VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

DIGITAL WATERMARKING AND STEGANOGRAPHY

(BCI3005)

TOPIC: -

Invisible watermarking and copyright analysis

(J Component Report)

Submitted by

TEAM MEMBERS:

ABHIJEET MARIKAL- 17BCI0009

K V NARENDRA KUMAR-17BCI0050

K Rajesh – 17BCI0147

NARENDRA – 17BCI0106

Faculty: Prof. THENMOZHI T

School of Computing Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

DECLARATION

I hereby declare that the J Component report entitled " " submitted by me to Vellore Institute of Technology, Vellore14 in partial fulfilment of the requirement for the award of the degree of B.Tech in Computer science and engineering is a record of bonafide undertaken by me under the supervision of ***Prof. THENMOZHI T*** I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

ABSTRACT

The rising potential of modern communications needs the exceptional means of security in the computer network. The network security is becoming more important and challenges of data exchanged on the Internet increases. Therefore, the confidentiality and data integrity are requiring protecting against unauthorized access and use. This has resulted in an explosive growth of the field of information hiding. Information hiding is an addition of application-oriented information to a multimedia signal, without causing any perceptible distortion. The energy of the embedded signal should be low enough when projected onto the human perception domain, but it should be strong enough for robust machine detection. Information hiding techniques have recently emerged in many different application areas. Digital audio, videos, and images are increasingly furnished with distinguishing but imperceptible marks, which may contain a hidden copyright notice or a serial number, which can even help to prevent unauthorized copying directly.

Steganography and watermarking are two methods used in information hiding, which focuses on secret communication. The use of watermarking and steganography, as a viable form of communication, has been largely propelled by the growth of the Internet. The Internet offers an opportunity to exchange large amounts of digital information over great distances. The prevalence of media, such as audio, videos, and images, on the Internet, provides an ideal channel for watermarking and steganographic communication.

Steganography or (Covert Communication) is an ancient art that has been reborn in recent years; it plays a very important role in protecting information in the current age of virtually connected system, organisation's secret information needs to be shared to its branches worldwide and copyright issues need to protect as well. Therefore, the increasing demand for protecting information will continue Steganography basically aims at hiding communication between two parties from the attackers

Steganography system involves inserting an identifier be it a message, the secret message or a marker into a medium that will not be affected. The most important object inserted into a cover medium is the secret message or serial numbers or secret code. When this information is embedded, the steganographic technique ensures that it will not be detected and assess by any unauthorized person. The cover message that hosts the embedded is called a Stego Object. The cover message is required to be disposed of after the receivers got it in order to prevent an accidental reuse.

LITERATURE REVIEW

Several approaches have been proposed for achieving reversibility. Some of it included a high secure reversible visible watermark algorithm. It can fully remove the watermark from the visible watermarked image such that the original image can be restored. To restore the original image, the difference of subtracting the approximated image from the original image and other side information are listlessly compressed to be embedded in the visible watermarked image by a reversible data embedding algorithm. A key-based scheme is used for the compromise between transparency and robustness. The key is a random variable with discrete normal distribution. Only users with correct key can restore the original image. Digital watermarking is one of the ways to prove the ownership and the authenticity of the media. There are mainly two types of watermarking algorithms: visible watermarking and invisible watermarking. For invisible watermarking, the watermark should be perceptually transparent and robustness. For visible watermarking, the watermark should be perceptually visible and robustness. Watermarking is performed by embedding a digital watermark signal into a digital host signal resulting in watermarked signal. Distortion is introduced into the host image during the embedding process and results in Peak Signal-toNoise Ratio (PSNR) loss. Reversible watermarking, which can recover the original host signal perfectly after the watermark extraction. There are two lossless visible watermarking algorithms. They are Pixel Value Mapping Algorithm (PVMA) and Pixel Position Shift Algorithm (PPSA). PVMA uses the bijective intensity mapping function to watermark a visible logo whereas PPSA uses circular pixel shift to improve the visibility of the watermark in the high variance region. Reversible watermarking using perceptual model is used. During data hiding, distortions are introduced in an original image because of quantization errors, bit replacement, or truncation at the grey-scale limit. These distortions are irreversible and visible, which is unacceptable in some applications such as medical imaging. A new method for protecting the copyright and verifying the integrity of BMP images by using removable visible watermarks and irremovable visible watermarks. Depending on the relationship between the embedded message and the cover image, data embedding applications could be divided into two groups. The first group is formed by steganographic applications. The second group of applications is frequently addressed as digital watermarking. In watermarking application, the message supplies additional information about the image, such as image caption, data about the image origin, author signature, image authentication code. There are some applications for which any distortion introduced to the image is not acceptable. A secure algorithm is used for watermarking. Conventional cryptographic systems permit only valid key holders' access to encrypted data, but once such data is decrypted there is no way to track its reproduction or retransmission. Therefore, conventional cryptography provides little protection

against data privacy, in which a publisher is confronted with unauthorized reproduction of information. It is a visible, or preferably invisible, identification code that is permanently embedded in the data

INTRODUCTION

STEGANOGRAPHY

The word 'steganos' means "covered or protected" and 'graphie' means "writing". Steganography is thus, not only the art of information hiding, but also the art and science of hiding the fact that communication is even taking place. Privacy is not the only motivation for steganography. By embedding one piece of data inside of another, the two become a single entity, thus eliminating the need to preserve a link between the two different pieces of data, or risk the chance of their separation. One application than exhibits the advantage of this facet of steganography is the embedding of patient information within the medical imagery. By doing so a permanent association between these two information objects is created the goal of steganography is to avoid drawing suspicion to the transmission of the secret message. The concept of "What You See Is What You get (WYSIWYG)" which we encounter sometimes while printing images or other materials, does not always hold true. Images can be more than what we see with our Human Visual System (HVS); hence they can convey more than merely 1000 words. For decades people strove to create methods for secret communication. A Steganographic system has two main aspects: Steganographic capacity and imperceptibility. However, these two characteristics are at odds with each other. Furthermore, it is quite difficult to increase the Steganographic capacity and simultaneously maintain the imperceptibility of a Steganographic system

WATERMARKING

Watermarks are identification marks produced during the paper making process. The first watermarks appeared in Italy during the 13th century, but their use rapidly spread across Europe. They were used as a means to identify the papermaker or the trade guild that manufactured the paper. Watermarks continue to be used today as manufacturer's marks and to prevent forgery. A watermark is a "secret message" that is embedded into a "cover source". Usually, only the knowledge of a secret key allows us to extract the watermark. Thus, the effectiveness of any watermarking technique depends on how robust the watermark is i.e. Even if someone knows that a watermark is exist (i.e. visible watermarking) in a given object, it should be impossible to remove the watermark from the watermarked object without causing a distortion or destroying the original (watermarked) object.

A. Requirements of a Steganographic System:

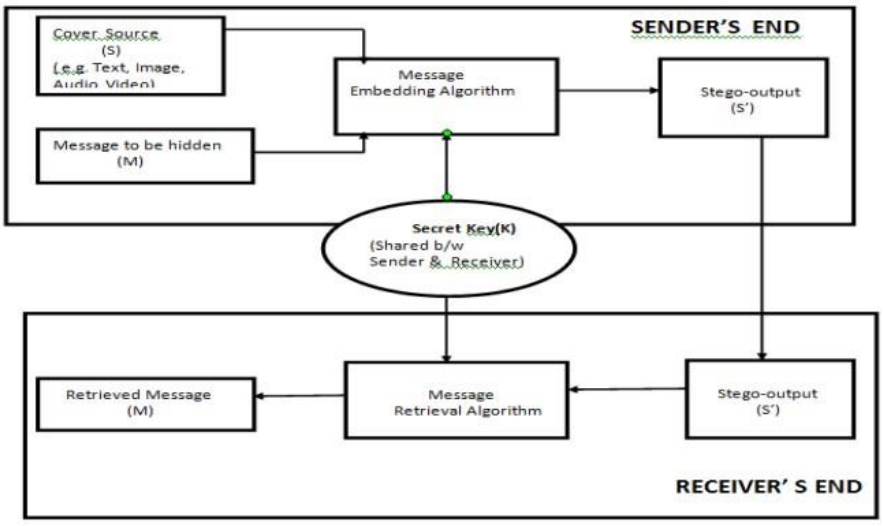
1. The most important requirement for a steganography system is that the presence of the hidden message be undetectable. This means that images with and without secret messages should appear identical to all, irrespective of the possible statistical tests that can be carried out.
2. Another important requirement is the capacity of the communication channel. The challenge is to embed as much information as possible.
3. The last important requirement is that it must be possible to detect the hidden message without the original image **B. Steganography Process:**

a. Message Insertion (Sender's end):

1. Cover source (e.g. image, audio, video) and secret message which is to be hidden are given as input to the Message Insertion Algorithm.
2. Use the secret key & Steganographic Algorithm to hide the message in the cover source
3. Stego Output is produced as result of step 2

b. Message Retrieval (Receiver's end):

1. Stego Output send by sender is given as input to the Message Retrieval algorithm.
2. Use the Message retrieval algorithm and secret key to retrieve the message from the Stego output
3. Secret message is retrieved as a result of step 2



Applications of Steganography:

- **Secret Communication Using Steganography:** two parties can communicate secretly without anyone knowing about the communication. Cryptography, only encode the message but its presence is not hidden and thus draws unwanted attention, Steganography, thus, on the other hand, hides the existence of message in some cover media.

- **Copyright Protection**: This is basically related to watermarking i.e. a secret message is embedded in the image which serves as the watermark and thus identify it as a intellectual property which belongs to a particular owner.
- **Digital Watermarking:** This is one of the most important applications of Steganography. It basically embeds a digital watermark inside an image. Digital watermarks may be used to verify the authenticity or integrity of the carrier signal or to show the identity of its owners. It is prominently used for tracing copyright infringements and for banknote authentication
- **Use by terrorists**: Steganography at a large scale can also be used by terrorists, who hide their secret messages in innocent cover sources to spread terrorism across the country. Rumours were spread about terrorists using steganography when the two articles titled "Terrorist instructions hidden online" and "Terror groups hide behind Web encryption" were published in newspaper. Other media worldwide cited these rumours many times, especially after the terrorist attack of 9/11, without ever showing proof.
- **Feature Tagging:** Captions, annotations, time stamps and other descriptive elements can be embedded inside an image, such as the name of the individuals in a photo or location in a map. Copying the stego image also copies all of the embedded features and only parties who possess the decoding stego key will be able to extract and view the features

Watermarking Algorithm for PDF Document

With the popularity of the network, delivery of electronic document is becoming increasingly more efficiency. A number of documents are share and spread on the network in the form of electronic documents. Currently, PDF (Portable Document Format) is the main file format to transmit on the network. It is widespread used in electronic file transfer, exchange and distribution. And almost all of the popular electronic publications are in .pdf format. The widespread use of PDF comes across some security issues which could be solved by digital watermarking. The core idea is to embed secret information about copyright into the digital products. When a digital watermark, or a piece of multimedia confidential information is embedded into a multimedia, only a special watermark detector can be used to extract the watermark in the case of the human perceptual system imperceptible. In recent years, more and more researchers began to study the PDF document watermarking algorithm, but still in infancy. In this paper, an object-oriented PDF document page watermarking algorithm is proposed. It masquerades watermark signal as legitimate PDF document page objects to achieve watermark embedding.

PDF STRUCTURE

The structure of PDF document contains two aspects; one is PDF file structures, called a physical structure. Another is the organized structure of PDF document

content, known as the logical structure. The physical structure consists of four parts: the file header, file body, cross-reference table, and the tail of a file. File header indicates the version number of the PDF specification complies. File body is the most essential part of the PDF document which is organized by a series of page objects. A record of the cross-reference table marks a page object entry that contains the page number of the object, offset and status, enabling random access to a page object. The tail of a file is an entrance to the entire PDF document declaring the address of the cross-reference table, and saving the encrypted information of the PDF document. The logical structure of a PDF document can be viewed as a tree structure for all objects, the root of the tree is the Catalog. There are four roots of leaf nodes: pages tree, outlines hierarchy, articles threads, names destinations. In addition, PDF document has eight basic objects: Boolean, Numeric, Stream, Dictionaries, String, Names, Arrays, and Null.

WATERMARKING ALGORITHM

The key idea of digital watermarking algorithm is to use the redundant information in the host file to add information for embedding watermark signal, such as image watermarking and video watermarking. The document watermarking algorithm developed slowly, because there is no redundant information in a document due to its tight structure, particularly in PDF document. At present, PDF document watermarking algorithm is divided into three categories: format-based, structure-based, and the natural language watermarking algorithm. This paper is based on the document structure algorithm, which is defined on the basis of a profound analysis of the document structure, without affecting the document content and format. The biggest advantage of such an algorithm is resistant to all kinds of format attacks, robust to most geometrical attacks, and keep the document display, but it is more difficult.

Watermark Embedding

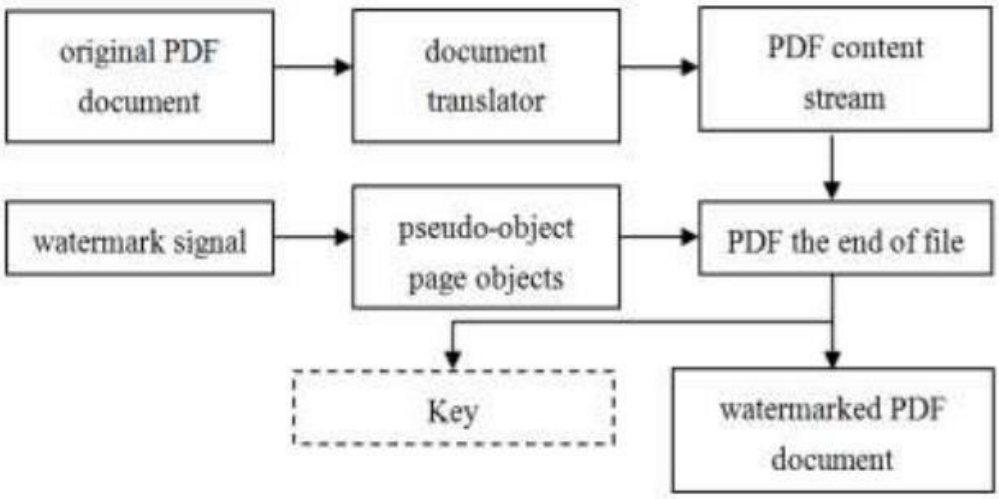
Watermark signal embedding algorithm is shown in Figure 2: The function of the document translator is to read the document in the form of binary data stream, and read sequentially from the tail of a file to header by moving the file pointer. The specific steps of the algorithm are as below：

Step 1: Document translator reads PDF a document test. pdf content stream in binary form and saves it to the document test1. pdf and test1.txt content stream;

Step 2: To determine whether the document has been read, that is, the file pointer return "EOL". If it is not finish, please return to step 1, if is finished, proceed to the next step;

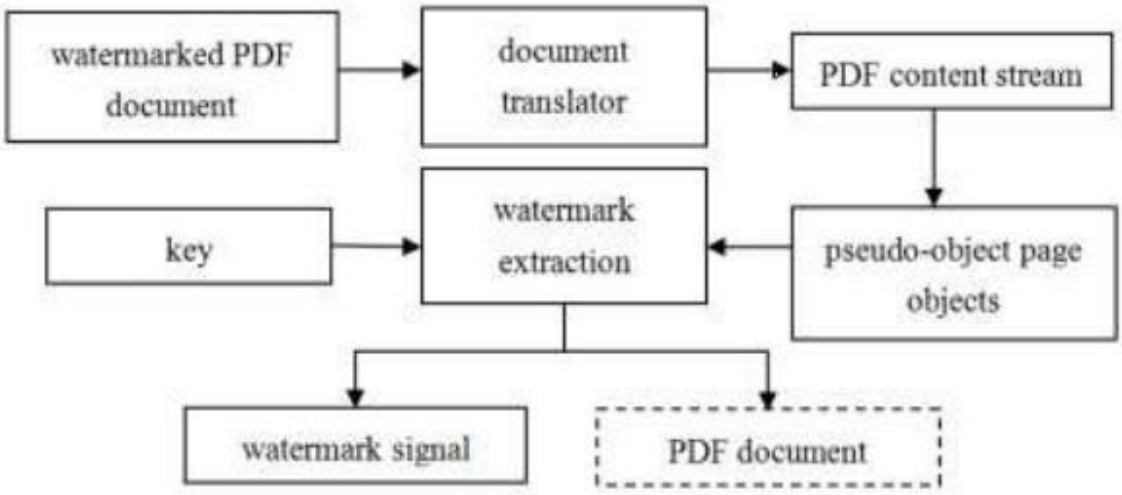
Step 3: “w” will be read into test1.pdf and test1.txt;

Step 4: Close the document translator, and store it as test.pdf, test1.pdf and test1. txt respectively, where the test1.pdf is the watermarked PDF documents.



Watermark Extraction

In the section of “Watermark Embedding”, the editable watermark can be successfully embedded into the original PDF document, and can meet the invisibility. The extraction process of the watermark signal is an inverse embedded process. The block diagram of our watermark extraction algorithm is shown in



When extracting the watermark, firstly, read watermarked PDF documents to get the document content streams via PDF document translator; then look for objects with camouflage page watermark signal to extract the watermark signal in the disguise page object. The specific steps are as follows:

Step 1: Document translator reads watermarked PDF document test1.pdf flow in the form of binary data stream and saved it to the document test2.pdf content stream;
Step 2: Flag detect whether the document contains pseudo-object page objects "0 0 obj", if there is, go on next step, otherwise go to step 5;

Step 3: Read the watermark signal of pseudo-object page, and writes it into test2.txt content stream;

Step 4: Determine whether the end of the watermark bit is read, if not, turn to the previous step, otherwise transfer it into the next step;

Step 5: To judge whether the document has been read, that is, the file pointer return "EOL"; if it is not finished, return to step 1, if it is finished, proceed to the next step;

Step6: Close the document translator, and save it to the three documents by the name test1.pdf, test2.pdf and test2.txt. Test2.txt is the stored watermark signal, test1.pdf and test2.pdf are the watermarked PDF documents.

PERFORMANCE ANALYSIS

In VC ++ 6.0 environment, this paper uses C language to read the document, embed watermark signals and make some extracting algorithm simulation experiments. In addition, utilizing the official PDF editing software Adobe Acrobat X Pro to attack document, and the attack types are: text delete, add text, text substitution, delete pages, crop pages and page rotation. The subjects were 50 Chinese-English paper related to the three key words of “video watermarking”, “document watermarking”, “pulse diagnosis”. The experimental results confirmed the reliability of the algorithm and the results of experimental statistics are shown in Table I.

TABLE I. THE ATTACK RESULTS OF 50 LAB DOCUMENT WATERMARKED.

Attack number	1	2	3	4	5	6
Number of articles detected watermark	50	50	50	42	47	48

Where attack No. 1-6 represent six kinds of attacks. They are delete text, add text, replace text, delete pages, crop pages and page rotation, respectively. And we will

evaluate the proposed watermarking algorithm from four aspects. Robustness: By analysing the data in Table 1, the proposed watermarking algorithm in this paper can completely resist the attack of text type, but it is kind of inadequate for page attack, because the watermark signal embeds into the document in the form of a pseudo-page objects. During the attack, word attack was confined within the pages of the object, and on the contrary, page attack is a kind of bulking page operations. It operates on page objects, so the pseudo-page objects will have a greater destructive power. However, the recognition rate of the watermark signal is still more than 90%; it can be seen that the algorithm has strong robustness. Security: In this paper, the algorithm embeds the watermark signal in the structure of the PDF document, so the embedding algorithm has strong anti-attack ability and the security is guaranteed.

Invisibility (Transparency): By observing Figure 2 and 4, we can conclude that the document will not display any significant changes after addition of the watermark. The display of the document will be not affected by the embedded watermark. The algorithm has invisibility. Watermark capacity: We can see the watermark signal has editable characteristic. Thus, theoretically the capacity of the watermark signal is unlimited; a very large capacity can be embedded.

Code:

Pdf_watermark_embed.py

```
import PyPDF2

from reportlab.lib.units import cm from

reportlab.pdfgen import canvas def

create_watermark(file_name, content):

    c = canvas.Canvas(file_name, pagesize=(30 * cm, 30 * cm))

c.translate(50, 400)


    c.setFont("Helvetica", 5)


    c.setFillColorRGB(255, 0, 0)


    c.setFillAlpha(0)


    c.drawString(2 * cm, 3 * cm, content)
```

```
    c.save()
return file_name
def
embed_watermark
(pdf_file_in,
pdf_file_mark,
pdf_file_out):


    pdf_input = PyPDF2.PdfFileReader(open(pdf_file_in, 'rb'))


    pdf_watermark = PyPDF2.PdfFileReader(open(pdf_file_mark, 'rb'))


    pdf_output = PyPDF2.PdfFileWriter()



    pageNum = pdf_input.getNumPages()
for i in range(pageNum):
        page = pdf_input.getPage(i)


        if i == pageNum - 1:
            page.mergePage(pdf_watermark.getPage(0))
page.compressContentStreams()
pdf_output.addPage(page)
pdf_output.write(open(pdf_file_out, 'wb')) def
enc():
    from cryptography.fernet import Fernet
key = Fernet.generate_key()     file =
open('key.key', 'wb')
    file.write(key) # The key is type bytes still
```

```
    file.close() # Use one of the methods to get a key (it must be the same when decrypting)
input_file = './test.txt'     output_file = 'test.encrypted'


    with open(input_file, 'rb') as f:
        data = f.read()

    fernet = Fernet(key)
encrypted = fernet.encrypt(data)
with open(output_file, 'wb') as f:
f.write(encrypted)     with
open(output_file, 'rb') as f:
        data = f.read()
return data if __name__ ==
'__main__':
    tmp_file = "rajesh."
watermark = enc()
    file_mark = create_watermark(tmp_file, watermark)
pdf_file = "./internet.pdf"     file_out = "./out.pdf"
    embed_watermark(pdf_file, file_mark, file_out)
```

pdf_watermark_extract.py:

```
import PyPDF2 def extract_watermark(file_watermarked):
pdf_input = PyPDF2.PdfFileReader(open(file_watermarked, 'rb'))
pageNum = pdf_input.getNumPages()
    extractedText = pdf_input.getPage(pageNum - 1).extractText()
print(extractedText.split()[-1]) if __name__ == '__main__':
origin_file = "./GCN.pdf"     file_watermarked = "./out.pdf"
extract_watermark(file_watermarked)
```

dec.py:

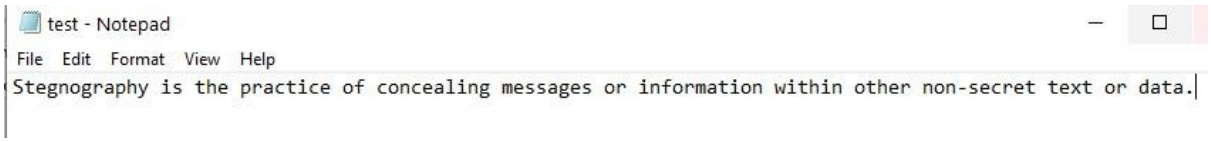
```
from cryptography.fernet import Fernet file
= open('key.key', 'rb')
key = file.read() # The key will be type bytes
```

file.close() # Use one of the methods to get a key (it must be the same as used in encrypting) input_file = 'test.encrypted' output_file = 'Decprted.txt' with open(input_file, 'rb') as f:

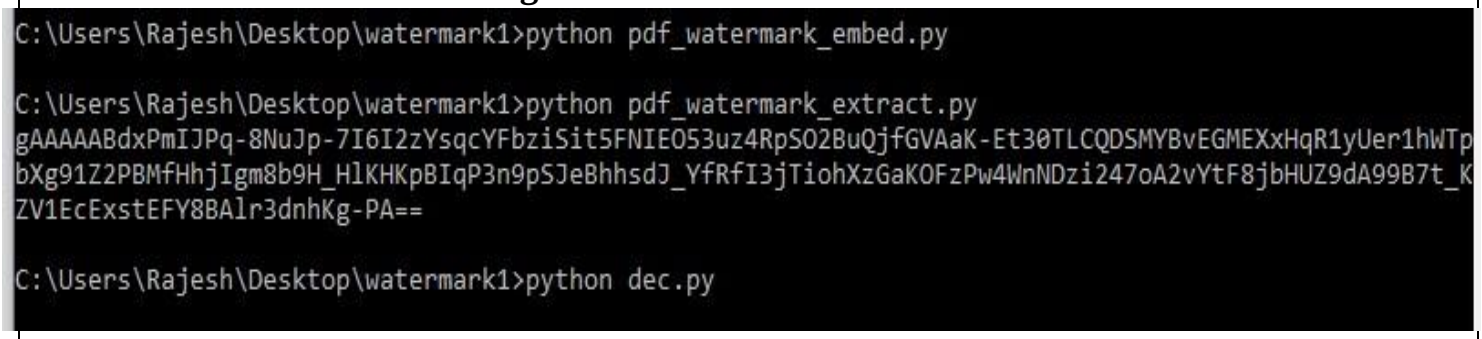
data = f.read() fernet = Fernet(key) encrypted = fernet.decrypt(data) with open(output_file, 'wb') as f:

f.write(encrypted)

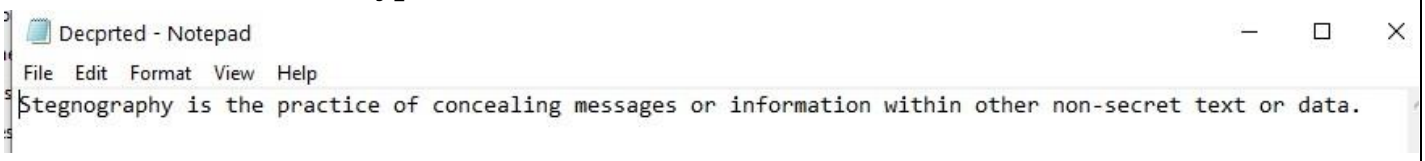
Test.txt file:-



Results of watermarking:-



Retrived and decrypted watermarked text:-



COPYRIGHT ANALYSIS:-

Most efficient watermarking techniques

(Used to copywrite a document)

1) Fingerprint as Watermark:

We discuss a copyright identification scheme for colour images that takes advantage of the complementary nature of watermarking and fingerprinting. It utilizes an authentication logo and the extracted features of the host image to generate a fingerprint, which is then stored in a database and also embedded in the host image to produce a watermarked image. When a dispute over the copyright of a suspect image occurs, the image is first processed by watermarking.

If the watermark can be retrieved from the suspect image, the copyright can then be confirmed; otherwise, the watermark then serves as the fingerprint and is processed by fingerprinting. If a match in the fingerprint database is found, then the suspect image will be considered a duplicated one. Because the proposed scheme utilizes both watermarking and fingerprinting, it is more robust than those that only adopt watermarking, and it can also obtain the preliminary result more quickly than those that only utilize fingerprinting.

The Proposed Copyright Identification Scheme

The proposed scheme contains two phases: the fingerprint and watermarked image generation phase and the authentication logo detection phase. The former phase extracts features from the host image, which, along with a logo image, is used to generate the fingerprint. The fingerprint also serves as the watermark, and the phase embeds it in the host image to produce a watermarked image. On the other hand, the latter phase extracts features and retrieves the watermark from the suspect image. The extracted features and the retrieved watermark are utilized to restore the logo image, which is used to identify the copyright. If it fails, the retrieved watermark then serves as the fingerprint and is compared with those in the database to determine if the suspect image is a duplicated one. The fingerprint and watermarked image generation phase (shown in Figure 3) works as follows. In the beginning, feature extraction extracts the features of the host image and then logo scrambling disarranges the authentication logo to a scrambled logo image. After that, fingerprint generation takes as input the extracted features and the scrambled logo to generate the fingerprint. Finally, the fingerprint serves as a watermark and is embedded in the host image, which becomes a watermarked image. The fingerprint is also stored in a database for later use in the next phase. The authentication logo detection phase (shown in Figure 3) checks the watermark first and, if necessary, the fingerprint next. In the beginning, watermark retrieval regains the watermark from the suspect image. Next, the features of the suspect image are extracted by feature extraction. After that, logo restoration takes as input the retrieved watermark (the expected fingerprint of the

suspect image) and the extracted features to recover and rearrange the scrambled logo to restore the authentication logo. The phase ends if the accuracy rate of the restored logo determined by logo comparison is high enough; otherwise, the process proceeds to retrieve the next available fingerprint from the database and then returns to logo restoration, which takes as input the retrieved fingerprint instead of the extracted watermark. The phase restores the logo from the retrieved fingerprint as well as the extracted features and proceeds to logo comparison. The looping process continues until the authentication logo is discovered or no fingerprint is available.

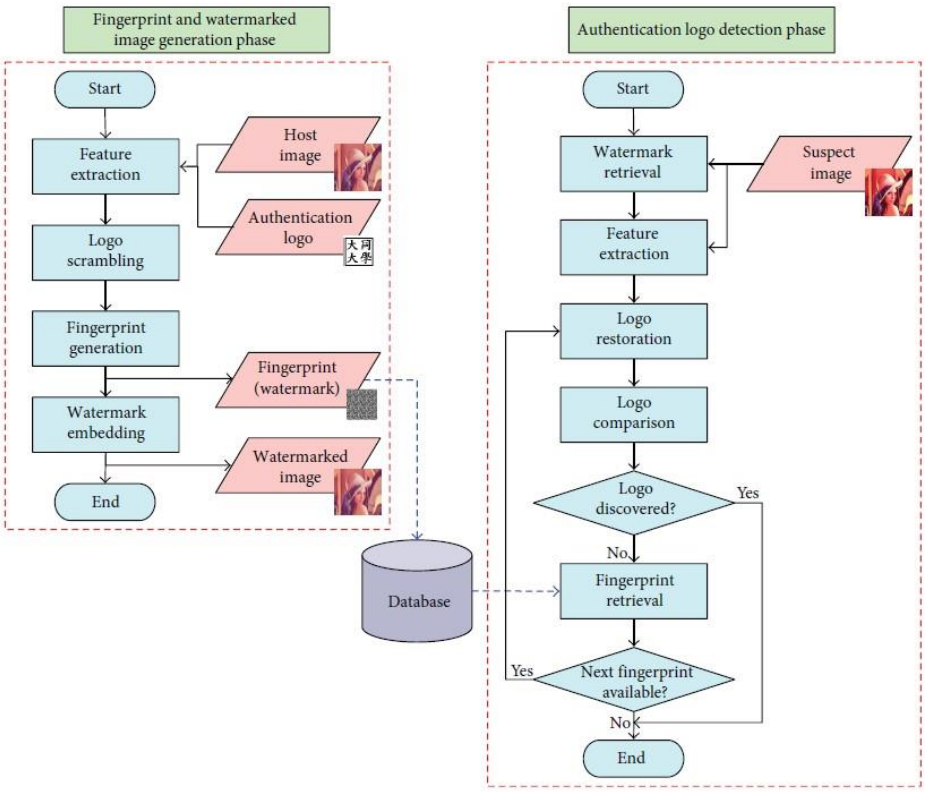


Figure 1 The phases of the proposed scheme.

Input: A color image H ($N \times N$).
Output: A feature image FT ($N/8 \times N/8$).
Convert H to the YCbCr color space
Partition each of the Y, Cb, and Cr channels into $N/8 \times N/8$ non-overlapping blocks of size 8×8
For each corresponding block of the three channels
    Take the first 4 samples from the Y channel, the next 2 from the Cb, and last 2 from the Cr as depicted in Figure 4 to form a packed block of size 8×8
End For
Apply 2D-DWT to each packed block to obtain $N/8 \times N/8$ LL_2 blocks of size 2×2
For each of the LL_2 blocks
    Compute the average A of the four coefficients
    Obtain the feature type T according to (1)
    Determine the FT-share according to T, A, and Table 1
End For
Assemble the FT-shares to form the feature image FT

ALGORITHM 1: Feature extraction.

Two kinds of experiments were conducted to prove the effectiveness of the proposed scheme. The first experiment shows the robustness of our scheme and the other demonstrates the capability of unique identification. In the experiments, the authentication logo used to generate the watermark
Two common measurements used to estimate the robustness of our scheme are described as below.

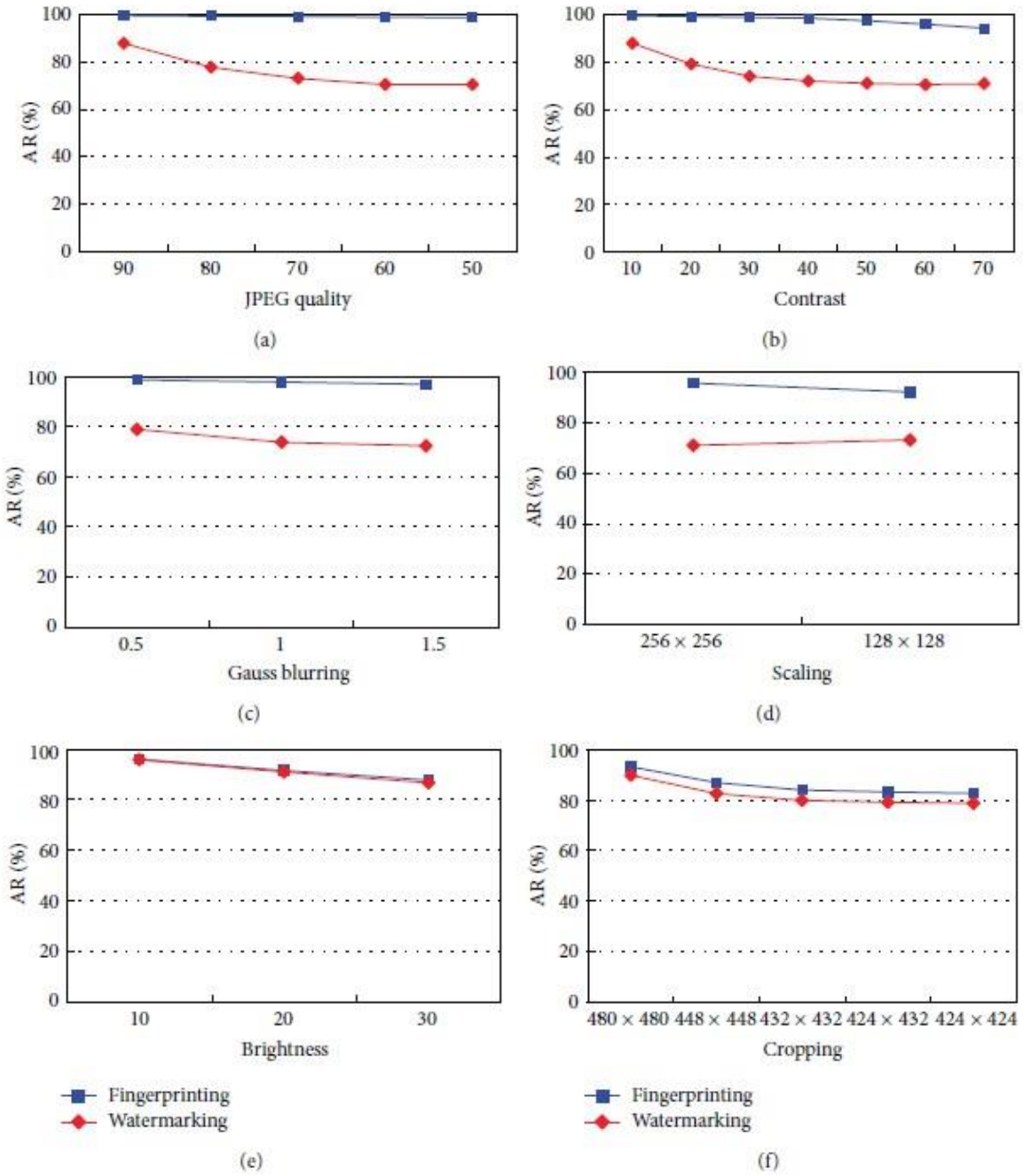
Peak Signal to Noise Ratio (PSNR). The measurement to estimate the colour image quality after image processing is a variant version of normal PSNR. The variant PSNR listed
below does not consider the influence of the green channel because the channel is not modified by our scheme. Consider

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{(\mathrm{MSE}(R) + \mathrm{MSE}(B))/2} \mathrm{dB},$$

where MSE is the *mean square error* between the originalmage and the modified image, which is defined as follows:

$$\mathrm{MSE} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \left(x_{ij} - x'_{ij}\right)^2}{N^2},$$

where x_{ij} represents the original pixel value and x denotes the modified pixel value.
According to the definition of PSNR, the higher the value is, the better the quality of the modified image is. Generally, if the PSNR is greater than 30 dB, the quality of the modified image is acceptable.



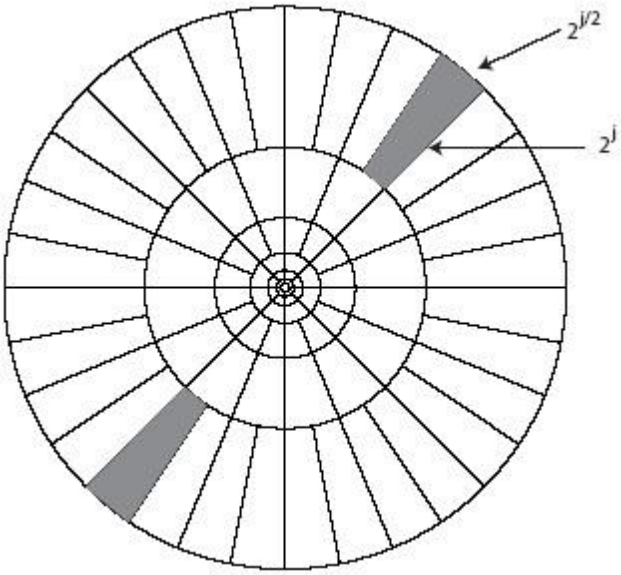
Barcode Watermarking:

Watermarking technique using the curvelet transform In that work, the new method was introduced in the context of the new watermarking framework developed in this project we analyse barcode watermarking as a method that can be applied independently of the watermarking framework of those papers

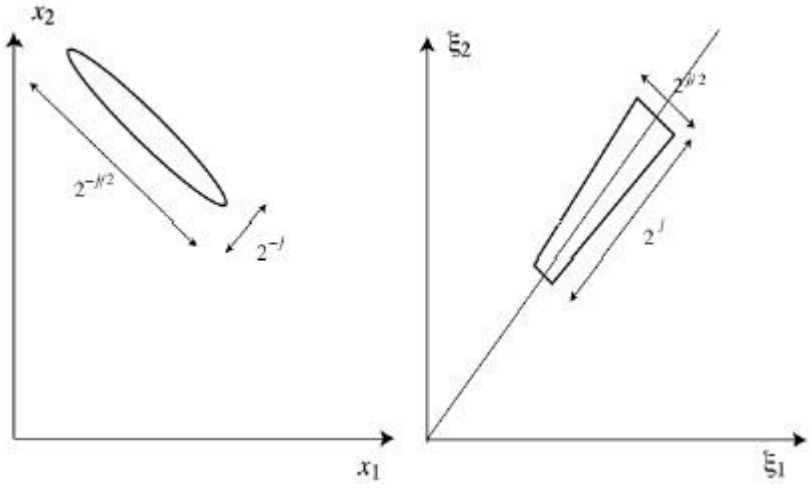
THE CURVELET TRANSFORM

Most first generation watermarking techniques are pixel oriented to free ourselves from individual pixel values we need concentrate on image features for a secondgeneration approach to do that we need to adopt an image description that concentrates on features The Continuous Curvelet Transform (is a multi-scale

transform with frame elements x μ γ indexed by a location scale and orientation parameter μ the transform combines the time frequency properties of wavelets with high directionality and anisotropy Curvelets are constructed by tiling the frequency plane with the polar grid shown in Figure:

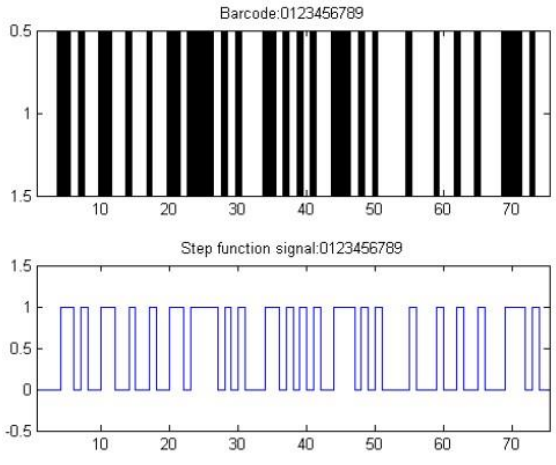


The Continuous Curvelet Transform (CCT) is a multi-scale transform with frame elements indexed by a location, scale and orientation parameter . The transform combines the time-frequency properties of wavelets with high directionality and anisotropy. Curvelets are constructed by tiling the frequency plane with the polar grid shown in Figure 1. Figure 2 shows the curvelet support in the spatial and frequency domains.



A CURVELET BASED WATERMARK

Barcode Watermarking (BW) is an entirely new method. It is a member of a new class of watermarks that modify a statistical distribution of a particular image characteristic. In this case it modifies the distribution of curvelet coefficient magnitudes. As curvelets of particular size are associated with image features on a particular scale this approach fits naturally with a feature-based approach [2}[3]. If we denote the forward FDCT of an image by C , we note that the transformation, $V\ CC+ = U \neq I$ is not invertible (is not a one-to-one mapping). This means that there are many possible choices of curvelet coefficients for a single image. As a result, it is difficult to mark individual curvelets and be able to recover them. To overcome this problem, we adopt a different approach. One way to characterize the set of curvelets of an image is to order them by magnitude and divide the resulting set into percentile ranges. The top 10 −15% of the curvelet coefficients determines its visual appearance. The remaining 85 − 90% relate to small features that generally go visually unnoticed. We break the curvelet structure of a given image into a sum of 100 separate curvelet structures $k\ F$. Each curvelet $k\ F$ is obtained by selecting the curvelet coefficients that lie in the k th curvelet percentile range with all other curvelet coefficients set to zero.



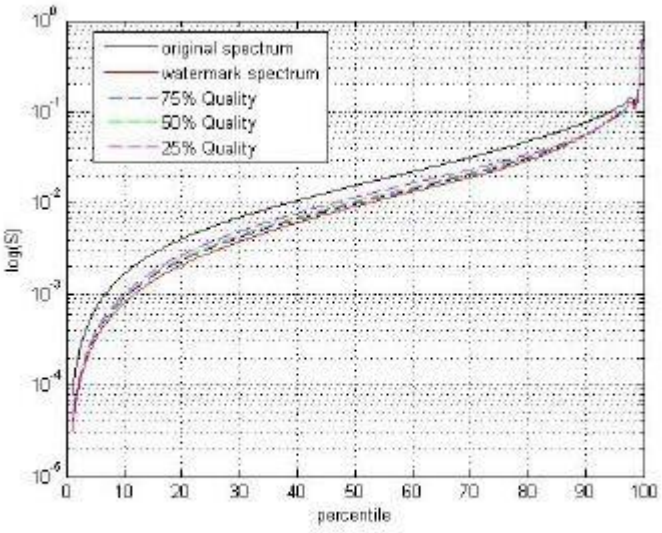


Figure 14 Barcode watermarked images

Bar Code Watermarking consists of the following steps

1. Determine all the curvelet coeffs of the image I.
2. Sort the coefficients by magnitude.
3. Divide the coefficients into 100 percentile ranges.
4. Remove a set of percentiles described by a barcode
5. Reconstruct an image consisting of the reduced set of coefficients

The encoding is not the standard binary, but rather a prescribed set of 7 bits. A "0" represents a white bar of unit length and a "1" represents a black bar of unit length.

For example, the digit 3 is represented by "0111101", which means a white bar of unit width, followed by a black bar of width four, then a unit width white bar, step function signal alternating between 0 and 1.
and finally a unit width black bar. See Table 1. A clean barcode signal will be a For our purposes we wish to leave the top 11 percentile untouched to we place a normal UPC barcode which is 75 units in length in the region just below the top 11 and fill the The full string has 100 bits but we are only interested in bits 16:90 of any extracted string.

Digit	Manufacturer's Number	Product Number
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

Table 1: Standard Industrial Bar Encoding

The watermarked images have very low PSNR around the mid-30 range and the watermark has very low visibility and is very difficult to remove. To summarize, we divide an image into 100 subimages, each one corresponding to a percentile range of curvelet coefficients. We leave the top 10 bands untouched and eliminate the bottom 15. In the central range of 75 percentiles we selectively delete percentile bands according to a barcode. Each 0 in the code corresponds to a band which is to be eliminated. Figure9. Show the sub images corresponding two the three principal curvelet magnitude ranges.

Palm Print asWatermarking:

In the field of biometrics, palmprint is relatively new compared to other biometric traits and is seen as a promising biometric technology. Although a number of works have been reported on palmprint identification and verification showing the importance of palmprint features for human recognition

Intense work has been reported in the last decade for person's recognition using palmprint biometric patterns. However, there has been a little interest in the security of palmprint recognition systems. In this paper we analyse a new blind watermarking technique to address the vulnerabilities of a palmprint recognition system against replay attacks.

In particular, we focus on protecting the system against changing or submitting a fake palmprint image at the enrolment and/or recognition stage while still not affecting the recognition performance. To achieve this, the watermark is embedded into the palm's region of intersect which inherently makes it robust to common attacks. The origin of the palmprint images is proven by checking the presence of the watermark in the watermarked palmprint image by comparing the

Maximum Likelihood (ML) ratio of a given binary hypotheses to a determined threshold based on the statistical model of the watermarked coefficients.

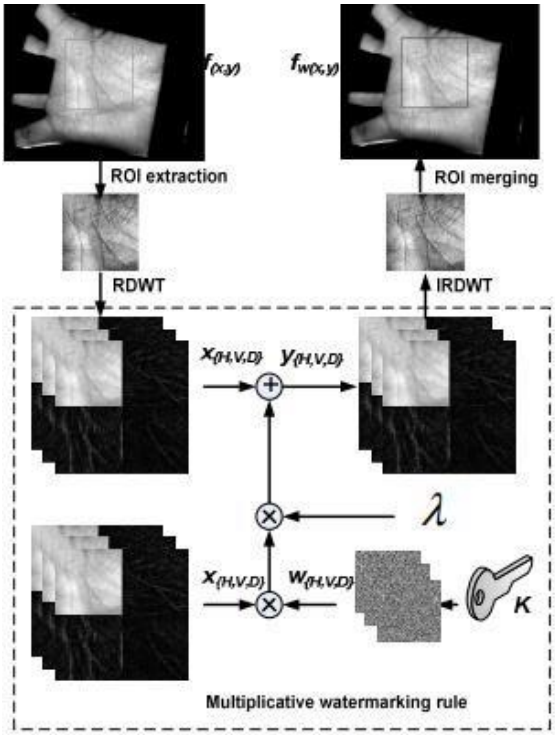


Figure 2. Watermark embedding process

The results obtained have shown that using watermarking concept can efficiently The proposed watermarking technique starts with a localization of a region of interest (ROI) in the palmprint image using the algorithm described in [12]. Embedding only in the ROI allows for more robustness and for a better imperceptibility of the watermark. Subsequently, a given watermark is embedded into the ROI using Redundant Discrete Wavelet Transform (RDWT). The originality of the palmprint images submitted to the recognition system or sorted in the template database can be checked by the detection of the given watermark. An optimum watermark detector has been devised using a Maximum Likelihood Estimation scheme and a modeling scheme for RDWT using Generalized Gaussian Distribution.

The following steps are used to embed the watermark (see Fig.2):

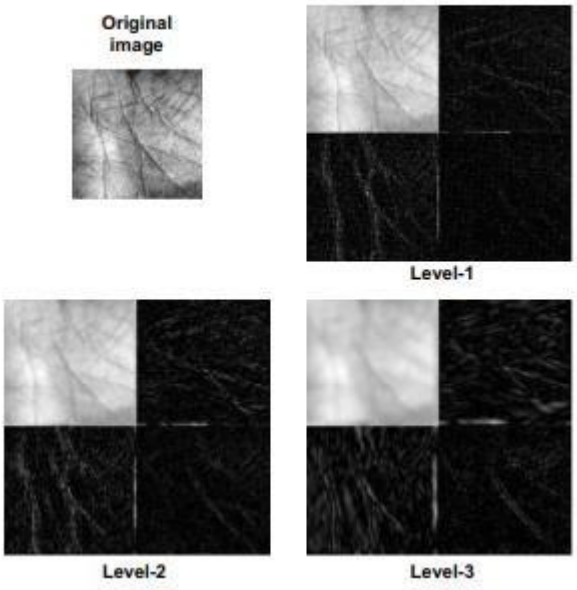
1) The original palmprint ROI is decomposed into 3 levels using RDWT (Fig.3). Only the detail sub-bands of the third level are used to embed the watermark signal.

2) The watermark is embedded into the detail sub-bands of the third level of the decomposition (horizontal (H), vertical(V) and diagonal (D)) using a multiplicative rule as follows:

$y\{H,V,D\} = (1 + \lambda w\{H,V,D\}) \cdot x\{H,V,D\}$ (1) where $x\{H,V,D\}$ are the RDWT coefficients of the given palmprint ROI f. $w\{H,V,D\}$ are the watermark signals distributed over the detail sub-bands while $y\{H,V,D\}$ are the watermarked sub-bands. The parameter λ controls the strength of the watermark.

3) An inverse RDWT is performed to retrieve the watermarked ROI and paste it onto the ROI location in the given/original palmprint image to obtain the watermarked version.

The watermark w is a pseudo-random sequence uniformly distributed in $[-1, 1]$ and generated using a secret key K.



One of the main requirement of watermark embedding relates to the watermark imperceptibility (invisibility) which can be controlled using the watermark strength λ of equation (1). There is a trade-off between the robustness and the imperceptibility of watermark; the higher the watermarking strength is, the better the robustness of the watermark is, but with less invisibility. Fig.7 shows the variation plot of λ against the Peak Signal-toNoise Ratio (PSNR) of a sample palmprint image P olyU 124 S 09.bmp. It can be clearly noticed that RDWT offers more imperceptibility and robustness than the DWT. It is also worth to mention that the invisibility of the watermark depends on the richness of the palmprint images in terms of texture.

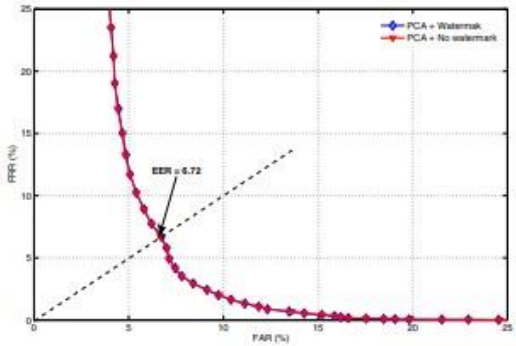


Figure 5. ROC curve, the effect of the watermark on the recognition accuracy using PCA algorithm

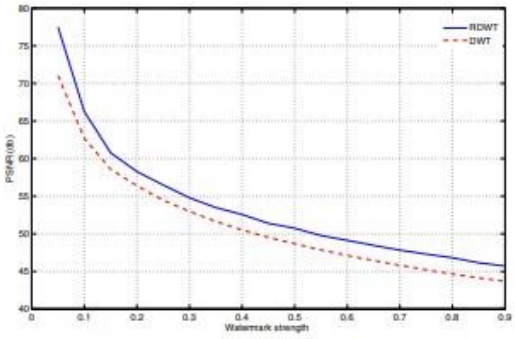


Figure 7. PSNR of watermarked image $PolyU\_124\_S\_09.bmp$ in RDWT and DWT

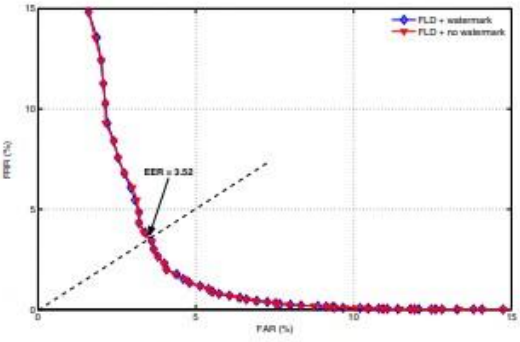


Figure 6. ROC curve, the effect of the watermark on the recognition accuracy using FLD algorithm

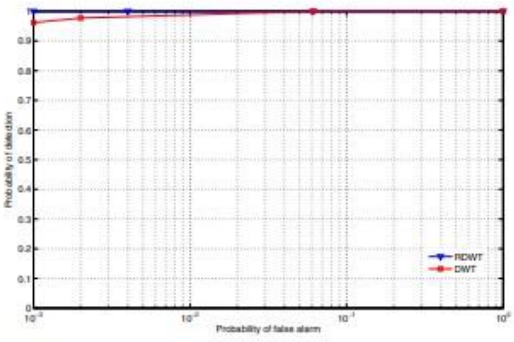


Figure 8. ROC curves: advantage of RDWT over the DWT without any attacks

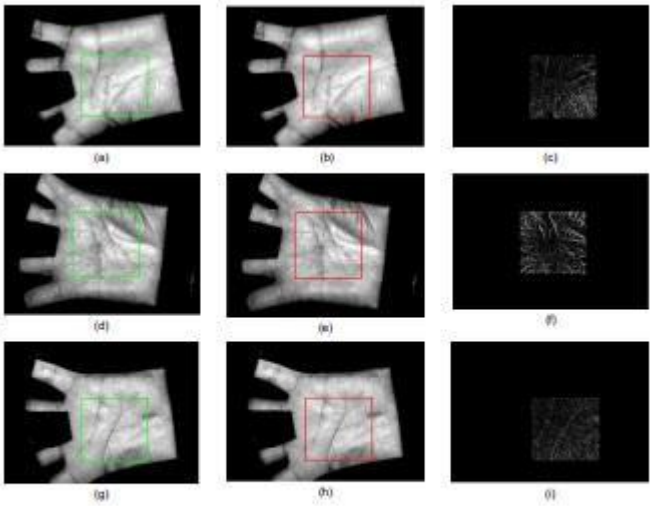


Figure 11. Visual effect of the watermark embedding in three different palmprint. (a),(d) and (g) Original palmprint images. (b),(e) and (h) Their watermarked version with a $PSNR$ of 46.96 *db*, 45.30 *db* and 47.98 *db*, respectively. (c),(f) and (i) The difference images magnified by 20.

The RDWT has been used to embed the watermark while the ML ratio and statistical model of the watermarked coefficients have been used to detect the watermark. The results obtained suggest that watermarking is a useful technique to protect palmprint recognition systems where invisibility, robustness and system performance can still be maintained. In the future works, the robustness of the watermark against geometrical attacks such as rotation and scale will be investigated.

Hand Written Signature Watermarking:

In this project we analyse a technique for embedding handwritten signature image data into color images and extracting embedded image data from colour images. For the sake of security, two watermarked images will be transmitted in different times from the sender's end to the receiver's end, in due course, from two transmitted images, at the receiver's end, original handwritten signature image data will be extracted; a new way of data hiding. From 1994 onward, the use and popularity of Internet in business particular has explored the area of Intellectual Property protection techniques.

This project presents a technique for watermarking Handwritten Signature that achieves robustness by responding to complexity of copy detection, vulnerability to mark removal after revelation for ownership verification and mark integrity issues due to partial mark removal; these three weaknesses.

We have designed a new Double Crossover Algorithm (DCA). This DCA has some rules, known as DCA Master Rules, those rules are illustrated below:-

Step1: Select two individuals: A, B

Step2: Get chromosome from both the individuals (each chromosome with 3 characteristics in both the individuals). A single byte (with 8 bits) is responsible for a character each chromosome made up of 3 bytes, i.e., with 3 characters chromosome from individual A designated as 'x y z' chromosome from

individual B designated as ‘p q r’ two points Crossover taken place between ‘x y z’ and ‘p q r’.
Recombination : ½ A with characteristic ‘x q z’ (where A Dominant and B recessive) and ½ B with characteristic ‘p y r’ (where B Dominant and A recessive).
Step3: First Generation (F1), heterozygous individuals are taken for selffertilization.
Step4: Heterozygous parent gives equal portions of gametes.
Step5: Segregation occurs in production of gametes. Step6: The progeny are then equally divided between the dominant phenotype and the recessive phenotype. Characteristic with original individuals will be derived (F2).
Rules are implemented and tested using 2 different pseudocodes, for security purpose

Watermarking Using Monohybrid Genetic Crossover:-

```
1. Open files in1, in2 in input mode
2. Open files out1, out2 in output mode
3. Read bytes one by one from both the
   input files until EOF
4.     increase the count by one
5.     if count > 54 then {
6.       if RGBCount = 1 then
7.          write byte of in1 to out1
           write byte of in2 to out2
8.       if RGBCount = 2 then
9.          write byte of in2 to out1
           write byte of in1 to out2
10.      if RGBCount = 3 then
11.         write byte of in1 to out1
            write byte of in2 to out2
12.      increase RGBCount by 1
13.      if RGBCount > 2 then
14.        assign 0 to RGBCount
       }
15.    Else
16.      write byte of in1 to out1
         write byte of in2 to out2
```

There were two innovations to the science of genetics: • developed pure lines
• counted those results and kept statistical notes
Pure Line - a population that breeds true for a particular trait [this was an important innovation because any non-pure (segregating) generation would and did confuse the results of genetic experiments] F1 Generation possesses the information needed to produce both parental phenotypes in the following generation. The F2 generation always produced a genotype ratio (1:2:1) and phenotype ration (3:1) where the dominant trait is present three times as often as the recessive trait. Dominant - the allele that expresses itself at the expense of an alternate allele; the phenotype that is expressed in the F1 generation from the cross of two pure lines.

Recessive - an allele whose expression is suppressed in the presence of a dominant allele; the phenotype that disappears in the F1 generation from the cross of two pure lines and reappears in the F2 generation.

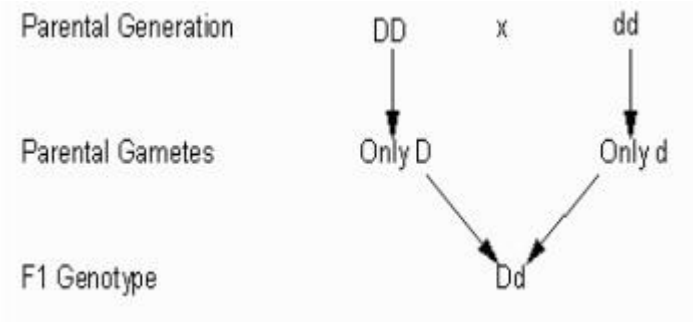
Watermark Detection Using Self Crossover:-

```
1. Open files in1, in2 in input mode
2. Open files out1 in output mode
3. Read bytes one by one from
   both the input files until EOF
4.     increase the count by one
5.     if count > 54 then {
6.       if RGBCount = 1 then
7.          write byte of in2 to out1
8.       if RGBCount = 2 then
9.          write byte of in1 to out1
10.      if RGBCount = 3 then
11.         write byte of in2 to out1
12.      increase RGBCount by 1
13.      if RGBCount > 2 then
14.         assign 0 to RGBCount
       }
15.    Else
16.       write byte of in1 to out1
17. Close in1, in2, out1
```

Thus the following points are important in our work:

- The hereditary determinants are of a particulate nature. These determinants are called genes.
- Each parent has a gene pair in each cell for each trait studied. The F1 from a cross of two pure lines contains one allele for the dominant phenotype and one for the recessive phenotype. These two alleles comprise the gene pair.
- One member of the gene pair segregates into a gamete, thus each gamete only carries one member of the gene pair.
- Gametes unite at random and irrespective of the other gene pairs involved.

Using symbols we can depict the cross of Image (where the signature will be marked) and Handwritten Signature Image, these are Parental Generation, in the following manner:



DD : Image (where the signature will be marked).

dd : Handwritten Signature Image.

Dd : Watermarked Image(s).

For self crossover, Dd x Dd, has to be taken, The F2 generation was created by selfing the F1 Images. This can be depicted graphically in a Punnett square.

Union of Gametes At Random		***D***	***d***	Punnett Square
	D	*DD*	*Dd*	
	d	*Dd*	*dd*	

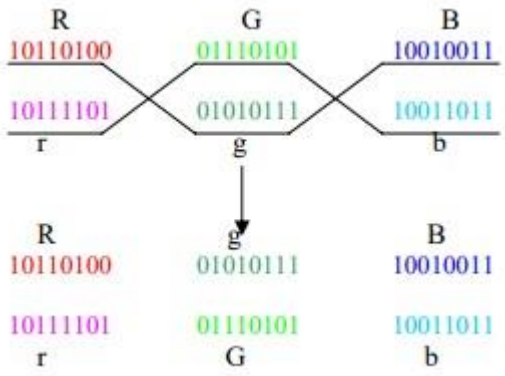
The Punnett Square allows us to determine specific genetic ratios. Genotypic ratio of F2, 1 DD : 2 Dd : 1 dd (or phenotype ratio, 3 D_ : 1 dd)

So, our intension is to get, dd (Pure line homozygote recessive) : Handwritten Signature Image.

½DD (Pure line homozygote dominant) : Image (where the signature is marked), not required in our case.

Heterozygous F1 generation, from 2-points crossover of Parental Chromosomes, Solution:

Final output of Selfing from F1 Generation:-

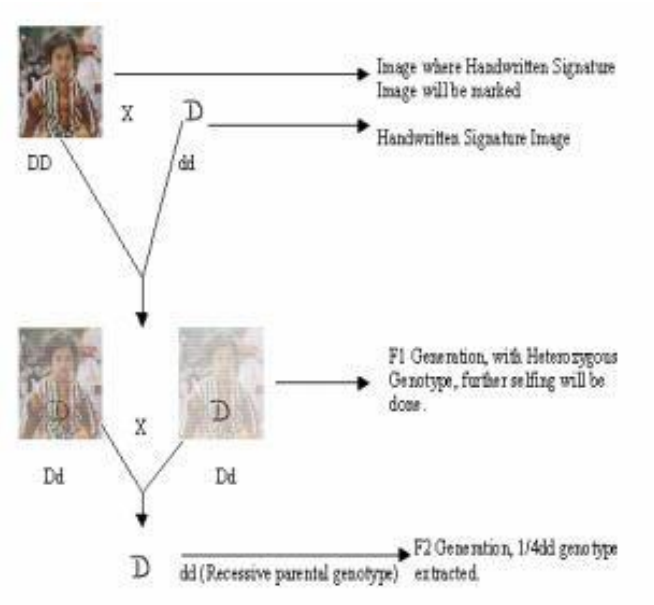


From the product of F2 Generation we have got our desired output, Pure line homozygote recessive, with combination as follow:

½DD (Pure line homozygote dominant) : Image (where Handwritten Signature will be marked).

¼Dd (Heterozygotes) : Watermarked Images.

½dd (Pure line homozygote recessive): Handwritten Signature Image.



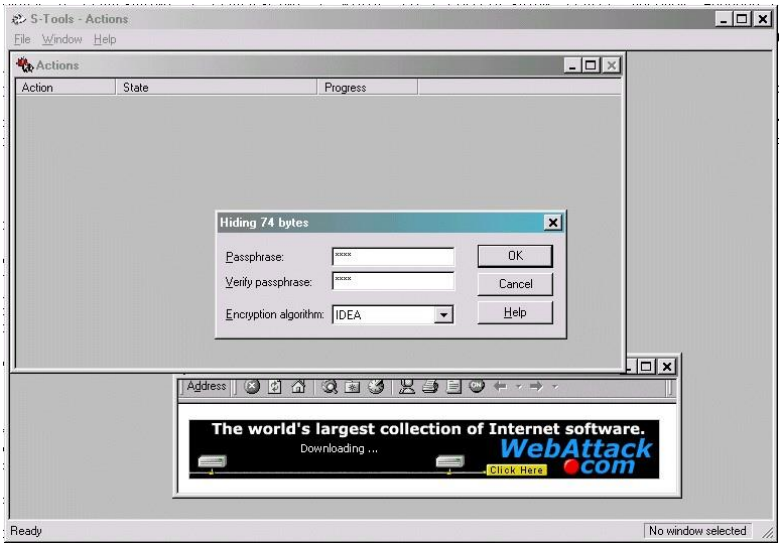
Technique about watermarking is that are more efficient for detection, more convincing for ownership and recipient verification, and more secure and robust against mark removal than 43 existing techniques. Both Heterozygous images are used for mark removal. This technique is evolved as result of extensive study of genetical behavior of living organisms

ONLINE STEGNOGRAPHY TOOLS COMPARITIVE ANALYSIS:- Problem Statement:-

Numerous steganographic software tools are available on the internet today. The basic idea related to these different tools is the same: to create a steganographic software that can hide image or text in another medium. This medium can be another image, video file, or even a sound file. For our project, we have decided to use 5 steganographic software tools namely STools, VSL, OpenPuff, CryptaPix, and Quick Crypto.

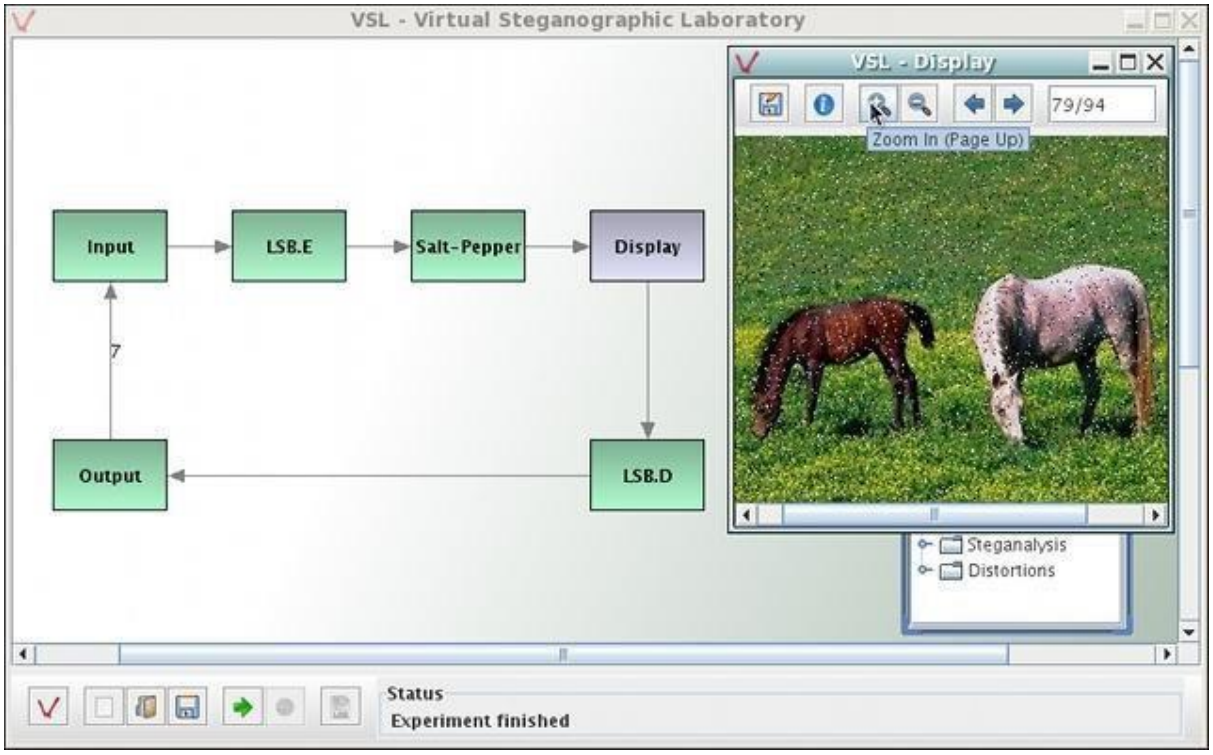
STEG-TOOL – 1(S-Tools)

S-Tools is a steganography tool created by Andy Brown in 1996. This software is able to use images or audio files to hide other images of audio files and vice versa. There are some limitations as to what format the file types have to be that is the audio file must be of WAV type and the image file must be a BMP or GIF file; however, this limitation only applies to the base image which is the image that will hide the secret image. Furthermore, this software has an Action Window that shows the users what steps are being carried out by the software. Moreover, this software also makes use of passphrases and encryption techniques like IDEA, DES, Triple DES, and MDC [17]. Lastly, this software provides a function for revealing or decoding data. A screenshot of the homepage of S-Tools is shown in Fig



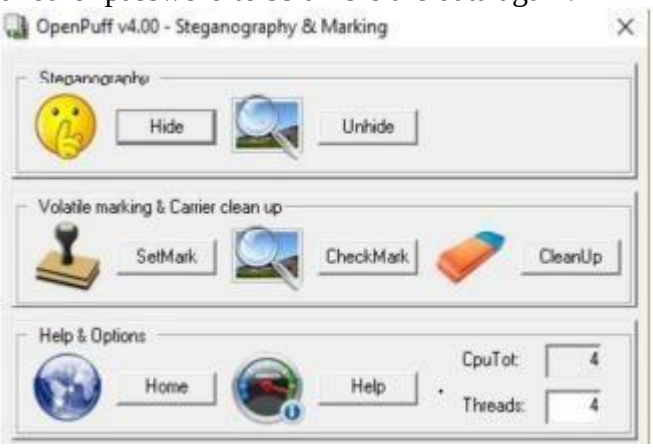
STEG-TOOL – 2(VSL)

VSL is a steganography tool created and maintained by Technology in Szczecin, Poland. This software uses the LSB technique in order to apply steganography. Moreover, this software can also use more advance encoding techniques like Karhunen-Loeve Transform technique and F5 Algorithm. VSL can use any image file format since it uses the raw bytes of the file. This software is created using Java and is an open source software. This software can also use more than one input images. Furthermore, VSL uses a variety of distortion filters which are used to test resistance of Steganographic techniques. These filters include Salt-Pepper, Gauss Noise, Crop, Resize, Median, Gamma, JPEG, Gauss Blur, and Sharpen. Lastly, this software provides features of decoding and analyzing data . A screenshot of VSL is shown in Fig.



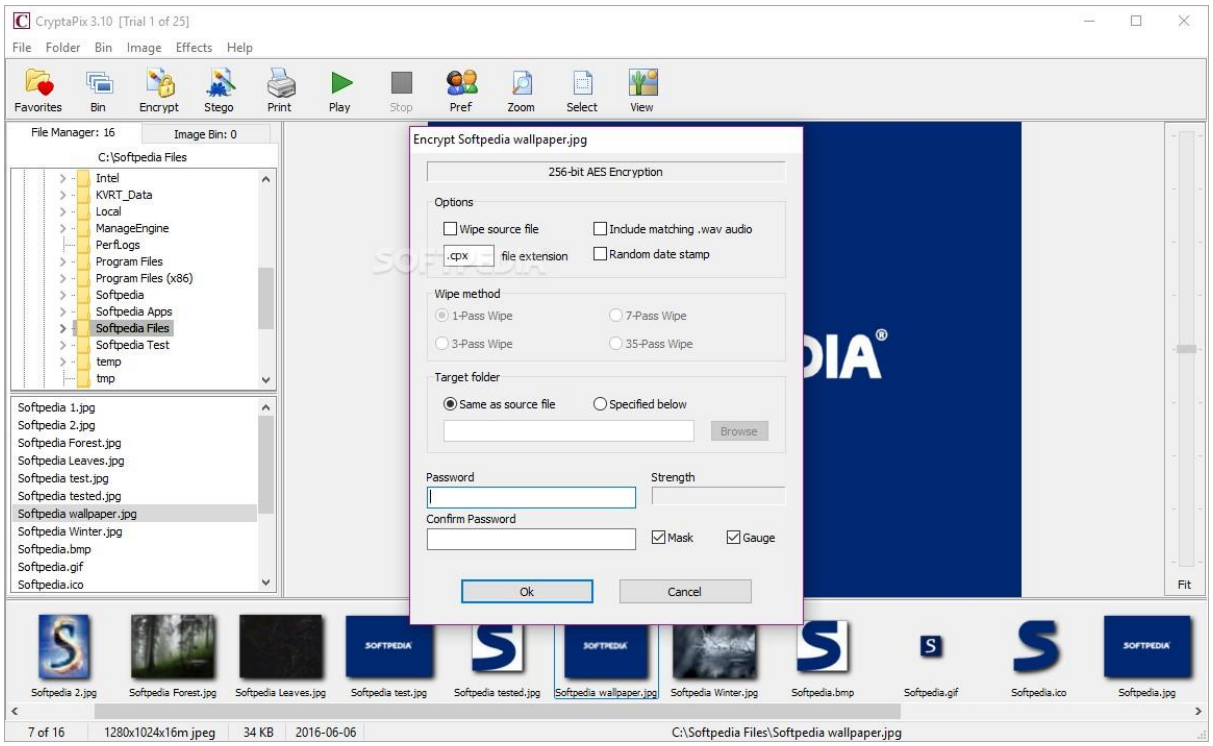
STEG-TOOL – 3(Open puff)

Open Puff is a steganographic software that was founded by the company EmbeddedSW in 2004. Open Puff makes use of carrier chains by dividing the data into carrier chains and then hiding data. Furthermore, 256 Mb of data can be hidden using this software. Image (including JPG, PNG), video (including 3GP, FLV, MP4), audio (including MP3, WAV), and pdf files can be hidden using Open Puff and most of the famous formats are accepted. Moreover, Open Puff makes use of four layers of security. The first layer provides multicryptography which uses an algorithm create using 16 different encryption software products that include AES, Anubis, Camellia, Cast-256, Clefia , FROG, Hierocrypt3, Idea-NXT, MARS, RC6, Safer+, SC2000, Serpent, Speed, Twofish, and Unicorn-A, and uses double passwords. The second layer scrambles the encrypted data and then uses a cryptographically secure pseudo random number generator (CSPRNG) and another password to scramble the data again. Layer 3 combines this scrambled data with noise that is acquired from another CSPRNG in order to whiten the data. The last layer uses the whitened data and encodes it by using a function that is not linear and that makes use of the original carrier bits as well. Lastly Open Puff is open source and free for public use. A screenshot of the Open Puff is shown in Fig.



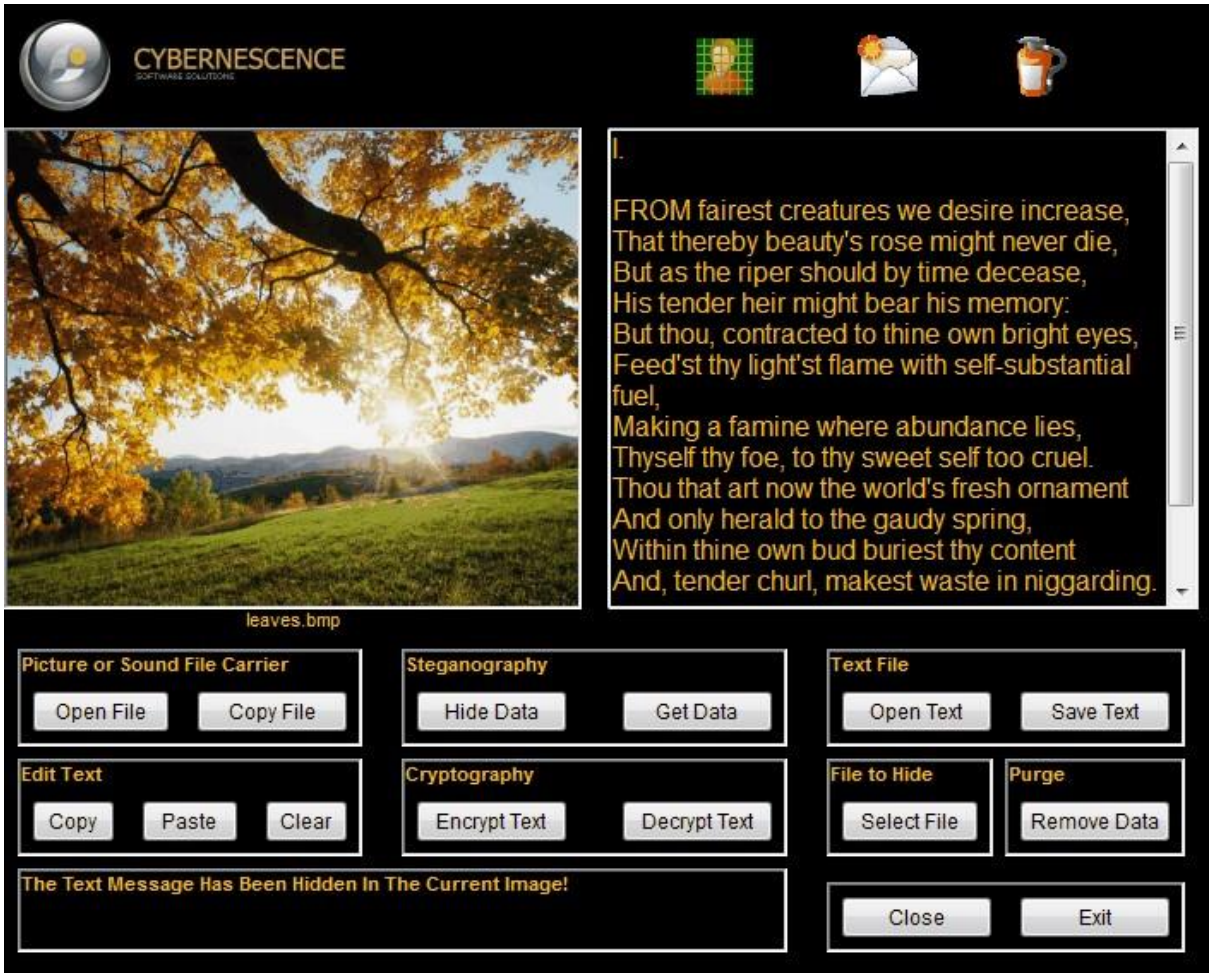
STEG-TOOL – 4(CryptaPix)

CryptaPix is a software created by Kent Briggs for managing and encrypting images. This includes features for editing images like resizing, rotating, cropping, and removing red eye from images. Furthermore, the images can be printed, secured, and organized in slideshows. Moreover, images, text, and data can be encrypted using the 256 bit AES encryption algorithm and then hidden into other image files. A screenshot of the homepage of CryptaPix is shown in Fig



STEG-TOOL – 5(CryptaPix)

Quick Crypto is a multi-feature software created by the company Cybernescence that includes features like file, volume, and email encryption, steganography, password managing, file recovery and shredding, system cleaning, and internet privacy. This software allows any type of file to be encrypted which includes image files. The encryption algorithm that this software uses for single password is AES, Triple DES, and Blowfish. Moreover, a password is required by software to make the encryption even stronger. The encrypted file can only be decrypted if the correct password is entered. Furthermore, this software also offers Steganography for numerous file types. For images, this software supports jpg, gif, and bmp file types, and for audio, this software supports wav and mp3 file types. This restriction applies to just the carrier files and any file type can be hidden in these carrier files. A screenshot of the homepage of Quick Crypto is shown in Fig



Feature Analysis:-

For the first software. S-Tools, the encryption method Triple DES is choosen and a passkey “secret key” was choosen to hide the secret image in the carrier images. The second software, VSL, does not provide encryption or a passkey. Thus, the secret image was simply hidden in the carrier images without any added security measures. In the third software,Open Puff, a passkey “secretkey” was used and the software used its own unique encryption algorithm. Furthermore, there were added security measures provided if multiple passwords were entered. However, only one password was used. The fourth software, CryptaPix was used to first encrypt the secret image using AES encryption algorithm with 36 bits since the trial version of the software was used. The password for the encryptionwas kept as “secretkey”.This image was then saved and then hidden in the carrier images using the password and the secret image was then hidden in the carrier images.

Feature Analysis of Different Stegnographic Tools:-

Steganographic Software	Carrier Image Formats	Memory Usage	Encryption Support	Steganographic Algorithm
S-Tools 4.00	BMP, GIF	1.6 MB	IDEA, DES, Triple DES, MDC	LSB
VSL 1.1	Any	43.8 MB	None	LSB, KarhunenLoeve Transform technique, F5 Algorithm.
Open Puff 4.00	BMP, JPG, PCX, PNG, TGA	36.0 MB	Algorithm created using AES, Anubis, Camellia, Cast-256, Clefia , FROG, Hierocrypt3, Idea-NXT, MARS, RC6, Safer+, SC2000, Serpent, Speed, Twofish, and Unicorn-A	LSB
CryptaPix 3.10	BAY, BMP, CRW, CR2, CUR, DCR, DCX, DIB, EMF, FAX, GIF, G3F, G3N, ICB, ICO, JIF, JPC, JPE, JPG, JP2, J2C, J2K, MRW, NEF, ORF, PBM, PCX, PEF, PGM, PIX, PNG, PPM, PSD, PXM, RAF, RAW, RLE, SRF, TGA, TIF, VDA, VST, WBMP, WMF, XIF, X3F	6.3 MB	256 bit AES encryption (32 bit AES in the trial version)	Dividing plaintext into 3 bit segments
Quick Crypto 4.1t	JPG, GIF, BMP	4.7 MB	Blowfish, AES, Triple DES	LSB

Steganalysis:-

PSNR and SSIM values between the original carrier images and the images that hid the secret images were collected and the results are shown in Tables According to Tables , S-Tools has the highest PSNR values for all the three images. This means that the highest quality of images have been produced by S-Tools.

The second highest values for PSNR have been produced by Quick Crypto. We can see that there is just a minor difference in the values between Quick Crypto and S-Tools. Thus, both these software tools have produced high quality images.

Furthermore, VSL had the third highest values for PSNR although they are considerably lower than that of S-Tools and Quick Crypto. Lastly, the last two software tools that is Open Puff and Crypta Pix have similar values of PSNR which although are the lowest values, but still are considered as high quality images.

From the Tables the SSIM values for the various steganographic software tools that were used in this paper. SSIM is a better metric to use rather than PSNR because SSIM measures the similarity between images in a way that the human eye does and thus is more accurate.

According to Table , the highest SSIM values can be seen for S-Tools and Quick Crypto where the values are very close to 1 and some of the values are actually 1. According to the SSIM value of 1, the images are exactly alike. This means that the software tools have created images that are the exact same even after steganography had occurred. Furthermore, VSL, Crypta Pix and Open Puff have similar SSIM values that are lower than STools and Quick Crypto but are still close to 1. It is to be noted that all the software tools have SSIM values that are very close to 1 and thus, the images produced by these software tools are extremely similar.

PSNR VALUES:-

Image	STools 4.0	VSL 1.1	Open Puff 4.00	CryptaPi x 3.10	Quick Crypto 4.1t
Image 1: Lena Image	66.32	57.09	56.76	56.34	65.02
Image 2: Homogeneous Image	67.02	58.76	57.35	57.26	66.01
Image 3: Googled nature Image	67.1	57.95	56.65	57.03	66.25

SSIM VALUES:-

Image	STools 4.0	VSL 1.1	Open Puff 4.00	CryptaPi x 3.10	Quick Crypto 4.1t
Image 1: Lena Image	0.9999	0.9988	0.9989	0.9988	0.9998
Image 2: Homogeneous Image	0.9998	0.9987	0.9981	0.9981	0.9998
Image 3: Author' Image	1.0	0.9997	0.9997	0.9997	1.0

CONCLUSION:-

We found that the best performing software in terms of creating an image that resembled the most with the original image and had the best quality was S-Tools. Furthermore, Quick Crypto was the second best performing software and its values were very close to that of STools. Finally, the remaining three software that is VSL, Open Puff, and Crypta Pix had similar performances which were not as good as STools or Quick Crypto but were still extremely good. An improvement that can be made to the existing steganographic software tools is that multiple passwords can be used like in the case of Open Puff. This ensures that even if the hidden data is extracted and one correct password is entered, the data is still protected by two more passwords. Furthermore, some of the software tools used had restrictions for the format of the carrier images. These restrictions meant that the initial image had to be converted to a different format thus altering the quality of the image. Since steganography is intended to keep the image as similar to the initial image as possible, this change needs to occur.