Whitepaper
Mirage

A Steganographic Data Security Algorithm with Reduced Steganalysis Threat

Mohammad Fahmi Alalem M6d.Alalem@gmail.com

Abdallah Muhanah Manasrah A.Manasrah@birzeit.edu.ps

Birzeit University, Birzeit - Palestine

Abstract

Steganography is a method allowing its user to conceal some data to be sent or stored from the sight of an unwanted investigator inside an innocent vessel. If it is used wisely the message should not even be noticed to those unwanted parties not mentioning trying to attack it.

Steganalysis is the branch of data processing that seeks the identification of carrier vessels and retrieval of message hidden. In this paper we present the Mirage Image Steganography algorithm, an algorithm that we claim to be extremely safe, built over DCT (Discrete Cosine Transformation) frequency domain mutation, the algorithm uses error reductive measurements such as signal matching and statistical recovery to obtain insertion rate up to 13% of output vessel with reduced detectability. The algorithm also is a multi-messageFile multi-messageVessel capable.

Table of Contents

1 Introduction:

Steganography is the art and science of writing hidden messages inside innocent looking containers in such a way that no one apart from the sender and intended recipient even realizes the existence of a hidden message. The technique is ancient emerging monster that have gained immutable notice as it have newly penetrated the world of digital communication security.

Steganalysis is a newly emerging branch of data processing that seeks the identification of steganographic covers, and if possible message extraction. It is sinominous to cryptanalysis in cryptography.

Steganography differs from cryptography in that the first makes the message unreadable while the second makes it unseen. It is nevertheless possible to use both techniques to add security to our messages. For more details see the section 1.2.

The word *steganography* is of Greek origin and means *"covered, or hidden writing"*. Its ancient origins can be traced back to 440 BC When Demeratus sent a warning about a forthcoming attack to Greece by writing it on a wooden panel and covering it in wax.

Another ancient example is that of Histiaeus, who shaved the head of his most trusted slave and tattooed a message on it. After his hair had grown the message was hidden. The purpose was to instigate a revolt against the Persians.

Steganography used in electronic communication include steganographic coding inside of a transport layer, such as an MP3 file, or a protocol, such as UDP. [1]

Wide varieties of steganography implementations in sound files, movies, exe files, videos and many other file types exist.

The technique have had a lot of attention after the USA government had claimed the technique was used by al-Quada terrorists in there communication, Claims that were afterwards proven to be false.

Steganography can be also look to as a branch of digital watermarking. While there is some differences between the two concepts the two works in a similar way. For further information see section 1.3.

Steganography also can be implemented to cryptographic data so that it increases the security of this data.

1.1 Applications of Digital Watermarking and Steganography

Digital watermarking is the technique of embedding digital marks inside a container so that there is a logical way of extracting the data embedded, while not harming the container in any perceived way.

This is achieved by modifying some of the original file's redundant data so that the message is embedded in it. This technique is primarily used for protection of royalties and copyrights.

While Steganography dose use redundant portions of the container file to embed a message, the two field orientation are totally different. Watermarking regard the vessel file as the important data that is to be preserved, steganography on the other hand uses such files to deliver it's messages.

Watermarking Focuses on the inspiration of the embedded message and the vessel, such concern is very important for the field of use of watermarking.

Though it is not recommended steganography and watermarking are often used exchangeable. Applications of digital steganography and watermarking cover a very wide range of fields that include:

- Copyright protection
- Electronic commerce and copy control
- Forensic Image authentication
- Personal identification documents authentication.
- Cartography
- Medical imaging
- Broadcasting monitoring
- Network attacks tracing
- Covert communication
- Honeypoting.
- Anti-spoofing & masquerade attacks algorithms

Many other intelligence, counterintelligence, and security applications have been suggested or implemented.

As an example Microsoft Co. had announced its intention of making a prototype of an application watermarking tool that address the problem of software licensing theft.

A paper presented at the Fourth Information Hiding Workshop, principally authored by Microsoft's Ramarathnam Venkatesan et. al., says analyzing program flow through graph theory perspective and then subtly altering it to hide a secret watermark is a very useful way to make sure no code alternation, reverse engineering, and tempering "Since our watermark approach is based on generating hard instances of the location problem, it may be used to embed crucial checks in the program such that is hard to locate and hence temper with them." [2].

Probably the most likely application is using the watermark to encode licensing information, such as date of expiration or Ethernet addresses of computers permitted to run the program. When a user double-clicks on a program, Windows would read in the license information and could prevent the program from executing.

1.2 Steganography vs. Cryptography

Steganography and Cryptography are parallel data security techniques, both techniques can be implemented side by side, in fact most steganographic utilities implements cryptographic data security.

While the two fields aren't contradicting each other they have different qualities.

- Steganography can use cryptography but not the other way around.
- With cryptography we can protect the message but not hide its existence.
- Steganography has a very expensive payload as compared to cryptography.
- Steganography demands vessel files to be delivered in addition to the key and data required normally

Of course there is no benefit from trying to hide a message that is logically anticipated, that is for example a governmental portal known to send or/and receive secret massages wouldn't benefit from this technique, but a business firm seeking hiding of some document or a spy would certainly appreciate such a technique.

1.3 Steganography vs. Watermarking

Steganography pay attention to the degree of Invisibility while watermarking pay most of its attribute to the robustness of the message and its ability to withstand attacks of removal, such as image operations(rotation, cropping, filtering), audio operations(rerecording, filtering)in the case of images and audio files being watermarked respectively.

It is a non-questionable fact that detectability of a vessel with an introduced data (steganographic message or a watermark) is a function of the changeability function of the algorithm over the vessel.

That is the way the algorithm changes the vessel and the severity of such an operation determines with no doubt the detectability of the message, since detectability is a function of file characteristics deviation from the norm, embedding operation attitude and change severity of such change decides vessel file detectability.

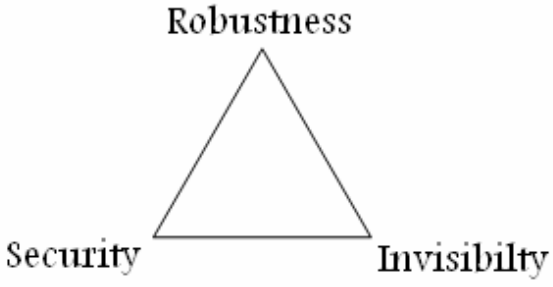


Figure 1 The Triangle of conflict

A typical triangle of conflict is message Invisibility, Robustness, and Security. Invisibility is a measure of the innotability of the contents of the message within the vessel.

Security is sinominous to the cryptographic idea to message security, meaning inability of reconstruction of the message without the proper secret key material shared.

Robustness refers to the endurance capability of the message to survive distortion or removal attacks intact. It is often used in the watermarking field since watermarking seeks the persistence of the watermark over attacks, steganographic messages on the other hand tend to be of high sensitivity to such attacks.

The more invisible the message is the less secure it is (cryptography needs space) and the less robust it is (no error checking/recovery introduced).The more robust the message is embedded the more size it requires and the more visible it is.

Those trade offs had separated the worlds of the two fields.

1.4 Steganography Overview

The key concept of steganography is the ability to hide and communicate information without the potential risk of detection of the communication. To clear the steganography concept let us make an example.

Assume that Alice and Bob are two parties who wish to communicate secretly, let Eva be an attacker monitoring the parties, had Bob and Alice try to communicate through encryption this might be caught by Eva.

Though Eva might be not able to unlock the content of Bob's and Alice's communication, but three nonnegotiable

truths are automatically and logically deducted:

- Alice knows Bob.
- Alice and Bob had communicated some information
- This information is probably important since it is encrypted

Standing on those grounds more and more attention will be dragged towards attacking such an important message.

Now let us say that Alice and Bob are communicating using a steganography algorithm S, using S Alice or Bob could encode his/her message into some carrier medium (plain, text, photo, film, exe file, or a song) or a combination of them, now when communicating the secret message what is apparent to be communicated is the cover medium and Eva probably won't suspect the transaction.

Even if for some reason the transaction should be suspected, there should be no possible reasonable way of telling if that there was a secrete information exchange or a normal file transaction.

Unless there was a way for Eva to tell wither the cover medium is really a dummy message and that there is some data hiding inside it, the secrete message exchange is totally secured and thus the message.

For this to be true the currier medium or the dummy message must be indistinguishable from its same type files. This could be perceived by two ways; the first way is that the human scenes shouldn't be able to suspect the dummy message to be not a true message, second there should be no algorithm to reasonably tell.

1.5 Bases of Secure Steganography Algorithms

1.5.1 Steganography Algorithms Invisibility

Totally secure steganography systems are those systems that their messages cannot be identified as steganographic messages with any rational means better than random guessing.

This fact was stated by Christian Cachin in an information-theoretic model for steganography in which he related the security of such systems against passive eavesdroppers, in this model it is assumed that the attacker has a complete knowledge of the algorithm and he only hasn't the secret key material.

The attacker can use detection theory to decide between hypothesis C (that message contains no hidden message) and hypothesis S (that a hidden message exists inside the dummy message), the steganography algorithm is perfectly secure if no decision rule exists that can perform better than random guessing. [4]

To achieve this goal the dummy message containing steganographic message shouldn't differ in anyway form matching type files. This simple rule was constantly been violated in all steganalysised algorithms as shown in the numerous steganalysis algorithms and papers.

The detectability of a message as earlier stated is also a function of the change of the vessel characteristics; the less error introduced to the vessel characteristics the less detectable it is. And so in order to reduce detectability one can make one of the counter measurements falling in one of those acts category:

- Make the insertion rate statistically insignificant to the size of the vessel.
- Select a vessel that best suites your message with minimizing change.
- Select a dynamic insertion schema that seeks minimization of error.

The first way of act is indeed taken by a steganography algorithm called steghide. Its author claims that by limiting the insertion rate to only 5% of the message size it cannot be identified by statistical analysis.

The second way of act can be implemented through some sort of a database of pictures, when the user needs to send information the algorithm searches the database for a perfect matching or a very close matching picture and then make any necessary alternation and the message should be good to go.

The main drawback of this technique is the need for an enormously large database of images on the sender size.

For a database of images that are able to hide a stream of M bits. The possible number of entries of the database is 2^M. This number is sure an enormously gigantic number.

Number of entries can be safely reduced by reducing number of stream bits but this way either a very small data insertion rate is gained, or a very huge transaction of very small images (suspicious).

Nonetheless this method has very appreciated fidelity that is the detectability of these container files are converging to zero. Change in the images after writing is a function of the database size, the more possibilities and combinations of the M bit stream the database have, the less change is introduced and thus the less detectable.

The third way of act is by trying to adapt the message insertion schema to the vessel data schema, algorithms like signal matching could solve quite a deal of the problem, this can be achieved by a generated random walk through the image which minimizes the error value.

Other character preserving measurements such as statistical recovery of the numerical and statistical representation of the vessel data can help recover whatever deviations introduced by embedding operation.

1.5.2 Steganography Algorithms Security

Information theory tells us through Kerckhoffs' principle of cryptography that the security of the system should rely only on the secret key material [4]. And this should be the case in any steganography system.

Using this cryptographic principle there should be a steganographic key governing the distribution of message data through container vessel so that access should be granted to the key holder only.

This fact was neglected by most steganographic algorithms, while most of those algorithms request a key from the user; most of the algorithms use this key as a cryptographic key.

While cryptography can be used as a second defense wall in case of system failure, or as a brutal-force counter measurement, this shouldn't cancel the security of the steganographic system.

1.6 Steganography Algorithms Bounds

The Corner stone of a security system is of course its safety, and this is indeed the most important aspect of a steganography system.

Security algorithms usually handle relatively small messages, Steganography in particular is usually used in critical high security profiles where the transmission often include bytes or kilobytes rather than Megabytes, and so it is safe to assume that the message size is not of a very alarming effect.

On the other hand it is very desirable that the transmission of relatively big messages should be enabled, probably with very big sizes. Message segmentation among numbers of carrier files could achieve.

Multi file message segmentation might in addition reduce vessel file error and build another level of security (if a cryptographic message segmented in N files, and file k is not detected, message cannot be rebuilt).

2 Steganography / Steganalysis Historical Review

2.1 1G Steganography

First generation steganographic algorithms considered only human abilities to spot irregularities as the only detection technique.

Those algorithms implemented in the Image steganography relied on the fact that computer images normally have quite a bit of redundant data and that changing the contents of those data (as pixels or color plate elements) could make us enough space to embed a considerably large message (50% data rate with BPCS steganography[5]).

Not all the first generation Steganographic algorithms had the huge data rate of the BPCS, but all did embed their data in very unique manner that created some unique irregularities. Examples of first generation Image Steganography algorithms include EzStego, JSteg, Steganos, and S-Tools.

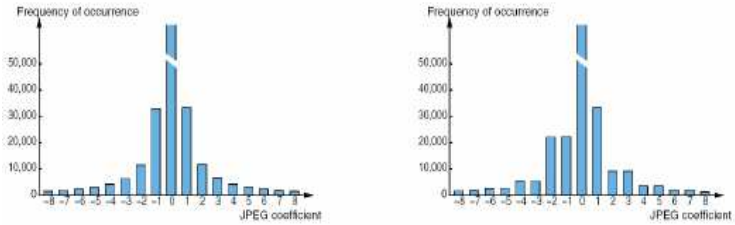
Fortunately this didn't last for ever as new evolving branch of data analysis was born "Steganalysis".

2.2 1G Steganalysis

The first released results of steganalysis maybe the paper by Andreas Westfeld and Andreas Gtzmann.

In there paper Gtzmann and Westfeld made clear how to attack a number of very famous image based steganography algorithms both visually by the use of some image techniques and naked eye, or by automated statistical algorithms.

Figure 2 JPEG coefficients deviations (left normal,



right steganography Image)

The two attacks introduced were the filter attack and the PoV statistical attack. Both attacks were designed to address the steganographic systems of spatial domain embedding [6].

The key idea here that there study have opened Pandora's Box, the application of there note wasn't bounded by the domain of image containers; it could be

implemented for any type of container files.

Following the attacks by Gtzmann and Westfeld new steganographic system emerged using frequency as embedding field those systems. Those systems were immune to the PoV and the filter attack, but had a lower data rate.

2.3 2G steganography

The birth of steganalysis was a very fortunate event as it helped develop second generation steganography. And like cryptography and cryptanalysis the war of the two worlds helped both fields.

There was a trend to elevate the container files insertion rates and minimizing naked eye visual attacks through the use of filtering noisy regions of image to embed data in it.

Though some achievements were made, this type of research was a huge fault, it ignored the simple fact that the core technique was compromised and that the algorithm was no more secure even to a small image manipulation script.

What could be called second generation steganography has steganalysis aware qualities, introducing statistical recovery to the message, and minimizing deviations. The two Examples that we consider to be of the second generation are OutGuess and F5.

OutGuess used a recovery optional function to recover its effects, while F5 built a well defined insertion schema that tend to recover the image it self.[7 & 8]

The techniques of the second generation wasn't all together very well as compared to the advance in the steganalysis work (part of this is the huge funds given to researchers at the steganalysis field, tens of millions of dollars were spent in those projects).

2.4 2G Steganalysis

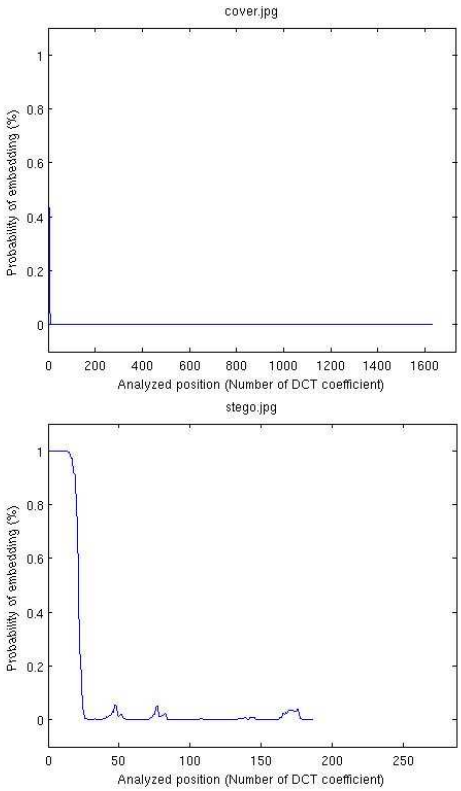


Figure 3 Chi-Square attack against a normal image (left) and a steganographic image (right)
As it is clearly seen the test show a high probability of embedding at the first of the stego image
The embedded message could be further more estimated in size and location

Following 2G steganography birth other studies were lunched. Niels Provos and Peter Honeyman in their paper "Detecting Steganographic Contents over the Internet" for example had made a wide scan of two web sites namely e-Bay and USENET after claims by an article in the USA-Today newspaper and multiple claims of the CIA and the American Department of Homeland Security that steganography is being in use by terrorist's sleepy networks.

The scan was over 3 Million images and had shown no evidence of wide use of

steganography; however the search contained some very valuable information. The project had succeeded to achieve high detection rates through what is called the Chi-Square attack. [9]

Provos along with other colleagues kept on working in steganography and steganalysis field producing a number of important papers that could be found on the internet, making two advanced steganalysis and steganography algorithms (stegdetect and OutGuess respectively).

Another project was lunched by the Department of Homeland Security by Dr. Hany Farid, in his paper "Detecting Steganographic Messages in Digital Images" Dr. Farid has implemented an AI two class Fisher Liner Discriminant Analysis based machine using 72 statistical features and a training data of 40,000 natural images. A sample of his work is summarized in the following table:

Embedding	Messsage	JPEG	GIF	TIFF
Jsteg	256 × 256	94.0	-	-
Jsteg	128 × 128	95.7	-	-
Jsteg	64 × 64	95.3	-	-
Jsteg	32 × 32	51.7	-	-
OutGuess$^-$	256 × 256	92.8	-	-
OutGuess$^-$	128 × 128	63.4	-	-
OutGuess$^-$	64 × 64	27.7	-	-
OutGuess$^-$	32 × 32	5.9	-	-
OutGuess$^+$	256 × 256	74.4	-	-
OutGuess$^+$	128 × 128	41.4	-	-
OutGuess$^+$	64 × 64	14.0	-	-
OutGuess$^+$	32 × 32	4.1	-	-
EzStego	194 × 194	-	45.2	-
EzStego	128 × 128	-	13.8	-
EzStego	64 × 64	-	2.9	-
EzStego	32 × 32	-	1.6	-
LSB	194 × 194	-	-	42.3
LSB	128 × 128	-	-	16.8
LSB	64 × 64	-	-	2.8
LSB	32 × 32	-	-	1.3

Figure 5 Detection rates of several Algorithms as detected by Dr. Farid

Dr. Farid was an entrepreneur in a way that not only had he detected first generation steganography tools successfully but also he had detected the well known OutGuess second generation Steganography algorithm [10].

Another study by Dr. Farid and Dr. Siwei Lyu had extended Dr. Farid work to the detection of F5 algorithm among others and elevating detection rates [11].

Like in cryptography and cryptanalysis, steganography and steganalysis will sure have a very long way to go.

3 The Mirage Algorithm

3.1 Algorithm Overview

Quite frankly the algorithm we implemented doesn't conclusively grantee that it is undetectable, it merely counter-measured the attacks publicly known.

Mirage algorithm has the following traits:

- Has a comparably high cover insertion rate as compared to other DCT based systems
- Enables multi-messageFile multi-carrierFile capabilities
- Uses signal matching message fitting
- Uses pseudo-random insertion for both data stream and header streams
- Anti-visual and filter attack
- Anti-Chi square and Chi square extended statistical attacks

Traits that we are welling to revel in the next sections.

3.2 Algorithm Illustrations

3.2.1 The Embed function

The embedding function used is an LSB DCT mutation.

```
Embed()
Begin
   order = getOrder(getBestFit());
   prepareHeader();
   bit messageBit;
   for (int I = 0 ; I < messageLength &&   I < imageCoffecientsLength; I++)
   begin
        messagBit ← getMessageBit(i);
        shortCircuit = (image[order[i]] ==0)
                  || (image[order[i]] ==1)
                  || DCcomponent(order[i]);

        if(shortCircuit) then
                  continue;
        end
        temp = image[order[i]]>>1<<1| messageBit;
        If(temp != image[order[i]]) then
                  addToRecovery(temp)
        end
        image[order[i]] = temp;
     end
   writeHeader();
end
```

Data of the image is obviously first transformed to the discrete cosine transformation (actually read from the JPEG file directly). And then the message is imbedded by the algorithm.

3.2.2 Signal Matching Function

```
bestFit() : integer
begin
  int BFK= MAX_INT + 1;
  int bestGainedKey = BFK;
  int bestError = MAX_INT;
  while(BEK < maxKeyValue)
  begin
        Inc(BFK);
        order ← getOrder(BFK);
        for( int I = 0 ; I < messageLen ; I++){
              if(image[order[i]]>>1<<1 | message[i] !=image[order[i]])
                    error++;
        }
        if(error < bestError){
              bestGainedKey = BFK;
              bestError = error;
        }

        If(bestError < MaxError)
              Return bestGainedKey;
  end
  return bestGainedKey; //best fit key
end
```

This function following the image and the data attribute can make error compression. The function was implemented at two random signals and it compacted nearly 10% of the error.

3.2.3 The Recovery function

```
Recover()
Begin
  Int I = 0;
  while(!recoveryQueue.empty())
  begin
        I = recoverQueue.deQueuue();
        For(int zz = lastWrittenOrder ;
        zz < imageLength ; zz++  )
        Begin
        If(I == image[order[i]]) then
              Image[order[i]] = Image[order[j]]>>1<<1 | ( (~I)
              <<31>>>31)
        end
     end
end
```

4 Algorithm Qualities Analytical and Statistical Analysis

4.1 Analytical Algorithm Capacity Bound Calculation

As we can see in the embed function we exclude only the Ac component and the 0 and 1 elements, (0 because writing in it will heart the image very much visually and 1 not to be confused to 0 when embedding with a 0 message bit) so the capacity of an Image I would be:

C = (H * W * 3) - (H * W * 3 / 64) - count (0) - count (1)

This insertion rate is the same insertion rate as compared to the OutGuess algorithm without recovery capabilities, while the F5 algorithm capacity is described by another equation, F5 use not the same insertion procedure and thus might have different insertion rate for the same vessel.

This insertion rate is the size of the whole modifiable area. For the recovery capability to work properly some of the file size need to be allocate as a recovery deposit.

4.2 Statistical Algorithm Capacity Bound Calculation

We had run our algorithm over a wide verity of natural images (1000 images) downloaded from the public domain photos website .

After analyzing the data gathered we had achieved a mean Insertion rate capacity of about 9.64% with a standard deviation of 1.087. Maximum insertion ratio was about 13.2%, minimum insertion ratio was about 7.062%.

This insertion rate is computed before the file compression and optional crypto. Insertion rate is the same as the F5 algorithm Insertion rate famously claimed by his owner to have the largest

insertion rate in the DCT mutation algorithms category.[8 & 7]

Though Insertion rate is not comparable to LSB or Filtered LSB embedding algorithms (insertion rates of 20% – 50 %) the higher undetectably of our algorithm justify the difference in insertion rates.

5 Other Built in Qualities

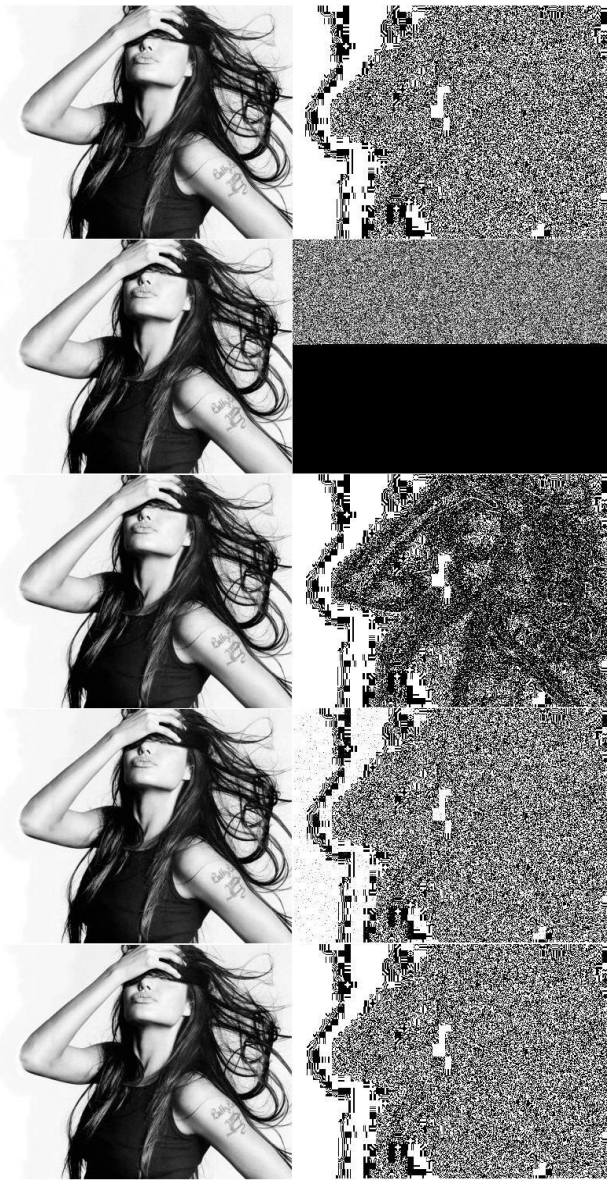
We had used a built in compression utility that uses an open source utility for ZIP file compression. We had add the ability to crypto the message before embedding, we used the java virtual machine built in algorithm DES 128 bit with fixed salting.

Although those utilities weren't essential for our algorithm working conditions but they are considered an important plus by most of the field researchers. Implementing a wider selection of crypto algorithms might be a good practice though not essential for coming upgrades.

6 Attacks Immunity

6.1 Visual Attacks Against Mirage

Visual attacks against Mirage such as PoV statistical attack and filter based visual attack aren't effective. The following figure is an example of the filter attack. The PoV attack is an automated version of the visual filter attack.



1	2
3	4
5	6
7	8
9	10

1 Original Picture **3** Invisible Secrets 2.1 Stego tool **5** Digital Invisible Ink toolKit 1.5 **7** S-tool Steganographic tool **9** Mirage, Even numbers are filters of the adjacent photo.

It can be seen clearly how the image attacks are working on 2 & 3 but not against 1 & 4 & 5, since the first is not a steganographic image and the later are frequency domain steganography algorithms.

6.2 Statistical Attacks Against Mirage

We have run the steg-detect utility over a Unix Ubuntu machine at a variety of 1000 natural images downloaded from public domain photos website.

Images were resized by scripting to the same size (400*640) to minimize the effect of size variation between different images, we used maximum embedding rate for the images.

A test on the 1000 images was automated by means of scripting over two copies of the images one with steganographic contents and another without such content, detection data was redirected to text files for analysis.

Analysis of the detection data revealed a suspecting rate of about 1.08% for both the two images categories, this rate falls within the false rate category of the test utility (nearly 1%) [9].

Further more the detection revealed that the detection result of the different images files were the same after and before embedding (if the file was suspected after embedding it was suspected before as well, and if it was suspected before embedding it is granted to be suspected after embedding).

Compared to experimental results conducted by Neils Provos here are the detection rates to a selection of other algorithms.

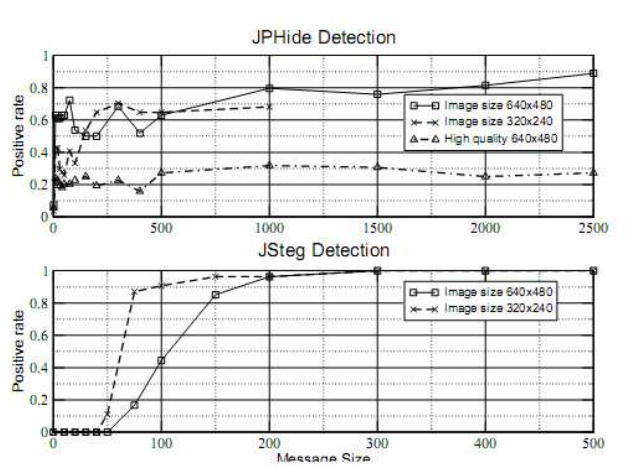


Figure 10: JSteg and JPHide detection for different test images and message sizes.

6.3 AI attacks Against Mirage

We have contacted Dr. Hany Farid via e-mail and he provided us with a copy of the source code of his project implemented in Mathlab.

Sadly Dr. Friad's machine need a supervised training over a huge number of images (originally Dr. Farid trained his machine over 40,000 images) this high computational requirement that is not available for us made us abounded the idea.

We have not however claimed our algorithm to be immune to such attacks.

7 Conclusions

Mirage is steganography algorithms immune to a number of steganalysis attacks amongst are:

- Filter attacks
- PoV attacks
- Chi-square statistical test
- Extended Chi-square Statistical test

Data rates of about 10% and up to 13% were found to be achieved through the

algorithm. Mirage algorithm is not a deductively proven total secure algorithm. In fact a very close review of the history of steganalysis and new studies reveal the size of threats (although none is specially implemented to Mirage) making maintaining the algorithm security an enormous task to undergo.

However this should not under estimate the potentials of our research and algorithm, no implemented or proposed algorithm in our knowledge is considered a total secured steganography algorithm, and our algorithm was built over all the proper foundations of data security we have knowledge in.

The war between steganography and steganalysis is not over. Steganalysis has succeeded in winning a number of points in that it has detected a number of Steganography algorithms.

Even at algorithms failure the use of random data distributions, data morphing, and even other cryptograph algorithms make such algorithms a considerably very high secure cryptography algorithms. Search for better steganography algorithm is constantly motivated by the high activity on the field of steganalysis, though the other might have a competitive advantage in financing and recruited experts.

8 Acknowledgments

We would like to thank our project supervisor Mr. Iyad Jaber for support, patience, and supervision. Dr. Wael Hashlamoon for devoting a lot of his dear time in explaining the statistical attacks, Muhana Manasrah,Mureed Al-Alem, and Muhmmoud Fathi for close attention, reviews and suggestion for our project and paper.

9 References

[1] Wikipedia contributors. Steganography [Internet]. Wikipedia, The Free Encyclopedia; 2008 Jan 15

[2] Ramarathnam Venkatesan et al, A Graph theoretic Approach to Software Watermarking, Fourth Information Hiding Workshop, March 23, 2000.

[3] C. Cachin, An Information-Theoretic Model for Steganography, Cryptology ePrint Archive, Report 2000/028, 2002

[4] A. Kerckhoffs, iLa Cryptographie Militaire (Military Cryptography), Sciences Militaires (J. Military Science, in French), Feb. 1883.

[5] Eiji Kawaguchi and Richard O. Eason, Principle and applications of BPCS-Steganography, University of Maine and Kyushu Institute of Technology, 2000

[6] A. Westfeld and A. Pfitzmann, Attacks on Steganographic Systems, Proc. Information Hidingo3rd Intil Workshop, Springer Verlag, 1999, pp. 61-76.

[7] Niels Provos, OutGuess - Universal Steganography, outguess.org, August 1998

[8] A. Westfeld, iF5 Steganographic Algorithm: High Capacity Despite Better Steganalysis, Proc. 4th Intil Work shop Information Hiding, Springer-Verlag, 2001, pp 289-302.

[9] N. Provos and P. Honeyman, Detecting Steganographic Content on the Internet, Proc. 2002 Network and Distributed System Security Symp., Internet Soc., 2002.

[10] H. Farid, DETECTING HIDDENMESSAGESUSING HIGHER-ORDERSTATISTICAL MODELS, DartmouthCollege Hanover, 2002

[11] S. Lyu and H. Farid, Detecting Hidden Messages Using Higher-Order statistics and Support Vector Machines, Proc. 5th Intil Workshop on Information Hiding, Springer-Verlag, 2002.