The African Journal of Information Systems

October 2013

Genetic Algorithm Based Model in Text Steganography

Christine K. Mulunda
University of Nairobi, christine.mulunda@gmail.com

Peter W. Wagacha
University of Nairobi, waiganjo@uonbi.ac.ke

Alfayo O. Adede
University of Nairobi, alfayaoyugi@gmail.com

Follow this and additional works at: https://digitalcommons.kennesaw.edu/ajis

Part of the Artificial Intelligence and Robotics Commons, and the Management Information Systems Commons

Recommended Citation

KENNESAW STATE UNIVERSITY
COLES COLLEGE OF BUSINESS
Department of Information Systems

Genetic Algorithm Based Model in Text Steganography

Christine K. Mulunda
University of Nairobi
christine.mulunda@gmail.com

Peter W. Wagacha
University of Nairobi
waiganjo@uonbi.ac.ke

Alfayo O. Adede
University of Nairobi
alfayaoyugi@gmail.com

ABSTRACT

Steganography is an ancient art. It is used for security in open systems. It focuses on hiding secret messages inside a cover medium. The most important property of a cover medium is the amount of data that can be stored inside it without changing its noticeable properties. There are many sophisticated techniques with which to hide, analyze, and recover that hidden information. This paper discusses an exploration in the use of Genetic Algorithm operators on the cover medium. We worked with text as the cover medium with the aim of increasing robustness and capacity of hidden data. Elitism is used for the fitness function. The model presented here is applied on text files, though the idea can also be used on other file types. Our results show this approach satisfied both security and hiding capacity requirements. Furthermore, we found that an increase in the size of the secret message resulted in an exponential increase in the size of the generated cover text. We also found a close relationship between the size of the chromosome used and the population size.

KEYWORDS

Steganography, Genetic Algorithm, Cover Medium, Elitism.

INTRODUCTION

With the widespread use of Internet and wireless networks, and the blooming growth in consumer electronic devices and advances in multimedia compression techniques, multimedia streams are easily acquired nowadays. In an attempt to ensure protection of the aforementioned multimedia contents and effective hiding of additional data into such digital content, several techniques emerged.

Steganographic techniques are a very important part of the future of Internet security and privacy on open systems such as the Internet because important data can be hidden inside a cover medium so that only the parties intended to get the message knows that a secret message exists. A cover medium acts as a carrier to embed messages into. Many different medium have been employed to embed messages for example images, audio, and video as well as file structures. The resulting media after the text message has been hidden in cover medium is called stego object (Anderson and Petitcolas 1998).

The mostly used medium include: text, video, audio and image. Despite availability of several steganography techniques, they are prone to visual, structural and statistical attacks. In relation to text steganography, texts with hidden data are expected to have higher entropy than those without. Thus the study will attempt to answer the following research questions:

  i. How are online users experiencing or addressing security and privacy issues in message/information transfer?

  ii. What are the available steganography algorithms?

  iii. How can the use of Genetic Algorithm be used to produce a secure and robust steganography tool?

  iv. How will the implementation of genetic algorithm based approach to text steganography reduce the likelihood of visual, structural and statistical attack to embedded messages?

Using genetic algorithms that are based on the mechanism of natural genetics and the theory of evolution, this paper discusses the process of designing a general method to guide the steganography process to the best position of data hiding. The cover text used is a set of random numbers. First, the secret text/payload is encrypted and then converted into its ASCII form. A Genetic Algorithm (GA) is then applied on the cover text obtained from a set of randomly generated numbers to embed the secret message (ASCII form) into the text data (random numbers). The cover text generated is dependent on the length of the secret message. Once optimal results have been produced the embedding process begins to produce a stego text. To add an extra layer of security the secret message is encrypted using playfair encryption method before converting to its ASCII representation.

Later, an extraction algorithm is applied to get the original secret message. The results show that the proposed approach satisfies security, robustness and hiding capacity requirements.

LITERATURE REVIEW

Text steganography involves hiding information in plain text. Some previous works on text steganography include:

Use of specific characters from words (Moerland 2003): In this method, some specific characters from certain words are selected and are used to hide the secret information. The first character of the first word of each paragraph can be used to hide a secret message one character at a time such that by placing these characters side by side, we get the whole message. Moerland also discusses about using punctuation marks. The idea behind this approach is to utilize the presence of punctuation marks like comma (,), semi colon (:), quotes (" ") etc. in the text for encoding a secret message. The use of punctuation marks is quite common in the normal English text and hence it becomes difficult for the intruder to recognize the presence of secret message in the text document. This accounts for the security of the technique.

Line shifting method (Low et al. 1995): Here the lines of the text are shifted to some degrees, such as 1/300 inch up or down. Then the information is hidden by creating a hidden unique shape of the text.

Word shifting method (Low et al. 1995): Unlike in the line shifting method, the information is hidden by shifting the words horizontally or by changing the distance between the words.

Use of synonyms of certain words to hide the message in the English text (Niimi et al. 2003): Certain words from the text are selected, their synonyms are identified, and then the words along with their synonyms are used to hide the secret message in the text.

Adding extra white-spaces in the text (Huang and Yan 2001): White spaces can be placed at the end of each line, at the end of each paragraph or between the words.

Persian/Arabic text (M. Shirali-Shahreza 2008) and Urdu/Arabic text (Memon et al. 2008): One of the characteristics of these languages is that they contain a number of dot symbols in their letters. One dot symbol in a letter can be used to hide the information by shifting the position of a dot symbol a little bit vertically high with respect to the standard point position in the text.

Hindi text Steganography (Alla and Prasad 2009); this technique is based on the fact that each language has its own characteristics. Every language is formed of combinations of one or more vowels and consonants. These vowels and consonants and the combination of the two form the basis of this Hindi text steganography technique. This technique makes use of two elements: simple letters (pure vowels and pure consonants) and compound letters (combinations of vowels, consonants, vowels and consonants).

Hiding secret message by using different spellings of the words (M. H. Shirali-Shahreza and M. Shirali-Shahreza 2008); most words have different spelling in UK and US. For example "dialog" has different terms in UK (dialogue) and US (dialog). This difference in spellings forms the basis of steganography.

Emoticon based text Steganography (Wang et al. 2009); emoticons are emotional icons that are used in online chatting. These emoticons express the feeling or mood of the persons communicating with each other.

METHODOLOGY

Genetic Algorithm (GA) is based on biological evolutionary theories and is often used to solve optimization problems. GA comprises of a set of individual elements (the population) and a set of biologically inspired operators. According to evolutionary theories, only the most suited elements in a population are likely to survive, generate offspring, and transmit their biological heredity to the new generations. GA's are much superior to conventional search and optimization techniques in high-dimensional problem spaces due their inherent parallelism and directed stochastic search implemented by recombination operators.

In a genetic algorithm, a population of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. A part of the chromosomes is called a gene.

In this paper we present a new method of hiding information in a text by use of genetic algorithm approach in text steganography. The secret message is first encrypted to give the message an extra layer of security.

Stenography Process

Figure 1 below describes the flow of the stenography process from encryption of the secret message to becoming a stego text.
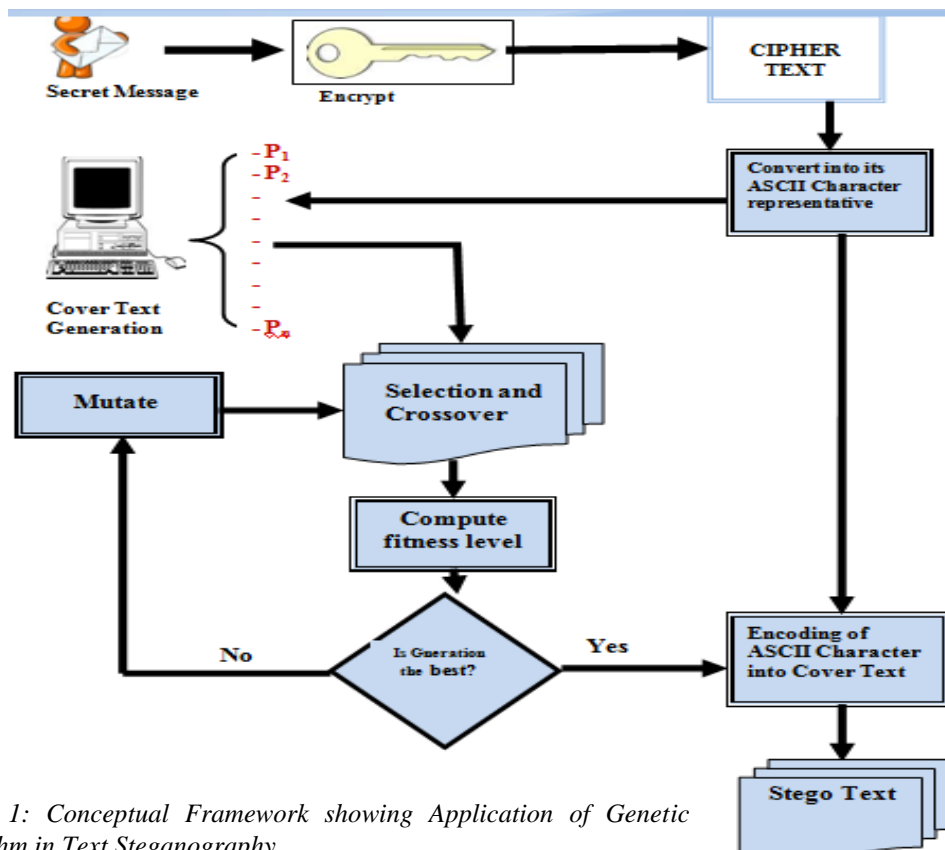

Figure 1: Conceptual Framework showing Application of Genetic Algorithm in Text Steganography

The initial stage of our steganography process was to encrypt the secret message by use of playfair cipher encryption algorithm to produce a cipher text (Jitendra et al. 2013). The cipher text was then converted to its ASCII Character representative. The number of characters obtained, *L*, forms the size of the population *(n)*. A random population of size *n* was then generated between the min and max values in *L*, with each individual member having *Z*-chromosomes (suitable solutions for the problem). Fitness function f(x) of each chromosome individual in the population was then evaluated. A new population was generated by repeating following steps until the new population was complete.

> *STEP 1: Encrypt the secret message*

> *STEP 2: Generate random population of size L (L=length of the Secret Message) with each individual member having n chromosomes (suitable solutions for the problem)*

> *STEP 3: [Fitness] Evaluate the fitness f(x) of each chromosome individual in the population*

> *STEP 4: [New population] Create a new population by repeating following steps until the new population is complete*

>> *i. [Selection] Select two parents from the population with the best fitness level*

>> *(the better fitness, the bigger chance to be selected)*

>> *ii. [Crossover] With a crossover probability, cross over the parents to form newoffspring (children). If no crossover was performed, offspring is an exact copy of parents.*

>> *iii. [Mutation] With a mutation probability, mutate new offspring at each locus*

>> *(position in chromosome).*

>> *iv. [Accepting] Place new offspring in a new population*

> *STEP 5: [Replace] Use new generated population for a further run of algorithm*

> *STEP 6: [Test] If the end condition is satisfied, stop and return the best solution to current population*

> *STEP 7: [Loop] Go to step 4*

After performing operations, some chromosomes might not satisfy the fitness and as a result the algorithm discards this process and uses the children chromosomes. The new generated population is again passed through a fitness function to find the best individuals for the population. Encoding of the secret message was performed once the optimum solution for the population was obtained.

Generating Population: Once the secret message is converted into its ASCII representative, the minimum and maximum values of the generated ASCII numbers are identified. The initial population will have a predetermined number of individuals which in this case is the total number of characters contained in the secret message. This population, which is a set of random numbers, is generated from the values that will fall between the identified minimum and maximum values. The individuals are grouped as a set of chromosomes containing genes. The two individuals with the highest fitness function will crossover to produce two offspring. The two offspring will undergo mutation, then will be assigned a fitness value before re-introduction into the population. From the population the two least fit individuals will then be discarded, as the original population size needs to be maintained. This will continue until an optimal solution is obtained.

Fitness Function: To get the individuals that are most fit, set operators (i.e. Intersection A∩B) are used to compare the ASCII values (elements) that are in the secret message with those contained within the individuals. The more values (elements) of the secret message contained in an individual, the higher the fitness function.

Mutation: Mutation process is used to introduce scarce genes to the population. This is achieved by using the set operator (i.e. difference) A-B: elements in A that are not in B. B in this case is the union of all the values contained in all individuals in the population, and A is the values in the secret message. This is done to get the scarce genes, which will be introduced to the produced offspring. Randomly a gene in the produced offspring will be selected and substituted with the scarce gene/allele. Each offspring is mutated before introduction to the population. To ensure that the two least fit individuals are not discarded with genes that are needed for optimization, set difference operation is reapplied to get the scarce allele/gene, if any.

Embedding process: Once an optimal solution is found, the embedding process begins. The first individual's genes are scanned to check if it matches with the secret message. If there is a gene that is similar, the message is embedded and the gene is substituted with the last gene. This continues until the last character in the secret message is embedded.

Steganalysis Process: To reverse the steganography process, the stego text is first decoded from its ASCII representative. The process of decrypting the Cipher text is performed to produce the secret message. The process of decrypting the text involves the use of a secret key used during the playfair encryption process together with cipher text extracted during steganalysis. This is as depicted in Figure 2 below.
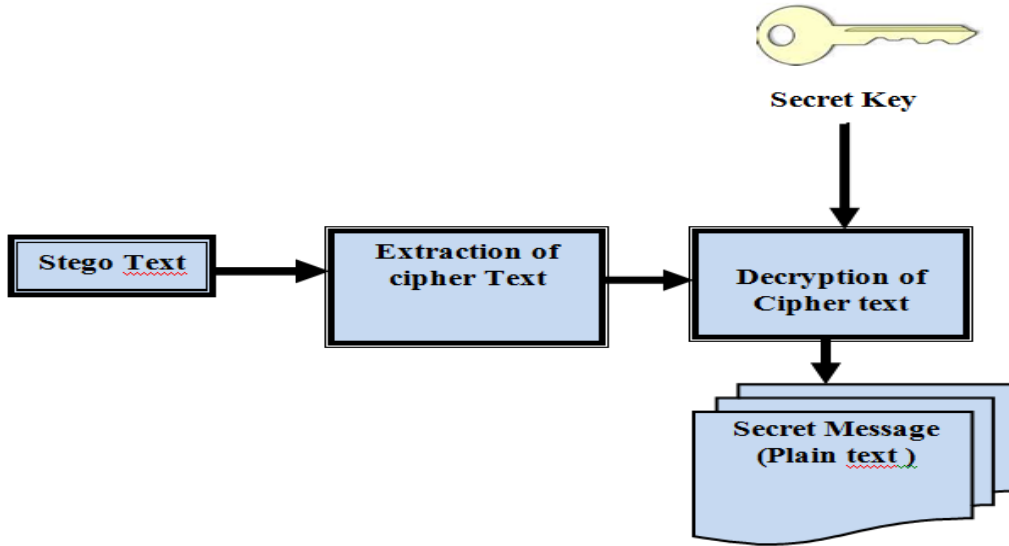
Figure 2: Conceptual Framework for Steganalysis

STEP 1: Retrieve the hidden text using the First In First Out (FIFO) algorithm by selecting n value from stego text (Where n is the number of an individual chromosomes used during STEGANOGRAPHY)

STEP 2: Extract the ASCII characters

STEP 3: Convert from the ASCII format to its representative character

STEP 4: Decrypt

STEP 5: Retrieve Secret Message

IMPLEMENTATION

This section aims at fulfilling our object on *how Genetic Algorithm can be used to produce a secure and robust steganography too?* The implementation of Genetic Algorithm was done on the cover text; in this case, a set of random numbers was used. The generated cover text depends on the length of the secret message. Once optimal results have been achieved, the embedding process begins to output stego text. An extraction algorithm is applied to reverse to the original secret message.

Genetic Algorithm Based Text Steganography Tool, developed to evaluate the proposed method, was implemented using Java programming language. The tool is a java Swing application developed using Net Beans Version 7.0.1 Integrated Development Environment (IDE) platform running on Java Development Kit Version 7.

RESULTS

Figure 3 below shows the steganography process of the cover text being passed into the embedding function with the secret message to encode resulting in a stego text containing the hidden message. A key is often used to protect the hidden message. This key is usually a password, so it is also used to encrypt and decrypt the message before and after embedding.
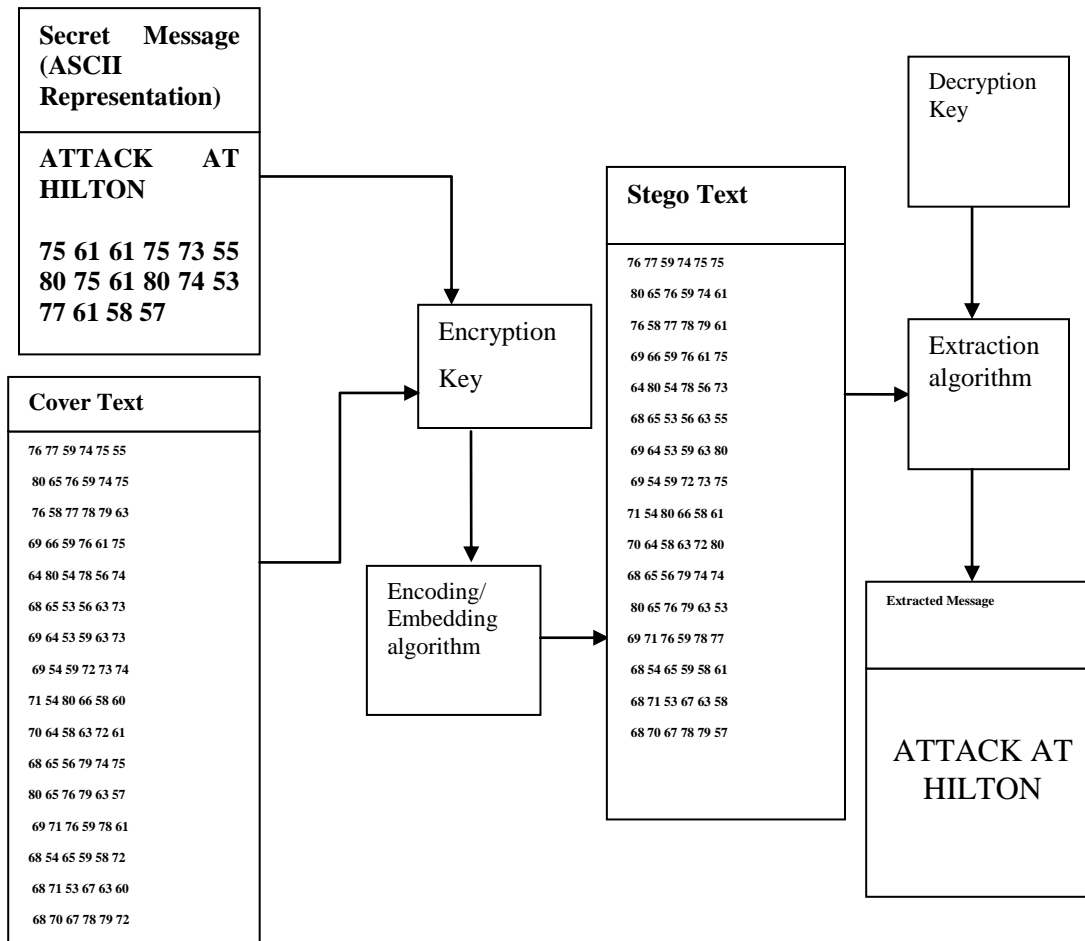
Figure 3: Represents the results obtained from the implementation process

Analysis of the Results

The experimental results showed that the approach used was found to satisfy both security aspects, hiding capacity requirements and minimal embedding time. It generated the stego text with minimum degradation and was not revealing about the existence of any hidden data, therefore maintaining its security. The analysis was done in two ways:

    a) Varying the size of secret message

    b) Varying the chromosome length

An increase in the size of the secret message shows that there is an increase in the size of the generated cover text. A chromosome of 4 bits, or 4 genes, was used on a population of 1 KB of file size to generate a cover text of 1 KB while a chromosome of 20 bits, or 20 genes, on a population of 1 KB of file size generated a cover text of 3 KB. See Appendix II.

The graph below shows that when the chromosome size was varied, there was a difference in the size of the cover text generated.
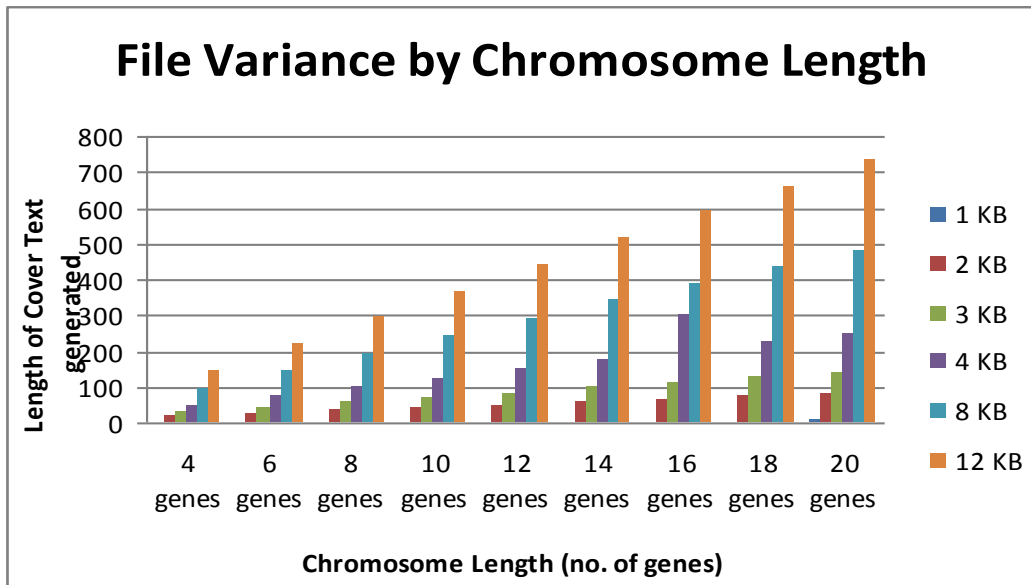

Figure 4: Graph represents file size variance by chromosome length

From the results obtained, it was found that best population size depends on the length of encoded message. That is, for a chromosome with 4 bits/genes, the population should be 4. Also, a chromosome of 20 bits/genes should have a population size of 20.

In relation to the embedding time, the results revealed that given a chromosome of 4 bits/genes on a population size of 4, the embedding time would be much faster than the same number of genes on a larger population size.

i.e. 4 bits/genes * Population size (4) = 16

When the numbers of chromosomes were increased to 20 bits/genes on the population size of 4, the embedding time was higher as compared to that of 4 bits/genes.

i.e. 20 bits/genes * Population size (4) = 80

This means that for best performance and/or speed of finding a solution, the population size should be almost equal to the chromosome size.

DISCUSSION

Online users find it difficult to trust the channels of communicating secret messages. This is because the available techniques are prone to attackers who intercept the message to reveal it, hence no security. This project therefore set out to investigate the available algorithm used to secure messages - in this case Steganography. Text Steganography was chosen from other cover mediums (video, image, audio) because of its smaller memory occupation and simpler communication. Genetic algorithm approach is not prone to visual, structural and statistical attack because of its use of random numbers and generation of random numbers between the minimum and maximum values of the secret message. Also, there are two layers of security being used; use of playfair encryption technique and then conversion of secret message to ASCII to generate the cover medium being used to embed the secret message bits/numbers.

Comparison of existing text based Steganography and the Genetic Algorithm technique used in this project was done in relation to robustness and capacity of hidden message. Robustness is the ability of a hidden message to not be detected either through visual, semantic, or statistical attack. The steganographic techniques available are prone to these attacks, unlike genetic algorithm approach, which makes used of random numbers.

Genetic algorithm technique used in this project is not prone to visual attacks because of its use of numbers. This is not the case for Format-Based technique that deals with modifications of existing text in order to hide the steganographic text by resizing of fonts, insertion of spaces or non-displayed characters, deliberate misspellings distributed throughout the text and resizing the fonts, among others. Insertion of spaces where extra space(s) between words is used - one space means that the transmitted information bit is \0", and two spaces mean \1", can easily be detected. The presence of errors in a document, such as deliberate misspellings when writing words, (\How *is* you" to \How *iz* you") may raise curiosity by someone intercepting the message.

As regards to statistical attack, character generation often takes the statistical properties of word-length and letter frequency to create "words" (with no lexical value) which will appear to have the same statistical properties as actual words in a given language. These words might convince a computer, which is only doing statistical analysis. Genetic algorithm technique used in this project uses random numbers to generate cover text, hence it is not prone to statistical attack.

Linguistic based methods deal with modifications of syntax and semantics of words and sentences to hide messages. Grammar-checkers used by modern word processors may be helpful tools in discovering ungrammatical texts. While legitimate ungrammatical texts certainly exist, given a certain context and threshold, such methods could be used to flag texts with no syntactic structure for further attention. In this project, random numbers are used. Syntax and semantics modification are not obvious, thus grammar-checkers cannot be used in structural attack.

As regards to capacity of hidden message, existing techniques are tedious to embed long messages and in some cases the meaning of the cover message changes completely until no sense can be made out of it. This is not likely to occur in the approach proposed in this project because one computer does all the work after the parameters for getting the population, size, fitness, and mutation values are pre-determined. Format-Based Method cannot be used to hide a long message as it is cumbersome and with its high affinity for visual attack, it cannot be effective. On the other hand, using linguistic method will

mean having a very long cover text that is prone to both syntactic and semantic errors and therefore also not very effective to use.

In order to fulfil our research question on how the use of Genetic Algorithm will be used to produce a secure steganography tool, a secret message was encrypted using the playfair encryption method, to add an extra layer of security to the message.

Comparisons of some Text Steganography Tools

The table below summarizes the comparison of some of the existing Steganography tools.

	GATS	**wbStego**	**SNOW**	**Stego**
Use of encryption/ decryption key	Yes	Yes/No	Yes/No	Yes
Cover file	System generated	Not System generated	Not System generated	Not System generated
File types	.txt	Image, pdf, txt	-	-
Visibility of secret message	Not visible	Not visible	Visible	Not visible
Type of encryption	Playfair	Various	ICE- Information Concealment Engine 64 BIT private key	-
Platform	JAVA: WIN	WIN : Delphi	C/C++: DOS WIN	C: DOS

**GATS – Genetic Algorithm Based Text Steganography Tool

CONCLUSION

A secure text Steganography algorithm based on the genetic method is proposed in this paper. The experimental results showed that this approach works, achieving effective optimization, security, and robustness.

Future work can be focused on exploring other search heuristics algorithms with an aim of improving the efficiency of the proposed algorithm in terms of robustness and capacity of hidden message. In addition, this technique can be extended to other types of files.

APPENDIX I: LIST OF ACRONYMS AND DEFINITIONS

ASCII – American Standard Code for Information Interchange

Cover Medium – it is used to embed messages and can either be video, text, or sound

Cover Text - Text containing an embedded message.

Cipher text – Refers to encrypted data.

Payload – Secret Message

Cryptography – The art of protecting information by encrypting it into an unreadable format, called cipher text. A secret key is used to decrypt the message into plain text.

Encryption – The translation of data into a secret code.

Decryption – The inverse of encryption

GA - Genetic Algorithm

Plain text – Refers to any message that is not encrypted - also called clear text.

Steganalysis – The art of discovering and rendering useless covert messages.

Steganography - A means of overlaying one set of information ("message") on another (a cover).

Stego/Steno text - The result of combining the cover text and the embedded message.

CFB - cipher-feedback

ICE - Information Concealment Engine

SNOW - **S**teganographic **N**ature **O**f **W**hitespace

APPENDIX II: LIST OF TABLES SHOWING FILE SIZE VARIANCE BY GENES

File size variance by 4 genes

Secret Text	1 KB	2 KB	3 KB	4 KB	8 KB	12 KB
Cover Text	1 KB	16 KB	28 KB	49 KB	95 KB	145 KB

File size variance by 6 genes

Secret Text	1 KB	2 KB	3 KB	4 KB	8 KB	12 KB
Cover Text	1 KB	25 KB	42 KB	74 KB	143 KB	219 KB

File size variance by 8 genes

Secret Text	1 KB	2 KB	3 KB	4 KB	8 KB	12 KB
Cover Text	1 KB	33 KB	56 KB	101 KB	194 KB	297 KB

File size variance by 10 genes

Secret Text	1 KB	2 KB	3 KB	4 KB	8 KB	12 KB
Cover Text	2 KB	41 KB	70 KB	124 KB	241 KB	367 KB

File size variance by 12 genes

Secret Text	1 KB	2 KB	3 KB	4 KB	8 KB	12 KB
Cover Text	2 KB	49 KB	84 KB	149 KB	289 KB	441 KB

File size variance by 14 genes

Secret Text	1 KB	2 KB	3 KB	4 KB	8 KB	12 KB
Cover Text	2 KB	58 KB	98 KB	176 KB	340 KB	515 KB

File size variance by 16 genes

Secret Text	1 KB	2 KB	3 KB	4 KB	8 KB	12 KB
Cover Text	2 KB	65 KB	112 KB	299 KB	386 KB	589 KB

File size variance by 18 genes

Secret Text	1 KB	2 KB	3 KB	4 KB	8 KB	12 KB
Cover Text	2 KB	74 KB	126 KB	225 KB	435 KB	663 KB

File size variance by 20 genes

Secret Text	1 KB	2 KB	3 KB	4 KB	8 KB	12 KB
Cover Text	3 KB	82 KB	140 KB	250 KB	483 KB	737 KB

REFERENCES

Alla, K. and Prasad, R.S.R. (2009). An evolution of Hindi text steganography, *Proceedings of the Sixth International Conference on Information Technology*, 1577–1578, Las Vegas, NV: New Generation.

Anderson, R.J. and Petitcolas, F.A.P. (1998). On the limits of steganography, *IEEE Journal of Selected Areas in Communications*, 16, 4, 474-481.

Huang, D. and Yan, H. (2001). Interword distance changes represented by sine waves for watermarking text images: IEEE, *Transactions on Circuits and Systems for Video Technology*, 11, 12, 1237-45.

Jitendra, C., Ravindra, K.G., and Shailendra, S. (2013). A survey of existing playfair ciphers, *International Journal of Engineering and Advanced Technology*, 2, 4, 658-9.

Low, S.H., Maxemchuk, N.F., Brassil, J.T., and O'gorman, L. (1995). Document marking and identification using both line and word shifting, *Proceedings of the Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies* (INFOCOM '95), Vol. 2, 853-60. Boston, MA.

Memon, J.A., Khowaja, K., and Kazi, H. (2008). Evaluation of steganography for Urdu /Arabic text, *Journal of Theoretical and Applied Information Technology*, 4, 3, 232-7.

Moerland, T. (2003). Steganography and steganalysis, *Leiden Institute of Advanced Computing Science*.

Niimi, M., Minewaki, S., Noda, H., and Kawaguchi, E. (2003). A framework of text-based steganography using SD form semantics model, *Pacific Rim Workshop on Digital Steganography*, Kitakyushu, Japan: Kyushu Institute of Technology.

Shirali-Shahreza, M. (2008). Text steganography by changing words spelling, *Proceedings of the 10th International Conference on Advanced Communication Technology* (ICACT '08), 1912-13, Phoenix Park, Korea.

Shirali-Shahreza, M.H. and Shirali-Shahreza, M. (2008). Steganography in Persian and Arabic Unicode texts using pseudo-space and pseudo-connection characters, *Journal of Theoretical and Applied Information Technology (JATIT)*, 4, 8, 682-7.

Wang, Z.H., Chang, C.C., Kieu, D., and Li, M.C. (2009). Emoticon-based text steganography in Chat, *Second Asia-Pacific Conference on Computational Intelligence and Industrial Applications* (PACIIA '09), Vol.2, 457-60, Wuhan, China.