A Novel: Introduction to Ciepiel's Cross Steganography Encryption & File Decreasing Mathematical Formula.

Created by: Marcin Ciepiel

Supervised by: Jin Zhang

Anglia Ruskin University – Cambridge

Student ID: 1716223

Table of Contents

Table of Contents 2

1) Abstract 4

2) Introduction 5

2.1) Problem Background 6

3) Research Aims & Objectives 7

3.1) Research Aims: 7

3.2) Research Objectives: 8

4) Scope and Delimitations 9

4.1) Scope: 9

4.2) Delimitations: 10

4.2.1) Full image size encryption software: 10

4.2.2) Military and confidential Ciepiel's Cross usage: 10

4.2.3) File Compression: 10

5) Literature Review 11

5.1) Why Ciepiel's Cross fills the gap: 11

5.2) Differences from other algorithms: 11

5.3) Wartime: 12

5.4) LSB Steganography: 12

5.5) Video Steganography: 13

5.6) Mathematical Formula Existence: 14

5.7) Summary: 14

6) Methodology 16

6.1) Discovery 16

6.1.1) Sum pattern discovery: 16

6.1.2) Ciepiel's Cross discovery: 20

6.1.3) Ciepiel's Cross formula creation: 20

6.1.4) Conclusion: 22

6.2) Research Framework 23

6.2.1) Ciepiel's Cross Formula – Further Research: 23

6.2.2) Ciepiel's Cross Steganography Encryption research process: 24

6.2.4) Digital Image's pixels representation 25

6.2.3) Rules of Ciepiel's Cross Steganography Encryption: 27

6.2.4) Encryption approach using MS Paint software: 31

6.2.5) Encountered issues and new discovery: 32
6.3) Data Collection Strategy 33
6.4) Gantt Chart 33
7) Design and Development 34
7.1) Coding Method Selection 34
7.2) Web-Application Design 35
7.2.1) Encryption Page Design: 35
7.2.2) Decryption Page Design: 41
Design Summary: 42
7.3) Web-Application Development (Approach) 43
7.3.1) Design Development: 43
7.3.1.1) EQ.PHP & DQ.PHP Design Code: 43
7.3.1.2) Layout.css Design Code: 45
7.3.2) Theoretical Functionality Development: 47
7.3.2.1) EQ.php Theoretical Functionality Code: 48
7.3.3) Encountered Issues: 52
8) Testing & Implementation 53
8.1) Encryption 54
8.1.1) Step 1: Photo needs to be respected by the Algorithm's rules 54
8.1.2) Step 2: Sum Pattern needs to be applied to each black pixel 54
8.1.3) Instruction of what values are being taken 55
8.1.4) Step 3: Image Encryption and new file creation 56
8.2) Decryption 60
8.2.1) Decryption Steps & RGB values extraction 60
8.3.2) Original Image Restoration 63
9) Results 65
9.1) Encryption Results 65
9.1.1) Results in Encrypted photo File Compression 65
9.2) Decryption Results 67
10) Discussion and Conclusions 68
10.1) Products of this study 68
10.2) Rules created for Ciepiel’s Cross Steganography Encryption 69
11) Future work 70
12) Bibliography 71

1) Abstract

New algorithms appearance is significant in digital security improvement. Several discoveries described in this study resulted in the creation of the new steganography encryption. Ciepiel's Cross contains three different steps that encrypt and decrypt a digital image using own, securely created rules. Discovery of the new mathematical sum pattern developed into the creation of new mathematical formula. Both had been used in the creation of a unique algorithm that successfully proved to be useful in steganography encryption.

Finalizing image encryption, research is being provided with decryption possibility. Allowing decryption only by understanding the Ciepiel's Cross Formula settles serious difficulty over this process. Creating a step-by-step encryption and decryption guide, this study covers the whole usage of Ciepiel's Cross algorithm.

This study covers the approach of web application creation to automate the encryption process. However unfinished, design shown in the research contributes an idea of the possible further development scheme.

Ciepiel's Cross algorithm fills the gap in steganography encrypted files being usually more extensive than the original photo. Encrypted using Ciepiel's Cross Steganography Encryption file, while saved in .JPG format decreases its size on disk.

2) Introduction

“Human communication was initiated with the origin of speech approximately 500,000 BC” (Gross, 2010). Time of people understanding each other, motivated an opportunity of secret language creation. When talking in language not understandable to other person, many secrets could get hidden. This is where steganography found its usage starting from 5th century BC (Alan Siper, 2005). Hiding a message using this method provides another layer of security to keep its secrets unrevealed. This study focuses on the novel’s Ciepiel’s Cross Algorithm and interprets it as an innovative and unique digital image encryption method.

Before reading this study, it is essential to understand what steganography is. “Coming from Greek word “στεγαυω” meaning “secret” or “covered”, steganography is the way of hiding the message or changing its original values so unauthorized people could not understand the meaning of it” (Kahn, 2005). An example of steganography could be explained by changing all "C" letters in this study to the letter “Z”. While being not very “practizal”, it already “zreates” the “zode” that, in many parts, might be understood only when knowing this rule.

“Steganography is the art of hiding of a message within some format so that presence of hidden message is indistinguishable. Cryptography involves converting a message text into an unreadable ciphertext. One hides the existence of the message and the other distorts the message itself. Many powerful & robust technologies of steganography & cryptography are developed. This paper is an attempt to analyze the steganographic & cryptographic techniques” (Pranali R. Ekatpure, 2015)

2.1) Problem Background

Steganography puts a significant mark on today's digital world. Privacy and work authority seems to be the primary concern within existing and widely used social media applications (Choudhary, 2017). There is an issue of people knowing their information are not secured on the Internet. This was the motivation in the Ciepiel's Cross algorithm creation.

There are multiple ways of using steganography to hide a message. "Steganography has developed a lot in recent years because digital techniques allow new ways of hiding information inside other information, and this can be valuable in a lot of situations. The first to employ hidden communications techniques -with radio transmissions- were the armies, because of the strategic importance of secure communication and the need to conceal the source as much as possible." (UKEssays, 2018)

Day of the Internet creation was the first step into the re-evaluation of privacy meaning (at that time mostly militarily). Humans, entering the technology age, put the beginning of feasibility of stealing data with ease. This is where file encryption found its existence. (FigLeaft, 2019)

While computers calculation speed and power are getting bigger every day, "new privacy algorithms are being introduced slowly. Possibly they appear every few years with huge worldwide accessibility in means of open-source usage" (Brian Beckett, 1997). This provides another concern of algorithms being breakable and easily understandable. When this encryption implementation approach gives positive results in everyone's favour, there are still cases for new algorithms to appear. Some life segments like online banking need a much more secure network that should not use any kind of public data encryption.

Discovery, creation, and documentation of Ciepiel's Cross Steganography Encryption is the first introduction into new algorithms appearance. Having "personal enigma" gives true privacy that might not be infiltrated by third party authorities like owners of social media networks.

Unknown encryption algorithm puts advantages, but also a huge risk over world's safety issues. When the privacy of personal messages or images gives mostly security, it may be used for detailed and structured data encryption, that is undiscoverable by any other person. Without a fully accessible understanding of each Algorithm, it might be possible to communicate secretly by any sender to any recipient worldwide.

Ciepiel's Cross creates a new solution in Steganography Encryption that may be used within any field of work when publicly accessible. Cryptographic algorithms are only robust when not discovered (Browinski, 2016). There are many infiltrated encryption algorithms, what puts undiscoverable solutions above the Algorithm's strength.

3) Research Aims & Objectives

3.1) Research Aims:

The main aim of this study is to discover a new mathematical formula. This can be achieved by either old formulas improvement or by looking to not resolved mathematical issues.

Following the formula's creation, rules allowing it to become an algorithm should be applied. Formula needs to be repetitive, continuous, and unique. The next step would be a need of finding finite sections within the repetitive pattern. Those sections describe calculations that user could compute by using the formula.

By having a working formula's solution, it would be necessary to prove its usage within any cryptographic method. Aiming for steganography, formula and Algorithm, both need to be useful around image files encryption.

That part sets the last aim of the research. Final image encryption with decryption possibility (using new formula) should prove its usage in cryptography. Any third-party person seeing the image output should not consider any valuable information in it. The message hidden in the image should be decryptable only by knowing the formula itself. By reconstructing an algorithm, it should proceed to the decryption, giving output same or similar to the given input.

Documenting all steps into Ciepiel's Cross creation is necessary to understand and reconstruct the encryption. Having both encrypted and decrypted images sets goal that finalizes this study aims.

3.2) Research Objectives:

- Discoverability of new formula. It needs to be repetitive, continuous, and unique for its task.

- Create an algorithm by using discovered formula. The algorithm solves an issue or creates new values by using input elements.

- Classify an algorithm as useful in steganography encryption. Producing a different output of the given image is necessary.

- Revise an algorithm uniqueness. Proving its non-existence in steganography or cryptography solutions.

- Construct an automated encryption within a software or web-based application. With this step being too complicated, encryption should be possible to be done manually using any available and public software like Paint, Photoshop, HxD etc.

- Examine steps taken under encryption process and use reverse engineering to decrypt an image the same way. That means, by using an algorithm and formula's output, it should be possible to re-create given as an input image.

- Develop steps and rules allowing to reproduce described tasks. Anyone having access to this document should be able to automatically or manually use Ciepiel's Cross for image encryption.

4) Scope and Delimitations

4.1) Scope:

This study is to focus on creating new steganography encryption. The cryptographic solution is going to be based on a newly discovered mathematical formula later evolved into an algorithm. Writing a manual guide for created encryption will define what software needs to be created or used for testing purposes.

The study covers rules to follow while creating and using later created cryptographic solution. Having an encryption availability sets the need for decryption possibility. That is covered within the research.

This study proves the usage of RGB pixels calculation over the Ciepiel's Cross formula as the encryption step. It resolves essential functions of steganography, allowing to change the image output.

The importance of new algorithms creation seems to be undoubtedly missing in works of literature. Ciepiel’s Cross sum pattern does not fill the gap but builds a new approach and improves undiscovered parts of steganography solutions. Research has taken a month to develop structures of Ciepiel's Cross encryption fully. Documentation is the essential part of providing guidance to the newly created algorithm.

Finalizing the study should result in Ciepiel’s Cross being available of image encryption and decryption.

4.2) Delimitations:

4.2.1) Full image size encryption software:

Creating a working software for complete, fully sized pictures is the part of the research I did not take under development. The algorithm proves to work over smaller pictures (e.g.:8x8 pixels), and around this image size, research will take consideration. Building encryption software for any found in the Internet image is a too expensive task that I am not available to cover financially. As well as time to focus on the research puts a limitation on possible more significant file encryption. This however, will be taken under the future development consideration.

4.2.2) Military and confidential Ciepiel’s Cross usage:

The study does not cover military usage for Ciepiel’s Cross algorithm over any form of confidential security. Believing in the fact that newly created steganography might be helpful in top-secret fields is confirming; however, the study covers only description and guidance over Ciepiel’s Cross encryption. Research is an introduction to Algorithm's explanation, and this is the focus of the work.

4.2.3) File Compression:

File compression is also not included as a separate chapter of Ciepiel’s Cross methodology. While it takes a huge part in Algorithm itself, it is just a result of encryption. This Ciepiel's Cross usage will be taken into consideration in future research.

5) Literature Review

5.1) Why Ciepiel's Cross fills the gap:

Keeping in mind that this study is a novel, literature review aims for resolving a gap of undiscovered cryptographic solutions. By reviewing existing algorithms, mathematical patterns and formulas, research should fill the gap by presenting a fully functional and unique algorithm. This, as followed, should be used in steganography subject. Ciepiel's Cross does not provide a non-existing functionality but introduces a new way of encryption approach.

The literature review confirms all discovered mathematical formulas and algorithms to avoid plagiarism. While Ciepiel's Cross sum pattern might not be a revelation itself, its algorithm evaluation, usage, and encryption implementation were undiscovered and not developed before. This puts a question of how many mathematical formulas were not yet found and how many of them could be used as separate and fully working cryptographic solution.

Huge review could be achieved by using a maths originality confirmation databases like: JSTOR, Gale, EBSCOhost, and Internet Archive. Those usually prove a previous existence of given formulas and check originality using their gigantic academic database. The Ciepiel's Cross has been submitted to Journal of Lost Formulae.

Awaiting confirmation, there are several existing issues that Ciepiel's Cross may solve. Looking at security issues, digital data has more than 19 different cryptographic algorithms implemented on the Internet (Kessler, 2021). Widely used are represented by password and text encryptions. Steganography encryption is mainly used within image watermarking or by providing a secret message inside the photo (EC-Council, N/A). This can include viruses, executive code or just a text not visible when opening an image file.

5.2) Differences from other algorithms:

Ciepiel's Cross uses an image to differ its RGB values completely. What is more – it also compresses the file to $1/2^{nd}$ of the size of its original pixels. While file compression is not the central part of this study, it does affect the image output. It only provides more evidence of possible; future uses of the Ciepiel's Cross algorithm. Looking at already existing steganography algorithms, the "encryption output is usually bigger in size" (Staff, 2019). However, there are cases when this way of image encryption keeps the file size by saving new data on top of existing bytes.

5.3) Wartime:

“During the World Wars, many different methods of sending hidden messages were used. Female spies would encode messages in knitted patterns (leading to a ban on new knitting patterns). Photosensitive glass (which shows an image when exposed to the correct wavelength of light) was used during World War II to send messages to Allied forces.” (Furuseth, 2005)

While Ciepiel's Cross steganography differs from described above, it applies the same reason for its existence. Physical objects like photosensitive glass are less likely to be exposed due to the limited tries that the product can be tested. However, those had been exposed. The importance of unbreakable encryption stands with its secret. Ciepiel’s Cross algorithm also uses a modern way of communication. Digital data seems to be more important than physical objects, as today’s world communicates by the Internet.

5.4) LSB Steganography:

LSB steganography hides text information by correlating pixel values inside of the image. This encryption usage can change file size only by increasing it (Team, 2020). Some values use more disk space which occur with file size increasement. In this case, changing the photo's values is dangerous and visible. Dangerous due to the one-way encryption as original values are not stored anywhere. Visible because of the ease of spotting the difference. While an image with hidden text can be no different from the original, it has different pixel values. Confronting two photos next to each other provides an understanding of where changes occur.



Figure 1 - Message baked inside the image. Source: https://www.mygreatlearning.com/blog/image-steganography-explained/

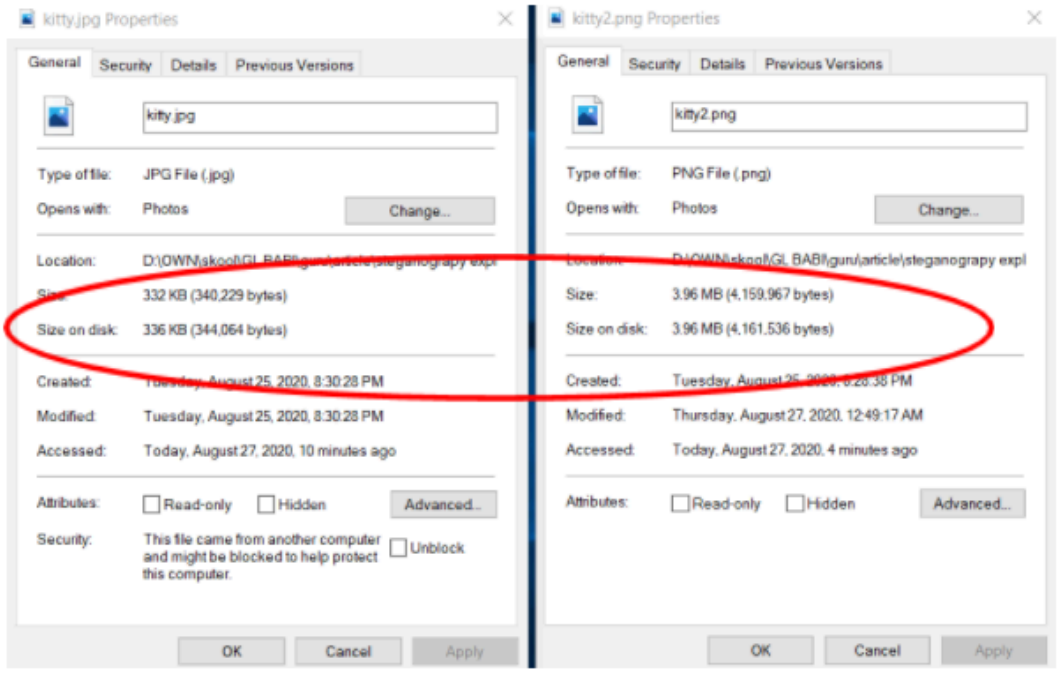


Figure 2 - Difference in size and format after baked message in kitty's image. Source: https://www.mygreatlearning.com/blog/image-steganography-explained/

This is different from what Ciepiel's Cross Steganography Encryption does. Ciepiel's Cross is a basic example of hiding the message, so the unauthorized party could not link the output by any way with its original input. By changing the file size to its $1/2^{nd}$ and changing pixel values, Ciepiel's Cross maximizes security by creating an entirely new picture that can only be decoded using its Algorithm's formula. It might look like a dark cube for anyone seeing the encrypted image, while after decryption, it shows a much bigger and fully understandable photo. It needs to be reminded that uses of both steganography algorithms differ. When some people want to keep the secret message inside the original image, Ciepiel's Cross ultimately secures all the information being put as an input.

5.5) Video Steganography:

“Human’s eye can see up to 1000 frames per second. That allows to hide a message inside the video after increasing its displayed frames to 3000 each second” (Liu Yunxia, Liu Shuyang, Wang Yong, Zhao Hongguo, LiuSi, 2019). Unfortunately, special software or any publicly accessible program allows to separate each frame with any video format. To hide a message, it needs to be separated and hardly accessible to both communication sides. Video steganography is much more secure due to the hardness of message extraction but only once done correctly. If a person puts the whole message in just one frame, it gives minimum to no security. Also, disadvantage of the video steganography might be the file size. To keep messages, disk space is required. (Maerian, 2014) says that “Modern days disks are much bigger than ten years ago”, but that does not help with invisibility. Movie file size is definite and usually exceed few megabytes. This change on disk could be noticeable if a file like this shows on an unauthorized computer. Speed in sending encrypted messages like that also does not work in favour of video steganography. Communication needs to be rapid. Sending bigger files takes time. When adding the decoding time, information might be read too late.

Comparing to Ciepiel’s Cross, video steganography seems to be slightly more time consuming to decode. Both approaches take time to decrypt the message manually. However, Ciepiel’s Cross algorithm can be automated without sending any additional information. When video editing needs extra message to decoder to automate its process (message like what frames should software stop the video to read the information), Ciepiel’s Cross has set rules in which image can be restored to its original values. This can work in favour only when Algorithm is not known and kept in secret.

Ciepiel’s Cross gives an advantage of sending the file that is automatically encrypted and does not include any clues of what file represents. Also, due to its file compression, the size is small enough not to be noticed on the computer's disk.

5.6) Mathematical Formula Existence:

Searching for any plagiarism, there was a need to cover most of the mathematical books to find Ciepiel's Cross sum pattern similarities. Looking on: Ronald J. Tallarida, 1992. Pocket Book of Integrals and Mathematical Formulas, no similar sum pattern exists (Tallarida, 1992). Further research shows that no algorithm has been created using it, and it was never used in cryptography before. Learning from this book proves, that Ciepiel’s Cross is a numbers pattern (sum pattern) and can be described as a formula itself. That allows to move work faster than expected.

Chris McMullen, 2015. 300+ Mathematical Pattern Puzzles: Number Pattern Recognition & Reasoning, proves that years later, sum pattern or similar formula has not been found or described anywhere (McMullen, 2015). This evidence the gap in mathematical formulas. Ciepiel’s Cross provides new evidence and new chapters to be written around mathematical books. Sum pattern and puzzles require to solve some issues using mathematical approaches.

Ciepiel's Cross puts another functionality that provides digital steganography on top of a sum pattern, later called an algorithm.

5.7) Summary:

Ciepiel's Cross is a sum pattern that can be described as a formula itself. It has a unique advantage of decreasing the output size of an encrypted image, which has not been seen around widely used digital steganography. Encrypted files are small and have no similarities with original photos. That provides privacy and security, which was not as strong in above examples. The disadvantage comes with keeping an algorithm hidden.

On the other hand, when knowing the decryption way, it still takes time to deconstruct encrypted image using reverse engineering manually. Another disadvantage might be completely different image output. In some cases, as an advantage, it can also be treated as a limitation. Ciepiel's Cross numbers pattern (Algorithm) cannot be used when someone expects a hidden message to be presented in identical as the input photo. This steganography solution should not be used to keep messages hidden but to keep them not understandable to unauthorized persons. It provides another layer of security that applies only to specific tasks.

6) Methodology

This section is considered as a theory and explanation of Ciepiel's Cross sum pattern. Process of discovery, development progress and results have been introduced in steps to allow complete understanding of Algorithm's creation and cryptography possibilities.

6.1) Discovery

Ciepiel's Cross has been found accidentally. By forcing the discovery to be useful in cryptography, it became classified as a valuable algorithm in steganography. To understand what Ciepiel's Cross does, it needs to be explained where it comes from.

Ciepiel's Cross is the name of a sum pattern that allows repetitively and continuously create the sum of adjacent numbers between two rows.

6.1.1) Sum pattern discovery:

There was a Youtube video showing brain puzzles to solve. One of the puzzles provided two rows of numbers, as shown in figure 3 (GijiGadu, 2020).

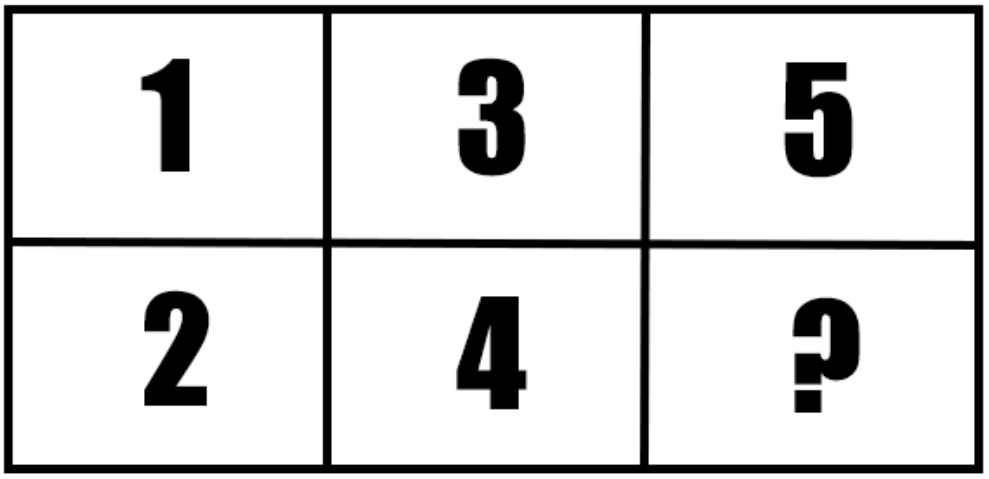


Figure 3 - Youtube quiz - brain puzzle from https://www.youtube.com/watch?v=rFUVqjktJUY

Two rows of numbers with one unknown were described as a puzzle to discover the missing block. First to come to mind is to put the number "6" where the question mark has been placed. Youtuber, who had created the video explained, that "to our shock" it will not be a number "6" but the letter "R", as "Reverse" in cars gearbox. His output is shown in figure 4 below.

1	3	5
2	4	R

Figure 4 - Youtuber's answer to the brain puzzle

Stopping the video, there was one more answer to discover that felt to be correct. Instead of either number “6" and letter "R", the number "5" could be an answer. With that statement figure 5 had been created.

1	3	5
2	4	5

Figure 5 - My approach to solve the brain puzzle

This solution comes with a reasonable cross number summing. It was possible to spot by adding numbers diagonally. In both rows, last columns give a number “5” as a solution. The statement shown in figure 6 and 7.

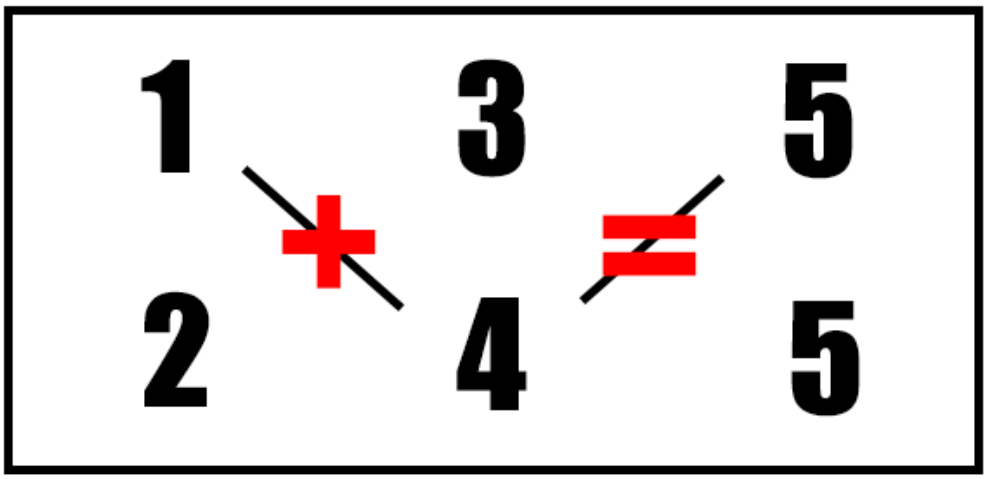


Figure 6 - Ciepiel's Cross sum pattern discovery. First row

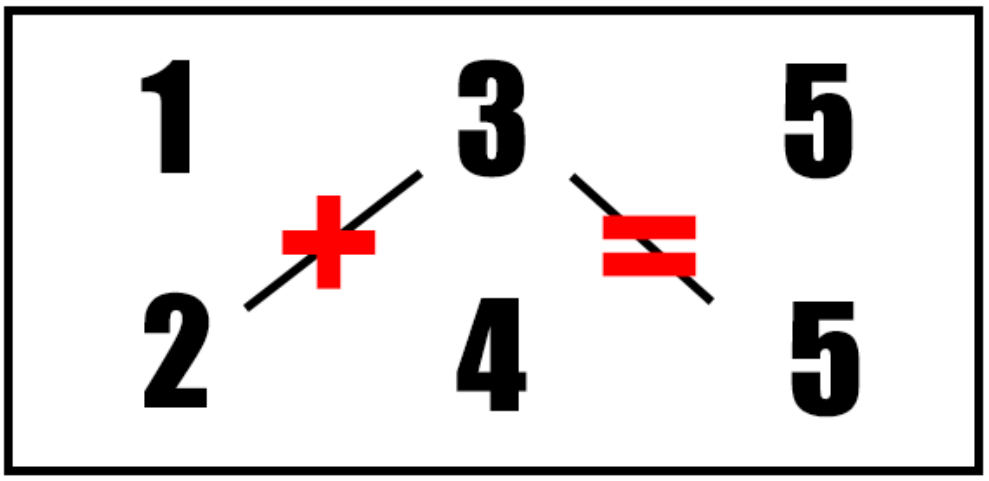


Figure 7 - Ciepiel's Cross sum pattern discovery. Second row

Having that answer, there were two more questions to be asked. Is this continuous? Furthermore, will it repeat for different numbers in both rows? By adding numbers diagonally in theory, it looks there is a way to create two infinite rows that are rationally explained.

For this to confirm, my approach was to diagonally add further numbers to create longer rows, as stated in figure 8.

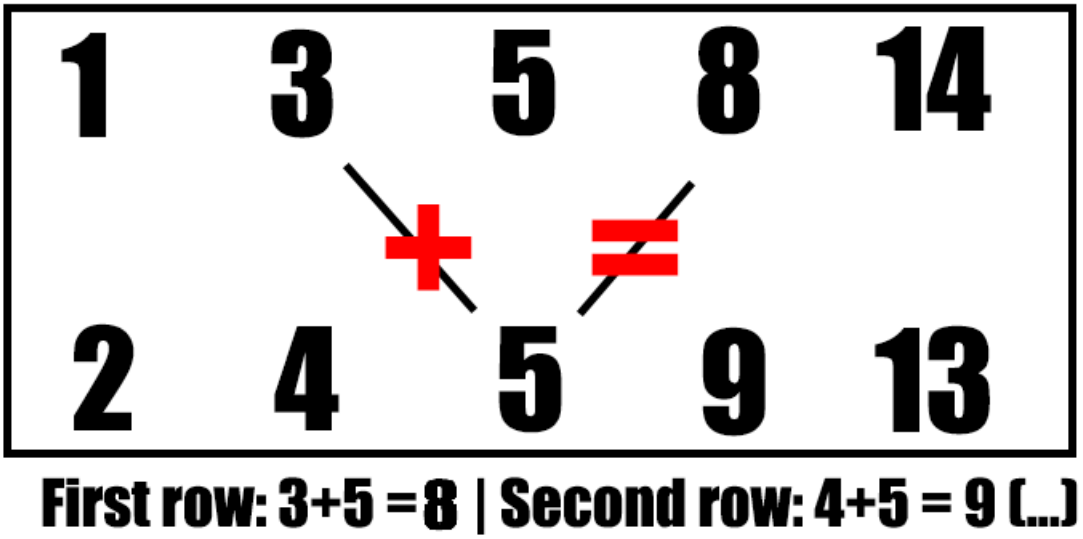


Figure 8 - Further brain quiz discovery research

At this moment it looks, that diagonal summing can create infinite in length rows of numbers. Next approach was to spot any repetitive patterns that occur when length of rows expands. It was visible that after the third column (where numbers "5" are placed), each following number in column A is alternatively bigger and smaller from row B. Figure 9 shows this rule.

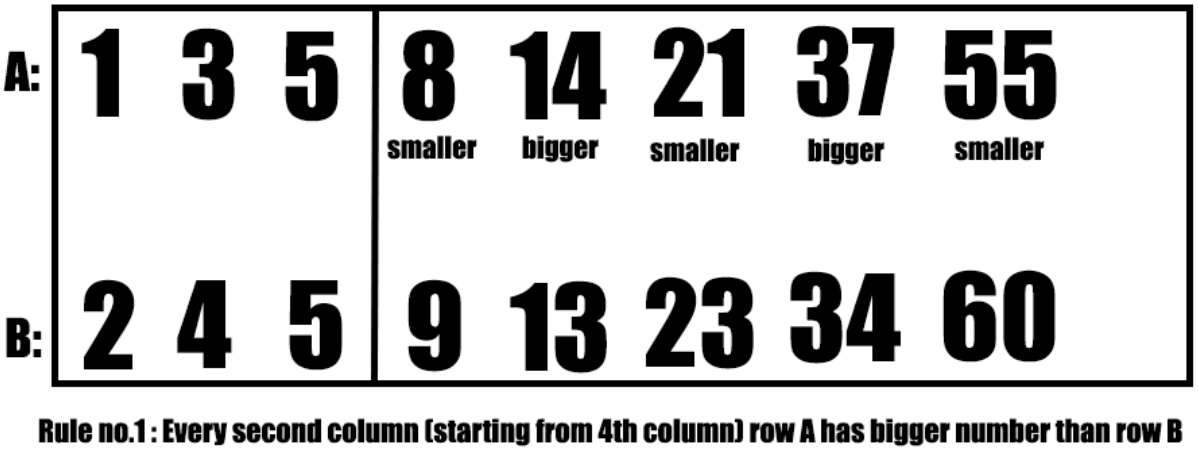


Figure 9 - Rule no.1 - Every second column (starting from 4th column) row A has bigger number than row B

This was the discovery that did not result in any helpful manner to steganography encryption. "Rule no.1" is just an extra function to the sum pattern described in figures 6 & 7.

6.1.2) Ciepiel's Cross discovery:

Continuous and infinite sum pattern was not the discovery I was looking for. Without any third-party inspiration, it took 20 more minutes to find another existing, repetitive pattern inside of described two rows.

The most significant discovery that led to encryption was found within the first three columns. Applying T-Cross-Summing (named as numbers are being added as a letter T) it was possible to spot another rule within this sum pattern.

Representing numbers by letters gives a better understanding of how T-Cross-Summing is applied to the pattern.

6.1.3) Ciepiel's Cross formula creation:

To create a formula, numbers should be represented as letters. This explains that the formula can be applied to any numbers within the sum pattern described above. I have chosen letters stated as follow: x, y, z and v. Those letters will represent numbers, depends on what values should be taken into consideration.

For the first example, let us apply letters to numbers using T-Cross-Summing order within first three columns of Ciepiel's Cross Sum Pattern rows.

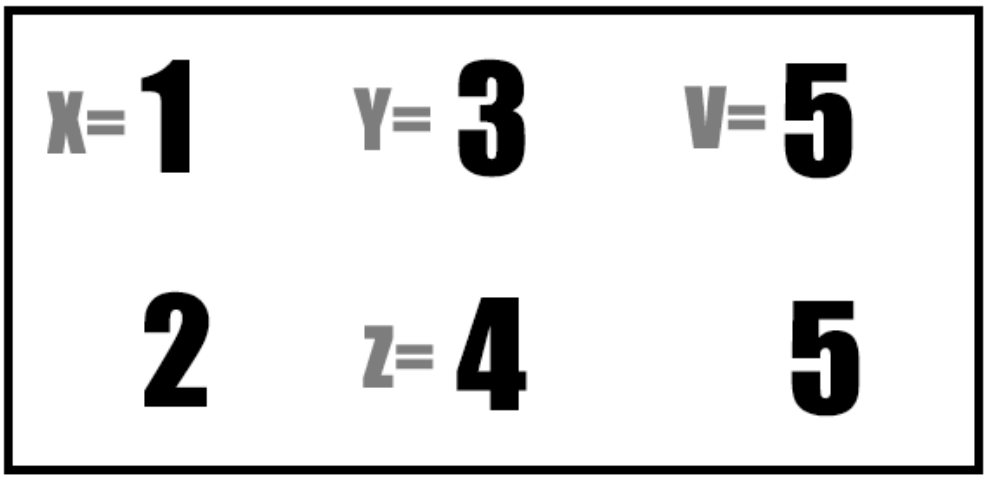


Figure 10 - T-Cross-Summing. Ciepiel's Cross discovery

X is equal to first in order number 1.

Y is equal to second in order number 3.

Z is equal to third in order number 4.

V is equal to fourth in order number 5.

This rule is explained in figure 10.

Applying the formula: **x + y + z = y + v** Where **x = 1, y = 3, z = 4, v = 5**, shows the correctness of given statement for first three columns within the sum pattern. Figure 11 proves the calculation.

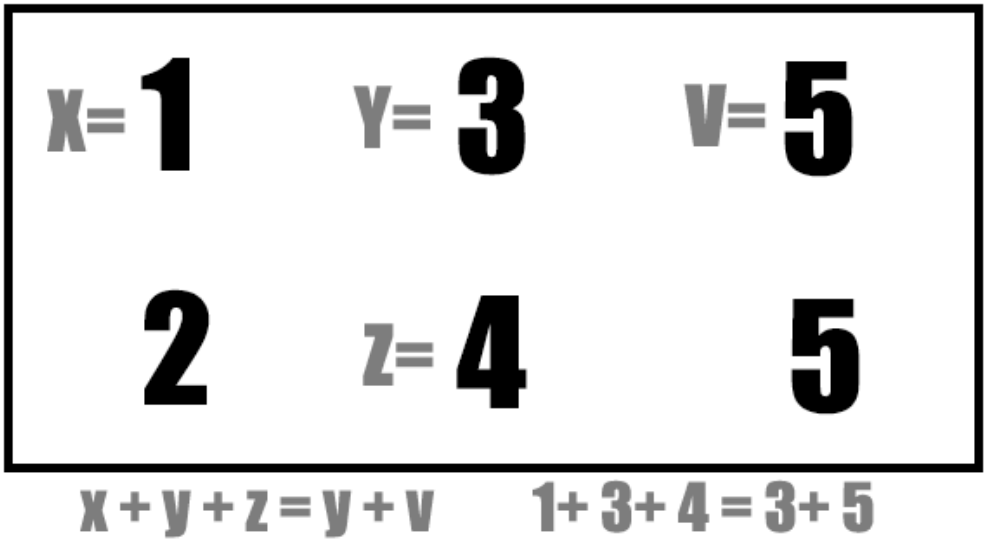


Figure 11 - Ciepiel's Cross formula proof

It looks like the formula is correct for the first statement by providing the number "8" as a solution. To prove that formula exists and is useful, it should be tested over any other place of the sum pattern described in figures 6 & 7.

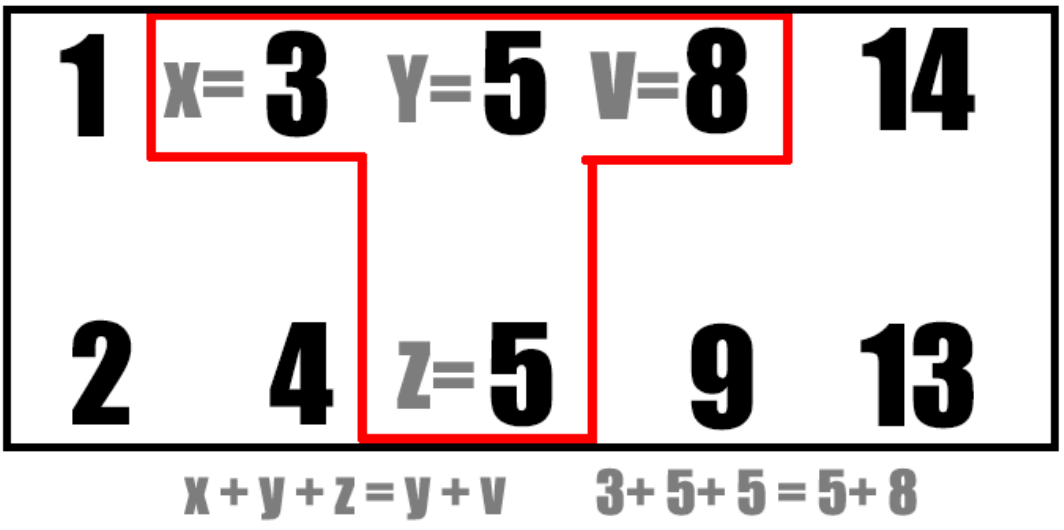


Figure 12 - Second test of Ciepiel's Cross formula

While first numbers had some differences before (example: Rule no.1, seems to work only after the 3rd column), let us take many later numbers from this sum pattern to confirm the formula's correctness.

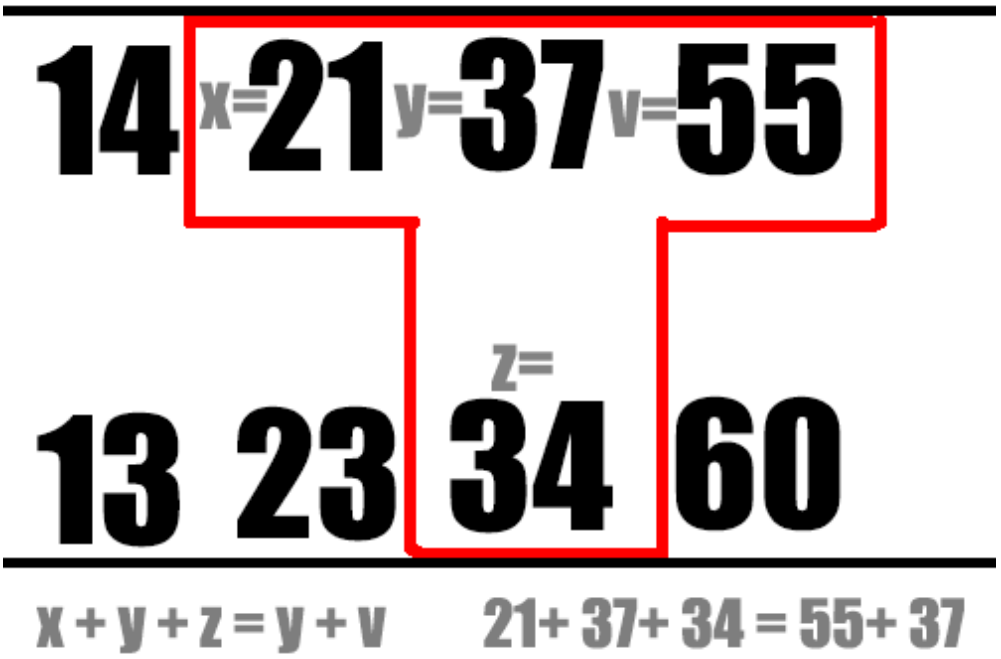


Figure 13 - Third test of Ciepiel's Cross formula

Fortunately, further numbers prove the existence of this pattern as well as confirm the formula existence. In figure 13th, the given output for both sides is number "92".

This discovery, however unique was not the last step in researching new possibilities of the two rows sum pattern. Using the author's name, this formula was named Ciepiel's Cross Formula and led to create a steganography algorithm based on T-Cross-Summing (also named by needs for this research).

Everything described above has been tested and documented within Paint software. Paint is a simple graphic design tool, allowing its users to draw, write or edit pictures with basic functionalities.

6.1.4) Conclusion:

Ciepiel's Cross formula works within Ciepiel's Cross Sum Pattern. It shows proof of being repetitive, continuous, and infinite in length. This solution solves the creation of new mathematical formula that evolves into an algorithm, as described in section 6.2 of this study. Sum pattern uses newly created T-Cross-Summing to identify, that three numbers being equal to two numbers within T letter shape of given two rows of numbers.

6.2) Research Framework

The research framework takes Ciepiel's Cross sum pattern and its formula under another functionality consideration. This section considers that the T-Cross-Summing can be used within encryption in steganography's field. The usability of this formula finds its existence in simple Image's RGB pixel replacement. Following sections will explain the usage of Ciepiel's Cross formula (later called an algorithm) to digital picture encryption and decryption.

6.2.1) Ciepiel's Cross Formula – Further Research:

Ciepiel's Cross sum pattern and formula were forced to be used in cryptography or steganography. There was no correlation between discovered T-Cross-Summing and the image encryption solution. Having in mind that formula cannot be used for anything forced me to develop further research, that evolved to conclusion of creating new file encryption.

Understanding that four numbers from T-Cross-Summing give the same output means, it was possible to represent the sum pattern's numbers as bytes of the computer's file. When this conclusion was not correct as bytes were represented by different numbering (1's and 0's), it was possible to apply sum pattern over the pixels on monitor's screen.

First pixel in the left upper corner of the monitor's screen represents the first number of the "A row" in sum pattern from Figure 4. Each following number from the "A row" represents every next pixel of first pixel's row in the upper monitor's screen. That means, the first upper row of pixels on the monitor, can be represented as numbers of the "A row" from Ciepiel's Cross Sum Pattern. Second row of pixels on the monitor is represented by B row numbers from Ciepiel's Cross Sum Pattern. This is being described in figure 14.



Figure 14 - Pixel's representation as a Ciepiel's Cross sum pattern

Image from figure 14th represents the Photoshop screenshot from the top of the monitor's screen. While the image is not sharp due to the rescaling, it gives an accurate picture of pixels being represented by rows from described above sum pattern. The red pixel represents the "A row" first's number from Ciepiel's Cross Sum Pattern (1). The yellow pixel represents the "B row" first's number from the same pattern (2). Each following from left to right pixel in the first row, represents the "A row" and the same rule occurs in the second row of pixels.

6.2.2) Ciepiel's Cross Steganography Encryption research process:

The first aim was to find the usability of the sum pattern in file encryption. The rule set at the beginning stands that encryption had to use the same numbers as Ciepiel's Cross Sum Pattern to decrypt a message successfully.

Tries in experimenting within the text encryption gave no successful results due to a no reasonable memorization of the pattern's code. While encrypting a text message, each letter had to be remembered as a number, and that concludes in must for letters to not be repeated. When letter in the text would be written twice, sum pattern had to use the same T-Cross-Summing across the message. However unreasonable, it was possible with making breaks between the letters. Nevertheless, this approach was not given into this study consideration. Future work considers, the Ciepiel's Cross Formula can be applied to text messages.

Reviewal and experimental method proved that message could be encrypted within the picture. Having a photo with a written sentence could be encrypted, resulting in not understandable output to any human being but an algorithm owner.

Applying numbers from the sum pattern on top of the image's pixels gives the following in figure 15th result.

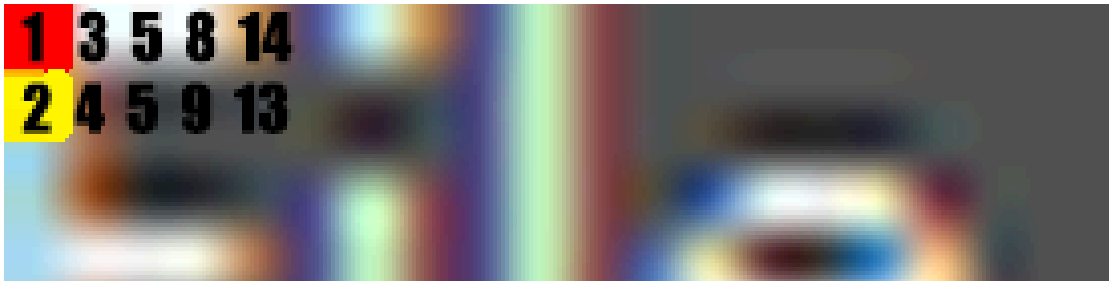


Figure 15 - Applying sum pattern on top of the image

The above's test, results in realization of finding a different way of pixel's representation. Pixels could be represented on top of a digital image of any format, e.g.: .PNG or .JPG, instead of a monitor's screen. Picture from the 15th figure is a .PNG image file that can be used to discover Ciepiel's Cross Steganography functionality.

Unfortunately, for testing purposes image is too large to easily describe and provide a good understanding of encryption over the given formula.

6.2.4) Digital Image's pixels representation

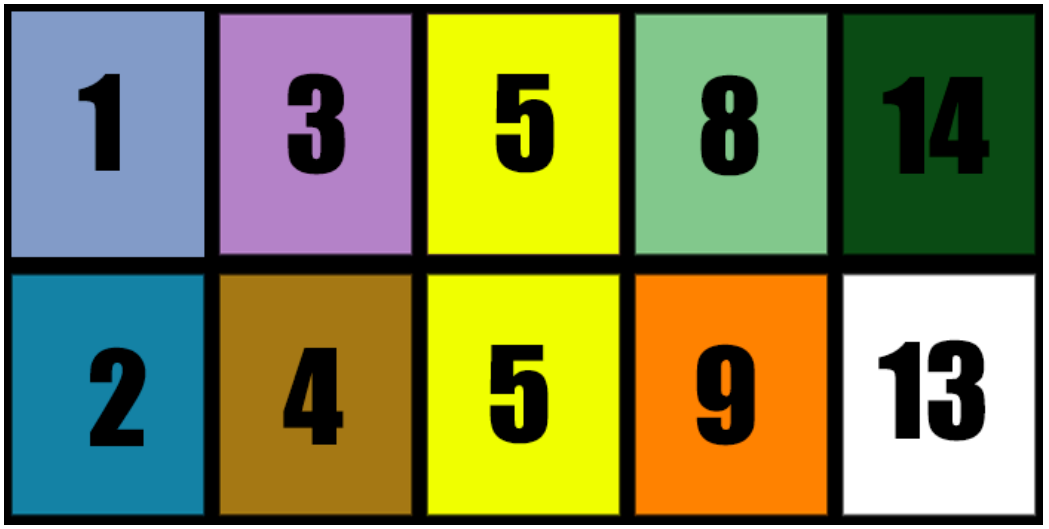


Figure 16 - Implementation of Ciepiel's Cross sum pattern on top of RGB photo

Experimenting with coloured photos and changing RGB of an unencrypted Image was too difficult to test. As a result, for testing purposes, only white and black pixels should be taken into consideration. Reason for this downgrade is only due to the massive numbers to colours memorization. Each number must own its private colour to encrypt the image successfully. Testing just two colours should provide a possibility of practical usage of Ciepiel's Cross Sum Pattern in steganography encryption.

The next rule has to be set. Black pixels need to represent the hidden message (anything that person wants to hide within the image). White pixels should represent blank space that is not being considered as a message itself.



Figure 17 - Black and White image representation of a secret message to be hidden

Figure 17th is a black and white image that shows a message to be encrypted. First row of pixels is all white, second row of the image has some black pixels. A sum pattern should consider those pixels as valid.

There is another issue with an encryption shown in above's figure. There are multiple rows, and Ciepiel's Cross contains only two rows within its pattern. While it can be infinitely wide, there are some limitations in the height of the image. That results in a simple solution of

representing each A and B rows repetitively under another A and B rows as shown in figure 18th.

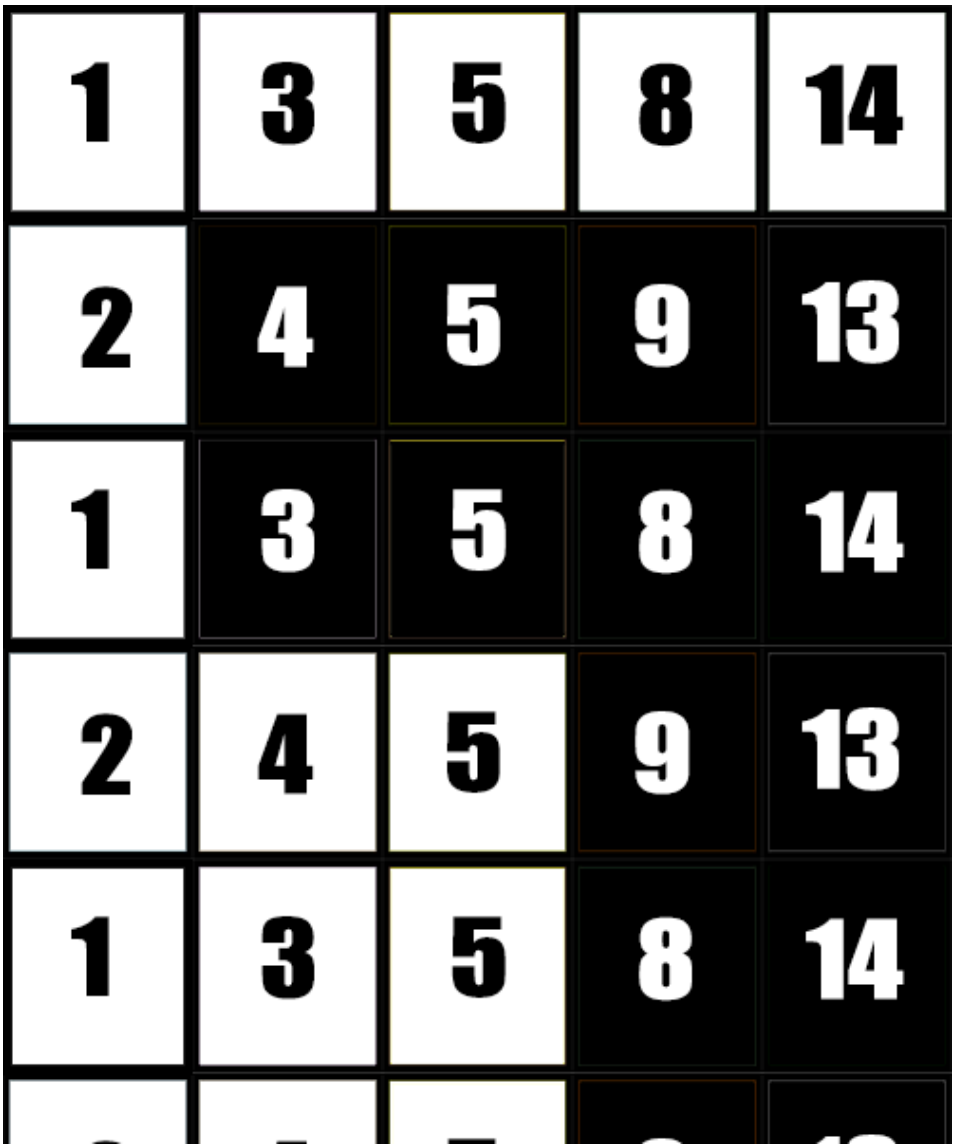


Figure 18 - Sum pattern represented in repetitive two rows over the image

The next step would be an implementation of the Ciepiel's Cross (T-Cross-Summing) pattern on top of the image to hide and encrypt the message. This can be done by selecting that only black pixels will represent correct values for the T-Cross-Summing. If Ciepiel's Cross formula provides incorrect calculations, that means the pixel is white. However, when numbers represented in pixels give correct to formula calculation, the pixel stands as black. Figure 19th explains the formula's pixel confirmation.

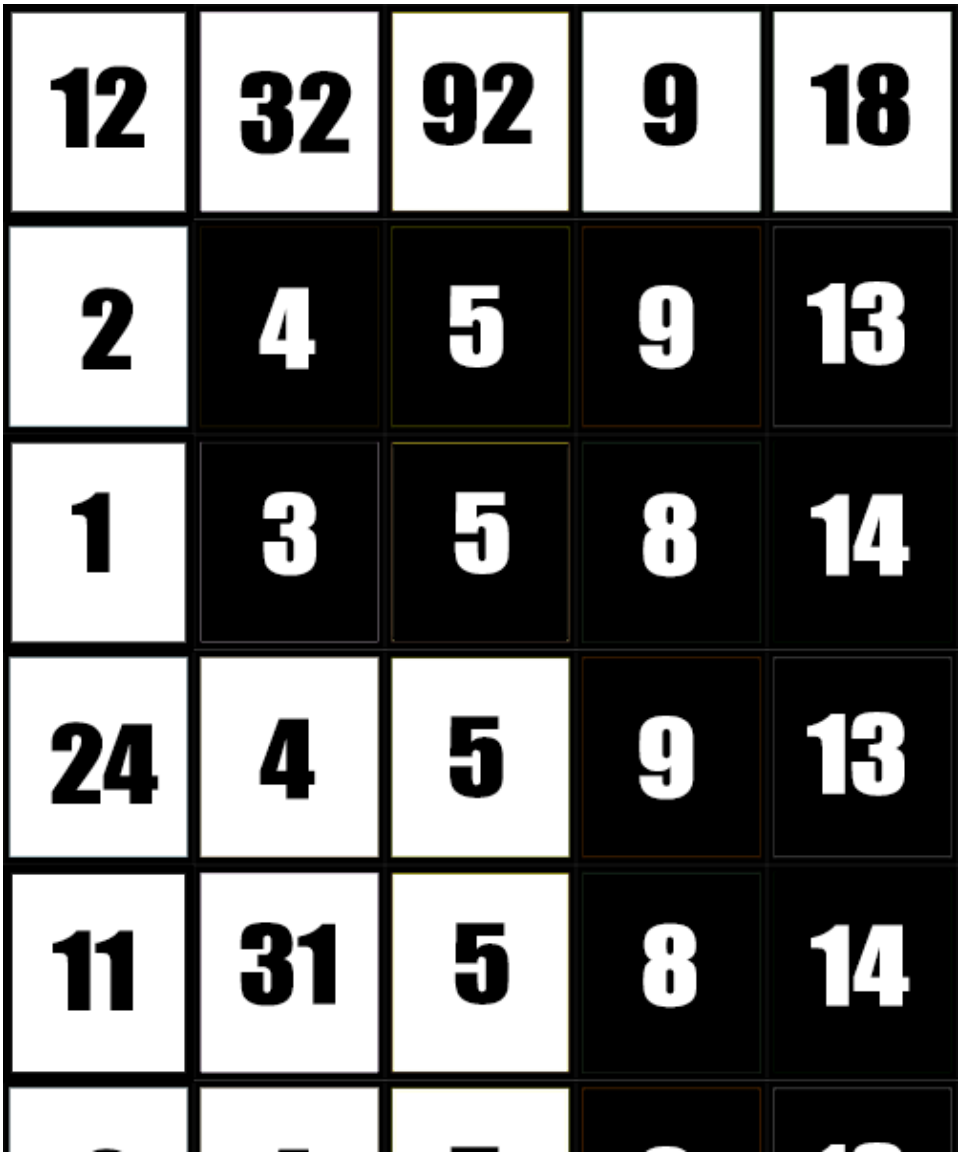


Figure 19 - Understanding the encryption method over Ciepiel's Cross formula

Figure 19th puts correctness on Ciepiel's Cross formula only for black pixels. While applying the formula's T-Cross-Summing on white pixels, output should give invalid to formula's calculations.

Example from 3rd row 2nd column from figure 19th:

1 + 3 + 4 is equal 3 + 5 (black pixel) and taken from different row numbers 24 + 4+ 31 are not equal to 4 + 5 (white pixel).

Unfortunately, it encounters issue where rows do not provide correct values for existing formula. Not everywhere formula gives black pixels, even when it should. Another rule can resolve this.

6.2.3) Rules of Ciepiel's Cross Steganography Encryption:

First encryption rule is that message needs to be written with white blank spaces in specified spots. Having three rows marked as a black pixel put a limitation on the steganography image encryption. The formula cannot be proved correct in this case, as in figure 19th. In figure 20th explanation of providing a correct message is provided.

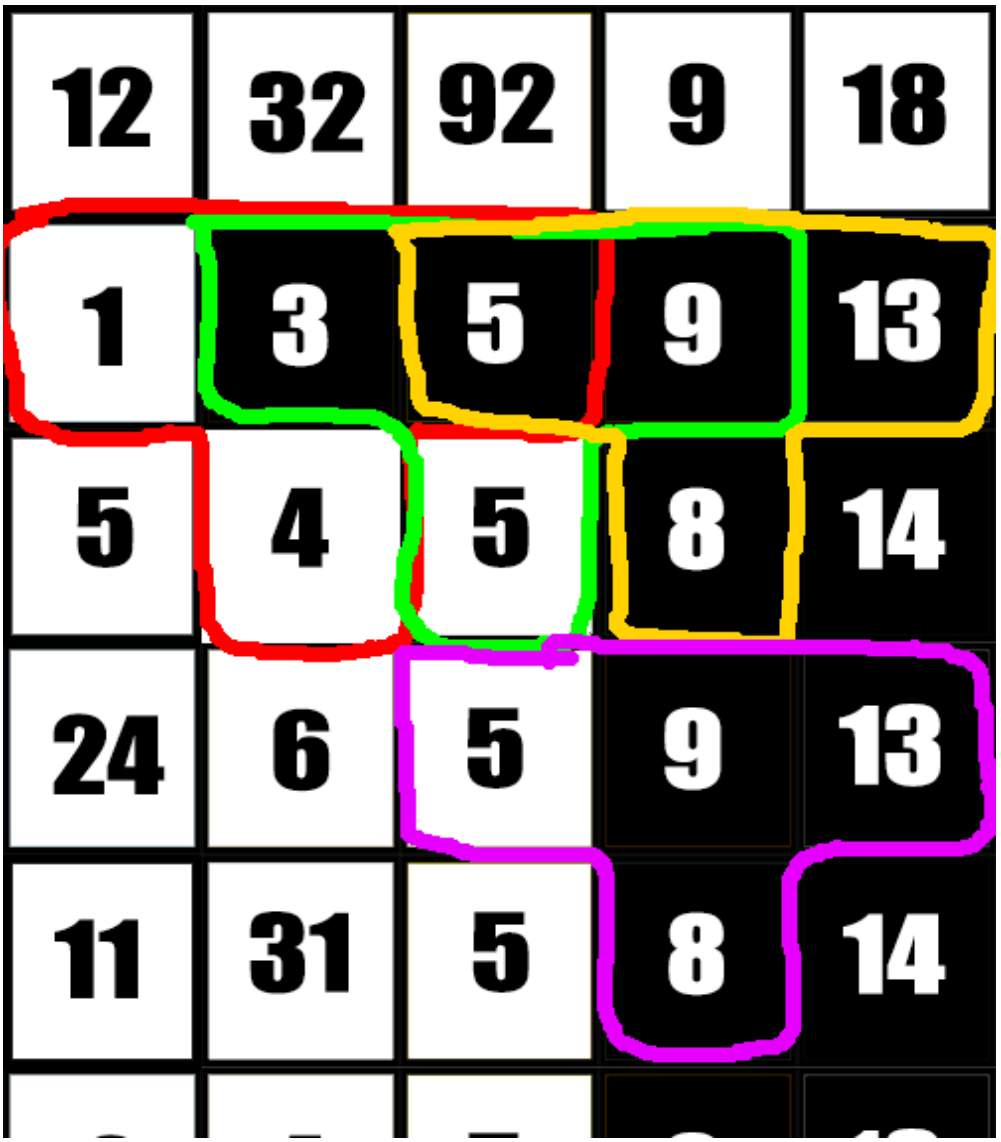


Figure 20 - Two black pixels removal confirms the working formula's order

Other rules have to be set. On the above's 20th figure, there is a missing last column that should allow numbers to check formula's correctness for the 5th column of black pixels. Without the last column being white, it is impossible to prove if the last column should be black or white. That sets a rule that first and last column of the image should always be white.

Figure 20th explains that black pixels should always be in the middle of the T-Cross-Summing. Black pixel should always be represented by the letter "y" in Ciepiel's Cross Formula. This puts another limitation where not every row sets a correct black pixel value. For having only letter “y” as the formula's representation, black pixels should always be written in a Ciepiel’s Cross Sum Pattern. That allows the black pixels to be confirmed by a formula, as stated in figure’s 21.

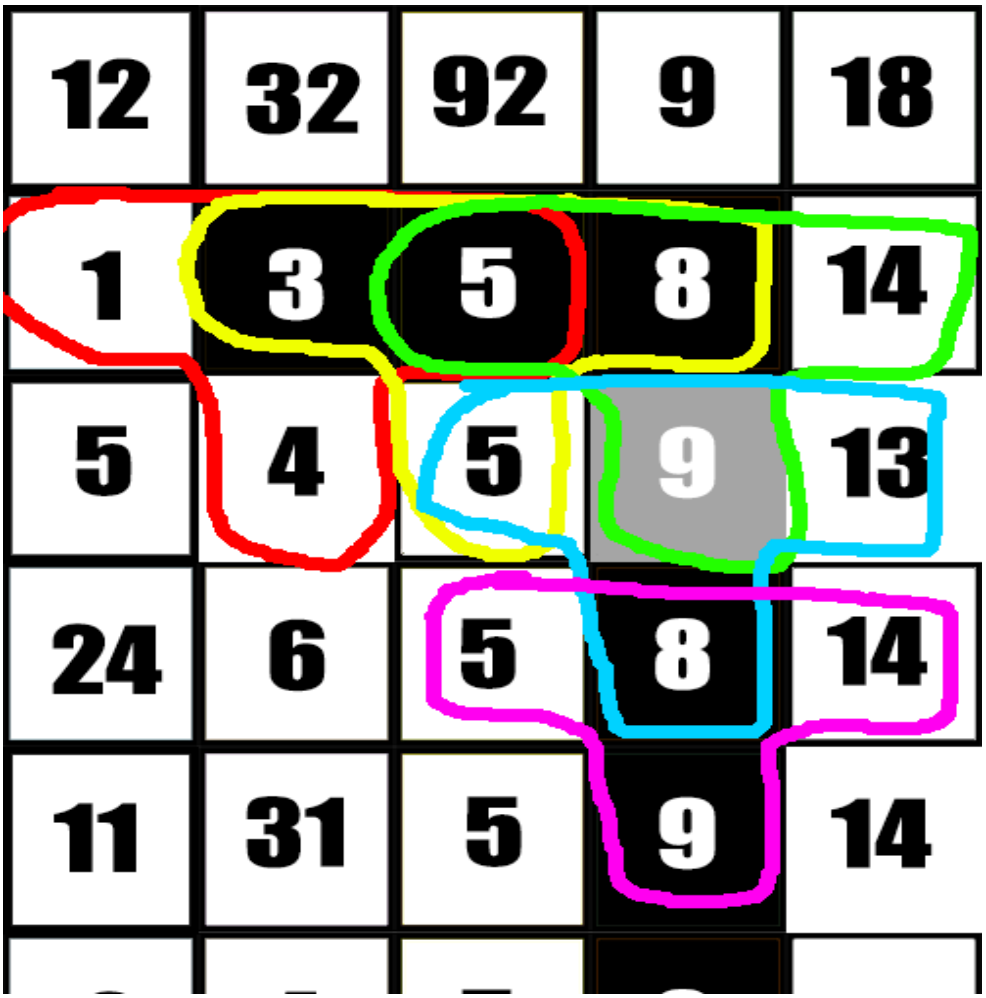


Figure 21 - Ciepiel's Cross formula proves to has more limitations.

Above’s 21st figure greys out the black pixel that is incorrectly treated as valid. Formula does not provide a correct output for the calculation, what puts another difficulty on the image preparation. Another rule says that every second row, picture has to provide a blank space within the message. While having a bigger picture it does not affect understanding of the message, for smaller (like shown on figure 21st) pictures, it cannot provide a detailed output. Figure 22 shows how the bigger message should be prepared to be treated as possibly valid for steganography encryption approach.



Figure 22 - Image almost ready to be encrypted using Ciepiel's Cross formula

It does not affect the message itself but extends the image preparation process. The next step would be the final approach of converting the message by changing its RGB values. This results in the last rule that provides Ciepiel's Cross formula's ground to be used in steganography.

First research has stated the letter "y" from the formula to be representative of the black pixels. While this was correct for detailed pixel by pixel future encryption, it does not work for bigger pictures. Having formula's letter "y" represented as a black pixel gives detailed and correct output only if each row is treated separately. That means, every third row, image should be white pixeled.

At this point, another rule should be created. It can be stated that every value of Ciepiel's Cross Formula is being treated as a black pixel. Each letter from the formula (x, y, z and v) is then black pixeled and treated as valid and not only the "y" letter as stated in figure 21.

Both approaches are correct; however, for testing purposes, I have considered all formula's values represented by pixels to be black coloured. This allows to test both small and bigger images.

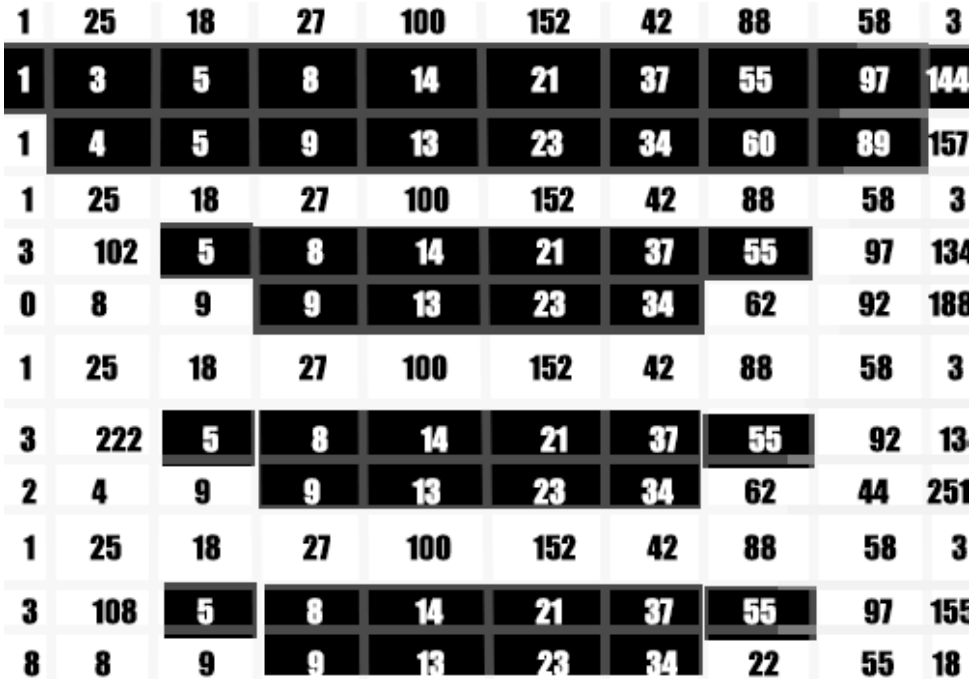


Figure 23 - Complete "T" letter ready for Ciepiel's Cross encryption based on a sum pattern

Figure number 23 represents just a letter "T" from the "This is secret message" image. By setting all black pixels represented as sum pattern values, message gets a sort of distortion that does not affect understanding of the message in the whole context. However, this approach is

perfect representation of how RGB values should be set on the image while encrypting it. Figure 24th shows a complete message with distortion occurred. Both encryption methods (formula's "y" value only and taking all formula values as black pixels) are correct. It depends on the encryption approach.



Figure 24 - Distortion applied to a testing image over a "T" letter

For comparison purposes, distortion has been applied only to a "T" letter (figure 24), as this is the subject of our encryption testing. Encrypting the whole message would be a time-consuming task that cannot be processed in the actual environment.

6.2.4) Encryption approach using MS Paint software:

The main subject of steganography encryption will be the letter "T" from the 24th figure's image. Being considered small enough to fit in tests will be an example and proof of Ciepiel's Cross's usefulness within steganography.

For testing purposes, used for encryption software is Paint. It provides the basic pixel's RGB information window needed for successful message encryption. Figure 25th describes how the original image can be encrypted using Paint's RGB values merge.

Research is being done on a colourful photo for better encryption process understanding. The whole encryption process is reproduced in the "Testing and Implementation" section of this study.

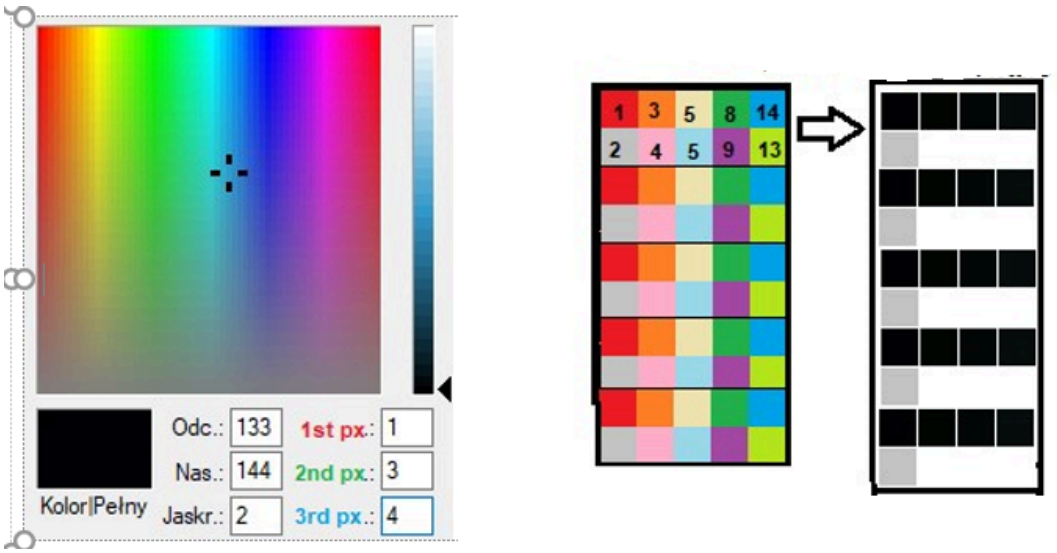


Figure 25 - First encryption approach using Paint. Colourful photo fully encrypted

Red colour is represented by first in order number within the Ciepiel's Cross sum pattern (1). Orange colour is represented by second in order number from the sum pattern (3) and pink colour is represented by third in order number from the sum pattern (4). Those three creates another pattern named for testing purposes: Upside Down Pythagoras Triangle. Encryption occurs by using only three values from Ciepiel's Cross sum pattern as representative of RGB colouring scheme. (RGB contains only three values).

In this step, we are not changing values but creating an entirely new colour by writing original image values into RGB Edit Colours Paint window.

As result, by taking first three in corresponding order pixels, it is possible to create a completely new pixel by setting its RGB values to Upside Down Pythagoras Triangle order. Output pixel of those three looks like a black pixel itself. This is where file compression finds its existence as well. Changing three pixels into one reduces the image's size.

To create second encrypted pixel, the following three in order sum pattern values need to be put in RGB creation. That means the second output pixel will take colours represented by numbers: 5, 8 and 9. Those also output a black pixel right next to the first one; however, both pixels have different values and are different in shading. Having created rules in mind, first column should always be missed and blank. That is why in the given in 25th Figure output, some grey pixels appear.

Results of this approach are being developed under the "Results and Discussion" section.

6.2.5) Encountered issues and new discovery:

The main issue is that steganography encryption could work without the Ciepiel's Cross Sum Pattern only with using its formula's rules. When numbers represented by x + y + z are equal to y + v, then pixel is being represented by black colour. When this equation is not correct, white pixel is being placed. Finding that formula can be used for encryption by itself, it is being covered as a future study that separates it from the sum pattern. With the formula being used without the Ciepiel's Cross Sum Pattern, it does provide detailed output without a distortion shown in figure 24th. This approach puts the sum pattern without concrete usage.

6.3) Data Collection Strategy

Data collection strategy has been set as a testing phase of different existing softwares. Used programs need to be useful in Ciepiel's Cross formula's encryption. Those should provide a window with a separate editor for RGB values of each pixel.

Creating a testing image is not a decisive action and can be processed using any publicly available software like Paint, Photoshop, or any graphical tool.

Rules created for Ciepiel’s Cross Steganography Encryption are the priority to make sum pattern and formula both useful in the encrypted message.

Covering the Literature Review section, this study does not gather any public information to be necessary or valuable while researching the steganography method. Ciepiel’s Cross is an independent sum pattern followed by the original formula. Those are not available in any third-party sources.

6.4) Gantt Chart

The process of development differs, and some gaps in the study creation occur. Ciepiel’s Cross Sum Pattern has been developed over a year ago, with only one year being researched and documented. This Gantt chart provides the work for eight months of study preparation and writing.

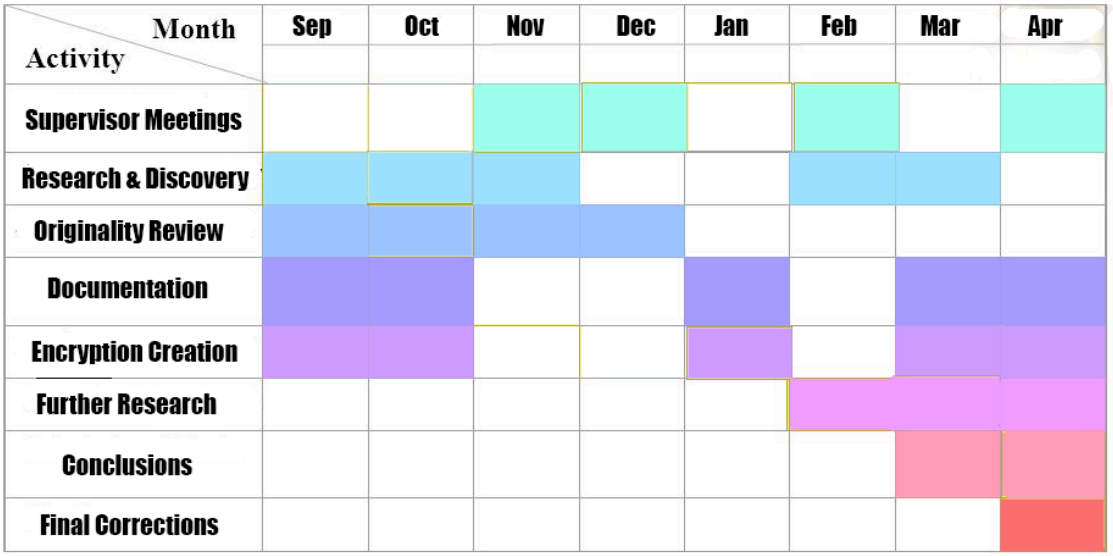


Figure 26 - Gantt Chart created using template found in: http://slideplayer.com/slide/7687098/25/images/28/Example%20of%20Gantt%20Chart%20Month%20Activity%20Sept%20Oct%20Nov%20Dis%20Jan%20Feb%20March.jpg

7) Design and Development

This part of the study does not include necessary steganography encryption information. It can be skipped to the "Testing and Implementation" section.

Decision of what application steganography encryption should follow was not an easy task. Some questions had to be asked before writing the code. Should it be a web application or software? What are the advantages and disadvantages of those solutions? Design and Development section is based on the research and followed by the implementation approach.

This section is only an approach to constructing a fully working application. Having a product of Ciepiel's Cross Algorithm, “Testing and Implementation” section is based on usage of already existing software called MS Paint. Paint allows to fully encrypt given picture and decrypt it by using the Ciepiel’s Cross Formula.

Following “Testing and Implementation” section is being taken as an official encryption approach.

7.1) Coding Method Selection

Being fully aware about issues of web-application, the decision has been made in its favour. Software should be written either in C# or C++ as those languages are more familiar to my expertise. However, as more secure solution, it is also time-consuming. Web method provides

more accessible and better solutions for coding the steganography application. It also can be secured by passwords or be set as inaccessible to public information from server-side.

As encryption could be created in publicly available software like Paint, I had to create RGB pixel values editor similar to existing ones. Paint, however useful is already an existing solution.

“Web application can run remotely as well as locally what differs it from software that can only be used on computer where it was installed” (Rana, 2017). Keeping in mind that computer accessibility is limited, the web application seems to be a good choice for proposing a wide usage of steganography encryption.

7.2) Web-Application Design

Design of the application should be simple, understandable, and intuitive. It needs minimum functionalities allowing the app to upload a picture, take and change its RGB values per pixel and output the result as a new, encrypted image. It needs to understand rules created in the Methodology section; however, those should be implemented last as future work. This study is not about creating a fully working software, but of Ciepiel’s Cross discovery and its encryption possibilities. Main reason for this section is to design a PHP, CSS & HTML application that allows changing RGB pixel values correspondingly to Ciepiel's Cross Formula.

"PHP Web Development always provides the needed support whenever one is stuck with errors and problems regarding the scripting language. Since this is a widely used language, it can be easily learned. The documentation that is found online for the PHP language is free of cost because it is free and has an open-source." (Rana, 2017)

Starting from the design. The application should have a simple playground window with a menu-bar at the top. This is shown in figure 27th below.

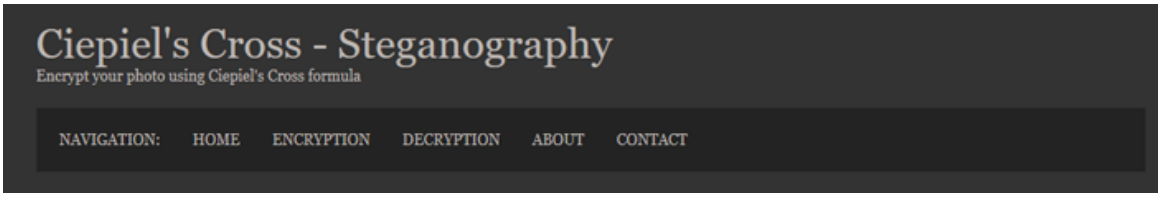


Figure 27 - Simple navigation bar to Ciepiel's Cross web-app

Simple navigation menu keeps the website organized and sectioned. The main focus of this study is around Encryption and Decryption pages. Home, About and Contact pages are strictly cosmetic and will not be considered in this documentation.

7.2.1) Encryption Page Design:

Next step would be an Encryption background page creation. This should contain simple features as:

- Area (In which photos are going to be displayed)
- Buttons: Upload Image, Encrypt Image and Download Image
- And simple instruction.

The mathematical formula, sum pattern, and distortion effects should be implemented automatically by the website in the back-end. Those functionalities should not be visible to the user. Having secured and not publicly visible steganography process, Ciepiel's Cross is hidden and cannot be cracked in different than web-application decryption way. That means only the Decryption page can bring back the original photo from an encrypted output.



Figure 28 - Encryption Page of Ciepiel's Cross web application

Figure 28 represents the main design of the Encryption page. After uploading the image using corresponding button, photo is being displayed in the square box. Next step would be to Encrypt the image using second button, as shown in Figure 29.

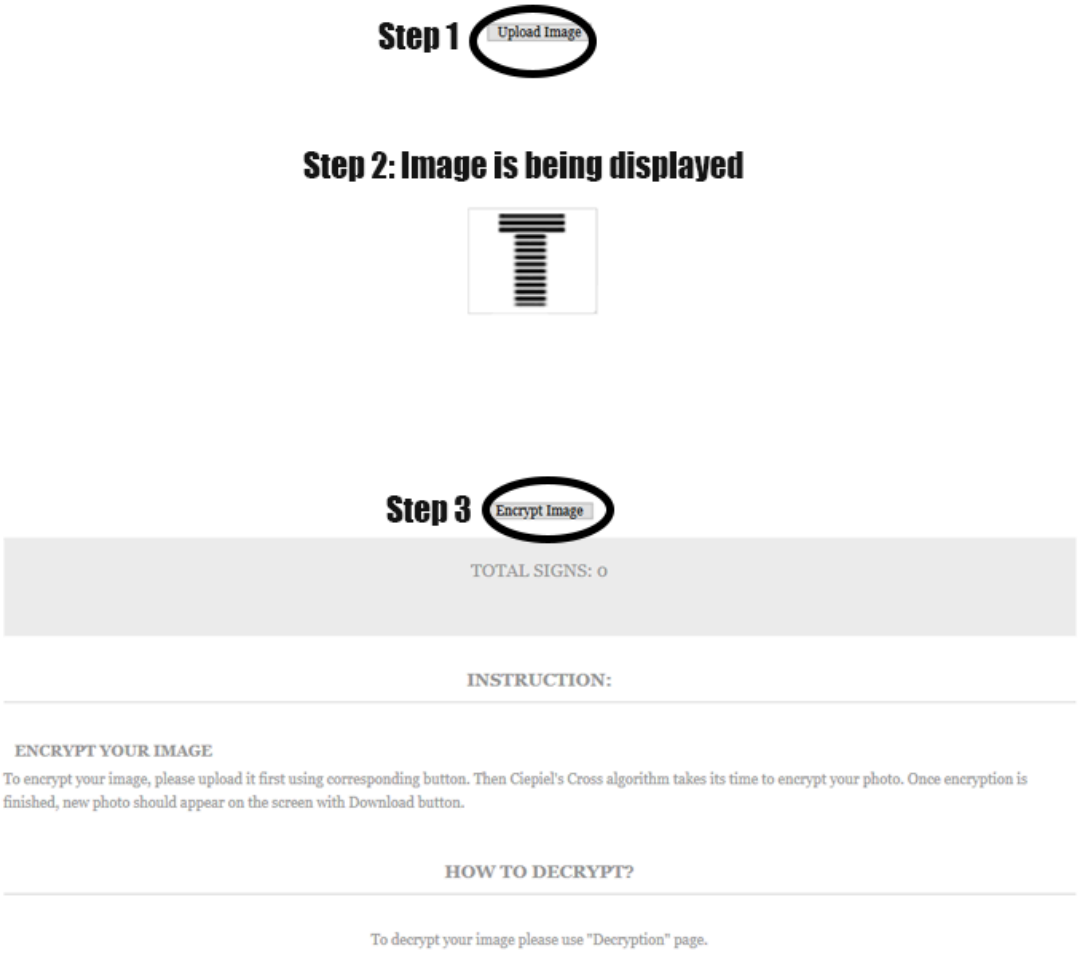


Figure 29 - Ciepiel's Cross web-app encryption instruction

As shown in Figure 29th image is already converted using Ciepiel's Cross's rules. It has been converted in the back-end of the web application to fit the Ciepiel's Cross formula. After pressing the "Encrypt Image" button, photo is being encrypted in the background. User can only see a loading screen that is followed by an output, already encrypted image. All maths is being done without the user intervention.

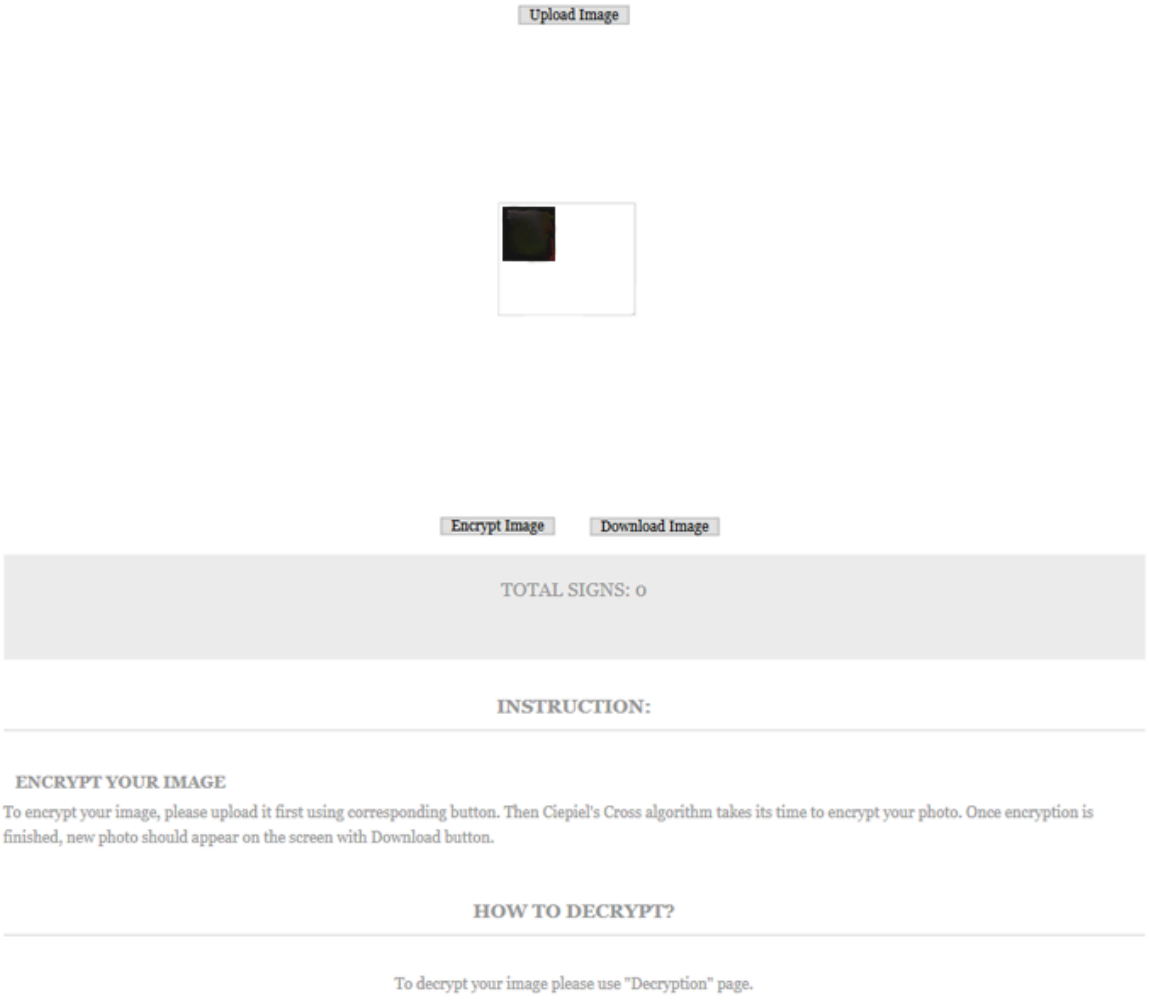
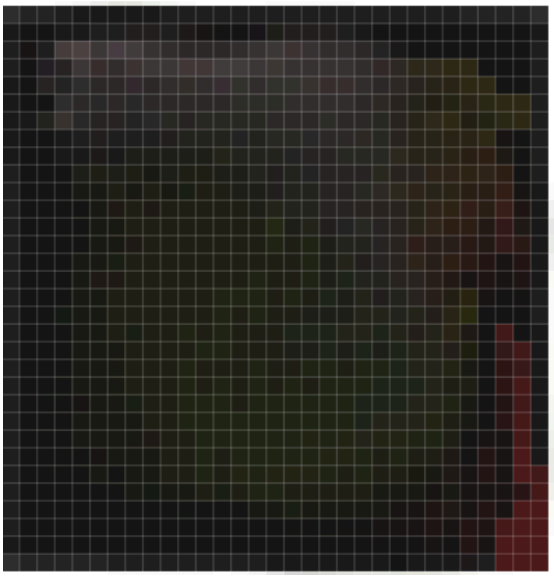


Figure 30 - Encrypted Image using Web-Application

Encrypted image is being displayed by replacing the photo that user uploaded. An additional "Download Image" button appears to download encrypted image to the user's disk.

Encrypted picture is much smaller in size as it uses three pixels and converts them into one. Even if it looks like a small, black square, by zooming the picture some differences appear. Figure 31 represents the zoomed-in encrypted photo.



Both “Upload Image” and “Download Image” functions use the operating system File Manager to get files from the disk. Windows has a file manager that looks like shown in figure 32. This figure presents the process of image uploading.

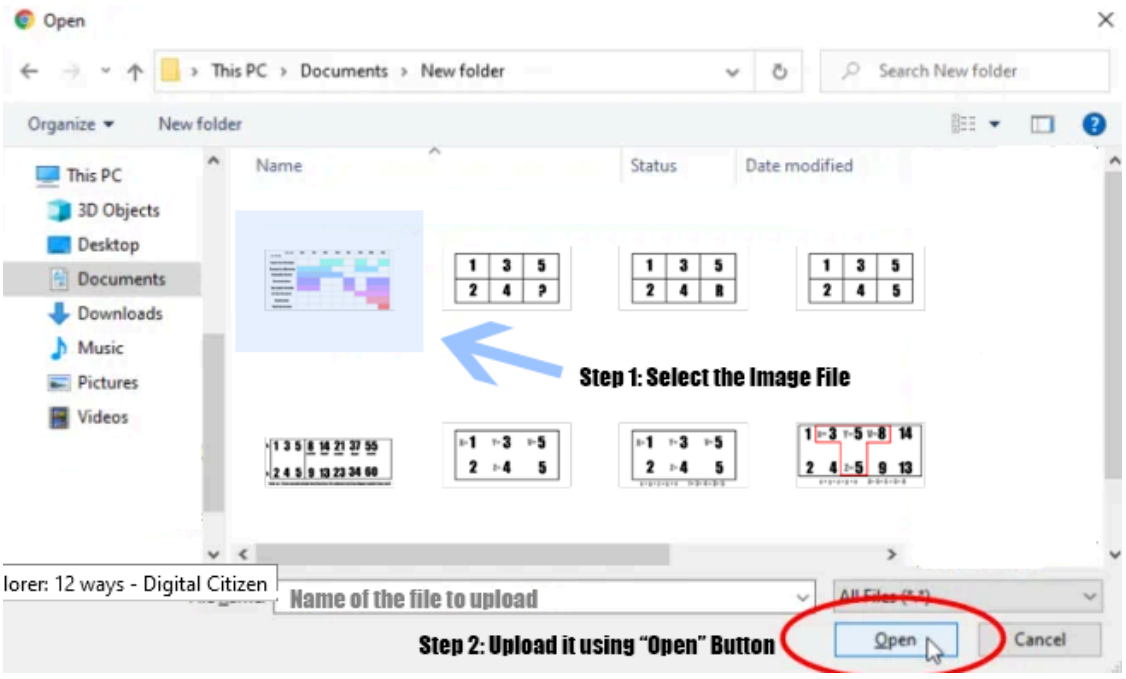


Figure 32 - Instruction of uploading the image file to encrypt it using web-application

Downloading the file looks almost identical, as shown in figure 33.

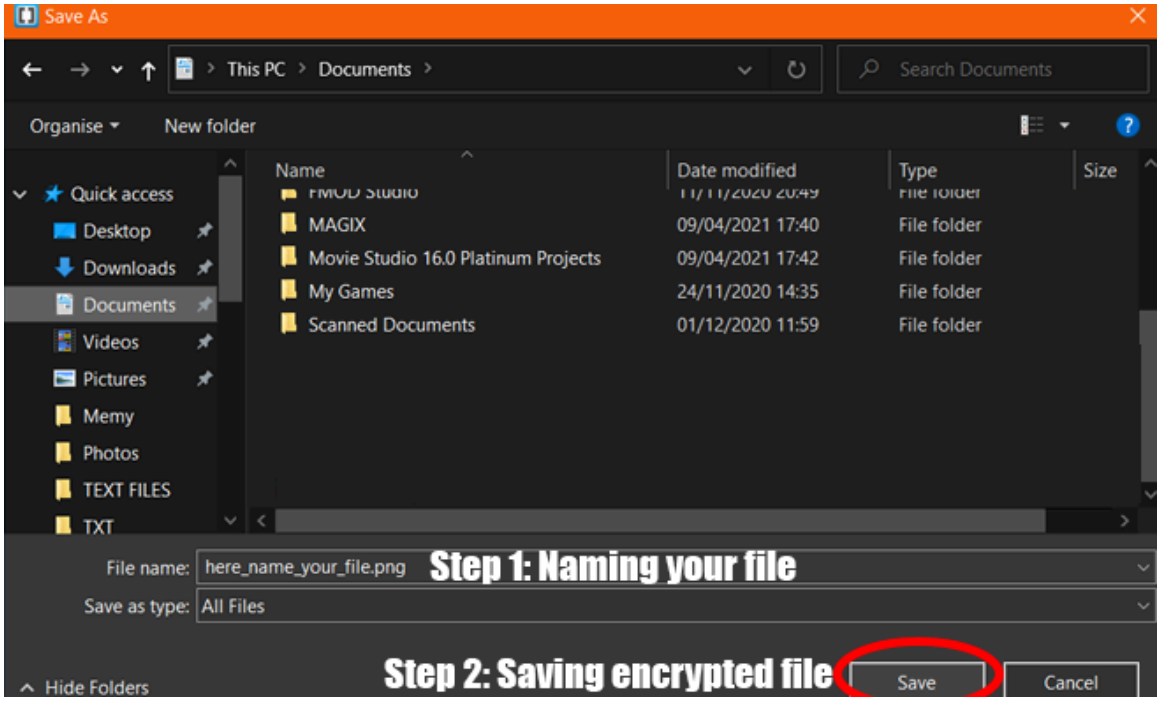


Figure 33 - Downloading already encrypted file from the web-application

Encryption page design ends at this point. The file has been uploaded, encrypted by the web application, and downloaded by the user.

7.2.2) Decryption Page Design:

Decryption Page looks almost identical as Encryption. The difference appears in files that user needs to upload and the process of the background image conversion. For decryption page, image has to be already encrypted to be displayed correctly by clicking the "Decrypt Image" button. Figure 34 is a representation of Decryption page on Ciepiel's Cross steganography application.

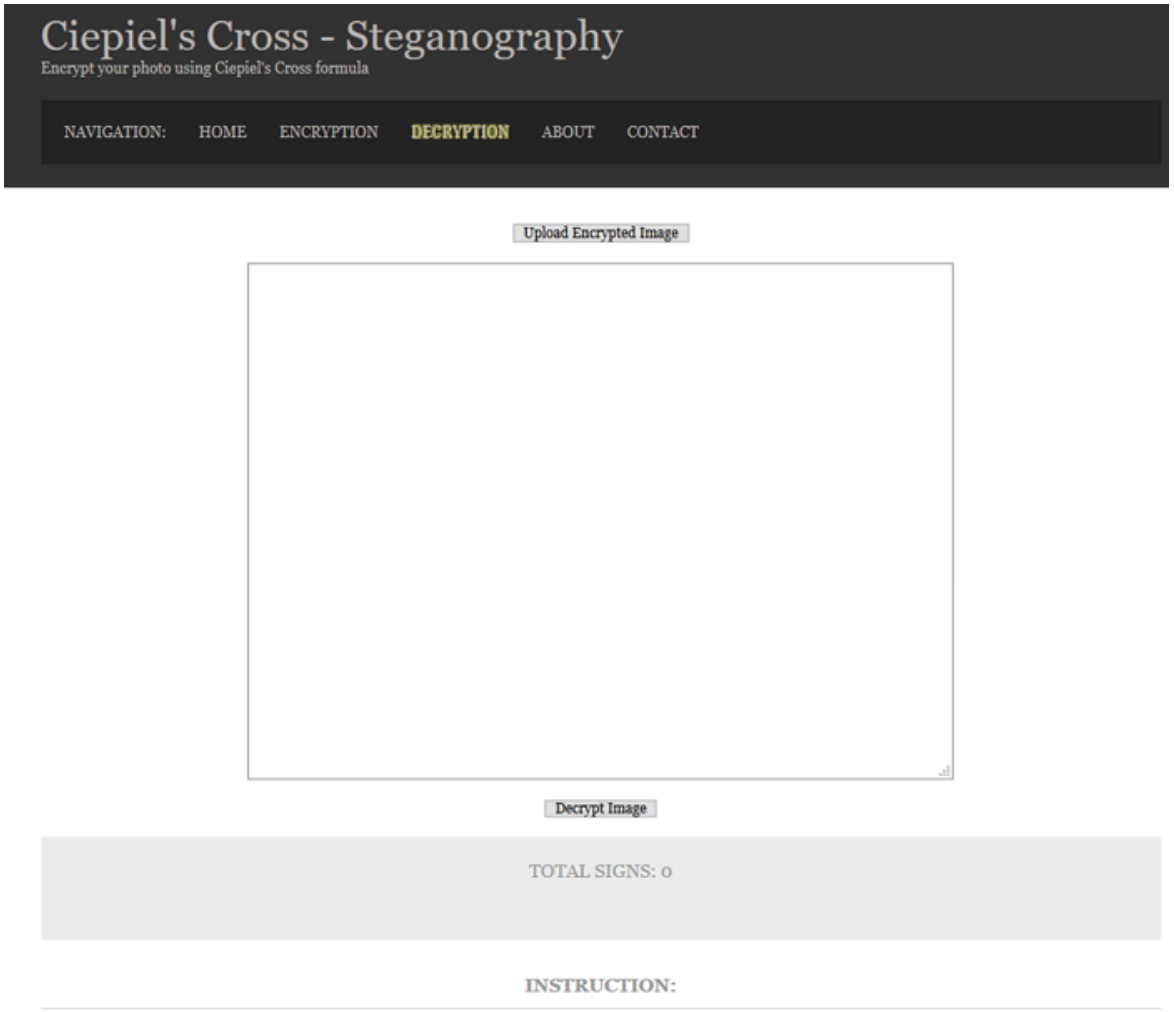


Figure 34 - Decryption Page on Ciepiel's Cross web-application

One extra feature has been added. For a better user experience, page that user is already on has been highlighted in the navigation menu. This allows the user to understand where precisely on the web application is he located. Also, buttons naming has changed to be stricter of what actions are being processed on the selected page.

Design Summary:

Complete web application design has been introduced and simplified to its minimum user intervention. User for both “Encryption” and “Decryption” pages needs in order:

- Upload the image using “Upload Image” button
- Select the image from its own device’s disk
- Wait for the Image Conversion
- Download the image

Same steps occur to both “Encryption” and “Decryption” functionalities. Implementation of this design is being shown in “Implementation” section.

7.3) Web-Application Development (Approach)

Ciepiel's Cross steganography is being written in HTML, CSS, and PHP programming languages. Implementation of this study is sectioned to "Design Development" and "Back-End Development". Web application is considered unfinished and has the potential to be expanded in future work.

7.3.1) Design Development:

Design takes a huge role in user-friendly software. In this section, the code for Ciepiel's Cross web-application page appearance is being described.

7.3.1.1) EQ.PHP & DQ.PHP Design Code:

EQ.php is the main PHP file containing all functionality for encryption page. DQ.php is the code containing all decryption functionalities. Those two are being put together as the overall design for both pages looks the same. To keep this study condensed, only EQ.php design will be documented.

Only design parts of the code are documented in this sub-section.



Figure 35 - Design Implementation Code EQ.php Header

```
<!-- content -->
<div class="wrapper row2">
  <div id="container" class="clear">
    <!-- content body -->
    <center>
        <br>
        <br>
        <!-- UPLOAD IMAGE FORM-->
    <form action="eq.php" enctype="multipart/form-data" method="post">
Select image :
<input type="file" name="file"><br/>
<input type="submit" value="Upload Image" name="Submit1"> <br/> <br> <br>
</form>
        <?php
if(isset($_POST['Submit1']))
{
$filepath = "images/" . $_FILES["file"]["name"];

if(move_uploaded_file($_FILES["file"]["tmp_name"], $filepath))
{
echo "<img src=".$filepath." height=200 width=300 />";
}
else
{
echo "Error !!";
}
}
?>

      <br>
        <br>
          <input type="submit" value="Encrypt Image" name="Submit2">
    </center>
    <br>
```

Container for uploaded images

Upload Image Button

Encrypt Image Button

Figure 36 - Design Implementation Code EQ.php Image container and buttons

```
  <section id="shout">
    </section>
    <!-- main content -->
    <div id="homepage">
      <!-- services area -->
      <center>
      <h1>Instruction:</h1>
      </center>
      <section id="services" class="clear">
          <center>
          <strong>Encrypt Your Image</strong>
          </center>
          <p>To encrypt your image, please upload it first using corresponding button. Then Cie
          its time to encrypt your photo. Once encryption is finished, new photo should appear
          button. </p>
          <br>
          <br>
            <center>
                <h1>How to Decrypt?</h1>
            <p>To decrypt your image please use "Decryption" page. </p>
            </center>
      </section>
    </div>
    <!-- / content body -->
  </div>
</div>
<!-- Footer -->
<div class="wrapper row3">
  <footer id="footer" class="clear">
    <p class="fl_left">Copyright &copy; 2021 - All Rights Reserved - <a href="http://mcgamestud
    </p>
    <p class="fl_right">Created by <a href="http://mcgamestudios.za.pl" title="Ciepiel">Ciepiel
  </footer>
</div>
</body>
</html>
```

Container for instruction note

Instruction for users

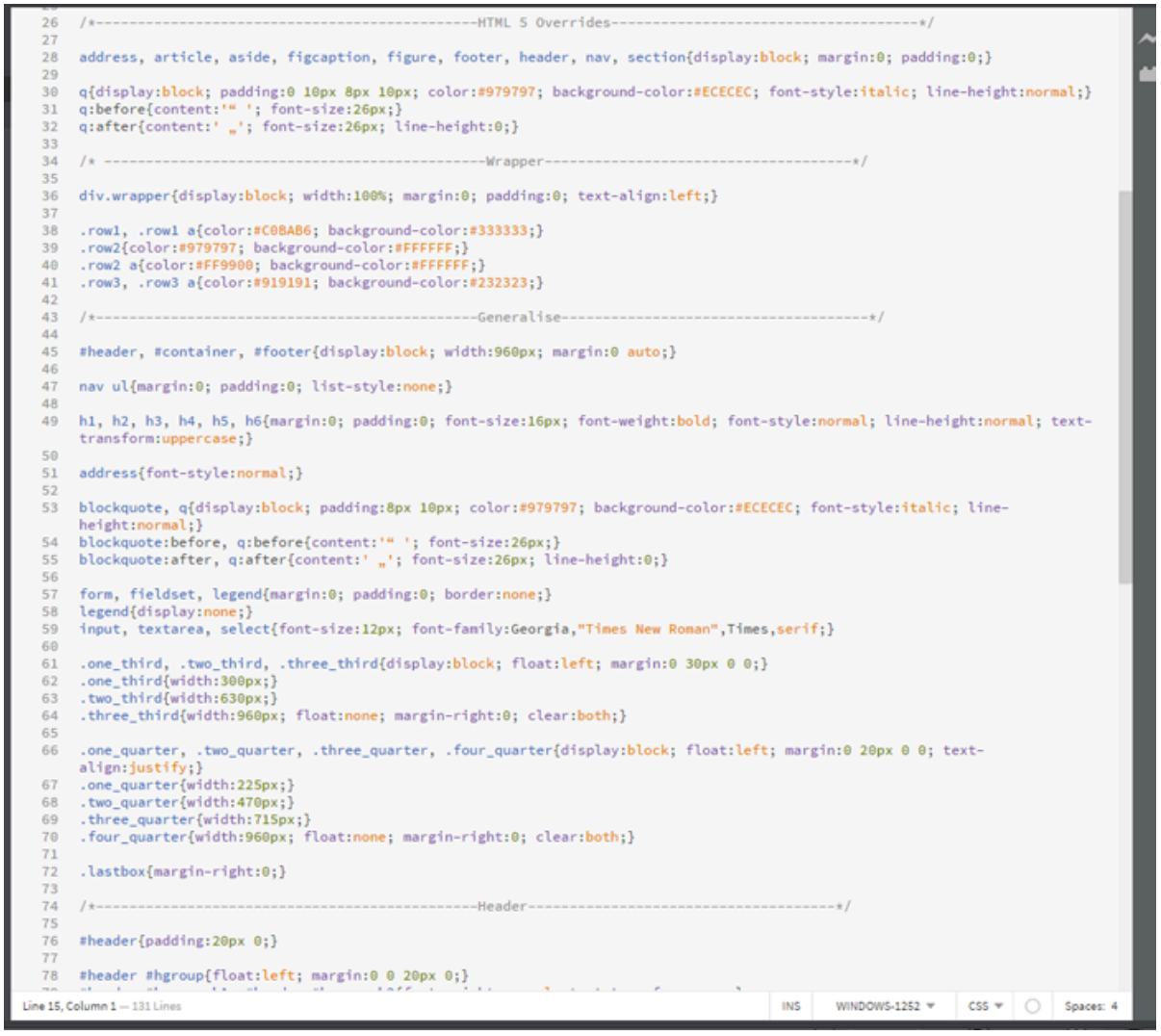
Footer

Figure 37 - Design Implementation Code EQ.php Instruction container and Footer Code

7.3.1.2) Layout.css Design Code:

Layout.css code has been changed using Bootstrap 3 framework found in https://getbootstrap.com/ website. This layout was created in 2016 for a personal project.

“Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. “ (mdo, 2013)



```
/*-----------------------------------------------HTML 5 Overrides-----------------------------------------*/

address, article, aside, figcaption, figure, footer, header, nav, section{display:block; margin:0; padding:0;}

q{display:block; padding:0 10px 8px 10px; color:#979797; background-color:#ECECEC; font-style:italic; line-height:normal;}
q:before{content:'“ '; font-size:26px;}
q:after{content:' „'; font-size:26px; line-height:0;}

/* ----------------------------------------------Wrapper----------------------------------------*/

div.wrapper{display:block; width:100%; margin:0; padding:0; text-align:left;}

.row1, .row1 a{color:#C0BAB6; background-color:#333333;}
.row2{color:#979797; background-color:#FFFFFF;}
.row2 a{color:#FF9900; background-color:#FFFFFF;}
.row3, .row3 a{color:#919191; background-color:#232323;}

/*-----------------------------------------------Generalise----------------------------------------*/

#header, #container, #footer{display:block; width:960px; margin:0 auto;}

nav ul{margin:0; padding:0; list-style:none;}

h1, h2, h3, h4, h5, h6{margin:0; padding:0; font-size:16px; font-weight:bold; font-style:normal; line-height:normal; text-
transform:uppercase;}

address{font-style:normal;}

blockquote, q{display:block; padding:8px 10px; color:#979797; background-color:#ECECEC; font-style:italic; line-
height:normal;}
blockquote:before, q:before{content:'“ '; font-size:26px;}
blockquote:after, q:after{content:' „'; font-size:26px; line-height:0;}

form, fieldset, legend{margin:0; padding:0; border:none;}
legend{display:none;}
input, textarea, select{font-size:12px; font-family:Georgia,"Times New Roman",Times,serif;}

.one_third, .two_third, .three_third{display:block; float:left; margin:0 30px 0 0;}
.one_third{width:300px;}
.two_third{width:630px;}
.three_third{width:960px; float:none; margin-right:0; clear:both;}

.one_quarter, .two_quarter, .three_quarter, .four_quarter{display:block; float:left; margin:0 20px 0 0; text-
align:justify;}
.one_quarter{width:225px;}
.two_quarter{width:470px;}
.three_quarter{width:715px;}
.four_quarter{width:960px; float:none; margin-right:0; clear:both;}

.lastbox{margin-right:0;}

/*-----------------------------------------------Header----------------------------------------*/

#header{padding:20px 0;}

#header #hgroup{float:left; margin:0 0 20px 0;}
```

Figure 38 - Layout.css Design Code Part 1

```
#header #hgroup{float:left; margin:0 0 20px 0;}
#header #hgroup h1, #header #hgroup h2{font-weight:normal; text-transform:none;}
#header #hgroup h1{font-size:36px;}
#header #hgroup h2{font-size:13px;}

#header form{display:block; width:290px; float:right; margin:20px 0; padding:0;}
#header form input{display:block; float:left; width:200px; margin:0; padding:5px; color:#C0BAB6; background-color:#232323; border:1px solid #666666;}
#header form #sf_submit{display:block; float:right; width:70px; font-size:12px; font-weight:bold; text-transform:uppercase; color:#FFFFFF; background-color:#FF9900; border:none; cursor:pointer;}

#header nav{display:block; width:100%; margin:0; padding:20px 0; color:#C0BAB6; background-color:#232323; clear:both;}
#header nav ul{padding:0 20px;}
#header nav li{display:inline; margin-right:25px; text-transform:uppercase;}
#header nav li.last{margin-right:0;}
#header nav li a{color:#C0BAB6; background-color:#232323;}
#header nav li a:hover{color:#FF9900; background-color:#232323;}

/*-----------------------------------------------Content Area-----------------------------------*/

#container{padding:30px 0;}
#container section{display:block; width:100%; margin:0 0 30px 0; padding:0;}
#container .last{margin:0;}
#container .more{text-align:right;}

/* ------Slider-----*/

#container #slider{}

/* ------Shout-----*/

#container #shout{display:block; width:930px; padding:15px; font-size:16px; line-height:1.8em; color:#979797; background-color:#EEEEEE; text-transform:uppercase; text-align:justify;}
#container #shout p{margin:0; padding:0;}

/* ------Main Content-----*/

#container #homepage{display:block; width:100%; line-height:1.6em;}
#container #homepage h1{margin-bottom:25px; padding-bottom:10px; border-bottom:1px solid #D6D6D6;}

#container #homepage #services{}
#container #homepage #services article{display:inline; float:left; width:300px; margin:0 30px 0 0; padding:0;}
#container #homepage #services .last{margin-right:0;}
#container #homepage #services figure{display:inline; float:left;}
#container #homepage #services strong{float:left; margin:8px 0 0 10px; padding:0; font-size:14px; font-weight:bold; text-transform:uppercase;}
#container #homepage #services p{display:block; margin:0; padding:5px 0 0 0; clear:both; line-height:1.6em;}
#container #homepage #services .spacer{clear:both; padding:8px 0; line-height:normal;}

#container #homepage #latest{}
#container #homepage #latest article{}
#container #homepage #latest figure{padding:4px; border:1px solid #D6D6D6; text-align:center;}
#container #homepage #latest figcaption{padding:5px; text-align:center; color:#979797; background-color:#EEEEEE; text-transform:uppercase; text-align:center;}

/*-----------------------------------------------Footer-----------------------------------*/

#footer{padding:20px 0;}
#footer p{margin:0; padding:0;}
```

Figure 39 - Layout.css Design Code Part 2

7.3.2) Theoretical Functionality Development:

Functionality implementation has changed the design presented in section “Design Development". Some solutions proposed by design could not work while coding the web application. Changes are primarily cosmetic and do not affect any above's descriptions.

After the design improvement, the website looks as shown in figure 40.

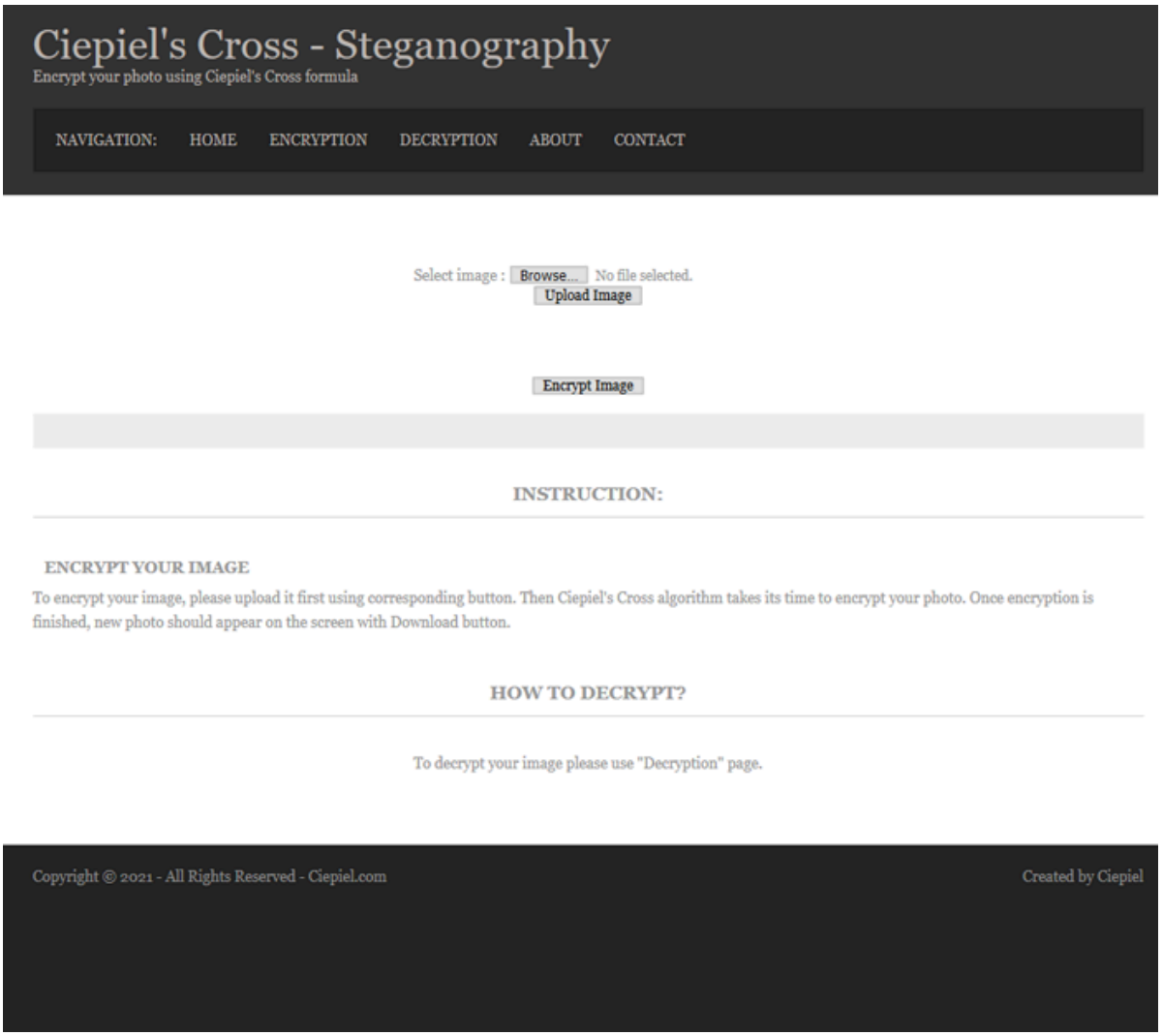


Figure 40 - Design changes during the Implementation phase

First user needs to browse the file, that he wishes to encrypt. Then by using an “Upload Image” button, the photo is being displayed at the centre of the page. Figure 41st shows how the design looks after the image has been uploaded by the user.

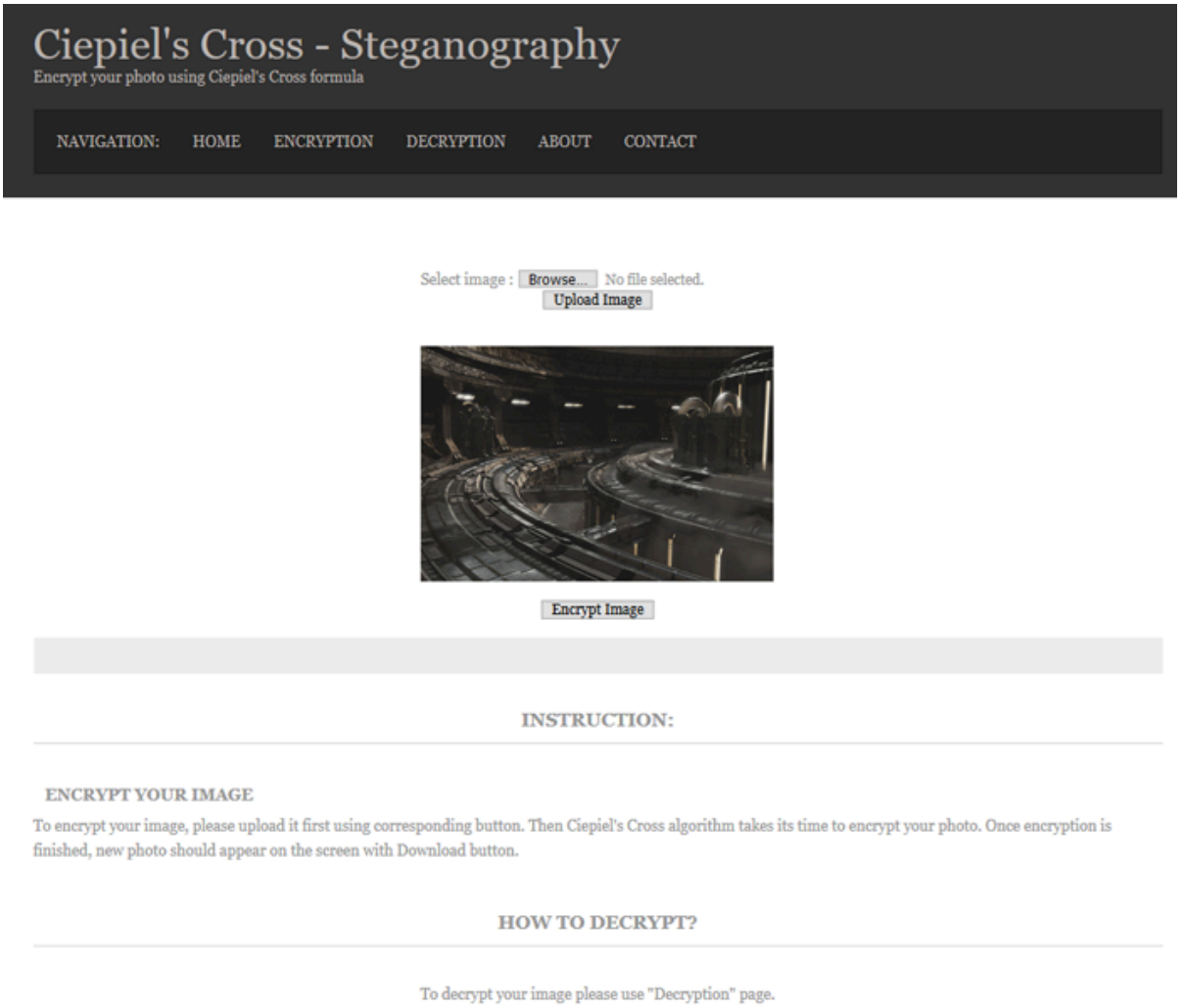


Figure 41 - Changes in Design after implementation. Uploaded and displayed image

7.3.2.1) EQ.php Theoretical Functionality Code:

For testing purposes, the uploaded image needs to be converted to a reasonable size. As a result, it loses its details. This step is also essential to keep the website's design tight. Future work allows to remove this feature and improve file importing.

In figure 41, EQ.php functionality is being described.

```
<!-- content -->
<div class="wrapper row2">
  <div id="container" class="clear">
    <!-- content body -->
    <center>
        <br>
        <br>
        <!-- UPLOAD IMAGE FORM-->
    <form action="eq.php" enctype="multipart/form-data" method="post">
Select image :
<input type="file" name="file"><br/>
<input type="submit" value="Upload Image" name="Submit1"> <br/> <br> <br>
</form>
        <?php
if(isset($_POST['Submit1']))
{
$filepath = "images/" . $_FILES["file"]["name"];

if(move_uploaded_file($_FILES["file"]["tmp_name"], $filepath))
{
echo "<img src=".$filepath." height=200 width=300 />";
}
else
{
echo "Error !!";
}
}
?>

      <br>
        <br>
          <input type="submit" value="Encrypt Image" name="Submit2">
    </center>
    <br>

  <section id="shout">
    </section>
    <!-- main content -->
    <div id="homepage">
      <!-- services area -->
      <center>
```

Uploading the image Button

Image is being taken from User's Disk

Image is being resized

If there is no image, show error message

Figure 42 - Eq.php Functionality Code - Uploading the image

At this stage, code does not apply any Ciepiel's Cross formula's rules to the uploaded photo (Figure 42). This needs to be improved by applying distortion to any image uploaded by the user. For testing purposes, image will be white and black only as the formula does not accept other colours for now.



In Figure 44, code that puts white lines every two rows of the newly created image is shown. First the new, entirely black image is being created as a 250x250 pixels square. Then white lines are being applied to the image every three rows. This leaves the space for Ciepiel's Cross Sum Pattern to be applied and fully understandable by Ciepiel's Cross formula.

```php
<?php
    // Generate image resource with a width an height of 250 pixels.
    $im = imagecreatetruecolor(250, 250);

    // Create a color.
    $white = imagecolorallocate($im, 255, 255, 255);

    // Draw a white line from 0x,0y to 300x,0y.
    imageline($im, 0, 0, 300, 0, $white);
    // Draw a white line from 0x,0y to 300x,0y.
    imageline($im, 3, 0, 300, 0, $white);
    imageline($im, 6, 0, 300, 0, $white);
    imageline($im, 9, 0, 300, 0, $white);
    imageline($im, 12, 0, 300, 0, $white);
    imageline($im, 15, 0, 300, 0, $white);

    // Write the image resource to a file called line.png.
    imagepng($im, 'disstorsion.png');

    // Destroy the image resource.
    imagedestroy($im);
?>
```

Figure 44 – Theoretical Image distortion by putting lines every 3 rows. Code found at: https://www.hashbangcode.com/article/drawling-line-pixels-php

Now, this code needs to be applied to an uploaded by user's image.

```php
<?php
if(isset($_POST['Submit1']))
{
$filepath = "images/" . $_FILES["file"]["name"];

if(move_uploaded_file($_FILES["file"]["tmp_name"], $filepath))

    //Adding a variable to uploaded image

    $dissortedImage = ($_FILES["file"]["tmp_name"], $filepath); //Uploaded File is being named as dissortedImage

    //Dissort Image by Ciepiel's Cross rules before displaying it

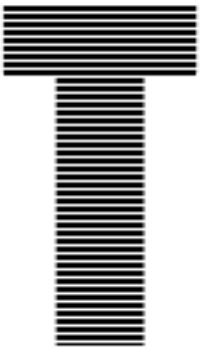
    $white = imagecolorallocate($dissortedImage, 255, 255, 255); //Creating a white color line
    imageline($dissortedImage, 0, 0, 300, 0, $white); //Apply white line from 0x,0y to 300x,0y of the uploaded image
    imageline($dissortedImage, 3, 0, 300, 0, $white); //Apply white line from 3x,0y to 300x,0y of the uploaded image
    imageline($dissortedImage, 6, 0, 300, 0, $white); //Apply white line from 6x,0y to 300x,0y of the uploaded image
    imageline($dissortedImage, 9, 0, 300, 0, $white); //Apply white line from 9x,0y to 300x,0y of the uploaded image
    imageline($dissortedImage, 12, 0, 300, 0, $white); //Apply white line from 12x,0y to 300x,0y of the uploaded image
    imageline($dissortedImage, 15, 0, 300, 0, $white); //Apply white line from 15x,0y to 300x,0y of the uploaded image
    (...)


{
echo "<img src=".$filepath." height=200 width=300 />";
echo "$dissortedImage";
```

Figure 45 – Theoretical Image distortion is being applied to user's uploaded image

Where "(…)" appears, other white lines should be created and code duplicated.

By expanding the code to cover all 200 in height pixels with white lines, figure 45 shows the expected result.



"Upload Image" button needs to contain all encryption functionalities. By clicking it, web application should process the image and output an already encrypted photo. To make this real, code must understand which pixels are white and which are black to apply sum pattern A & B rows only to black pixels. It should put random value from 0 to 255 as RGB's maximum number for any white pixels. This has been well shown in figure 23.

```php
<?php
$checkValue = $dissortedImage();

//Loop for going through all the pixels of the uploaded image
for($x=1;$x<=$width;$x++)
{
    for($y=1;$y<=$height;$y++)
    {
        $pixel=getPixel($checkValue, $x, $y);
        //Now check RGB of selected pixel
        $rgb = imagecolorat($checkValue, x, y);
        //Describe RGB for selected pixel
        $colors = imagecolorsforindex($checkValue, $rgb);
        var_dump($colors);
    }
}
?>
```

Figure 47 – Theoretical Function that goes through all image's pixels and checks their value in RGB

To improve this function, it should apply sum pattern numbers whenever it finds black (0,0,0) RGB colour in pixel. It should add numbers correspondingly to the sum pattern's order. Formula that checks the image in decryption needs to collect those numbers to understand where the black pixel occurs.

```
        //Theoretical black pixel numering
        if($y = $rgb(0,0,0) && $x = $rgb(0,0,0))
        {
            $pixel = imagecreate($y, $x);
            imagecolorallocate($pixel, 1, 3, 5);
            imagepng($image);
            imagedestroy($image);

        }
    }
}
echo '$pixel'
?>
```

This approach gives difficulties, where each black pixel will be replaced in its RGB values by numbers 1, 3 and 5. Code should read the whole A & B rows of Ciepiel's Cross sum pattern to read the whole image correctly and encrypt it correspondingly.

7.3.3) Encountered Issues:

At this level, web application seems to be too hard to develop on time. By having a complete design and partially working functionalities, I had decided to use already existing software to encrypt and decrypt digital image. For testing purposes, the software I have decided to apply my steganography encryption to is Paint. It contains straightforward functionalities allowing to encrypt and decrypt the given picture.

8) Testing & Implementation

Implementation section provides a guide of digital photo encryption using MS Paint software. “Microsoft Paint (Paintbrush, for MacOS users) is a simple raster graphics editor that has been included with all versions of Microsoft Windows. The program opens and saves files in Windows Bitmap (BMP), JPEG, GIF, PNG, and single-page TIFF formats. The program can be in colour mode or two-colour Black-and-White, but there is no greyscale mode. For its simplicity and that it is included with Windows, it rapidly became one of the most used applications in the early versions of Windows, introducing many to painting on a computer for the first time. It is still widely used for simple image manipulation tasks”. (Insider, 2021)

To fully encrypt an image, it should be black and white only. It should also be respected by rules created in Methodology section. For testing purposes, a description of the image will be prepared over the 10x10 pixels in size photo.



Figure 49 – Too big 10x10 Proposed picture to be encrypted using Ciepiel's Cross algorithm inside of the MS Paint software

This testing is going to be documented as step-by-step guidance to Ciepiel’s Cross steganography encryption.

8.1) Encryption

Encryption for testing purposes, states of using the image size of 10x4 pixels for better understanding of this study. This is being described in figure 59.

8.1.1) Step 1: Photo needs to be respected by the Algorithm's rules

Figure 50 represents how the picture should look to be encrypted using all created Ciepiel's Cross rules.

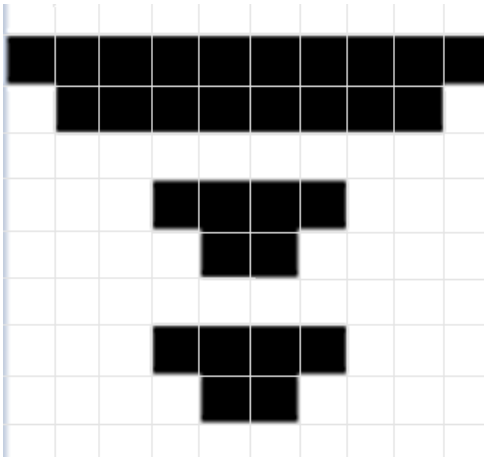


Figure 50 – Proposed, fully prepared to encryption 10x10 image

8.1.2) Step 2: Sum Pattern needs to be applied to each black pixel

To encrypt the photo, sum pattern values should be added to its black pixels only. To any other pixel, a random value should be generated. All black pixels are being coloured black what respects the Ciepiel's Cross algorithm rules.

105	2	12	8	14	222	0	4	5	8
1	3	5	8	14	21	37	55	97	144
0	4	5	9	13	23	34	60	89	200
125	9	22	0	18	422	8	15	2	9
12	4	6	8	14	21	37	59	27	124
6	6	0	9	13	23	34	60	89	200
105	2	12	8	14	222	0	4	5	8
12	4	6	8	14	21	37	59	27	124
6	6	0	9	13	23	34	60	89	200
125	9	22	0	18	422	8	15	2	9

Figure 51 - Ciepiel's Cross sum pattern applied to the image

8.1.3) Instruction of what values are being taken

This, however not a step, is a description and visualization of what values are being taken while encrypting the photo. Figure 52 provides a visual understanding of why black pixels should be considered as Ciepiel's Cross Sum Pattern values.

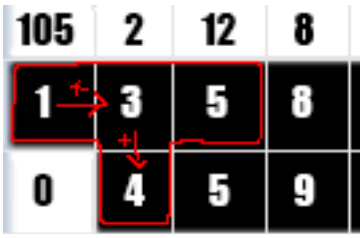


Figure 52 - Black pixels sum pattern explanation

All black pixels have to be represented by sum pattern values in corresponding order. Having a "**x + y + z = y + v**" formula, pixels should respect this formula to be taken as black coloured during the decryption. If the formula is not correct and the left side of the equation is not equal to the right side of the equation, then a white pixel is being put on all fields.

It is very important to note that if black pixel has been applied and another row's calculation proves that the same pixel should be white, it is still a black pixel. Whenever formula is correct, a black pixel is always put, no matter what comes from the other's row calculation. This rule is being described in figure 53.

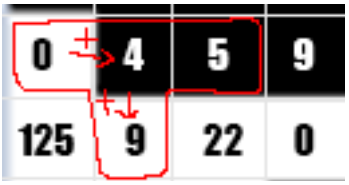


Figure 53 - Black pixels are never overwritten by white pixels

This rule of encryption says that black pixels are never overwritten by white pixels.

8.1.4) Step 3: Image Encryption and new file creation

This step provides all understanding of how new image is being created using Ciepiel's Cross Sum Pattern. For this, user is required to create a new image file, where new pixel values can be drawn and stored.

Paint has an "Edit Colours" feature, that allows to create new colour by providing R, G and B values. This is where all values from Figure 51 will be provided. Each pixel is created by taking "x, y and z" values from the original image correspondingly as "R, G, and B". This is described in Figure 54. New colour is being created using Upside Down Pythagoras Triangle or "Upside Down and mirrored "L" letter".

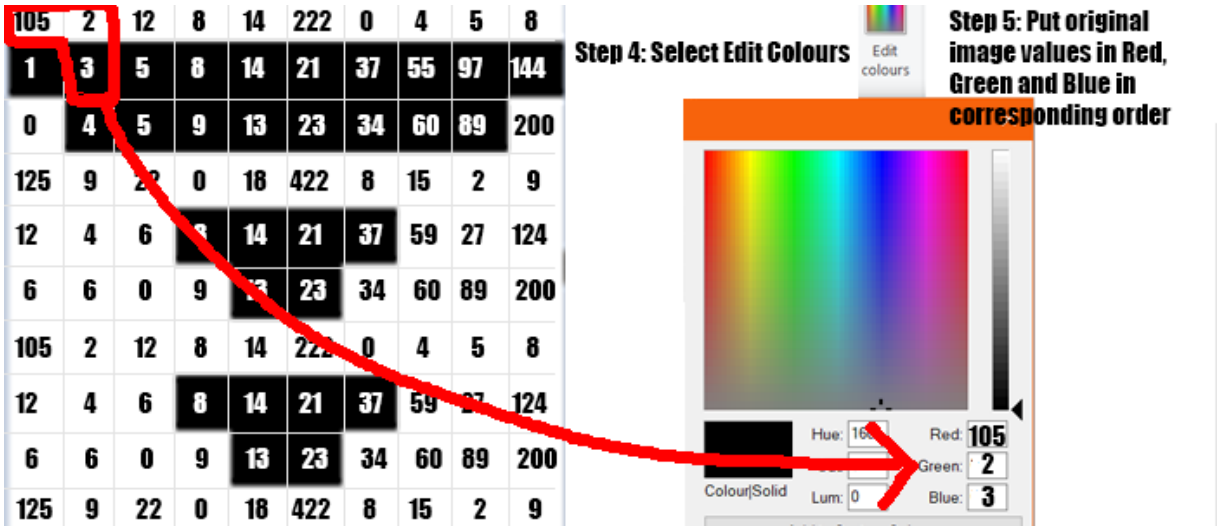


Figure 54 - Putting sum pattern values from original photo to RGB values of new colour creation

At this point, we should have new colour that is an encryption of first three pixels from our original image. Figure 55 provides an example of creating second-pixel encryption for a better understanding of this process.

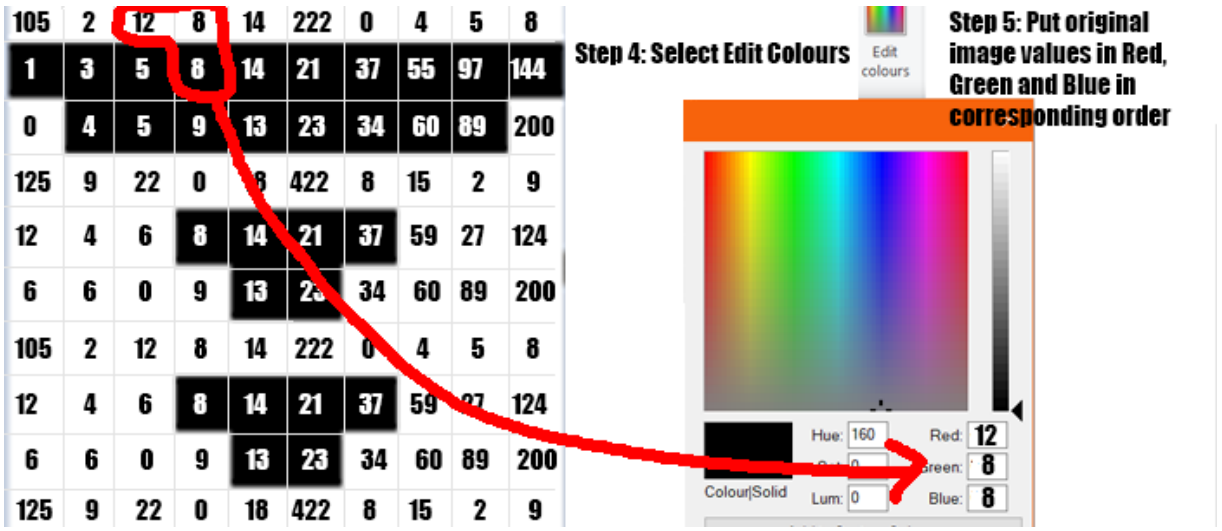


Figure 55 - Second-pixel encryption using RGB values

To compress the file and save space, only three further values of the original photo should be taken (as shown in figure 55). That means for the first pixel values, "105, 2, 3" (figure 54) were used, and as those had been used, the next unused ones should be taken “12, 8, 8” (figure 55).

From those 6 pixels, we should have the following in figure 56 output.

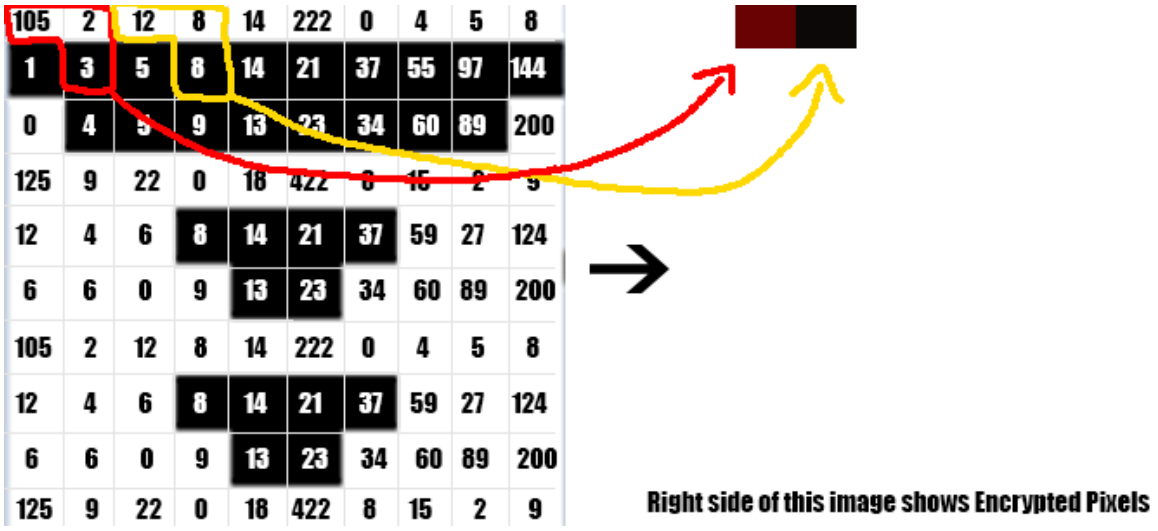


Figure 56 - First two pixels encrypted showcase

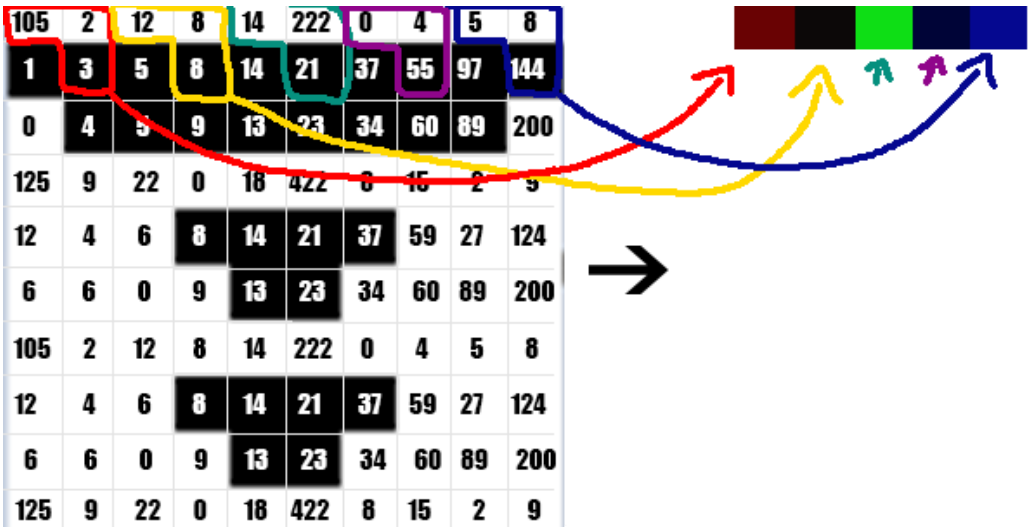


Figure 57 - Fully encrypted first pixel's row of the original image

Another important rule is that each row is being considered separately within Ciepiel's Cross encryption. This rule is being described in Figure 58.

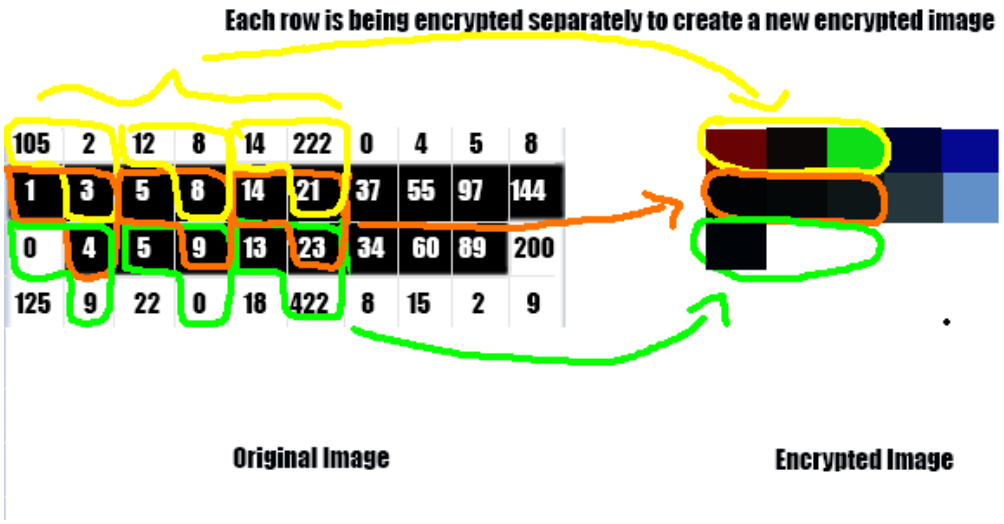


Figure 58 - Ciepiel's Cross encryption considers each row of the original photo as required for an encryption.

Note: Pixel in 6th column and 4th row is wrongly generated. It states as "422" and should be taken as "255" value, as this is the maximum value RGB colour pixel can be created with. This issue was proven to be automatically corrected by MS Paint software, and so the "255" value has been applied to this white pixel. The correction has been applied to the 59th figure result.

Figure 59 provides an image of fully encrypted first four rows of the original photo. For a better understanding of this study, only first four rows are being tested and documented.

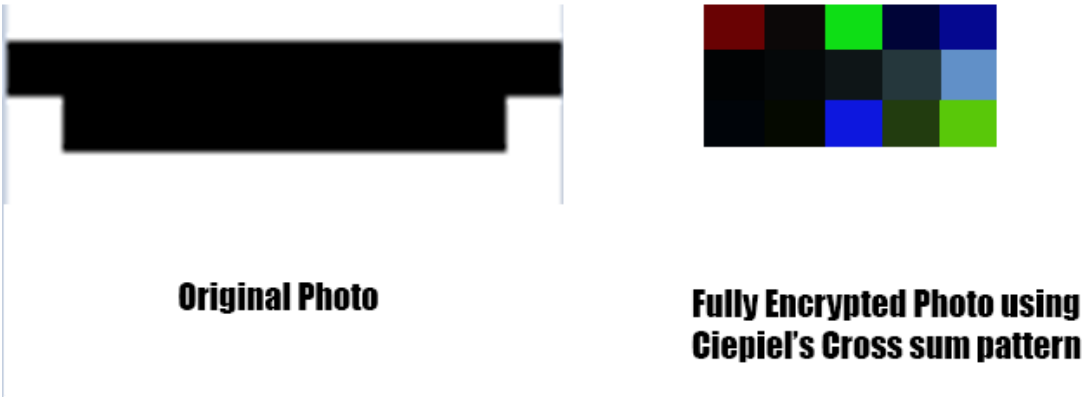


Figure 59 - Fully encrypted (10x4) photo using Ciepiel's Cross sum pattern

As the encryption result, the encrypted image looks completely different from the original input. It can be decrypted by using the pixel colour picker tool also accessible within MS Paint software.

8.2) Decryption

Decryption reverses the encryption process by taking the fully encrypted image's pixels and creating an original photo by placing R, G and B values in corresponding places. This step is being described in figure 60.

8.2.1) Decryption Steps & RGB values extraction

To decrypt an encrypted image, colour picker tool should be used. This tool copies the pixel's RGB values and takes its pixel colour. By checking pixel's colour R, G and B values original image can be restored.

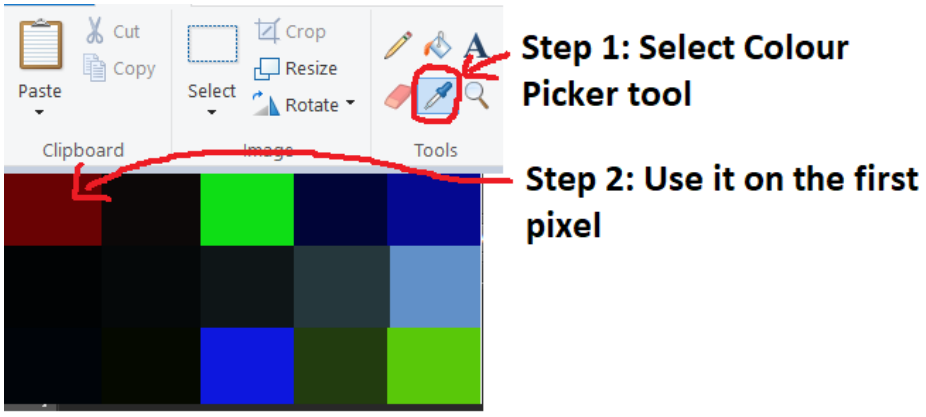


Figure 60 - Image decryption using colour picker tool

It is now possible to check RGB values of this colour by using "Edit Colours" function described in Figure 54.

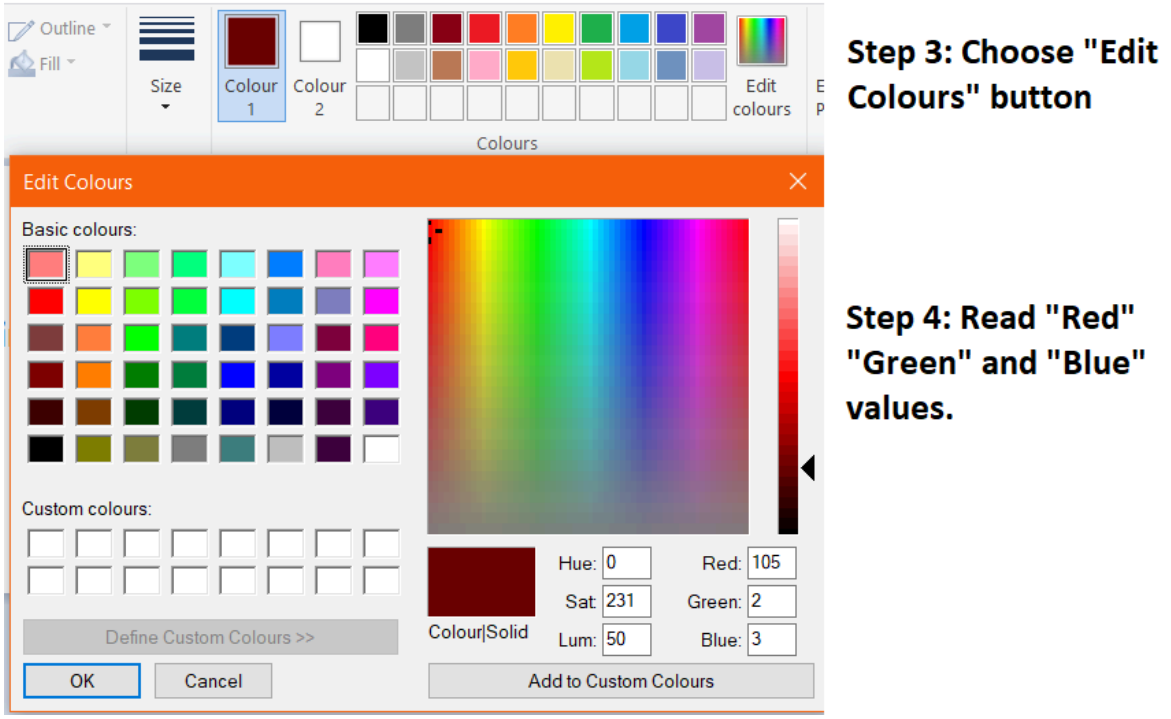


Figure 61 - Decryption instruction. Finding RGB values of encrypted image

Having first pixel values (105,2,3), it is possible to note those values in corresponding order within separate document. Chosen software to write these values is MS Paint. Figure 62 displays how the first pixel's values should be extracted.

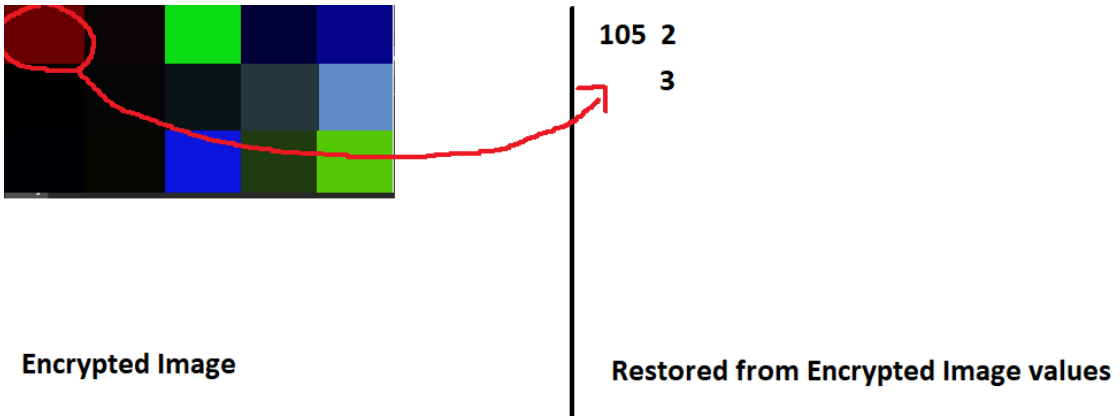
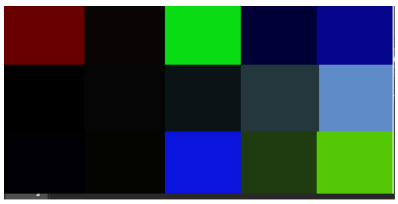


Figure 62 - Restored from encrypted image first pixel's values

At this point, decrypter does not know how big the image is. For this reason, values have to be restored using Upside Down Pythagoras Triangle used for encryption purposes. For each pixel decryption, steps 1, 2, 3 and 4 must be repeated.



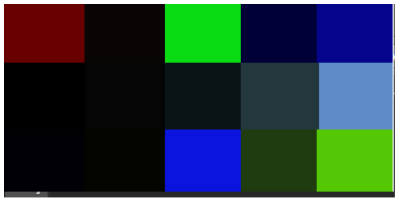
Encrypted Image

105 2 12 8 14 222 0 4 5 8
3 8 21 55 144

Restored from Encrypted Image values

Figure 63 - First encrypted row of image's pixels value restored.

The above's figure presents values restored from the first row of an encrypted image using repeatably steps 1 to 4 from Decryption section. Figure 64 displays all values being restored from each pixel of encrypted image.



Encrypted Image

105 2 12 8 14 222 0 4 5 8
1 3 5 8 14 21 37 55 97 144
0 4 5 9 13 23 34 60 89 200
9 0 222 15 9

Restored from Encrypted Image values

Figure 64 - All values restored from Encrypted Photo

The result of extracted values gives decrypter few information. First knowledge is that the original image was 10 pixels wide and 4 pixels height (10x4). To that point, statement is proved to be correct.

8.3.2) Original Image Restoration

Having extracted values, the next step is to put the Ciepiel's Cross formula (x + y + z = y + v) on restored numbers. "Original Image Restoration" section takes only two steps to be accomplished. Figure 65 provides step 1 of the photo's decryption.

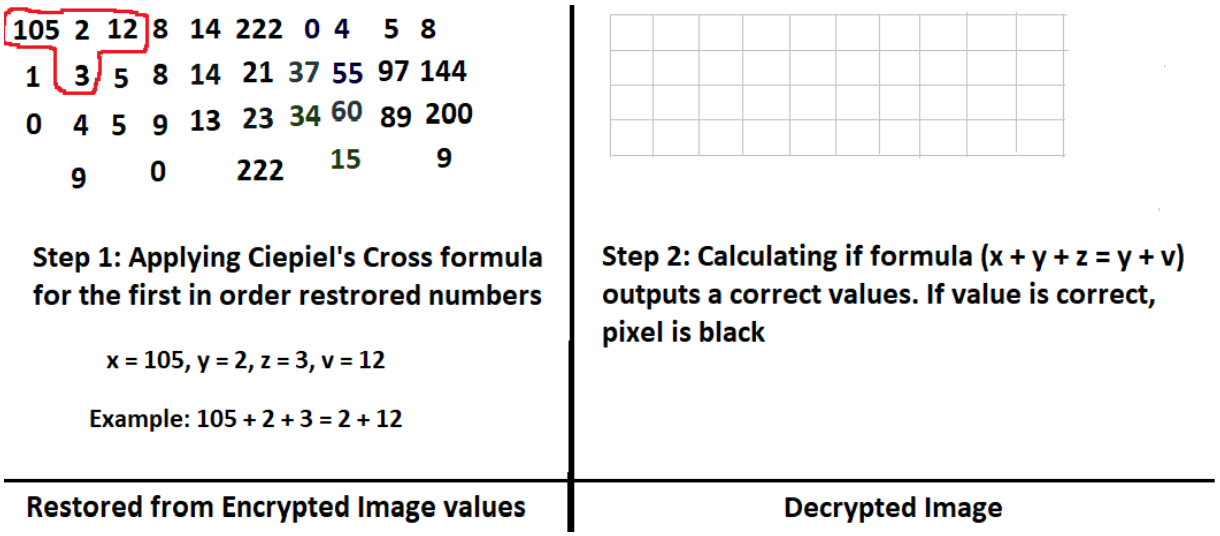


Figure 65 - Applying Ciepiel's Cross formula on extracted from encrypted image values

This gives the output, that first four in corresponding order pixels are white because statement "105 + 2 + 3 = 2 + 12" is not correct. By that, those pixels are staying white within Step 2. For testing purposes, some random values will be calculated, as shown in figure 66.

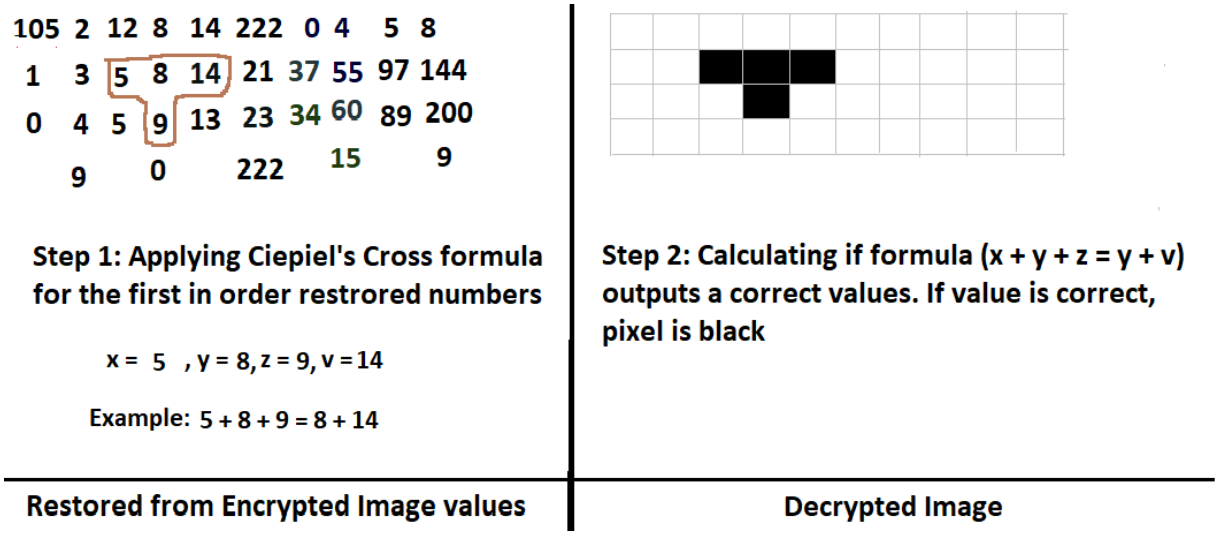


Figure 66 - Decryption of random values from extracted from original image values

Figure 66 takes values that are correct to Ciepiel's Cross Formula as stated in the figure. That results in those corresponding pixels being black.

Example shown in figure 67 reminds, that if the value "9" from selected pixels was previously black, even if applied for another row formula treats it as a white pixel, it still stays black. This is because black pixels are never overwritten by white pixels. This however, works in the other way. If white pixel was extracted and with another formula's usage the same pixel states as black, corresponding white pixel changes to black.

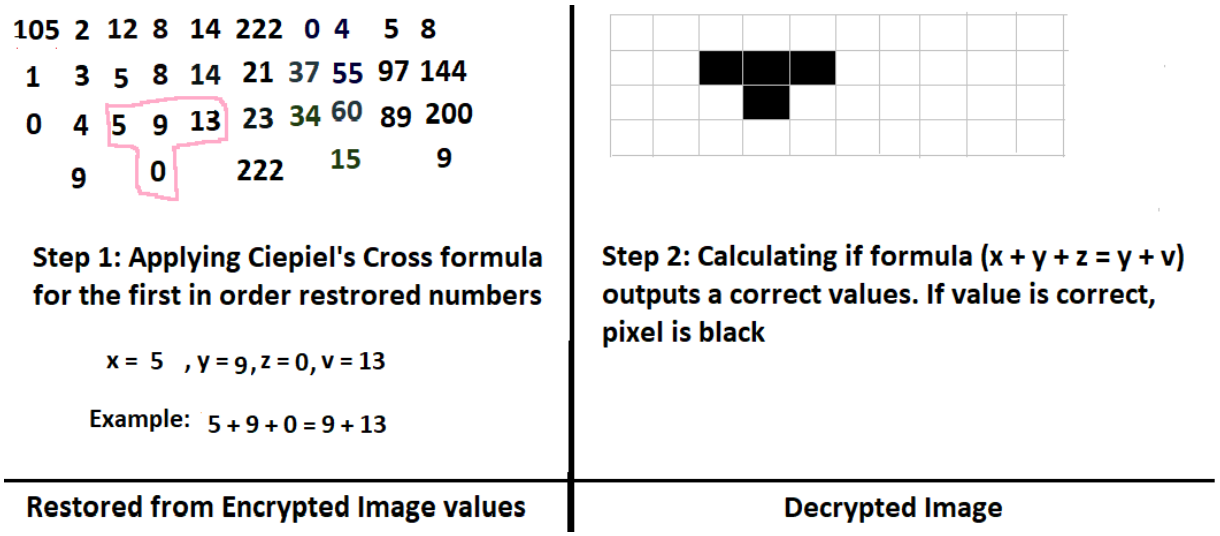


Figure 67 - White pixels do not overwrite black pixels – Decryption Example

Figure 68 displays a completely decrypted image using Ciepiel's Cross formula steps described in figure 65.

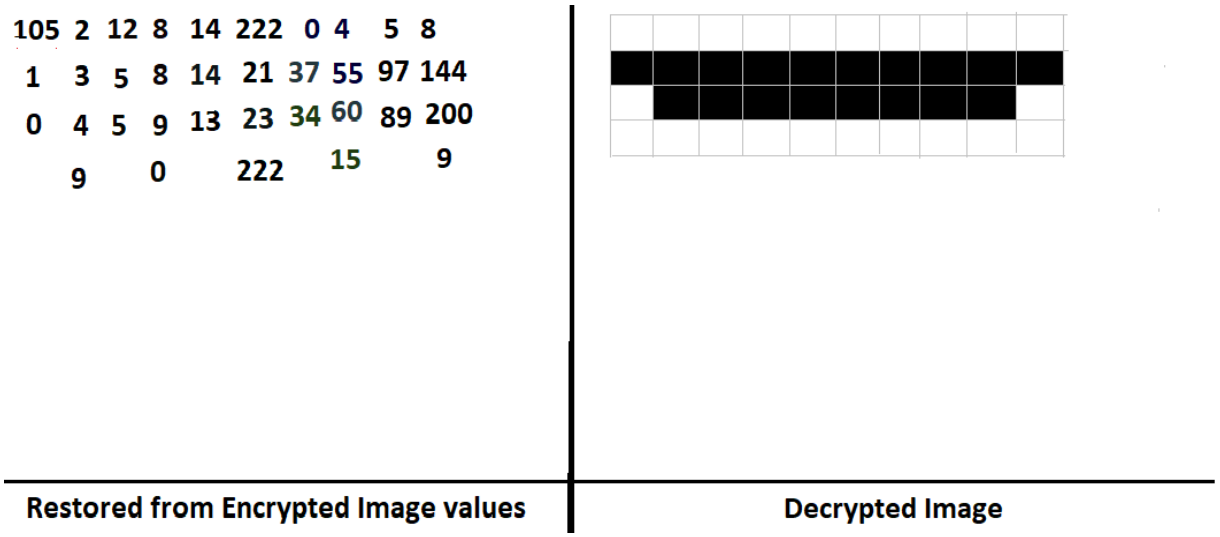


Figure 68 - Fully decrypted image using Ciepiel's Cross formula

9) Results

The main aim of this study was to prove that Ciepiel's Cross Algorithm can be used within steganography encryption and decryption. This has been tested and proved by re-creating the original image into non-understandable, encrypted output. As a result, encrypted photo could be decrypted to show given at the beginning photo.

9.1) Encryption Results

Using Ciepiel's Cross sum pattern, it was possible to encrypt the image. Applying the sum pattern on each black pixel and putting random (from 0 to 255) numbers on white pixels, image could be successfully hidden.

Encryption can be applied by using simple graphic design softwares like MS Paint, Photoshop, or Gimp. Every program that includes functionalities like image uploading, drawing, text editor or pixel colour picker can be used for successful Ciepiel's Cross Steganography Encryption.



Figure 69 - Results of Fully Encrypted Photo using Ciepiel's Cross sum pattern

9.1.1) Results in Encrypted photo File Compression

Image size compression seems to take a massive role in Ciepiel's Cross Steganography Encryption, as from a 10x4 pixels picture, it was possible to create a 5x3 pixels image (as shown in figure 69). File size however, has not been decreased using .PNG file formatting. Depending on the random value of white pixels, it could be possible, but for this testing, future research is required. For file to be smaller after encryption, all random values should generate lower than "10" or higher than "250" numbers.

File size of .PNG formatted; original photo is 163bytes. File size of .PNG formatted; encrypted photo is 196bytes.

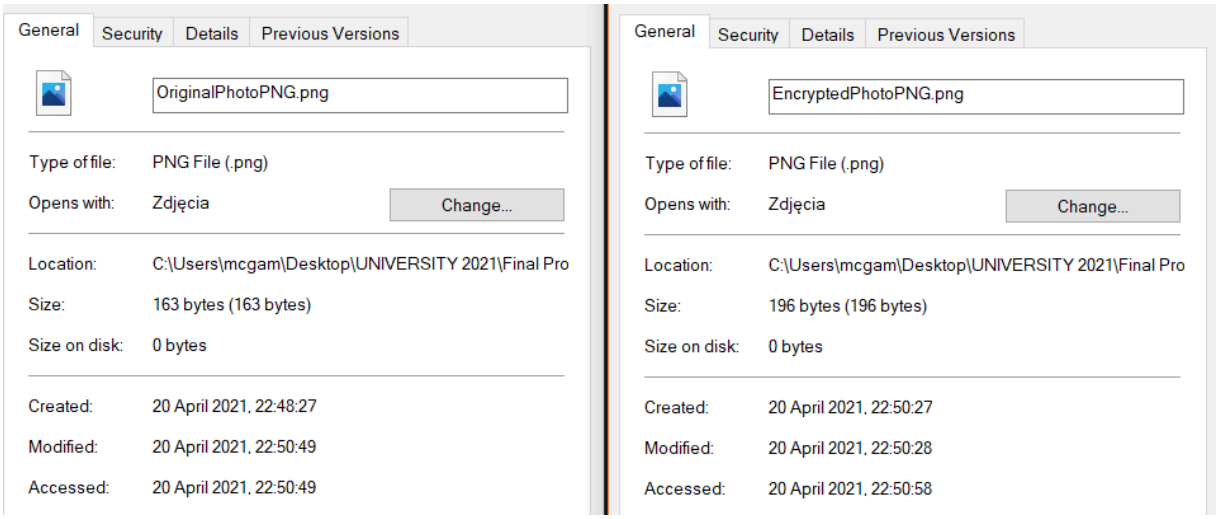


Figure 70 - File size difference between original and encrypted photo in PNG format

However, by saving both files in a .JPG format, encryption seems to decrease the file size by 17 bytes. That issue should be extended in future work.

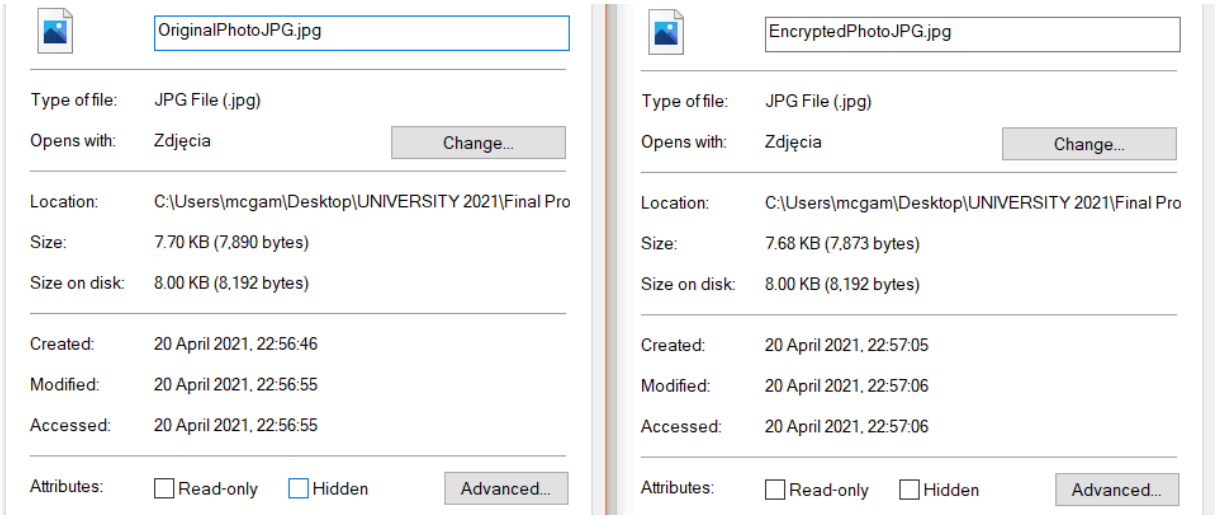


Figure 71 - File size difference between original and encrypted photo in JPG format

9.2) Decryption Results

Encrypted image could be fully decrypted using Ciepiel's Cross mathematical formula. By using the pixel colour picker tool, from each encrypted pixel, RGB values can be extracted. When put in corresponding (Upside Down Pythagoras Triangle) order, those values provide information allowing to define the size of the original image.

Having extracted from encrypted photo RGB values, using Ciepiel's Cross formula (x + y + z = y + v), it was possible to apply, which pixels were black, and which ones were white before the encryption.

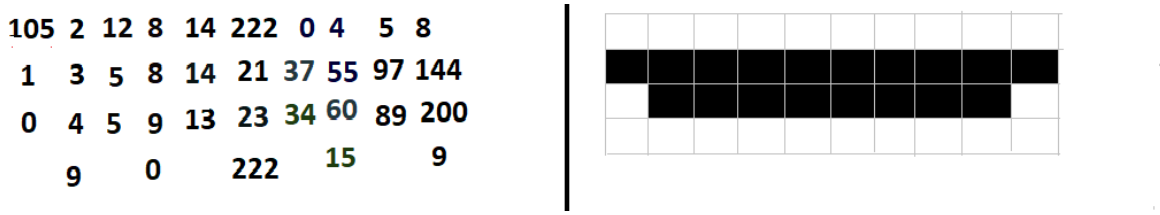


Figure 72 - Results of the Fully Decrypted image

10) Discussion and Conclusions

10.1) Products of this study

Summarizing the study, Ciepiel's Cross algorithm appeared to be new steganography encryption. Discovering the unique Ciepiel's Cross Sum Pattern and applying the formula to the newly created Ciepiel's Cross T-Cross-Summing Algorithm study has provided positive results in filling gaps around existing steganography solutions.

This study was written around several successive discoveries listed below:

- Ciepiel's Cross Sum Pattern
- Ciepiel's Cross T-Cross-Summing (Algorithm)
- Ciepiel's Cross Formula
- Ciepiel's Cross Steganography Encryption
- Ciepiel's Cross Decryption
- Ciepiel's Cross File Compression

Each discovery was a result of previous experimental testing. By guessing possibilities, the first three findings provided the final result in creating the Ciepiel's Cross Steganography Encryption & Decryption.

As new digital steganography encryption, Ciepiel's Cross proves its output to be smaller in size. Both original and encrypted pictures are almost identical in disk size when saving as .JPG format. Encryption to be revolutionary in file compression requires significant changes on the file's size or keeps the file's size unchanged.

Ciepiel's Cross decryption is not an easy task and is only possible by knowing the Ciepiel's Cross Sum Pattern, Ciepiel's Cross T-Cross-Summing Algorithm and Ciepiel's Cross Formula. That proves it does not outstand from other well secured and widely used steganography encryptions.

10.2) Rules created for Ciepiel's Cross Steganography Encryption

This study provides several rules created while developing and testing the Ciepiel's Cross Steganography Encryption. For the file to be successfully encrypted, strict rules must apply. That can be treated as an advantage of hardness in providing correct and secured encrypted message. Even if change of rules might occur in the future, for this moment, those had been written:

1. Sum Pattern is infinite in length but regular in height with having only two rows of infinite length numbers.
2. Sum Pattern values are alternately bigger and smaller in row A comparing to row B.
3. Rule no.: 2 works only starting from the 3rd column of each row.
4. T-Cross-Summing Algorithm applies to every place in Ciepiel's Cross Sum Pattern length.
5. There is only one existing formula for T-Cross-Summing (x + y + z = y + v)
6. Image to be encrypted needs to be black and white only.
7. Image to be encrypted should apply white lines distortion every three rows as wide as the length of the image.
8. On the image to be encrypted, sum pattern values should be put on every black pixel.
9. Image needs to have black pixels everywhere, where the sum pattern appears. All numbers from the T-Cross-Summing should be pixeled black.
10. Every white pixel should be valued with numbers starting from 0 to 255.
11. Decryption takes as black pixels only those pixels, giving correct formula calculations while checking their RGB values.
12. There are two decryption possibilities: 1) Only the "y" value in the formula is treated as a black pixel 2) "every" pixel from the formula is being treated as black.

11) Future work

There are multiple issues to be taken under consideration for future testing and development. Having so many rules, Ciepiel’s Cross steganography encryption creates a difficult image preparation task. Further research should simplify existing rules and expand its encryption functionality.

Software or web application can be created for more accessible image encryption and decryption. This task should state that all maths, distortions, image preparations and calculations are being done in the background and not visibly to the user. Having access to those documents and rules could make this study revealed and easily accessible to crack.

File compression should be taken as a considerable part of the Ciepiel's Cross Steganography Encryption. This field can be extended by testing several different file formats. Main Aim for this task is a result in encrypted image being same or much smaller in disk size than the original photo.

The Ciepiel’s Cross Sum Pattern might be used as a gap sequence in Shell Shorting algorithms. This should be taken into consideration for future evaluation of this study.

The last and the most crucial step would be finding a good source of publishing to this or future, related to Ciepiel’s Cross Algorithm study. Ciepiel's Cross can be publicly accessible if software or web application strictly secures all calculations. There is also a way of publishing this work as public, to match any project in the world.

12) Bibliography

Alan Siper, Roger Farley & Craig Lombardo, 2005. *The Rise of Steganography ,* s.l.: CSIS, Pace University.

Brian Beckett, McGraw Hill, 1997. *Introduction to cryptology & PC security.* s.l.:s.n.

Browinski, G., 2016. Encryption Methods of Yesterday and their Replacements. *BIO-key.*

Choudhary, K., 2017. *Why we need steganography,* s.l.: s.n.

EC-Council, N/A. *What is steganography and what are its popular techniques?.* [Online]

Available at: https://blog.eccouncil.org/what-is-steganography-and-what-are-its-popular-techniques/

FigLeaft, 2019. *How the definition of privacy has drastically changed over the last 20 years.* [Online]

Available at: https://figleaf.com/blog/perspectives/definition-of-privacy-has-drastically-changed-over-the-last-20-years/

Furuseth, A. G., 2005. *Digital Forensics: Methods and tools for re-trieval and analysis of security credentials and hidden data,* s.l.: NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET.

GijiGadu, 2020. *12345 puzzle, 12345 puzzle answer,what goes in the empty square.* [Online]

Available at: https://www.youtube.com/watch?v=rFUVqjktJUY

Gross, S., 2010. *Origins of Human Communication - by Michael Tomasello,* s.l.: s.n.

Insider, W., 2021. *Announcing Windows 10 Insider Preview Build 21354,* s.l.: s.n.

Kahn, D., 2005. *The history of steganography.* s.l.:s.n.

Kessler, G. C., 2021. *An Overview of Cryptography.* [Online]

Available at: https://www.garykessler.net/library/crypto.html

Liu Yunxia, Liu Shuyang, Wang Yong, Zhao Hongguo, LiuSi, 2019. *Neurocomputing.* s.l.:s.n.

Maerian, L., 2014. *Data storage -- then and now,* s.l.: Compute World.

McMullen, C., 2015. *300+ Mathematical Pattern Puzzles: Number Pattern Recognition & Reasoning.* s.l.:s.n.

mdo, 2013. *Bootstrap 3 released.* [Online]

Available at: https://blog.getbootstrap.com/2013/08/19/bootstrap-3-released/

[Accessed 21 April 2021].

Pranali R. Ekatpure, Rutuja N Benkar, 2015. *A Comparative Study of Steganography & Cryptography,* s.l.: ijsr.

Rana, M., 2017. *Why PHP Web Development is Widely Used by Most of The Developers?,* s.l.: s.n.

Rana, M., 2017. *Why PHP Web Development is Widely Used by Most of The Developers?,* s.l.: s.n.

Staff, P., 2019. *The Definitive 2019 Guide to Cryptographic Key Sizes and Algorithm Recommendations,* s.l.: s.n.

Tallarida, R. J., 1992. *Pocket Book of Integrals and Mathematical Formulas.* s.l.:s.n.

Team, G. L., 2020. *Image Steganography Explained | What is Image Steganography?*. [Online]

Available at: https://www.mygreatlearning.com/blog/image-steganography-explained/

UKEssays, 2018. *Examining The Importance Of Steganography Information Technology Essay.* [Online]

Available at: ttps://www.ukessays.com/essays/information-technology/examining-the-importance-of-steganography-information-technology-essay.php?vref=1