Sudan University of Science and Technology

Collage of Computer Science and Information Technology

Secure Data on HTML Web Page using Steganography with Encryption and Compression Technique

تأمين البيانات على صفحة الويب بإستخدام تقنية الإخفاء مع التشفير و الضغط

A Thesis Submitted in Partial Fulfillment of the Requirements of Master Degree in Computer Science

Prepared by:

Amel Elamin Elsheikh Elamin

Supervisor:

Dr. Faisal Mohammed Abdallah Ali

December 2019

Sudan University of Science and Technology

Collage of Computer Science and Information Technology

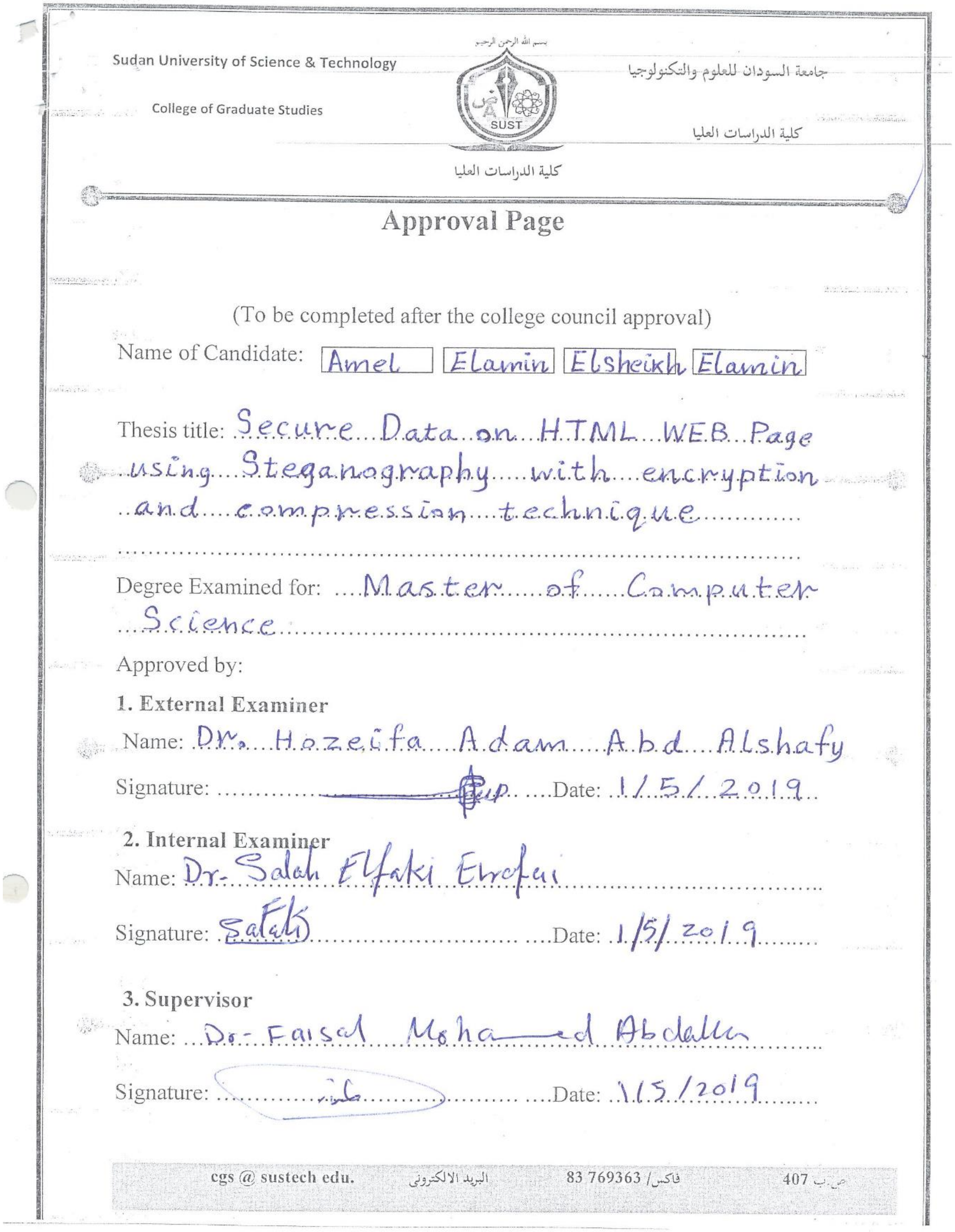
Secure Data on HTML Web Page using Steganography with Encryption and Compression Technique

تأمين البيانات على صفحة الويب بإستخدام تقنية الإخفاء مع التشفير و الضغط

A Thesis Submitted in Partial Fulfillment of the Requirements of Master Degree in Computer Science

Prepared by:	**Supervisor:**
Amel Elamin Elsheikh Elamin	Dr. Faisal Mohammed Abdallah Ali

December 2019



بسم الله الرحمن الرحيم

Sudan University of Science & Technology — جامعة السودان للعلوم والتكنولوجيا

College of Graduate Studies — كلية الدراسات العليا

SUST

كلية الدراسات العليا

Approval Page

(To be completed after the college council approval)

Name of Candidate: Amel | Elamin | Elsheikh | Elamin

Thesis title: Secure Data on HTML WEB Page using Steganography with encryption and compression technique

Degree Examined for: Master of Computer Science

Approved by:

1. External Examiner

Name: Dr. Hozeifa Adam Abd Alshafy

Signature: .... Date: 1/5/2019

2. Internal Examiner

Name: Dr- Salah Elfaki Elrofai

Signature: Salah .... Date: 1/5/2019

3. Supervisor

Name: Dr- Faisal Mohamed Abdalla

Signature: .... Date: 1/5/2019

cgs @ sustech edu. البريد الالكتروني 83 769363 / فاكس ص.ب 407

December 2019

الآيـــة

بسم الله الرحمن الرحيم

قَالُواْ سُبۡحَٰنَكَ لَا عِلۡمَ لَنَآ إِلَّا مَا عَلَّمۡتَنَآۖ إِنَّكَ أَنتَ ٱلۡعَلِيمُ ٱلۡحَكِيمُ ٣٢

صدق الله العظيم

DEDICATION

To my beloved Mother, you are all my life

To my Sibling.

To everyone who is close to me has given me support

To my Supervisor.

ACKNOWLEDGEMENT

I would like to thank my family for the great support.

Thanks to all my teachers especially my supervisor for teaching me how to be accurate in all the things that I do.

Thanks to Iyman Yousif, she helped me with the scholarship.

Thanks to all my friends especially (Rayan Alzain, Hind Mohammed, Marwa Hashim, Hind Adaln, and Mazaher Yousif) for the unconditional support along the way.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDIX

ABSTRACT

Many methods are used to securely transfer data, but the greatest challenge is to be careful of being detected by the intended recipient. The most famous methods are encryption and steganography. Encryption method draws the attention of third parties, but Steganography is a method to hide data inside a carrier such as Image, Audio, Video and Text without drawing any suspicion.

In this thesis, the data transfer as a text medium and in more accurate HTML file was used. This method is difficult type of the steganography because text files have a small amount of redundant data to hide a secret message and the structure of the document is observe. So to make detection more difficult both encryption and steganography are used with compression in addition to them, called Secure Hiding Data on HTML Web page using Stego-Crypto-Compression Schema.

The Schema has been applied in three levels, level one has been applied using encryption by ElGamal Elliptic Curve Cryptosystem, the input is a secret message and the output is a decrypt message. Level two is compression the decrypt message to generate compress message in format 0 and 1. In level three do steganography to hide the compress message in HTML file as a comment for tags.

The analysis shows that the model has a good result in carried the information, and increase measure of security.

المستخلص

تم استخدام العديد من الطرق لنقل البيانات بشكل آمن ، ولكن التحدي الأكبر هو أن يتم الكشف عنها فقط من قبل المتلقي المقصود. أشهر الطرق هي التشفير وإخفاء المعلومات. تلفت طريقة التشفير انتباه الأطراف الثالثة ولكن إخفاء المعلومات هي طريقة لإخفاء البيانات داخل ناقل مثل الصورة والصوت والفيديو والنص دون أي شك.

في هذا البحث ، تم استخدام وسيط نصي لنقل البيانات وأكثر دقة في ملفHTML . هذه الطريقة هي طريقة صعبة من طرق إخفاء المعلومات لأن الملفات النصية تحتوي على كمية صغيرة من البيانات المكررة لإخفاء رسالة سرية وبنية المستند تكون واضحة ، ولذلك ولجعل الكشف عنها أكثر صعوبة تم استخدام مزيج من الاثنين معا التشفير وإخفاء المعلومات وضغط البيانات بالإضافة إليهم، بما يسمى اخفاء البيانات في صفحة ويب HTML بتنفيذ المخطط Stego- Crypto- Compression.

تم تطبيق المخطط على ثلاثة مستويات ، المستوى الأول تم تطبيقه باستخدام التشفير بواسطة ElGamal Elliptic Curve Cryptosystem، والمدخلات هي الرسالة السرية و المخرج عبارة عن رسالة مشفرة. المستوى الثاني هو ضغط الرسالة المشفرة ويتم توليد رسالة ضغط بالتنسيق 0 و 1. في المستوى الثالث إخفاء المعلومات ، حيث يتم إخفاء الرسالة المضغوطة في ملفHTML كتعليق بالنسبة tags .

يظهر التحليل أن النموذج له نتيجة جيدة في نقل المعلومات ، وزيادة قياس الأمن.

CHAPTER I

Introduction

1. 1 Overview

Information hiding is a technique of covering the sensitive information within normal information. This creates a hidden communication channel between the sender and receiver such that the existence of the channel is unnoticeable (Kumar et al, 2013).

Steganography and cryptography are fields related to information hiding. The difference between these two methods is in terms of how to protect information's.

Cryptography is used to encrypt information based on some mathematical formulas. It is widely used to protect information exchanged over the Internet. World Wide Web (WWW) and e-mail are both public channels for transferring information. Meanwhile cryptography protects data by altering information into a form that unreadable or cannot understood by unauthorized people (Zaidan et al, 2009).

Steganography is used to disguising the information on the other media so that people do not feel the existence of such information's behind (Aboalsamh et al, 2008). But, sometimes steganography used in combination with cryptography that offer privacy and security are higher through the communication channel (Por et al, 2008).

1.2 Research Important

The importance of data hiding techniques comes from the fact the there is no reliability over the medium through which the information is send, in other words the medium is not secured. So, some methods are needed so that it becomes difficult for third party to extract the information from the message.

1.3 Problem Statement

Use Hyper Text web page (HTML) files as a carrier for text steganography systems is easy to detect and had very simple and weak approaches to conceal data. In addition we adhere to a certain size of the message that been hidden (in text) since any increase would have the risk of changing the shape of the page. Also text steganography had many constraints such as the languages, grammars and others.

The most commonly steganography algorithms used in HTML web page which is the tags and their attributes method is proved to needs the HTML web page must be larger with more attributes. For this reason the proposed design new algorithm and modified more levels secure.

1.4 Research Hypotheses

- Using compression algorithm will gives increase of storage capacity.
- Using cryptography algorithm will increase security.

1.5 Research Objective

The aim of this research is to develop a tool by use a new method that increases the storage space to accommodate the secret message while ensuring that the quality of the carrier is not affected**.**

The objectives of research are the following:

- Ensure that the message is read by the persons concerned only when exchanging messages
- Keep others away by ensure they do not suspect the existence of anything unusual is going on.
- Extract the message safely from the stego-object.

1.6 Research Methodology

First of all I will read the literature review and related works and applying deep study of steganography in text in order to determine the strength and weakness of each algorithm. Then I will design a new algorithm to develop a system protect the data and cover the common weakness and code it, first encrypt it, in addition the compression of encrypted text done after the first step. The output of those operations will be used as input to the third hiding operation in the HTML file. The extraction operations work as an opposite way.

1.7 Research Boundaries

The scope of the research is implementation of steganography tools for hiding information includes any HTML file and save as the same file path. Methodology: User needs to run the application. The user has two tab options – hide and unhide. The hiding of secret information (text) will be achieved by three levels of text steganography. The secret text massage used here is English language text. The file used are HTML file (web page). In ".htm and .html" extensions.

1.8 Research Contents

Chapter two contain two parts, part one represent a general techniques that associated with research, part two is the related studies that used text steganography. Chapter three explains the proposed algorithm that used in proposed methodology. Chapter four contains the project tool, implementation and result and finally Chapter five contains the conclusion and future work.

CHAPTER II

Literature Review

2.1 Introduction

This chapter shows some techniques of information security that associated with this research. These are the technique of steganography, cryptography and compression. Brief overview and the basic concepts which will be applied ending in HTML web page files. Then, it describes the related work of these techniques and the most important result that have been reached.

Many studies at the end of this chapter present how they applied some of these techniques and the most important result that have been reached.

2.2 Steganography

Steganography, originating from the Greek word “steganos” which means “covered” and “graphy” which means “writing or drawing”, is the art and science of hiding the existence of communication. The techniques used in steganography make it difficult to detect that there is a hidden message inside an innocent file. This way you not only hide the message itself, but also the fact that you are sending the message. This characteristic makes steganography the ideal science for hiding messages on the web, which is widely seen as a mass communication outlet (Patel).

Basic model of steganography the idea about steganography scheme in which first step is to embed original message in the carrier using any embedding technique then embedded message travel through the transmission media. At the receiver side receiver decodes the message which is the reverse process of embedding and gets the original message (Dhanani).

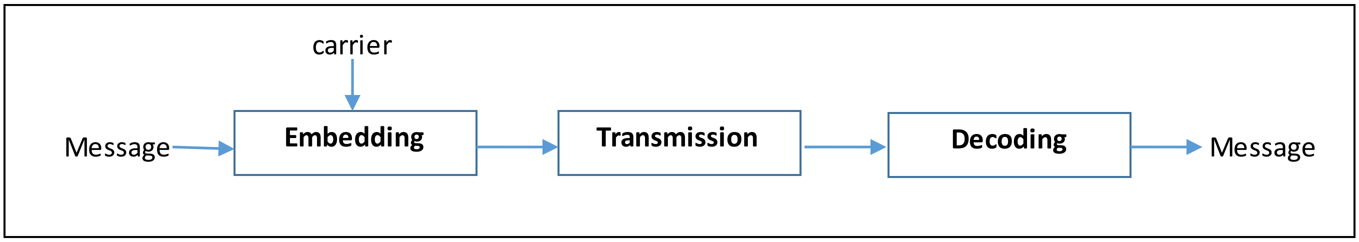


Figure 2.1 Basic model of Steganography (Dhanani)

In steganography, the possible cover carriers are innocent looking carriers (images, audio, video, text, or some other digitally representative code) which will hold the hidden information. A message is the information hidden and may be plaintext, cipher text, images, or anything that can be embedded into a bit stream. Together the cover carrier and the embedded message create a stego-carrier. Hiding information may require a stego key which is additional secret information, such as a password, required for embedding the information. For example, when a secret message is hidden within a cover image, the resulting product is a stego-image (Kumar, 2010).A possible formula of the process may be represented as:

Cover medium + embedded message + stego key = stego-medium

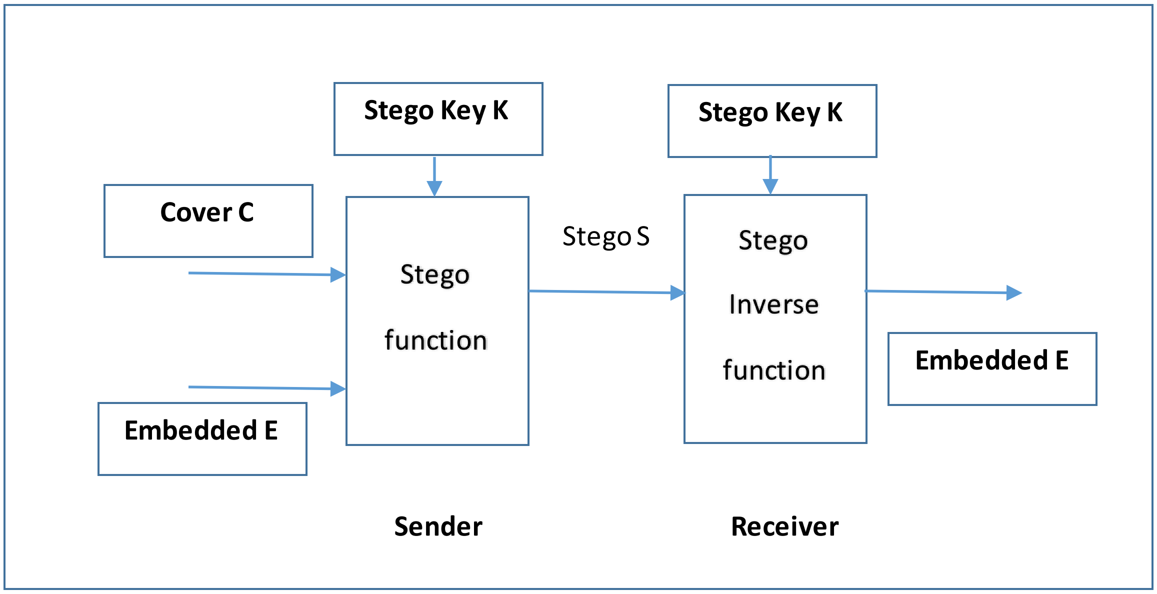


Figure 2.2 Steganography Mechanism (Kumar, 2010).

Steganography can be broadly classified into three types on the basis of the type of the cover media. Text steganography, image steganography, and audio steganography. A steganography technique that uses text as the cover media is called a text steganography. It is one of the most difficult types of the steganography technique. This is because text files have a very small amount of redundant data to hide a secret message (Garg, 2011).

2.2.1 Goal of Steganography

The primary goal of steganography is to hide a message inside another message in a way that avoids drawing suspicion to the transmission of the hidden message. If suspicion is raised, then the goal is defeated. Furthermore, actual detection of an embedded message renders the primary goal of steganography useless (**Patel**).

2.2.2 Types of Steganography

Steganography can be broadly classified into many types: **First Text Steganography** it consists of hiding information inside the text files. In this method, the secret data is hidden behind every nth letter of every words of text message. Numbers of methods are available for hiding data in text file. **Second Image Steganography h**iding the data by taking the cover object as image is referred as image steganography. In image steganography pixel intensities are used to hide the data. In digital steganography, images are widely used cover source because there are number of bits presents in digital representation of an image. **Third Audio Steganography i**t involves hiding data in audio files. This method hides the data in WAV, AU and MP3 sound files. There are different methods of audio steganography. **Forth Video Steganography i**t is a technique of hiding any kind of files or data into digital video format. In this case video (combination of pictures) is used as carrier for hiding the data. H.264, Mp4, MPEG, AVI are the formats used by video steganography. **Fifth Network or Protocol Steganography i**t involves hiding the information by taking the network protocol such as TCP, UDP, ICMP, IP etc., as cover object. . In the OSI layer network model there exist covert channels where steganography can be used (Kour et all, 2014)

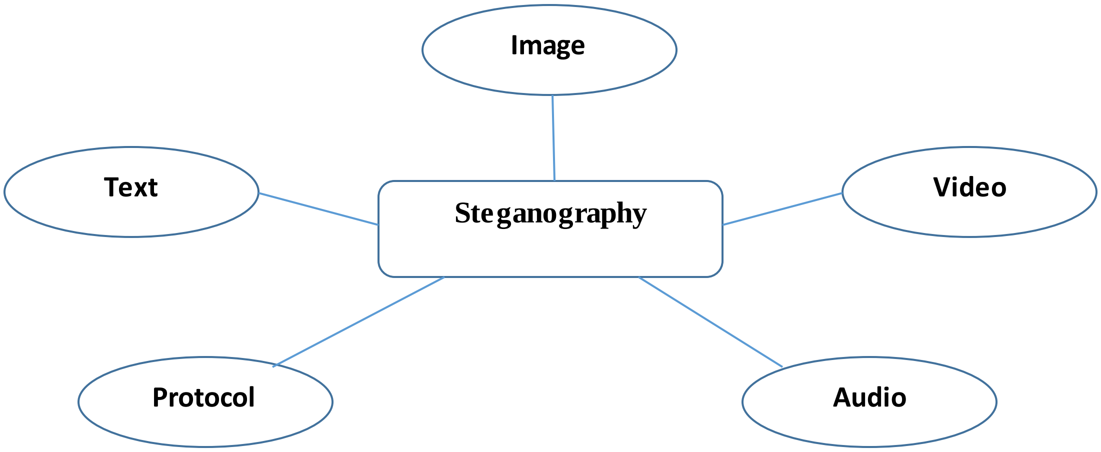


Figure 2.3 Types of Steganography (Kour et all, 2014)

2.2.3 The Steganography approaches

The steganography has mainly three approaches : **Pure steganography** This technique simply uses the steganography approach only without combination with other methods. It is working on hiding information within cover carrier. **Secret key steganography** The secret key steganography uses the combination of the secret key cryptography technique and the steganography approach. The idea of this type is to encrypt the secret message or data by secret key approach and the hide the encrypted data within cover carrier. **Public key steganography** The last type of steganography is to combine the public key cryptography approach and the steganography approach. The idea of this type is to encrypt the secret data using the public key approach and then hide the encrypted data within cover carrier. Further direction can be done by using small size of encrypted data to hide it within multimedia cover (Doğan, 2016).

2.2.4 Steganography Techniques

Hiding information in plain text can be done in many different ways. **Injection-based Technique t**his technique is known as ‘insertion’. It consists of injecting the secret message into the cover object. The secret message is hidden in an invisible part of the cover object. In other words, the data is embedded in areas that are ignored by the processing application. For instance, some cover-objects consist of end-of-file flags that tell the processing application to stop when reaching such flags. Therefore the secret message can be inserted after that flags.

The disadvantage of using this approach is that the stego-object size increases according to the amount of the embedded information. Therefore, the stego-object can be suspicious due to its large size once detected and compared to its original. **Substitution-based Technique** substitution-based technique is very popular. This technique consists of modifying the data of the cover-object and replacing it with the data of the secret message. For example messages can be hidden in byte of images by altering each bit. The substitution-based techniques gain is that the cover object size does not increase after embedding the hidden information. However, it may cause some distortion to the cover object. The degradation of the quality of the cover-object depends on the cover and algorithmic technique used. **Generation-based Technique** is different from the injection-based and substation-based techniques. It doesn't require to have a specific cover-object. It generates the cover-object for the reason of hiding the secret information based on some information structure. One benefit of this technique is that the stego-object and the original-object cannot be compared together. Thus we cannot discover the existence of hidden message in the stego-object due to the uniqueness of the carrier (Ashok, 2010).

2.3 Text Steganography

Text Steganography is the most difficult kind of steganography because a text file lacks a large scale redundancy of information in comparison to other digital medium like image, audio and video (Kour et al, 2014). The structure of the text document remain same throughout i.e. text document file is transparent during saving, written and retrieval phase. While embedding data in a text file, the main concern is its structure, which should not change.

Text steganography can be classified three basic categories: **First Format-based methods** use physical text formatting of text as a place in which to hide information. Generally, this method modifies existing text in order to hide the stenographic text. Insertion of spaces, deliberate misspellings distributed throughout the text, resizing the fonts are some of the many format-based methods being used in text steganography (Por et al, 2008). **Second Random and statistical generation** is generating cover text according to the statistical properties. This method is based on character sequences and words sequences. The hiding of information within character sequences is embedding the information to be appeared in random sequence of characters. This sequence must appear to be random to anyone who intercepts the message. A second approach to character generation is to take the statistical properties of word-length and letter frequency in order to create "words" (without lexical value) which will appear to have the same statistical properties as actual words in a given language. The hiding of information within word sequences, the actual dictionary items can be used to encode one or more bits of information per word using a codebook of mappings between lexical items and bit sequences, or words themselves can encode the hidden information (Por et al, 2008). **Third Linguistic method** which specifically considers the linguistic properties of generated and modified text, frequently uses linguistic structure as a place for hidden messages. In fact, stenographic data can be hidden within the syntactic structure itself .The problem in this approach is that replacement of synonyms may change the meaning or structure of the sentence (Por et al, 2008).

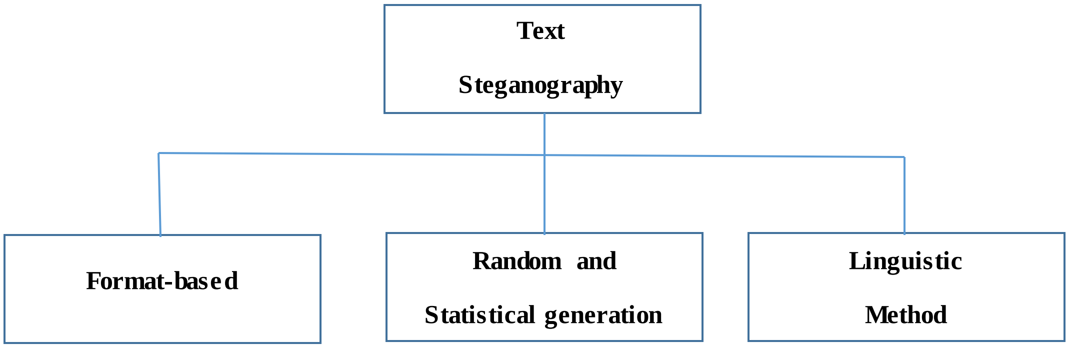


Figure 2.4 Three Basic Categories of Text Steganography (Por et al, 2008)

2.3.1 HIDING Text using Markup Language

Considering methods of hiding information in HTML documents, two main groups can be identified. The first group comprises techniques originating from the classical text steganography while the second group includes methods which make use of mark-up languages specific properties. The former group methods treat HTML documents as text files and consist in embedding secret in a file by changing its content in a particular way, depending on information one wants to carry.

2.3.1.1 Hyper Text Markup Language (HTML)

HTML (from HyperText Markup Language) is a Language that process not only plain text but also formatted data written. HTML is widely regarded as the standard publishing language of the World Wide Web.

HTML gives authors the means to publish online documents with headings, text, tables, lists, images, etc. Retrieve online information via hypertext links, design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc. Include spread-sheets, video clips, sound clips, and other applications directly in their documents.

Each HTML files (Web pages) must start with an HTML element (tag), that containing a HEAD element (tag) and then a BODY element (tag).

```
<HTML>
   <HEAD>
   <TITLE>A simple web page</TITLE>
   ... other head elements
  </HEAD>
  <BODY>
   ... document body
 </BODY>
</HTML>
```

2.4 Cryptography Overview

Steganography and Cryptography are two different approaches to ensure secure communication. Each approach has different purposes, aspects and limitations. As mentioned previous section the purpose of steganography is to hide the covert communication information, while the purpose of cryptography is to protect the contents of the secret information. Steganography and cryptography are related to two parties who wish to communicate securely to exchange some confidential information.

2.4.1 Cryptography

Is the process by which the data to be transmitted is hidden in a manner such that only the intended recipient can understand it. The initial data is called as plaintext and the encrypted data is called as cipher text. A key is used to hide the data .There are different types depending on the number and way in which the keys are used.

There are two types of cryptographic techniques: **Symmetric Key Cryptographic** actually the technique by which identical cryptographic keys are used for the purpose of both encryption and decryption. The receiver can get back original data by using the key. The symmetric key cryptography provides high data rates, usage as primitives to construct various cryptographic mechanisms and can be combined to produce stronger ciphers. The main fact here is that the security of data depends on the security of the key. So, care should be taken while exchanging keys between the sender and the receiver (Ushll et all, 2011). Symmetric cryptosystem have a problem of key transportation. The secret key is to be transmitted to the receiving system before the actual message is to be transmitted. Every means of electronic communication is insecure as it is impossible to guarantee that no one will able to tap communication channels. So the only secure way of exchanging keys would be exchanging personally. Symmetric cryptosystem can't provide digital signatures that can't be repudiated (Zhang et all, 2005). **Asymmetric key cryptographi**c the technique where two keys are used. One key is used to lock or encrypt the plaintext, and another to unlock or decrypt the cipher text. Neither key can do both the functions. One of these keys is published or made public and the other is kept private. This technique has comparatively slower data rate throughputs than the symmetric key technique (Ushll et all, 2011)..

2.4.2 Elliptic Curve Cryptography (ECC)

Elliptic Curve is an approach to public key cryptography based on the algebraic structure of elliptic curves over finite fields. With smaller key sizes and lower processing requirements than other public key cryptosystems, elliptic curve cryptography lends itself well to sending information securely over the internet where bandwidth and processing capabilities are limited. Ensuring the timely and reliable access to make use of information. ECC offers security with smaller key sizes, faster computation, lower power consumption as well as memory and bandwidth saving (Gajbhiye).

Elliptic curve cryptography is an asymmetric key cryptography. It includes public key, private key and set of operations associated with the keys to do cryptographic operations. Public key may be freely distributed where as private key is kept secret. The public key is used for Encryption, while the private or secret key is used for decryption. Some public key logarithms may require a set of predefined constants to be known by all the users taking part in communication. Domain parameters in ECC is an example of such constants (Jagdale et all, 2010).

The choice of the type of elliptic curve is dependent on its domain parameters, the finite field representation, elliptic curve algorithms for field arithmetic as well as elliptic curve arithmetic (Shankar, 2010). An elliptic curve in its “standard form” is described by:$y^2 = x^3 + \mathrm{ax} + \mathrm{b}$.A simple elliptic curve with points is shown in Figure 2.5.

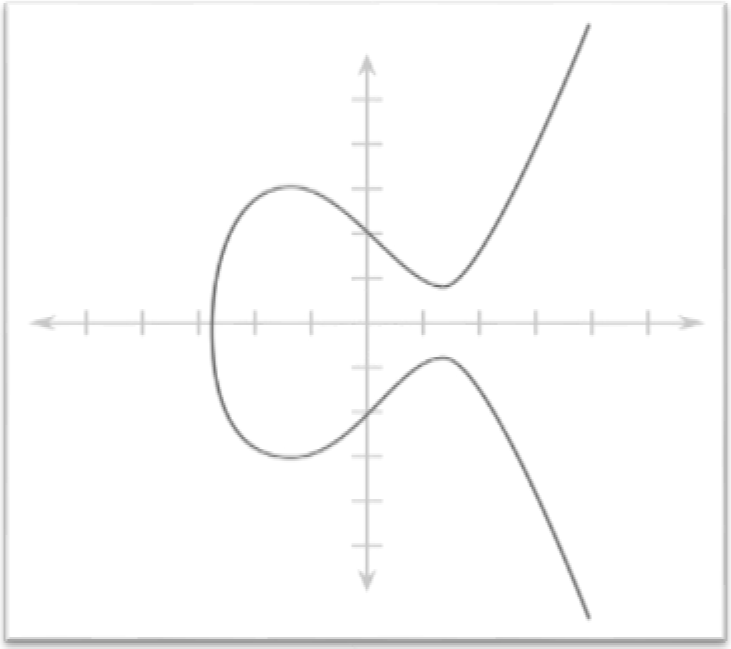


Figure 2.5 A simple elliptic curve (Shankar, 2010)

2.4.2.1 ECC STANDARDS

There are three immediate applications for ECC in cryptography, as it is described in this section. **Elliptic Curve Diffie-Hellman** The main objective of key exchange protocols is to put in contact two or more entities communicating through an open and insecure channel, sharing a secret key that will provide data confidentiality and integrity to any information exchanged using that channel.

ECDH denotes the generic key exchange scheme based on the Diffie-Hellman mechanism applied to elliptic curves (Malik). **Elliptic Curve Digital Signature Algorithm** ECDSA is the elliptic curve analogue of the Digital Signature Algorithm (DSA). Key pair in ECDSA is generated the same way as of that in ECDH (Malik). **Elliptic Curve Integrated Encryption Scheme t**he most extended encryption and decryption scheme based on ECC is the Elliptic Curve Integrated Encryption Scheme (ECIES). This scheme is a variant of the ElGamal scheme (Malik).

2.4.3 ElGamal Elliptic Curve Cryptosystem

2.4.3.1 Classic ElGamal Cryptosystem

ElGamal introduced a cryptosystem which depends on the Discrete Logarithm Problem. The ElGamal encryption system is an asymmetric key encryption algorithm for public key cryptography which is based on the Diffie-Hellman key exchange. ElGamal depends on the one way function, means that the encryption and decryption are done in separate functions.

The ElGamal solved the Diffie-Hellman key exchange algorithm by presenting a random exponent Type k. This exponent is a replacement for the private type of the receiving entity.

2.4.3.1.1 Mathematical Steps:

ElGamal encryption consists of three components: the key generator, the encryption algorithm, and the decryption algorithm.

2.4.3.2 Elliptic Curve ElGamal Cryptography

ElGamal Elliptic Curve Cryptography is a public key cryptography analogue of the ElGamal encryption schemes which uses Elliptic Curve Discrete Logarithm Problem.

2.5 Compression Overview

Data compression offers an attractive approach for reducing communication costs by using available bandwidth effectively. Compression algorithms reduce the redundancy in data representation to decrease the storage required for that data. Over the last decade there has been an unprecedented explosion in the amount of digital data transmitted via the Internet, representing text, images, video, sound, computer programs etc.

Data compression implies sending or storing a smaller number of bits. Compression is the 4reduction in size of data in order to save space or transmission time. Many methods are used for this purpose, in general these methods can be divided into two broad categories: Lossy and Lossless methods (Dhanani).Lossy compression means that some data is lost when it is decompressed. Lossy compression bases on the assumption that the current data files save more information than human beings can "perceive”. Thus the irrelevant data can be removed. Lossless compression means that when the data is decompressed, the result is a bit-for-bit perfect match with the original one. The name lossless means "no data is lost", the data is only saved more efficiently in its compressed state, but nothing of it is removed.

2.5.1 Huffman coding

Huffman coding is a popular method for compressing data with variable-length codes. Given a set of data symbols (an alphabet) and their frequencies of occurrence (or, equivalently, their probabilities), the method constructs a set of variable-length codewords with the shortest average length and assigns them to the symbols.

Huffman coding is a lossless data compression. It uses a variable-length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. It was developed by David A. Huffman (Ushll et all, 2011)..

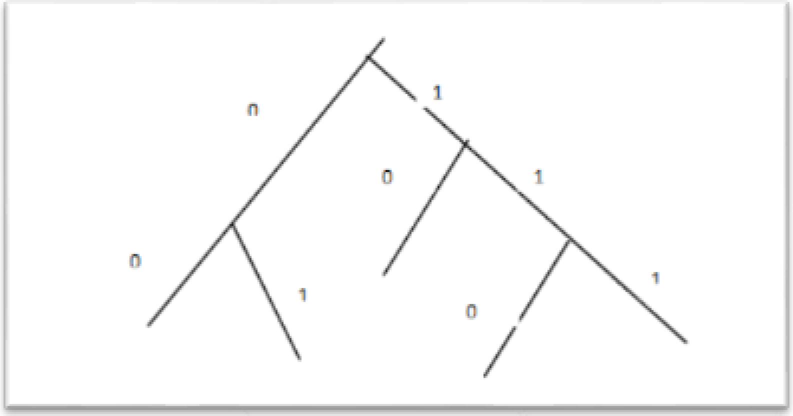


Figure 2.6 Huffman Coding Binary Tree (Ushll et all, 2011).

2.5.1.1 Huffman Encoding

The Huffman encoding algorithm starts by constructing a list of all the alphabet symbols in descending order of their probabilities. It then constructs, from the bottom up, a binary tree with a symbol at every leaf. This is done in steps, where at each step two symbols with the smallest probabilities are selected, added to the top of the partial tree, deleted from the list, and replaced with an auxiliary symbol representing the two original symbols. When the list is reduced to just one auxiliary symbol (representing the entire alphabet), the tree is complete. The tree is then traversed to determine the codewords of the symbols.

2.5.1.2 Huffman Decoding

Before starting the compression of a data file, the compressor (encoder) has to determine the codes. It does that based on the probabilities (or frequencies of occurrence) of the symbols. The probabilities or frequencies have to be written, as side information, on the output, so that any Huffman decompress or (decoder) will be able to decompress the data.

The algorithm for decoding is simple. Start at the root and read the first bit off the input (the compressed file). If it is zero, follow the bottom edge of the tree; if it is one, follow the top edge. Read the next bit and move another edge toward the leaves of the tree. When the decoder arrives at a leaf, it finds there the original, uncompressed symbol (normally its ASCII code), and that code is emitted by the decoder. The process starts again at the root with the next bit.

2.6 Related Works

Premingh, Rajat Chaudhary and Ambika Agarwal (Singh et all, 2012) proposed paper in hiding information method within spaces between words, this seems hardly to know about the existence of the hidden bits, but requires a great deal of spaces to encode few bits.

This thesis shows that one space is interpreted as “0” whereas two spaces are interpreted as “1”. This embedding scheme was applied in the space which appears between the words.

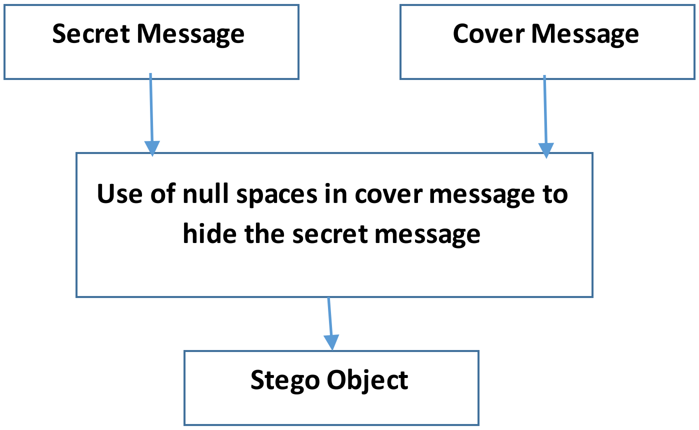

Secret Message
Cover Message
Use of null spaces in cover message to hide the secret message
Stego Object

Figure 2.7 Broad level steps in Text Steganography

Garg, M paper (Garg, 2011).proposed paper uses the html tag attributes to hide the secret messages after encrypted by using playfair cipher encryption algorithm. The technique has key file generated by scanning of the html documents, contain a table from two types of attributes , represented in two columns as Primary Attribute and Secondary Attribute. The format of key file is shown in table 2.1. The hiding of a bit is determined by the order of the attributes combination . If primary attribute is followed by a secondary attribute, it can hide a bit 1; else it can hide a bit 0.The secrecy of the hiding process to hide secret messages is high.

Table 2.1 Key File Format

First Attribute (Primary)	Second Attribute (Secondary)

Karan H. Parmar and AvanibaParmar paper (Karan et all, 2015) use Hybrid techniques using in HTML file, by combination multiple methods (whitespace between attributes, using attributes quotation and upper case on tags) together to hide a secret message. Largest embedded capacity and security is increasing.

Fatma Abdalla Mabrouk Kheiralla and DrMudawi Mukhtar Elmusharaf paper (Kheiralla, Elmusharaf, 2016) use Generic algorithm, concepts is apply in HTML file by considered that any tag represents gene and an attribute represents chromosome. A relation table have been created, it consists of a primary data of two columns, and each row consists of two chromosomes represent gene. The gene contains a set of properties. Hexadecimal encode have been chosen. Encoding chromosome are represented using Hexadecimal numbers (0-9, A-F) so the gene can hide 8 bits on the project, each chromosome in the row is supposed to hide 4 bits. The relation table stores the primary tag, from which it is supposed to start and then make a random search, the algorithm method examines each chromosome of each HTML (attribute), to examine the existence of the chromosome in the primary field of the key file. If the chromosome exists in the primary field, the algorithm will search its corresponding secondary chromosome in the corresponding HTML tag, if it found secondary chromosome, then this combination of chromosomes will be used to hide the bit. If not, then the algorithm will skip this chromosome. Hiding of a bit is determined by the order of the attributes in the attribute combination. If the primary attribute is followed by a secondary attribute, it can hide bit „1‟ in hexadecimal number, if not it can hide bit „0‟ in hexadecimal number. Largest embedded capacity and security is increasing.

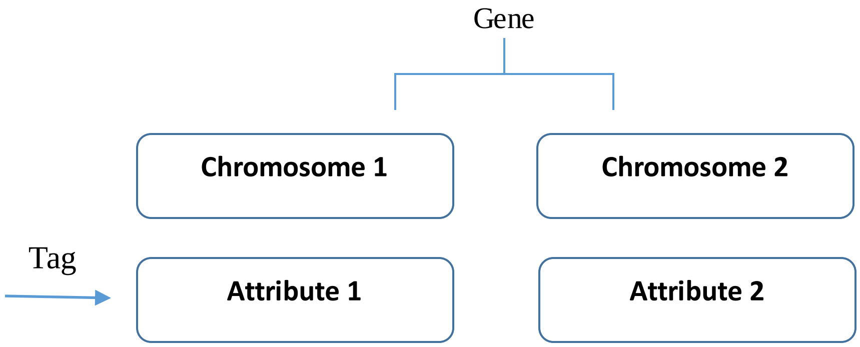


Figure 2.8 Tag characteristic technique

Chintan Dhanani and Krunal Panchal Embedding paper (Dhanani et al) deal with a secret data in multiple HTML web pages by generating a table having acolumns (page_name, page_link, page status, page_visiting_no). Hide the data in the page until median href tag encountered, then transfer control to page having relative link= median href tag. And before encrypt secret data using RSA algorithm. This method will increase Largest Embedded Capacity (LEC) & security of data.

Table 2.2: The summarization of related work

Paper Name	Author	Technique	Result	Open issues
A Novel Approach of Text Steganography based on null spaces	Premingh, Rajat Chaudhary and Ambika Agarwal	Uses the spaces in plaintext between words appears in HTML web page to hide the secret message , by represent one space as "0" while two spaces as "1" .	Hardly to know about the existence of the hidden bits.	Requires a great deal of spaces to encode few bits.
A novel text steganography technique based on html documents	Garg, M	Uses the html tag attributes to hide the secret messages after encrypted by using playfaircipher encryption algorithm. The technique has key file generated by scanning of the html documents, contain a table from two types of attributes , represented in two columns as Primary Attribute and Secondary Attribute .The hiding of a bit is determined by the order of the attributes combination . If primary attribute is followed by a secondary attribute, it can hide a bit 1; else it can hide a bit 0.	High secure in hiding process.	Requires a great number of attributes because it deals with a pair of attributes to represent just one bit.
Web Based Steganography Using Combination of Three Methods	Karan H. Parmar, AvanibaParmar	Hybrid techniques using in HTML file, by combination multiple methods (whitespace between attributes, using attributes quotation and upper case on tags) together to hide a secret message.	Largest embedded Capacity and High security.	Upper case on tags generate not valid file for W3C and XHTML, and It is noticeable.

Data Hiding On Web Page using Steganography by Genetic Algorithm	Fatma Abdalla Mabrouk Kheiralla , DrMudawi Mukhtar Elmusharaf	Generic algorithm concepts is apply in HTML file by considered that any tag represents gene and an attribute represents chromosome.a relation table have been created, it consists of a primary data of two columns, each row consists of two chromosomes represent gene.The gene contains a set of properties. Hexadecimal encode have been chosen.encoding chromosome are represented using Hexadecimal numbers (0-9, A-F) so the gene can hide 8 bits on the project, each chromosome in the row is supposed to hide 4 bits.The relation table stores the primary tag, from which it is supposed to start and then make a random search, the algorithm method examines each chromosome of each HTML (attribute), to examine the existence of the chromosome in the primary field of the key file. If the chromosome exists in the primary field, the algorithm will search its corresponding secondary chromosome in the corresponding HTML tag, if it found secondary chromosome, then this combination of chromosomes will be used to hide the bit. If not, then the algorithm will skip this chromosome. Hiding of a bit is determined by the order of the attributes in the attribute combination. If the primary attribute is followed by a secondary attribute, it can hide bit “1‟ in hexadecimal number, if not it can hide bit „0‟ in hexadecimal number.	experimental results showed that this approach works, achieving effective optimization, security, and robustness.	It is consist a complicated way for hide and extract the secret data.

HTML Steganography using Relative links & Multi web-page Embedment	ChintanDhanani, Krunal Panchal	Embedding secret data in multiple HTML web page by generating a table having acolumns (page_name, page_link, page status, page_visiting_no). Hide the data in the page until median href tag encountered, then transfer control to page having relative link= median href tag. And before encrypt secret data using RSA algorithm.	Gives high LEC & strong security in compare to other methods.	Complicate method on Data embedding and extracting because data will divided in multiple HTML web page.

CHAPTER III

Methodology

3.1 Introduction

This chapter describes the new model that is proposed to secure hiding data focuses on hiding text (the Secret-Message) into another HTML text web page (the Cover-text), using Stego-Crypto-Compression schema.

The model is divided into two major side the sender side that deals with the embedding processes of the secret message, and the receiver side that deals with the extraction processes.

The proposed method utilizes the html tags, white space to hide the secret message. It embeds in a way that the properties in the html tags have no effect on the content and therefore do not raise the suspicion of the site visitors. These properties of html tags can be utilized to hide the secret messages effectively.

3.2 Proposed Method

The proposed method is using multilevel text steganography (three levels). In level one encrypt the secret message (text), while level two compress encrypted text, and level three is stego text (Sender Side) and decompress-decrypt (Receiver Side) will be apply to the Stego-text. The output from level one is encrypt text will be converted into binary text by level two compress text which in turn will work as input in level three for conceals text in another text.

Figure 3.1 below shows the conceptual framework for the algorithm approach to text steganography. It demonstrates the flow of the secret message from encryption, to compression until finally reaches to stego system on the cover text to produces a stego text.

3.2.1 Steganography

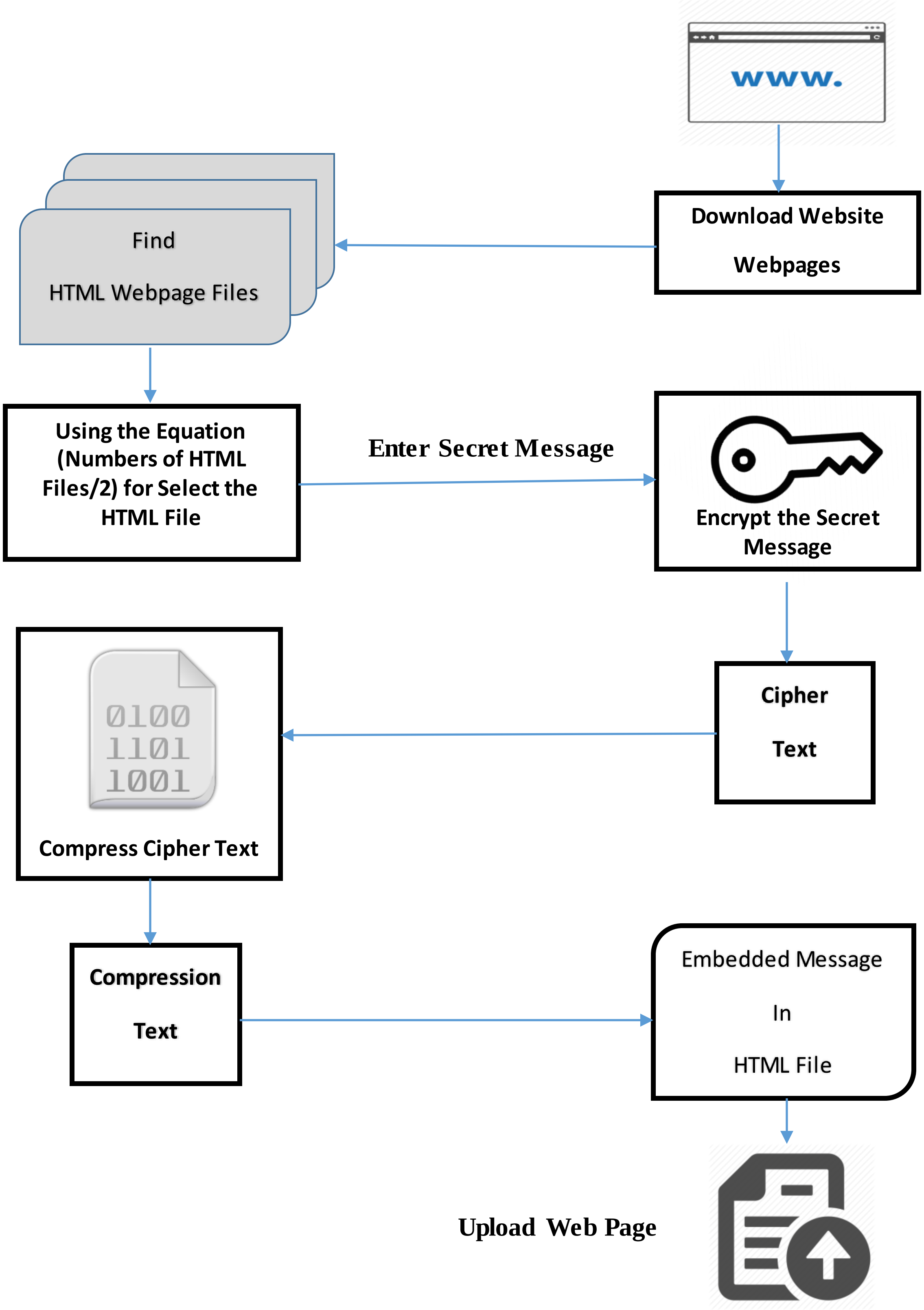


Figure 3.1 Model for Steganography

3.3 System Analysis

Three levels showing in figure 3.2.

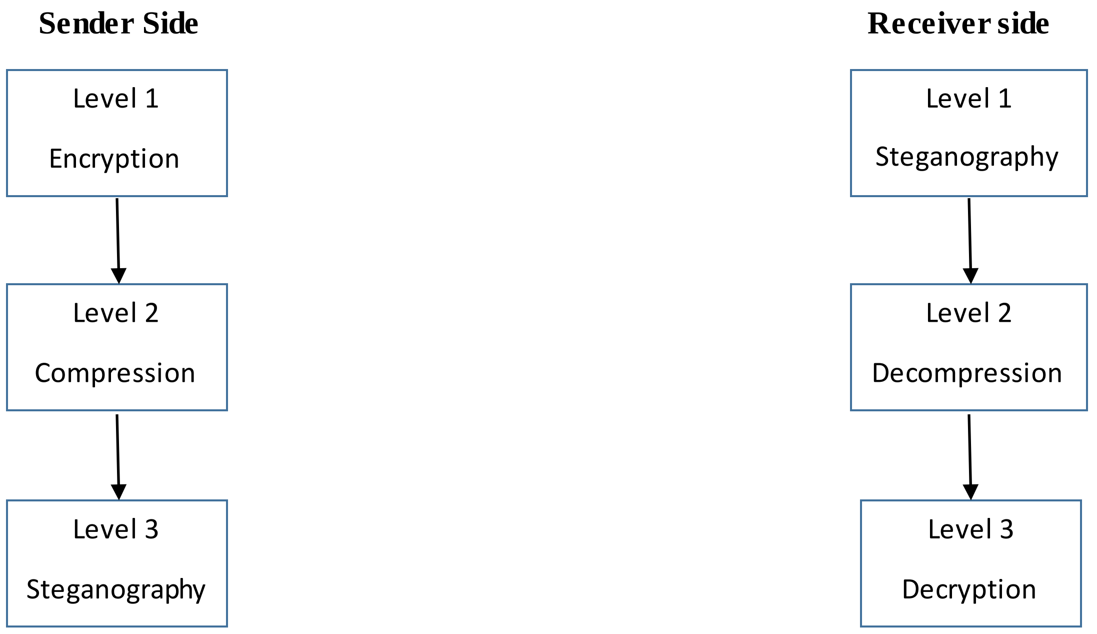


Figure 3.2 Stego-Crypto-Compression schema (Sender and Receiver side)

3.3.1 The Comments Hide Algorithm

The main function of the third level or level three is hide, the message or the compressed encrypted-text hide into an HTML text (it considered the final step) in the sender site, and the receiver extracts the compressed Encrypted-text from the HTML file (the first step) in the receiver site.

At the (Sender Site) receives the message entered from the second level and hide it in HTML file. At the (Receiver Site) the hide step is the first level, the output for this level is the input for decompression second level.

3.3.1.1 Scenario

The information to be hidden was divided into two parts. The first represents the keys used to decompress the message, and was stored as a comment represented by coordinated stars to produce a definition similar to what is added in the code as a definition of the programmer. The second contains the secret message after encrypted and compressed.

All previous data is represented in a HTML file chosen later by comment and is considered one of the most important components of a HTML file. One of its advantages is that they does not appear in the web page for the visitor only when asked the source code, in addition it does not affecting them.

Input: Compressed-text, Cover

Process: Hide

Output: Stego-text

Pseudo code: Hide Process (Sender Side):

```
-   Receive HTML file input from construction level.
-   Read HTML file source code.
-   Reset HTML file from comments.
-   Receive header and compressed message from level two.
-   Gets all unique tags on HTML file body.
-   Prepare header as a comment from header symbols and their length and add comment
    before head tag.
-   Compressed message hide as a comments in body before tags extract previously.
-   It is done by translate zero "0" to one space and one "1" to two spaces.
-   If compress message finish add three spaces as sign for ending.
-   Return new source code.
-   Write source code to HTML file.
```

Output: Compressed-text

Process: Unhide

Input: Stego-text, Cover

Pseudo code: Unhide Process (Receiver Side):

```
- Receive HTML file input from construction level.
- Read HTML file source code.
- Extract comments from source code.
- First comment using as key to uncompressing by read stars and calculate it by summation
  every stars parts to extract ASCII code and number of frequency.
- The reset of comments read spaces between every one comment.
- If one space it translate to “0” and two spaces to “1”.
- Stop extract if read three spaces.
- The text generated from “0” and “1” uncompressing using Huffman coding algorithm.
- The text result from previous step decryption using ElGamal Elliptic Curve algorithm.
- Original Message generate.
```

3.3.2 The Encryption

The first level at the (Sender Site) receives the message entered from the construction input level and encrypts it by ElGamal Elliptic Curve Cryptography, the encrypt message output from this step is ready to be used in the next compression step. At the (Receiver Site) the decryption step is the third level, the input for this level is the output from decompression second level. The output of decryption is the same in construction message input.

Input: Message

Output: Encrypt-text

Output: Message

Input: Decrypt-text

3.3.2.1 Encryption / Decryption:

A wants to send the message to B using elliptic curve ElGamal encryption. A chooses the elliptic curve.

$$y^2 = x^3 + \mathrm{ax} + \mathrm{b}$$

Choose the point G on the elliptic curve. A selects a private key “a‟ and generates the public key A=‟aG‟ and B selects a private key “b‟ and generates the public key B= “bG‟.

Algorithm (Sender Side):

Participant B encrypts a message m to A

- Obtain A’s authentic public key (p, g, g^a).
- Represent the message as integers m in the range {0, 1,… p-1}.
- Select a random integer k, 1 ≤ k ≤ p-2.
- Compute |Y= g^k mod p and S = m * $(g^a)^k$.
- Send cipher text c = (Y, S) to A.

Algorithm (Receiver Side):

Participant A receives encrypted message m from B

- Use private key a to compute (Y^{p-1-a}) mod p

  Note Y^{p-1-a} = Y^{-a} = Y^{-ak}

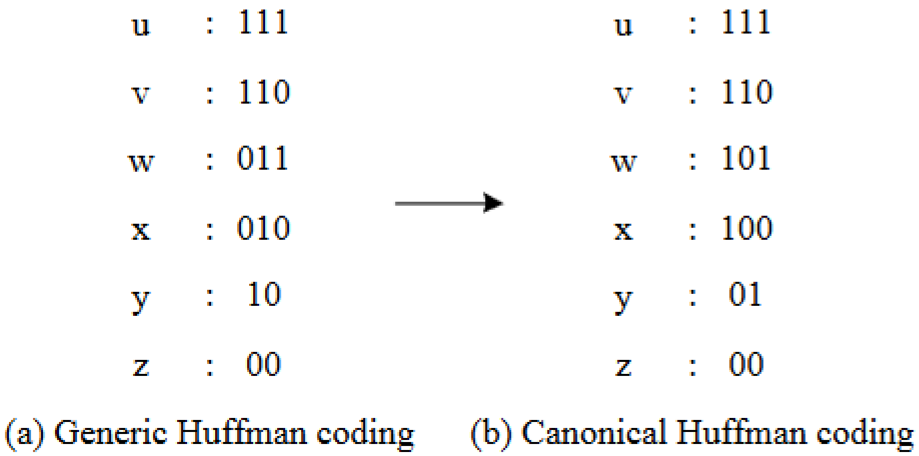
- Recover m by computing (Y^{-a}) * S mod p.

3.3.3 The Compression

The second level at the (Sender Site) receives the message output from the encryption first level and compress it by Huffman Coding using the Canonical Huffman Coding type, the compress message output from this step is ready to be used in the next step. At the (Receiver Site) the decompress step is the second level, the input for this level is the output from extract message from HTML file first level.

3.3.3.1 Canonical Huffman coding

Among many alternatives in Huffman coding, Canonical Huffman coding (CHC) is a particular type with unique properties which facilitate both encoding and decoding processes. Before generating the Canonical Huffman codewords, their length must be first calculated. This information is provided by the Huffman tree by counting the number of pointer jumps from the root to each leaf when traversing the Huffman tree (Arelakis, 2013). Then the value symbols are sorted in ascending order according to their code-length.



u	:	111	u	:	111
v	:	110	v	:	110
w	:	011	w	:	101
x	:	010	x	:	100
y	:	10	y	:	01
z	:	00	z	:	00

(a) Generic Huffman coding (b) Canonical Huffman coding

Figure 3.3 The transform from generic to Canonical Huffman code (Arelakis, 2013)

The codewords (CW) can be transferred to be Canonical Huffman code by the process is shown in Figure 2.7. The first value-symbol is z, its length is 2 so it's CW is “00” and The CW of y is

“01”. Then the CW length increases by 1, the FCW of length 3 is 100”, and it is assigned to x. Finally, w, v and u are assigned with “101”, “110” and “111”.

Input: Encrypt-text

Output: Compressed-text

Output: Decrypt-text

Input: Compressed-text

3.3.3.1 Compression / Decryption:

Algorithm (Sender Side):

- Build Huffman tree and get every character coding.

- Create a leaf node for each unique character and build a min heap of all leaf nodes (Min Heap is used as a priority queue. The value of frequency field is used to compare two nodes in min heap. Initially, the least frequent character is at root)
- Extract two nodes with the minimum frequency from the min heap.
- Create a new internal node with a frequency equal to the sum of the two nodes frequencies. Make the first extracted node as its left child and the other extracted node as its right child. Add this node to the min heap.

```
Compute the frequency f(c) of each character of c of X

Initialize a priority queue Q

For each character c of X:

  Create a single-node binary tree T sorting c.

  Insert T into Q with key f(c)

While Q.size>1 do

  Entry e1 = Q.removeMin () with e1 having key f1 and value T1

  Entry e2 = Q.removeMin () with e2 having key f1 and value T1
```

- Calculate the Canonical Huffman Coding for any character coding.
- Sort characters coding length ascending (if two characters coding is same length sort by character ascending).
- First character fill zero based on length of code.
- For the next character add one “1” to code if in same code length.
- Then add zero “0” to the code if the length of the code larger than previous.
- Encode any character in encrypt message with the corresponding code.
- Return header (symbol + code length) and compress message.

3.3.4 The Construction Level

The construction level specialized in inputs for two sides.

Algorithm (Sender Side):

- Secret Message

$$M \in (!\, arabic\ char)$$

- Browse to select web folder.

$$F_s \in WF$$

$$F_s \in (.html, .htm)$$

$$F_{position} \equiv \sum \text{№}\ F_s /2$$

$$\text{F} \equiv F_{position}$$

Algorithm (Receiver Side):

- Browse to select web folder.

$$F_s \in WF$$

$$F_s \in (.html, .htm)$$

$$F_{position} \equiv \sum \text{№}\ F_s /2$$

$$\text{F} \equiv F_{position}$$

CHAPTER IV

THE Implementation and Result

4.1 Introduction

In the previous chapter were described the structure of the thesis model and how the secret message passed between different levels. In the sender part it flows encryption-compression-steganography in the HTML text as a cover-object, and in the receiver part start with a cover-object, extract the message then returns again from those levels by reversing the order of the levels decompression-decryption.

This chapter shows the implementation of the model. Explain how the program works for both sender and receiver side, list the necessary steps and the results that have been reached will be included in this chapter as well.

4.2 The Implementation Tools

The previous model was executed by Python language version 3, under the Windows platform, using Jet Brains PyCharm Community Edition 2017 editor. The webpage is greed upon with both parties.

4.3 The Implementation

The most important properties of a cover medium is how that can be stored inside it without obfuscation the properties of the cover, security of information and amount of data.

The implementation of algorithm was done on the cover text; in this case, a HTML file was used. The generated cover text depends on the secret message. Once results have been achieved the embedding process begins, to output stego text. An extraction algorithm will be applied to reverse to the original secret message.

The tool was designed has two main interfaces, first one is specialized to sender part, and the second is for the receiver part. As shown in figures 4.1 and 4.2.

In the figure 4.1 the Sender screen the user must enter two inputs, first enter his secret message, the second input is the web directory by using the browsing button (Select web Folder).

After that the program will accept your inputs if they are legal, and go through all the levels that belong to the sender side (as it is shown in chapter three).

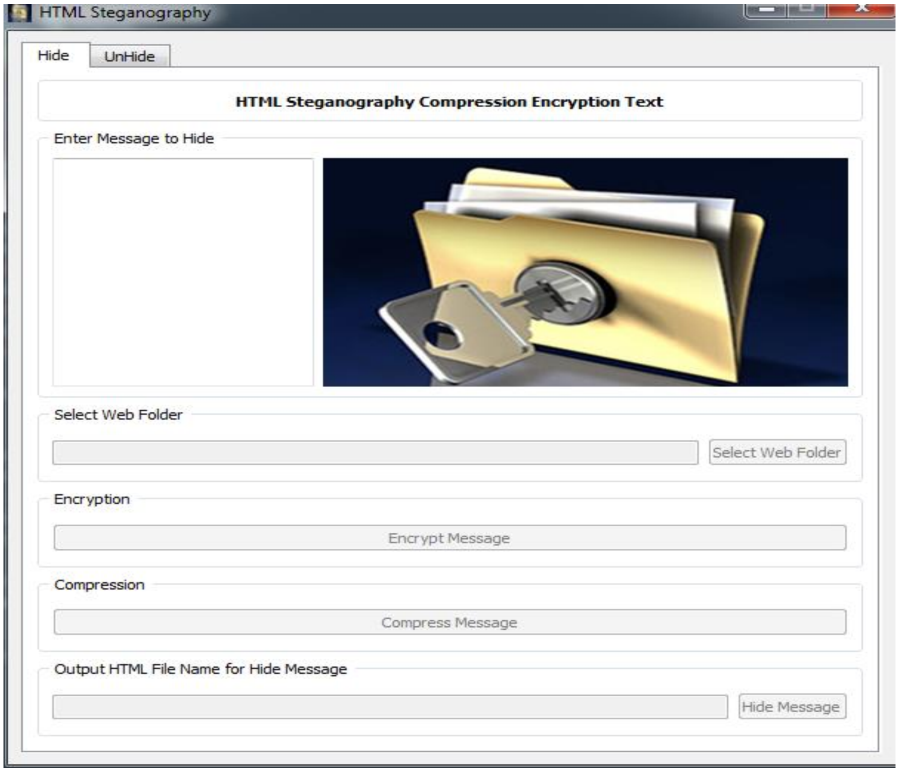


Figure 4.1 The sender part

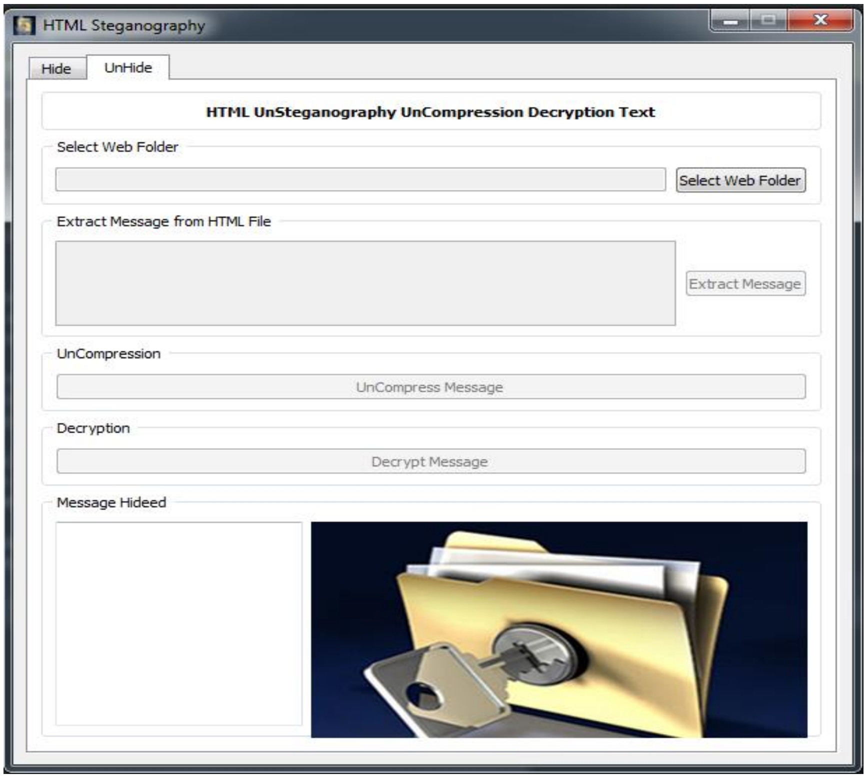
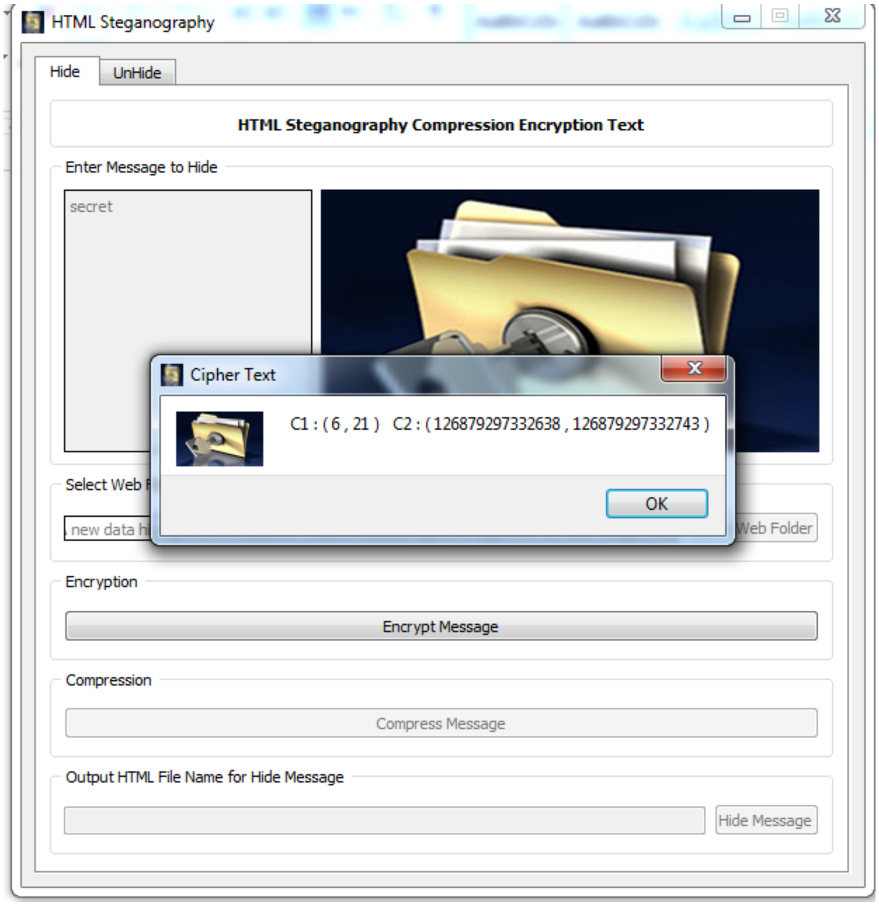


Figure 4.2 The receiver part

The encryption step will receive the message input by click button (Encrypt Message) and this will encrypt the message, see encrypt result is show in figure 4.3. Then the sender transfer to the next step compression that take the previous result to compress and prepare for hiding in the last step show figure 4.4.



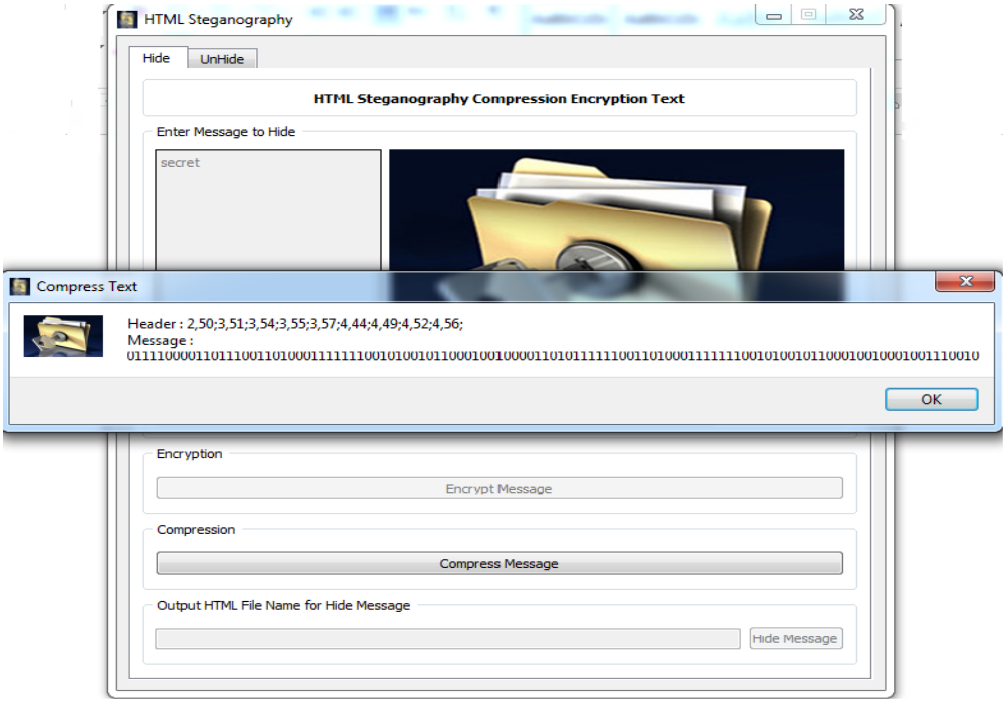
secret

new data h

C1 : (6 , 21) C2 : (126879297332638 , 126879297332743)

Figure 4.3 Encrypt result (sender part)

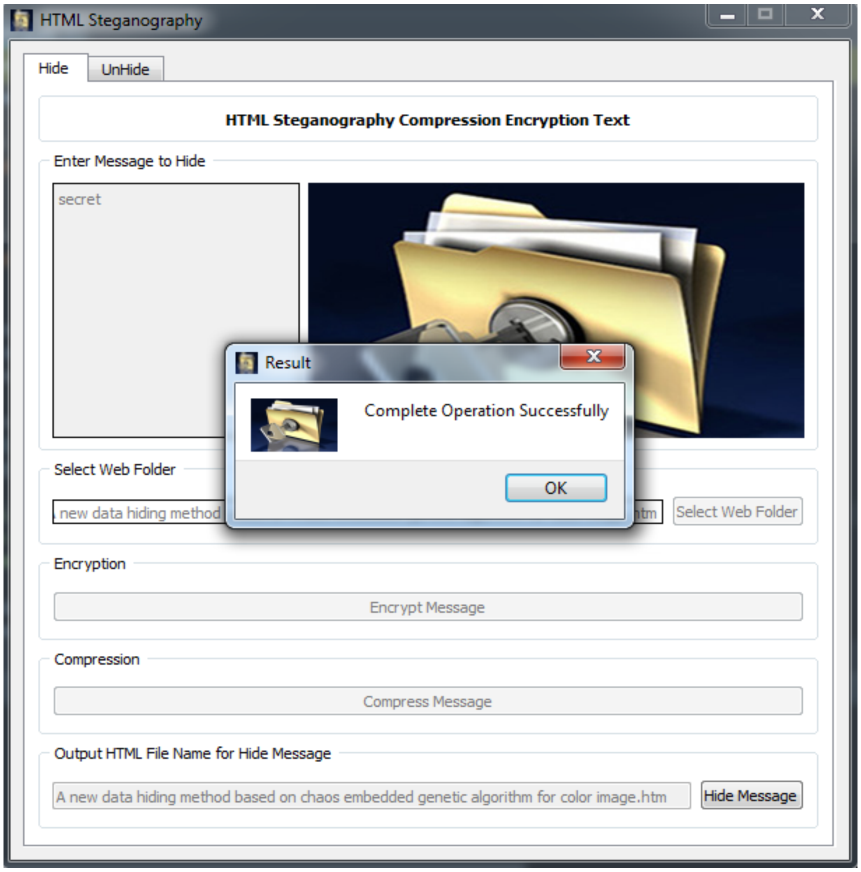
The last step take the second input was entered by sender to hide the final form of the previously generated message. Once the operation is completed successfully the window appears as shown in the figure 4.5.



secret

```
Header : 2,50;3,51;3,54;3,55;3,57;4,44;4,49;4,52;4,56;
Message :
0111100001101110011010001111111001010010110001001100001101011111100110100011111110010100101100010010001001110010
```

Figure 4.4 Compress result (sender part)



secret

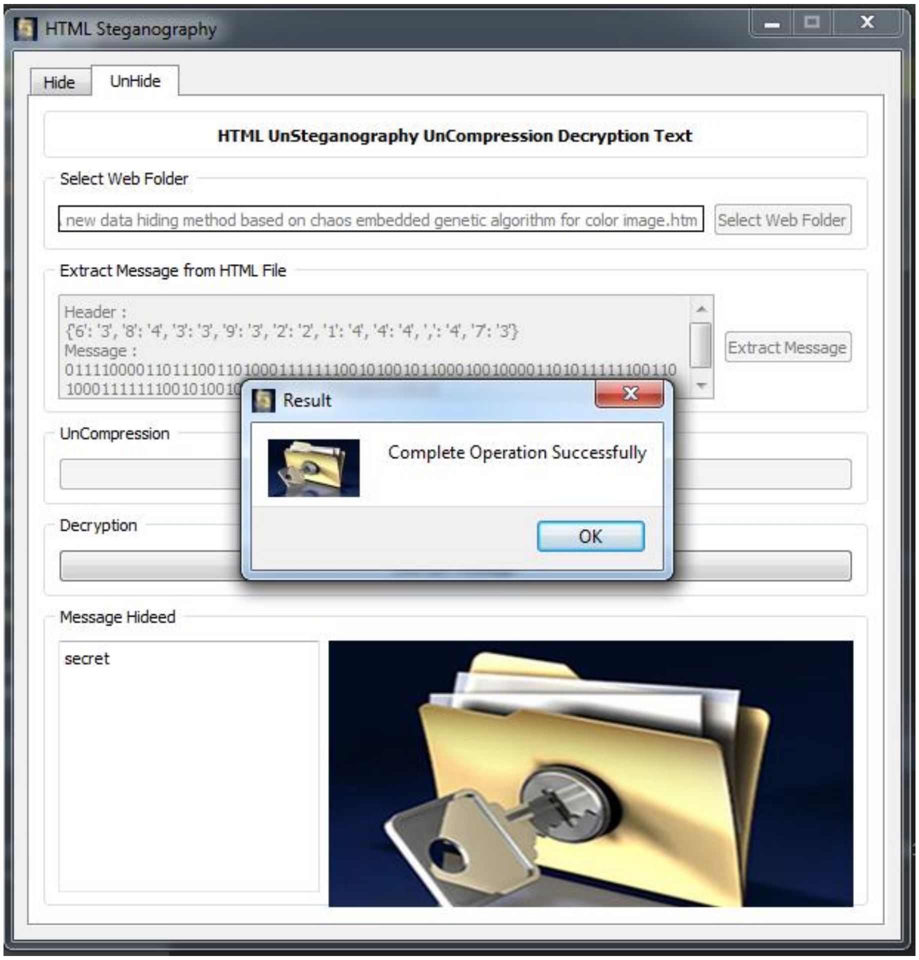
Complete Operation Successfully

new data hiding method

A new data hiding method based on chaos embedded genetic algorithm for color image.htm

Figure 4.5 Final successful alert (sender part)

In the figure 4.2 the Receiver screen, the user enter just one input centered on select the web directory by using also the browsing button (Select web Folder). If the operation is successful, the rest of the steps are inversely followed until the receiver reach the hidden message and confirmation message dialog appear show in the figure 4.6.



new data hiding method based on chaos embedded genetic algorithm for color image.htm

```
Header :
{'6': '3', '8': '4', '3': '3', '9': '3', '2': '2', '1': '4', '4': '4', ',': '4', '7': '3'}
Message :
0111100001101110011010001111111001010010110001001000011010111111100110
10001111111001010010
```

Complete Operation Successfully

secret

Figure 4.6: Final successful alert and secret message (receiver part)

4.4 Result

The test environment has been taken to test the proposed method in this thesis: Core i5 3230M CPU @ 2.6 GHz RAM is 4GB, Window 7 operating system, Firefox browser HTML web page source.

The different experiments were done by applied three different of secret messages to file in order to measure a variety parameters such as capacity, security and imperceptibility.

1- Capacity

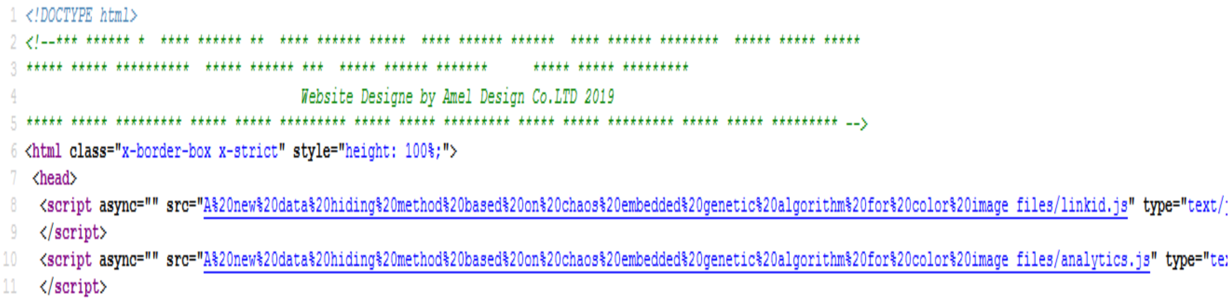
An increase in the size of the secret message shows that there is an increase in the size of the generated cover text

Table 4.1: The first secret message

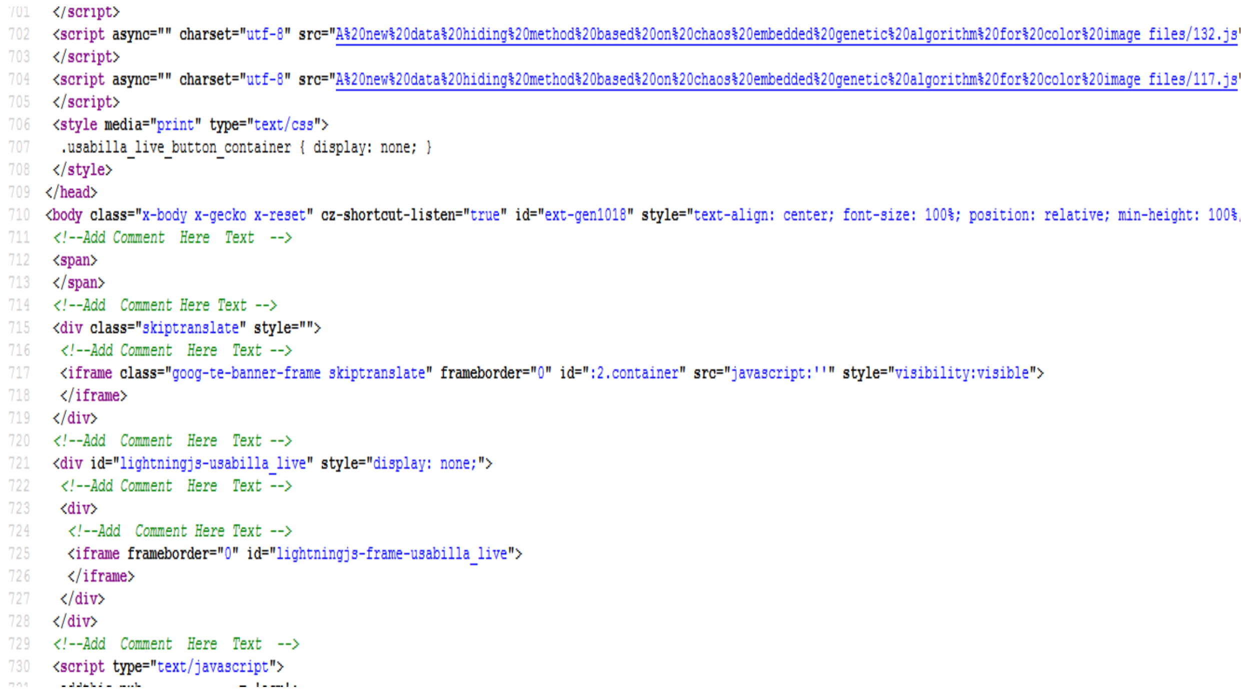
Experiments Message 1:
secret

Table 4.2: Experimental results-1

Secrete Message	Web Page File size	Comment Spaces	LEC(Largest Embedding Capacity)	Embedded Web Page File size
48 **Bits**	118 KB	8 Bits	4480 Bits	129 KB



```
<!DOCTYPE html>
<!--*** ****** *  **** ****** **  **** ****** *****  **** ****** ******  **** ****** ********  ***** ***** *****
***** ***** **********  ***** ****** ***  ***** ****** *******      ***** ***** *********
                                 Website Designe by Amel Design Co.LTD 2019
***** ***** ********* ***** ***** ********* ***** ***** ********* ***** ***** ********* ***** ***** ********* -->
<html class="x-border-box x-strict" style="height: 100%;">
 <head>
  <script async="" src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image files/linkid.js" type="text/;
  </script>
  <script async="" src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image files/analytics.js" type="te:
  </script>
```



```
  </script>
  <script async="" charset="utf-8" src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image files/132.js'
  </script>
  <script async="" charset="utf-8" src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image files/117.js'
  </script>
  <style media="print" type="text/css">
   .usabilla_live_button_container { display: none; }
  </style>
 </head>
 <body class="x-body x-gecko x-reset" cz-shortcut-listen="true" id="ext-gen1018" style="text-align: center; font-size: 100%; position: relative; min-height: 100%,
  <!--Add Comment  Here  Text  -->
  <span>
  </span>
  <!--Add  Comment Here Text -->
  <div class="skiptranslate" style="">
   <!--Add Comment  Here  Text -->
   <iframe class="goog-te-banner-frame skiptranslate" frameborder="0" id=":2.container" src="javascript:''" style="visibility:visible">
   </iframe>
  </div>
  <!--Add  Comment  Here  Text -->
  <div id="lightningjs-usabilla_live" style="display: none;">
   <!--Add Comment  Here  Text -->
   <div>
    <!--Add  Comment Here Text -->
    <iframe frameborder="0" id="lightningjs-frame-usabilla_live">
    </iframe>
   </div>
  </div>
  <!--Add  Comment  Here  Text  -->
  <script type="text/javascript">
```

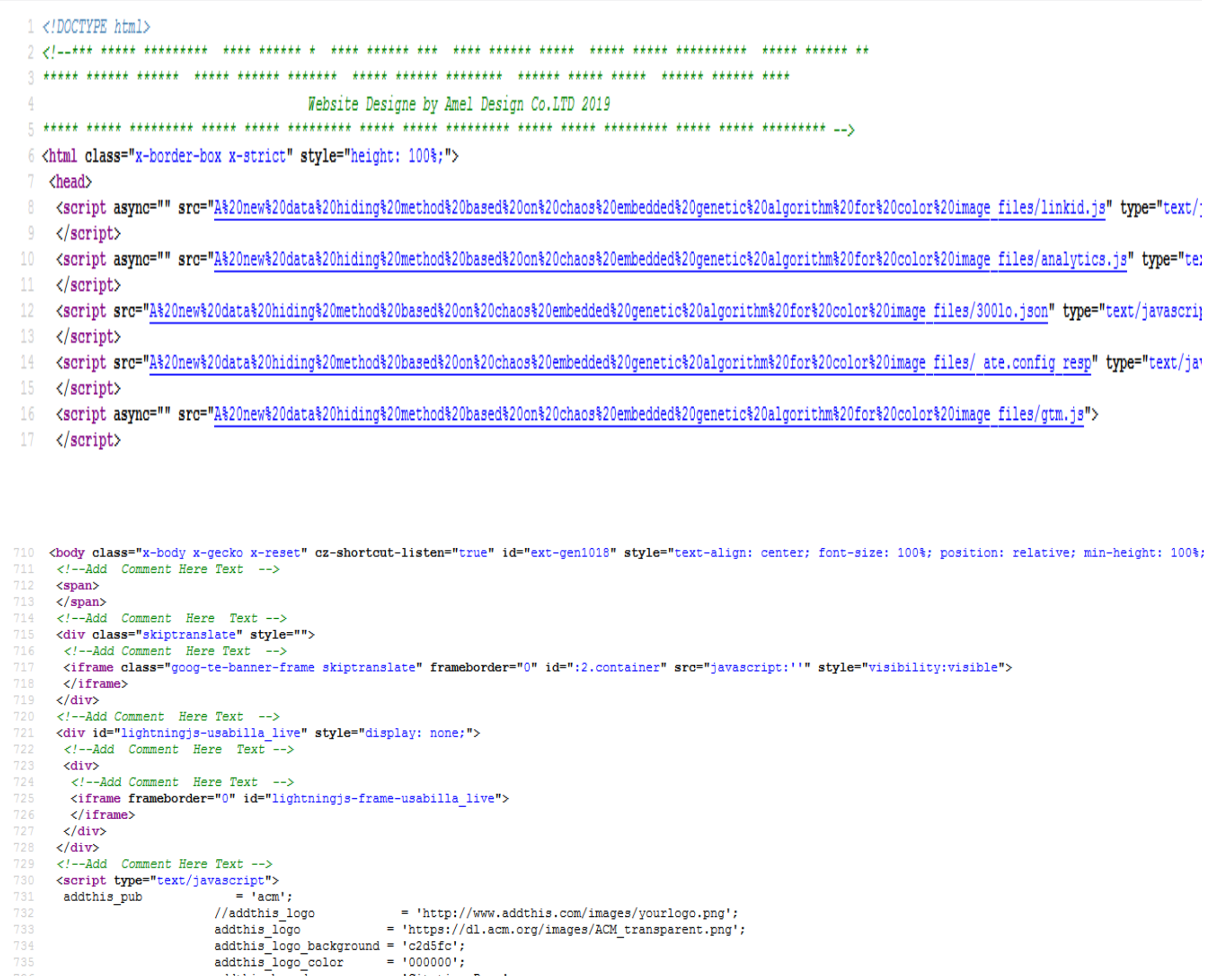
Figure 4.7: Source code of HTML Page after embedding result-1

Table 4.3: The second secret message

Experiments Message 2:
Welcome Home

Table 4.4: Experimental results-2

Secrete Message	Web Page File size	Comment Spaces	LEC(Largest Embedding Capacity)	Embedded Web Page File size
96 Bits	118 KB	8 Bits	4480 Bits	131 KB



```
<!DOCTYPE html>
<!--*** ***** *********  **** ****** *  **** ****** ***  **** ****** *****  ***** ***** **********  ***** ****** **
***** ****** ******  ***** ****** *******  ***** ****** ********  ****** ***** *****  ****** ****** ****
                                  Website Designe by Amel Design Co.LTD 2019
***** ***** ********* ***** ***** ********* ***** ***** ********* ***** ***** ********* ***** ***** ********* -->
<html class="x-border-box x-strict" style="height: 100%;">
 <head>
  <script async="" src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image_files/linkid.js" type="text/
  </script>
  <script async="" src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image_files/analytics.js" type="te
  </script>
  <script src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image_files/300lo.json" type="text/javascri
  </script>
  <script src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image_files/_ate.config_resp" type="text/ja
  </script>
  <script async="" src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image_files/gtm.js">
  </script>
```

```
<body class="x-body x-gecko x-reset" cz-shortcut-listen="true" id="ext-gen1018" style="text-align: center; font-size: 100%; position: relative; min-height: 100%;
 <!--Add  Comment Here Text  -->
 <span>
 </span>
 <!--Add  Comment  Here  Text -->
 <div class="skiptranslate" style="">
  <!--Add Comment  Here Text  -->
  <iframe class="goog-te-banner-frame skiptranslate" frameborder="0" id=":2.container" src="javascript:''" style="visibility:visible">
  </iframe>
 </div>
 <!--Add Comment  Here Text  -->
 <div id="lightningjs-usabilla_live" style="display: none;">
  <!--Add  Comment  Here  Text -->
  <div>
   <!--Add Comment  Here Text  -->
   <iframe frameborder="0" id="lightningjs-frame-usabilla_live">
   </iframe>
  </div>
 </div>
 <!--Add  Comment Here Text -->
 <script type="text/javascript">
  addthis_pub             = 'acm';
                    //addthis_logo            = 'http://www.addthis.com/images/yourlogo.png';
                    addthis_logo            = 'https://dl.acm.org/images/ACM_transparent.png';
                    addthis_logo_background = 'c2d5fc';
                    addthis_logo_color      = '000000';
```

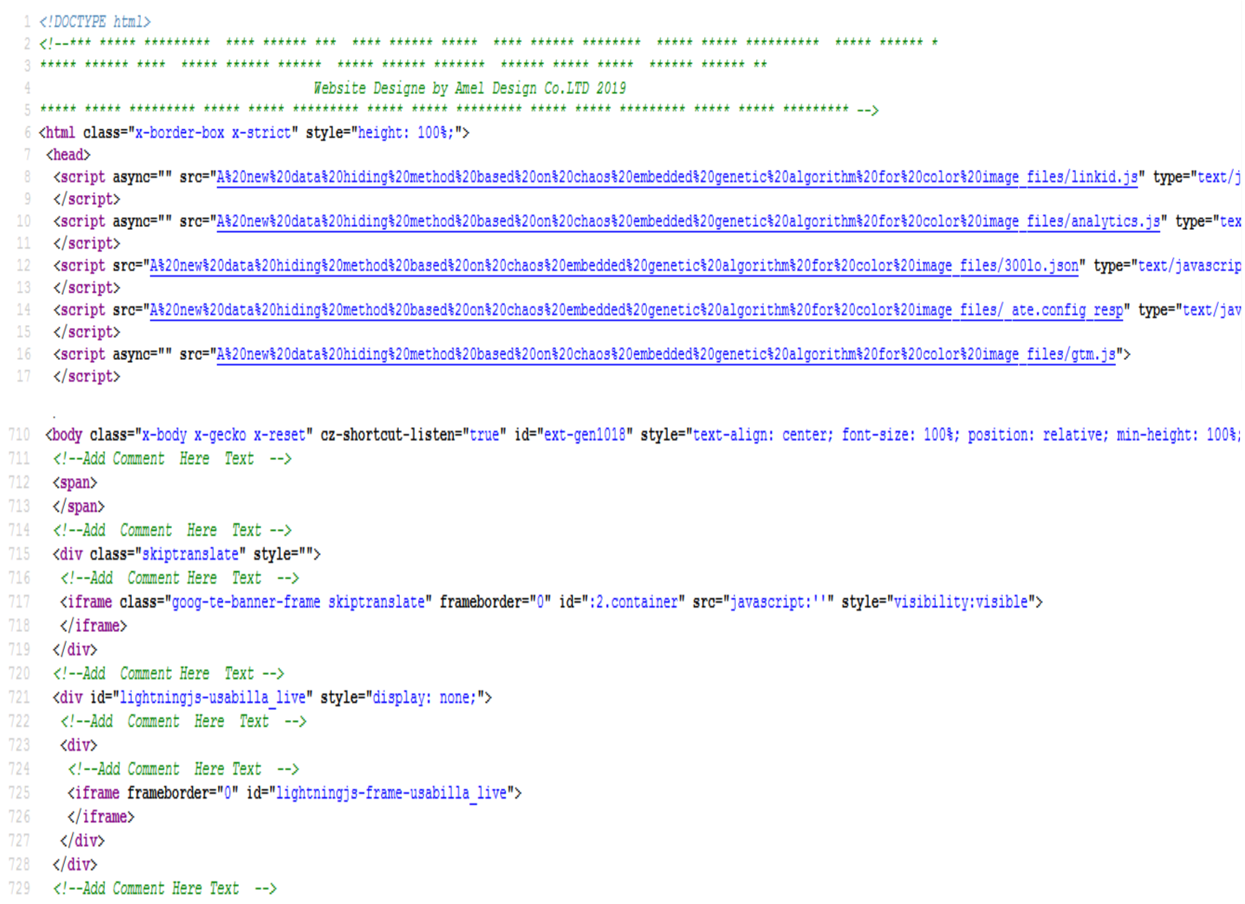
Figure 4.8: Source code of HTML Page after embedding result-2

Table 4.5: The third secret message

Experiments Message 3:
Compression and Encryption

Table 4.6: Experimental results-3

Secrete Message	Web Page File size	Comment Spaces	LEC(Largest Embedding Capacity)	Embedded Web Page File size
208 Bits	118 KB	8 Bits	4480 Bits	134 KB



```
<!DOCTYPE html>
<!--*** ***** *********  **** ****** ***  **** ****** *****  **** ****** ********  ***** ***** **********  ***** ****** *
***** ****** ****  ***** ****** ******  ***** ****** *******  ****** ***** *****  ****** ****** **
                                   Website Designe by Amel Design Co.LTD 2019
***** ***** ********* ***** ***** ********* ***** ***** ********* ***** ***** ********* ***** ***** ********* -->
<html class="x-border-box x-strict" style="height: 100%;">
 <head>
  <script async="" src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image_files/linkid.js" type="text/j
  </script>
  <script async="" src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image_files/analytics.js" type="tex
  </script>
  <script src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image_files/300lo.json" type="text/javascrip
  </script>
  <script src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image_files/_ate.config_resp" type="text/jav
  </script>
  <script async="" src="A%20new%20data%20hiding%20method%20based%20on%20chaos%20embedded%20genetic%20algorithm%20for%20color%20image_files/gtm.js">
  </script>

 <body class="x-body x-gecko x-reset" cz-shortcut-listen="true" id="ext-gen1018" style="text-align: center; font-size: 100%; position: relative; min-height: 100%;
  <!--Add Comment  Here  Text  -->
  <span>
  </span>
  <!--Add  Comment  Here  Text -->
  <div class="skiptranslate" style="">
   <!--Add  Comment Here  Text  -->
   <iframe class="goog-te-banner-frame skiptranslate" frameborder="0" id=":2.container" src="javascript:''" style="visibility:visible">
   </iframe>
  </div>
  <!--Add  Comment Here  Text -->
  <div id="lightningjs-usabilla_live" style="display: none;">
   <!--Add  Comment  Here  Text  -->
   <div>
    <!--Add Comment  Here Text  -->
    <iframe frameborder="0" id="lightningjs-frame-usabilla_live">
    </iframe>
   </div>
  </div>
  <!--Add Comment Here Text  -->
```

Figure 4.9: Source code of HTML Page after embedding result-3

By getting the previous results, the method gives high LEC. In additions it increases whatever the numbers of tags in web page and the length of comments that will add before the tags.

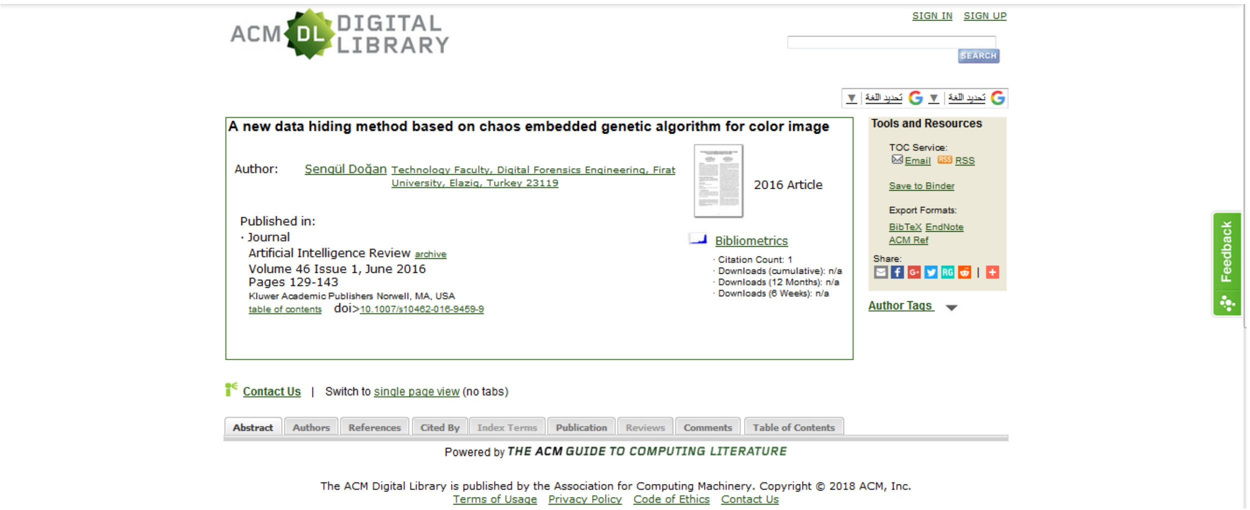
2- Security

The complexity is so obvious when applied encryption and the compression, this avoids ability to retrieve the original word during transformation process.

More difficult to decode, the comments inside the body tag are different from comment before head tag, that different lies in what those comments bear, the first carried the secret message after encryption and compression and they were represented using space between the words. The second the stars contain the key that help the receiver to uncompressed the message and it created by specific equation mentioned previous chapter. This stars do not raise doubt because they put as a definition of the programmer and this is a common practice in codes scripts pages.

3- Imperceptibility

Thesis method is embeds the secret message in the source code of the HTML page (figure 4.7) as a comments tag. In figure 4.8 show how the result of this approach. Imperceptibility test is measures to check the effectiveness of the proposed approach. Fig 4.9 and 4.10 show the screen shot of the perceptibility test with result 100% match between them before and after embedded.



A new data hiding method based on chaos embedded genetic algorithm for color image

Author: Sengül Doğan Technology Faculty, Digital Forensics Engineering, Firat University, Elazig, Turkey 23119

2016 Article

Published in:
· Journal
Artificial Intelligence Review archive
Volume 46 Issue 1, June 2016
Pages 129-143
Kluwer Academic Publishers Norwell, MA, USA
table of contents doi>10.1007/s10462-016-9459-9

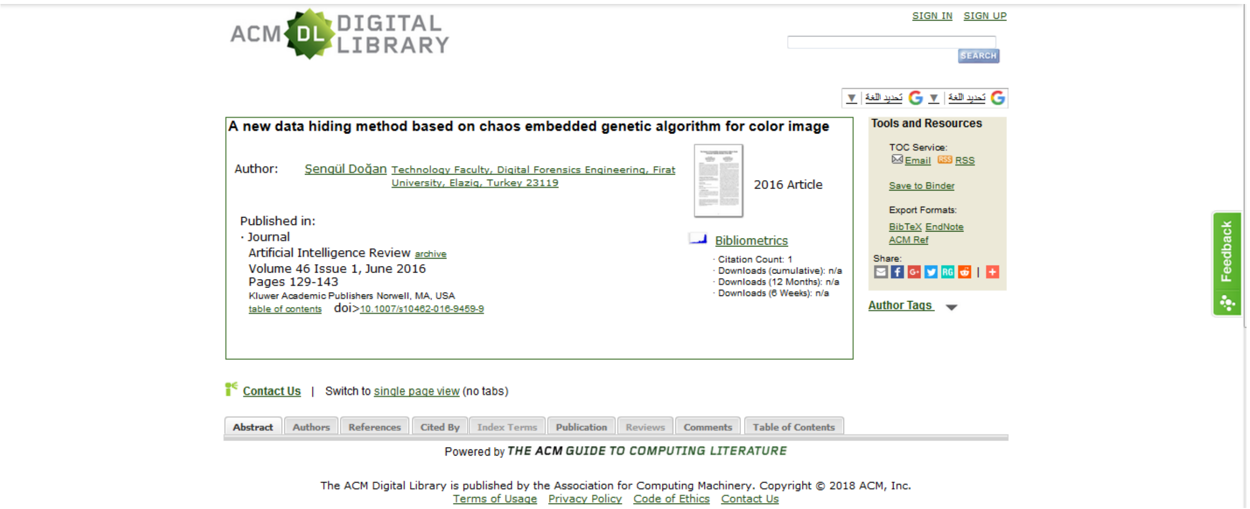
Bibliometrics
· Citation Count: 1
· Downloads (cumulative): n/a
· Downloads (12 Months): n/a
· Downloads (6 Weeks): n/a

Contact Us | Switch to single page view (no tabs)

Powered by THE ACM GUIDE TO COMPUTING LITERATURE

The ACM Digital Library is published by the Association for Computing Machinery. Copyright © 2018 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Figure 4.10 Web page before hide



A new data hiding method based on chaos embedded genetic algorithm for color image

Author: Sengül Doğan Technology Faculty, Digital Forensics Engineering, Firat University, Elazig, Turkey 23119

2016 Article

Published in:
· Journal
Artificial Intelligence Review archive
Volume 46 Issue 1, June 2016
Pages 129-143
Kluwer Academic Publishers Norwell, MA, USA
table of contents doi>10.1007/s10462-016-9459-9

Bibliometrics
· Citation Count: 1
· Downloads (cumulative): n/a
· Downloads (12 Months): n/a
· Downloads (6 Weeks): n/a

Tools and Resources

TOC Service:
Email RSS

Save to Binder

Export Formats:
BibTeX EndNote
ACM Ref

Share:

Author Tags

Contact Us | Switch to single page view (no tabs)

Powered by *THE ACM GUIDE TO COMPUTING LITERATURE*

The ACM Digital Library is published by the Association for Computing Machinery. Copyright © 2018 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Figure 4.11 Web page after hide

CHAPTER V

The Conclusion and Future Work

5.1 The Conclusion

The purpose of all this thesis secure communicate between the sender and receiver. A public internet website is the only media they have, so the content of HTML website must carrier the secret data without other website visitor notice any change. Steganography can protect data by hiding it but using it alone may not guarantee total protection, also using cryptography make the enemy detects presence of something stranger happens.

The website is generally classified under the script medium, it is quite difficult than other mediums because of less amount of redundancy and changes can be detected quite easily.

The advantages of thesis model are lead to 'security in depth', three levels was Applied; first it encrypts the secret message using ElGamal Elliptic Curve Cryptography algorithm so the receiver cannot obtain the message unless by the decryption. Second level compress the output from the previous level result using Huffman Coding by applied Canonical Huffman Coding to reduce the size of compress header. The last level uses Steganography techniques to hide the compressed message into one of the HTML webpage contents by adding comments for tags.

5.3 Future Work

Based on current work this thesis can be extended to include several suggested points to improve the performance of this algorithm:

First, work on processing other types of messages to hide all other data types such as audio, video images not only text data. Second, use integrity technique that makes the receiver for check if compress was done correctly. Third, use special logical comment for different tags to add more security and no attention if someone visit the source code. Forth, add special comment as other recommended for every different tags. Fifth, investigating alternative embedding techniques for compress header. Sixth, addition another standard security layer to the secret text.

References:

Kumar, Suresh. Sinkgh, Ganesh. Kumar, Tarun. and Singh, Maninder. (2013). Hiding the Text Messages of Variable Size using Encryption and Decryption Algorithms in Image Steganography .International Journal of Computer Applications.

Aboalsamh. Hatim, A. Mathkour, I. Mursi, Mona F.M. and Assassa, Ghazy M.R. (2008) . Steganalysis of JPEG Images: An Improved Approach for Breaking the F5 Algorithm. WSEAS International Conference on COMPUTERS, Heraklion, Greece.

Zaidan, B.B. Zaidan ,A.A. Taqa, Alaa. and Othman, Fazida . (2009). Stego-Image Vs Stego-AnalysisSystem. International Journal of Computer and Electrical Engineering.

Por, L.Y. Ang, T.F. and Delina, B. (2008) .White Steg: A New Scheme in Information Hiding Using Text Steganography. WSEAS Transactions on Computers.

Kheiralla, Fatma. and Dr. Elmusharaf, Mudawi. (2016). Data Hiding On Web Page using Steganography by Genetic Algorithm. International Journal of Innovations & Advancement in Computer Science.

Doğan, Şengül. (2016). A new data hiding method based on chaos embedded genetic algorithm for color image. Journal Artificial Intelligence Review archive.

Patel, Asha A. Information Hiding –The Art of Steganograph." .GSEC Practical.

Dhanani ,Chintan. and Panchal, Krunal. HTML Steganography using realative links and multi web page embedment. International journal Of Engineering Development and Research.

Kumar, Arvind. and Pooja, Km. (2010).Steganography-A Data Hiding Technique. International Journal of Computer Applications.

Garg, M. (2011). A novel text steganography technique based on html documents. International Journal of advanced Science and Technology.

Kour , Jasleen. and Verma , Deepankar. (2014).Steganography Techniques –A Review Paper. International Journal of Emerging Research in Management &Technology.

Doğan, Şengül. (2016). A new data hiding method based on chaos embedded genetic algorithm for color image. Journal Artificial Intelligence.

Ashok, J. Raju, Y., Munishankaraiah, S. and Srinivas, K. (2010). Steganography: An overview. International Journal of Engineering Science and Technology.

Por,L, Y. and Delina, B. (2008). Information Hiding: A New Approach in Text Steganography.Conf. On Applied Computer & Applied Computational Science (ACACOS '08).

Zhang, X. and Wang, S. (2005). Steganographyusing multiple-base notational system and human vision sensitivity, IEEE Signal Process. Lett.

Ushll, S. SathishKumal, G.A. and Boopathybagan, K. (2011).A Secure Triple Level Encryption Method Using Cryptography and Steganography.20 II International Conference on Computer Science and Network Technology.

Gajbhiye, Samta. and Dr. Karmakar, sanjeev. Application of Elliptic curve Method in cryptography:A Literature Review. (IJCSIT) International Journal of Computer Science and Information Technologies.

Jagdale, B.N. and Bedi, R.K . (2010). Securing MMS with high performance Elliptic curve cryptography.International journal of computer applications.

Shankar, Tarun. and Sahoo, G. (2010). Cryptography by Karatsuba Multiplier with ASCII Codes. International Journal of Computer Applications.

Malik, Muhammad. Efficient Implementation of Elliptic Curve Cryptography Using Low-power Digital Signal Processor.National University of Science and Technology (NUST), Pakistan.

Singh, Prem. Chaudhary, Rajat. and Agarwal, Ambika. (2012). A Novel Approach of Text Steganography based on null spaces. IOSR Journal of Computer Engineering (IOSRJCE).

Karan, H. Parmar. and Parmar, Avaniba . (2015). Web Based Steganography Using Combination of Three Methods. Hasmukh Goswami College of Engineering,Vahelal, Ahemdabad, Gujarat, India.

Arelakis, A. (2013). Design Considerations of Value-aware Cache. Licentiate degree thesis.

APPENDIX

Technical Proposed Levels Methods Algorithms

The Construction Level

The construction level specialized in inputs for two sides.

Algorithm (Sender Side):

- Enter the Secret Message.
- Check message format (English Characters).
- Do until the characters in right format.
- Browse to select web folder.
- Do until the web folder contain at least one HTML file.
- HTML file select from other html files base on equation (number of HTML files/2).

Algorithm (Receiver Side):

- Browse to select web folder.
- Do until the web folder contain at least one HTML file.
- HTML file select from other html files build on equation (number of HTML files/2).

Encryption / Decryption:

Algorithm (Sender Side):

- Encryption method receive the secret message input from construction level.
- Encrypt the message using ElGamal Elliptic Curve Cryptography.
- Return encrypt message.

Algorithm (Receiver Side):

- Decrypt method receive the decompress message output from second level.
- Decrypt the message using ElGamal Elliptic Curve Cryptography.
- Return message (match input message from sender construction level).

Compression / Decryption:

Algorithm (Sender Side):

- Receive encrypt message from level one.
- Calculate frequency for every character in encrypt message.
- Build Huffman tree and get every character coding.
- Calculate the Canonical Huffman Coding for any character coding.
- Sort characters coding length ascending (if two characters coding is same length sort by character ascending).
- First character fill zero based on length of code.
- For the next character add one “1” to code if in same code length.
- Then add zero “0” to the code if the length of the code larger than previous.
- Encode any character in encrypt message with the corresponding code.
- Return header (symbol + code length) and compress message.

Algorithm (Receiver Side):

- Receive extract header and compress message from level one.
- Calculate Canonical Huffman Coding from header.
- Decode message.
- Return decompressed message.