August 2021

ENIGMA

The new creative way to Compress & Encrypt secret messages in an image

Ohad Mavdali & Dotan Gotshtein

INTRODUCTION

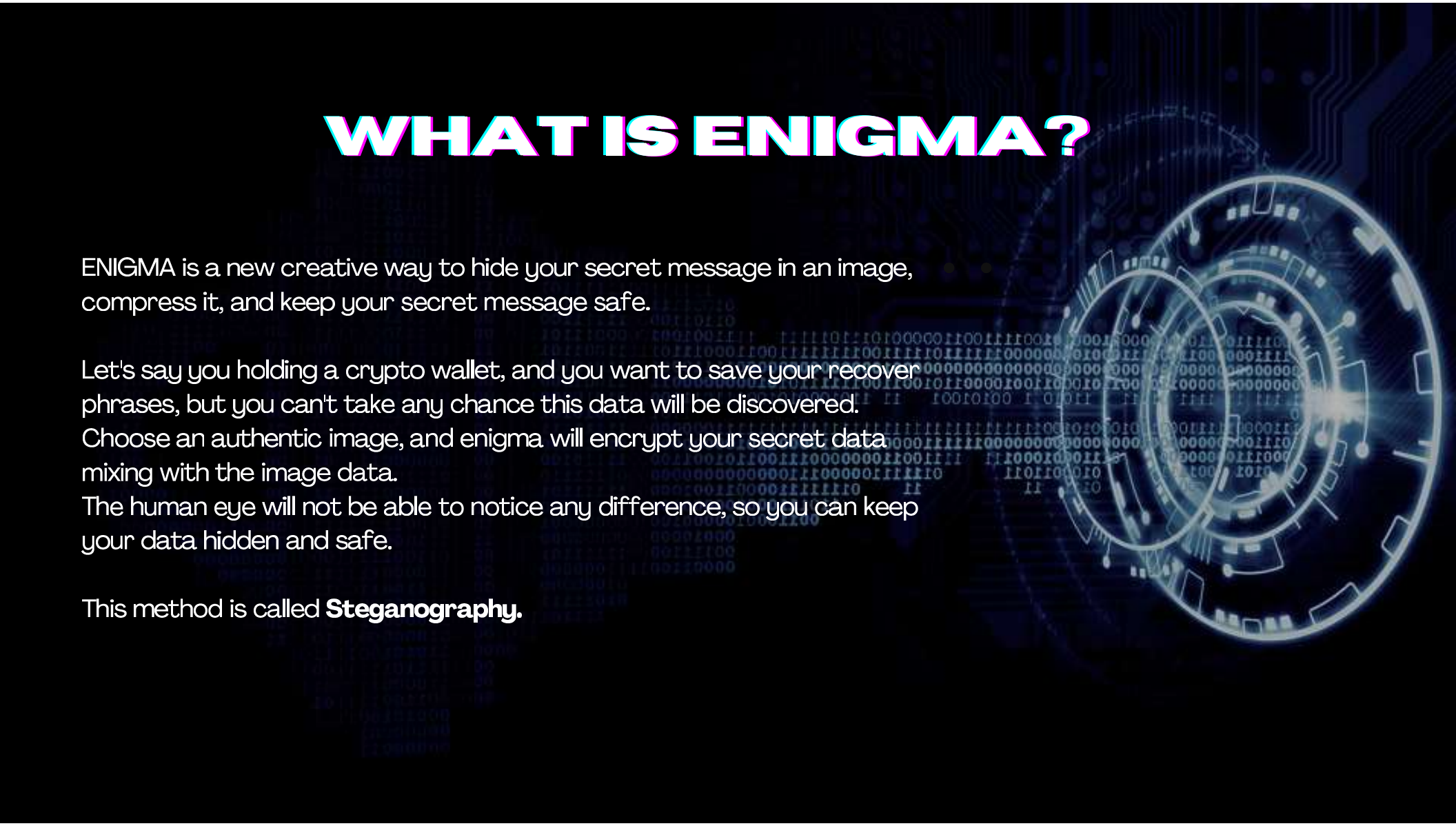
In the era, where privacy becomes more important,
we want to keep our data safe.

Data is been trading in high volumes to anyone who is
interested and can pay for it.

Our data becoming digital, and we are producing
and sharing more information online,
without taking into account the dangers involved.

That's why we developed the ENIGMA

WHAT IS ENIGMA?



ENIGMA is a new creative way to hide your secret message in an image, compress it, and keep your secret message safe.

Let's say you holding a crypto wallet, and you want to save your recover phrases, but you can't take any chance this data will be discovered.
Choose an authentic image, and enigma will encrypt your secret data mixing with the image data.
The human eye will not be able to notice any difference, so you can keep your data hidden and safe.

This method is called **Steganography.**

ENIGMA

PRIVACY IS DEAD?

Let's begin.

ENIGMA

TODAY'S AGENDA

Steganography is one of the most important fields of information security.
It is the art and science of secret communication between two sides.

The tools outside usually focusing on simple encryption such LSB manipulate support mainly Lossless compression and giving less importance to algorithms such JPEG compression which is a lossy compression.

We decided to take it one step forward, and encrypt the data while compressing the image so we reduce the image file size and still keep the secret data safe.

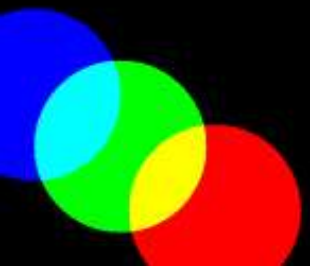
ENIGMA

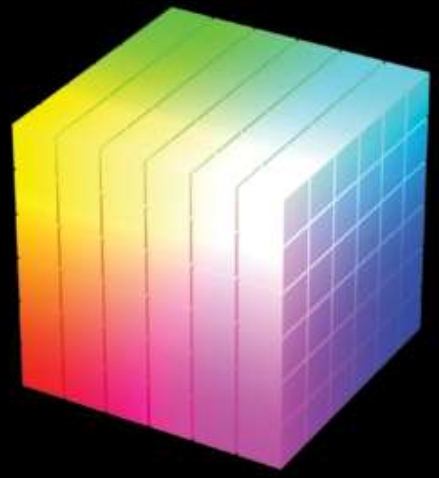
WHAT IS DIGITAL IMAGE

A digital image is a representation of numbers that can be stored and handled by a digital computer. In order to translate the image into numbers, it is divided into small areas called **pixels.**

We can think of an image as a tensor or a three-dimensional array where each channel represents a color - **R**ed, **G**reen and **B**lue.

RGB channels roughly follow the color receptors in the human eye.





ENIGMA

HOW DOES IT WORKS?

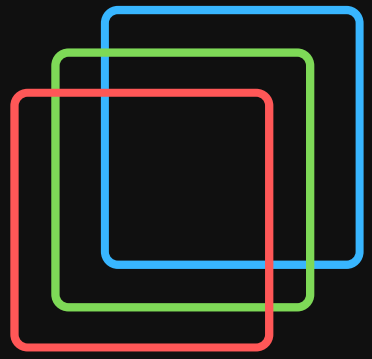
ENIGMA

ENCRYPT & COMPRESS

step by step:

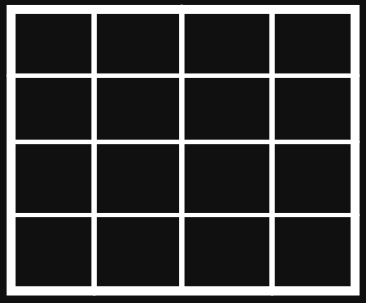
1. Resize the image so it can be divided by 8 both height and width.
2. Transform the BGR image into YCrCb.
3. The cover image is dived into 8×8 blocks of pixels.
4. DCT is applied to each block.
5. Each block is compressed by division in the JPEG quantization table - lossy compression
6. Convert each compressed block into 1D array - ZigZag
7. Each block contains High, Mid and Low frequency coefficients
8. Determine & replace LSB of the Mid coefficients with MSB of secret message
9. One binary character is replaced to avoid changing the original coeff

ENIGMA

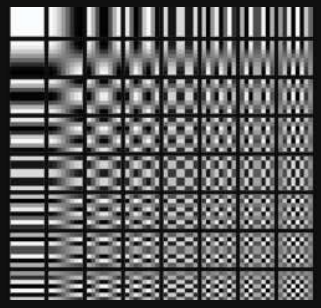


RGB to YCrCb

Divide to 8x8 blocks



FOR EACH 8X8 BLOCK



DCT

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Quantization

-24	-23	0	0	0	0	0	0
-19	4	1	0	0	0	0	0
5	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

ZigZag

We use the coefficients that are greater than 1 means we use only the medium frequency coeffs and replace their LSB with the secret text MSB

ENIGMA

ENCRYPT & COMPRESS

step by step:

10 Convert the array back to 8x8 2D blocks

11 Multiply each block with the quantization table

12 Apply IDCT on each block

13 Combine the 8x8 blocks back together.

14 Transform the YCrCb image into BGR.

15 Display the stego image as RGB and write it as a JPG file

ENIGMA

STEGANOGRAPHY USING DCT

The main idea in our project is to manipulate the Discrete Cosine Transform (DCT) portion of the standard JPEG encoding process in order to embed secret data in the lossy compression procedure.

DCT is applied to transfer the digital image data from the spatial domain to the frequency domain.

So the secret message is embedded by adjusting the coefficients of the middle-frequency sub-band in the luminance layer so that the visibility of the image will not be affected after compression.

ENIGMA

EXTRACTION

step by step:

1. Transform the encrypted BGR image into YCrCb
2. Divide into 8×8 block of pixels
3. Apply DCT on each block of the Luminance layer
4. Divide each block with the JPEG quantization table
5. Convert each compressed block into 1D array - ZigZag
6. Determine the LSB of the Mid coefficients
7. Concatenate each binary character to a string
8. Convert the string to characters by the ASCII table
9. Return the secret message and display it

ENIGMA

1. Upload a PNG image

Upload Image

2. Enter the secret message

3. Get a compressed JPEG image with secret encrypted

Yes!

Original file size 11.5MB

Compressed file size 2.6MB

The secret message

"The maximum text length depends on the image file size and the relevant coefficients amount.

The max coefficients available for encryption in this example are around 200K which represents around 25K chars"

ENIGMA

CONCLUSIONS

We took a PNG image, encrypted the secret message, and successfully returned a compressed JPEG image without any human-eye visible difference.
The original file size at start was 11.5MB.
The new compressed file size now is 2.6MB.

ENIGMA

IN THE FUTURE

Our vision of further development for this project is to take it to the next level and encrypt an image inside another image through the same process. The only different step in the algorithm will be needed is to format the hidden image pixels to binary and encrypt them in the same DCT coefficients

ENIGMA

THAT'S A WRAP!

Thank you for participating.

 YouTube

 GitHub

Ohad Mavdali
Dotan Gotshtein