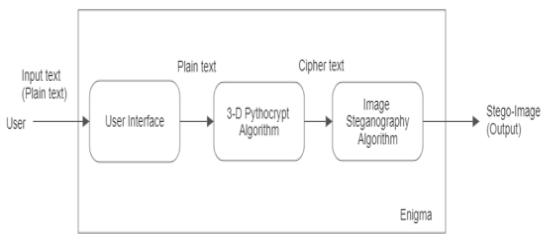
Here Alan Turing successfully decrypts the messages that was sent using ENIGMA machine which was impossible to crack during the world war II.

We decided that our project will be named as

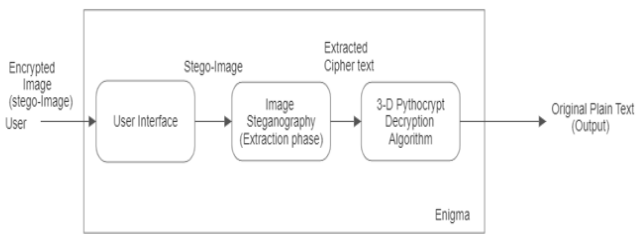
ENIGMA – A TOOL FOR CYBERSECURITY

Image Steganography





a. Encryption Phase in Enigma



b. Decryption Phase in Enigma

JAVA / PYTHON?

We initially started with java programming language but, due to the less overhead and less configuration we implemented in python programming language. Also, we have developed a custom CLI tool using python programming language which acts as a user interface for out API.

Process:

Encryption:

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 1. is chosen and is used for the encryption process. Id column in Table 1. Contains the id’s assigned to the shapes which will help us to communicate the shape used at encryption to the receiver. The cipher text obtained will be sent to the Image steganography algorithm where it will be converted to binary values and added in the LSB bit of each pixel in the image.

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	$(5 * \tan(54^o) * h * a^2)/12$

Decryption:

The cipher text will be extracted from the image by reversing the Image Steganography encryption phase.

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	$a^2 = (a * a)$ $h = (12 * v)/(5 * \tan(54^o) * a^2)$	$h = (12 * v)/(5 * \tan(54^o) * a^2)$	$a^2 = (12 * v)/(5 * \tan(54^o) * h)$ $a = \sqrt{a^2}$

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.

Key Points:

The 3-D Pythocrypt algorithm is a new and young technique and is infeasible for many cryptographic attacks.

Only some part of the numerical values can be obtained and also it is infeasible to find the original plain text by observing the pattern of ciphertext.

The octahedron shape was implemented in Java Programming language where we have used BigDecimal data type to store and process the intermediate results. And it was found that we can process $2^{2147483647}$ ASCII values that is approximately 646456993 digits.

We have considered 4K .png images to hide the cipher text. We can store more data in 4K images and .png image format support lossless compression. Approximately 4M characters can be stored inside the image.

Implementing 3D Pythocrypt along with Image Steganography provides better security and is faster than many existing cryptosystems.

CHAPTER 2

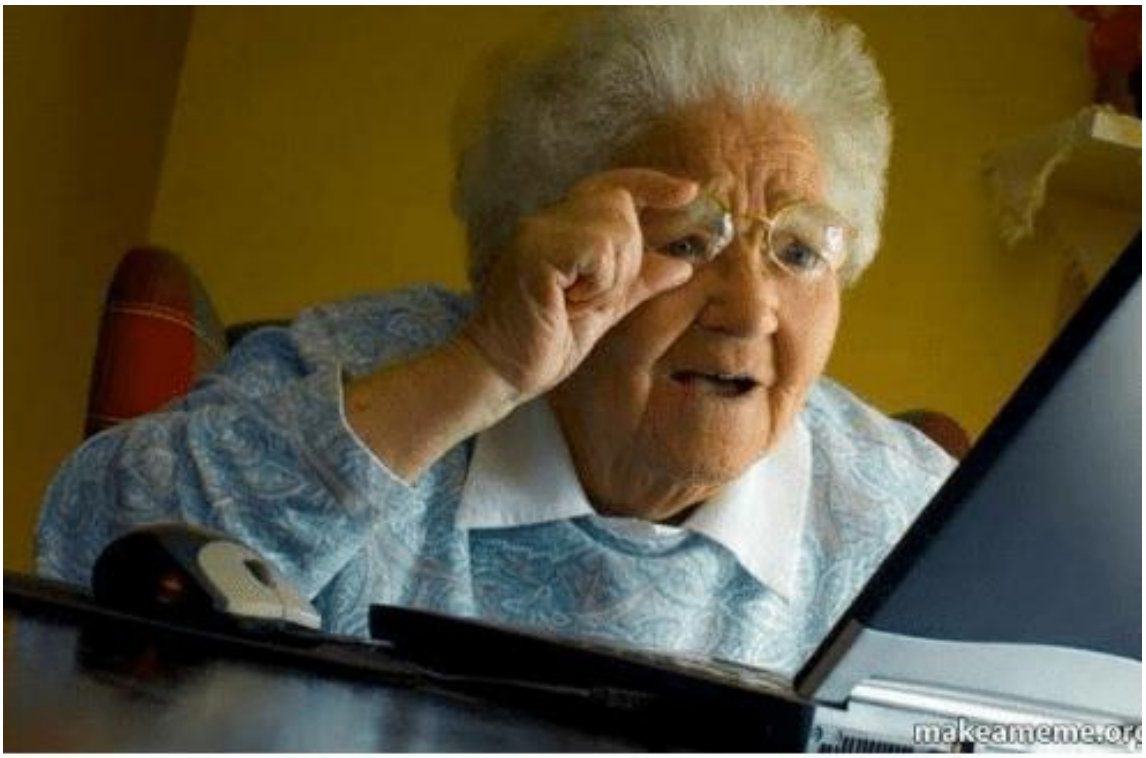
But not every expectation was satisfied.

Images cannot be used everywhere. Example, if we wanted to store data in a cloud there we can use 3D Pythocrypt algorithm but Image Steganography was not necessary. Most of the transmission medium will compress the image even though lossless images were used there are chances that the data can be lost.

And also, we wanted to contribute to the existing cryptographic system without replacing the entire system.

At the right time we got to know about Chitrakāvya.

Chitra Kavya is an ancient book, which is written in Bhandra language. This book contains different daily seen shapes like Umbrella, Snake. These shapes are called as “Bandhas”. The Sanskrit verse is splitted to individual alphabets and is hidden inside that shape. If we traverse this shape in a particular way and read that alphabets, then only we can decrypt the verse.



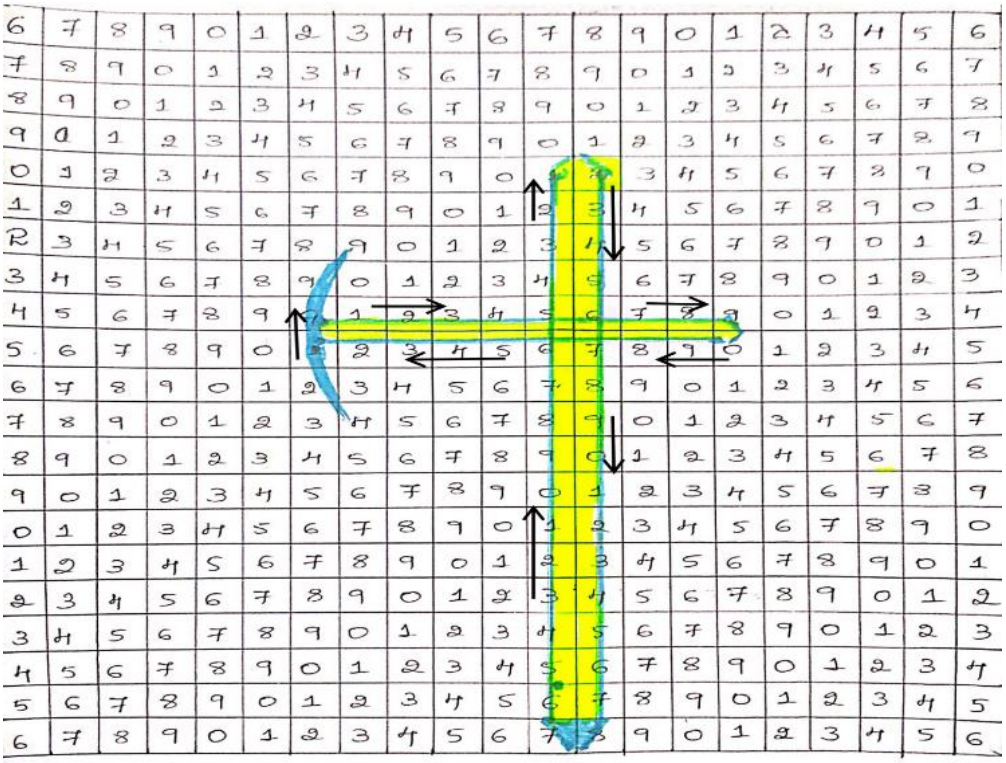
In our research work we have considered parashu bandha.

Key Generation using Chitrakāvya technique:

Instead of hiding the cipher text inside the image, we again encrypt the result using AES encryption algorithm. AES is a symmetric encryption algorithm that uses same key for both encryption and decryption process. The key will be generated using a new technique called Chitrakāvya.

Chitra kavya contains different designs (bandha) where the verse (shloka) is encrypted in those designs. If we traverse those designs (bandha) in a particular order we can decrypt the verse(shloka/message). Here we construct a 21 x 21 matrix (dimension is not fixed, it can be altered) in which the elements are an orderly arrangement of numbers from 0 to 9. A dynamic seed point is obtained as the initial node and the matrix is traversed in a specific order such that the path forms the chosen design.

Example:



Encryption:

We consider element 2 (5,13) (5th row and 13th column) as the seeding point. From this initial node, we traverse through the matrix in such a way that the path from the parashu bandha. The values encountered while traversing is taken as the key for the AES symmetric encryption algorithm. From fig. 1, we generate the key as **234567890987890123456787654321098765432124999012345432 1**. But, for the AES encryption, we only need a 256-bit key. Hence, we consider

only the first 32 digits of the above-generated value as the secret key for the encryption and decryption process.

Decryption:

The encrypted cipher text of the 3-D Pythocrypt algorithm along with the seeding point used in generating the key is fed as an input to the AES Decryption algorithm.

Using this seed point**, the key is generated at the receiver side**. Hence, **the key is not transmitted from the sender to the receiver** however, some knowledge about the key generation (matrix dimension, the design(bandha) used for encryption) should be known to the receiver.

Using the key and the message received, the cipher text obtained from the 3-D Pythocrypt encryption technique is recovered.

Sample Input and Output:

The plain text can be a sequence of any character which has equivalent ASCII values.
For example: Let the plain text be, "**Hello, this is a sample message**".
This sample plain text is 31 characters long, and its equivalent ASCII value will be 72101108108111443211610410511532105115329732115971091121081013210910111511597103101.

When this plain text is encrypted using the volume of octahedron geometric shape, the corresponding cipher text obtained from the 3-D Pythocrypt algorithm will be,
1311130474473615078252873891093659587289903933262200641513949618898664159933953864880737000302345572630791701897355670597012073.32115971091121081013210910111511597103101
Note that the dot (".") operator is used as the delimiter between cipher text and the key. Both cipher text and the key will be sent to the AES Encryption algorithm.

From Fig. 3. the key for AES encryption can be 234567890987890123456787654321098765432124999012345432 1, However, for the AES encryption 256-bit (32 digits) key is sufficient. Hence, we consider only the first 32 digits of the above-generated key (23456789098789012345678765432109).
The cipher text and the 32-digit key are passed as the input to the AES Encryption algorithm, the final encrypted cipher text will be,

uqXRqTULcMqYKwOiyaZnxwfDXMIC89lsmnyTt3orjH+WlfsQp0jlEMFU3CRGZbcv5VkeK0f9E5iANZ+od1hWbxhu9nkX/VagEeSdEgDXQt0khvPv3U12+nWNdr3TPE38V+hd/j/Jv9bUn8aKf92aID+5bcxWCpbK7QBNU13LtW9bQN0OhWz4UVTv+HrzVzgbNy0F4g7evOBdxfI4VMqMNW7Wo8uAUTQ4OuT0puPxfms=

This message is transmitted to the receiver, along with the information about the seeding point to generate the key at the receiver.
The original plain text is obtained at the receiver side by reversing the encryption process.

Conclusion:

- 3D Pythocrypt is a new and young technique in cryptographic field and is infeasible for many cryptographic attacks.
- In a given time, 3D Pythocrypt can handle huge amount of plain text, and is powerful and efficient than existing cryptographic systems.
- Generating keys using Chitra kavya is a new and powerful technique and it can be used in any scenario where dynamic key generation is required.
- There are many bandhas in Chitra kavya that can be implemented to generate the key and can be the next big thing in cryptography field.


THANK YOU!