Text Steganography Methods and its Tools

Abhishek Koluguri [#1], **Sheikh Gouse** [#2], **Dr. P. Bhaskara Reddy**[#3]

#1 Department of Computer Science and Engineering , MLR Institute of Technology,Hyderabad, Andhra Pradesh, India.
#2 Assistant Professor, Department of Computer Science and Engineering , MLR Institute of Technology,Hyderabad, Andhra Pradesh, India.
#3 Driector, MLR Institute of Technology,Hyderabad, Andhra Pradesh, India.

ABSTRACT

Steganography is art or science of hiding the data in embedded cover messages without altering it. The cover media may be text, image, voice, video streams in a digitized format. This paper proposes to hide the textual data. This paper helps the researchers, student's novel Steganography method with text media in a picture or image format and academicians to work towards Steganography techniques. We describe different methods and tools which are helpful to do research in the area text Steganography.

Key words: Steganography,stego,cover,Cipher, Linguistic, Lexical.

I. INTRODUCTION

In today's digitized world, due to tremendous increase in electronic communication technology, now it is a real and hard problem to send some sensitive data or information through a secure communication channel. This can be obtained by means of two techniques. One-Cryptography and second- Steganography. Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. Steganography is an ancient art or practice. Steganography is a branch of information hiding and its main goal is to communicate or transit the data securely in a completely undetectable manner. Literally meaning writing in a cover is the practice of hiding messages within other messages in order to conceal the existence of the original. Steganography refers to data or a file that has been concealed inside a digital image, video or audio file. Examples of its use can be found throughout history, dating as far back as ancient Greece. However, with the digital media formats in use for data exchange and communication today providing abundant hosts for Steganography communication, interest in this practice has increased. Couple this fact with the multitude of freely available, easy to use steganography software tools on the internet, the ability to exchange secret information without detection is available to virtually anyone who desires to do so, and provides unique challenges and opportunities for the security professional.

Steganography technologies are a very important portion of the future of security and privacy on open systems such as the Internet. Steganography can be used to hide important data inside another file so that only the parties intended to get the message even knows a secret message exists. In this paper, we review the Steganographic methods and tools in detail. The Steganography methods are very feasible to transit the data without altering it or encrypting and send it in a secure fashion

II. BACKGROUND

Steganography comes from the Greek and literally means "covered writing". The first the use of Steganography dates back to the Greeks. Historical methods relied on physical steganography – the employed media were: human skin, game, etc. Herodotus tells how a message was passed to the Greeks about Xerses' hostile intentions underneath the wax of a writing tablet, invisible inks and describes a technique of dotting successive letters in a cover text with a secret ink, due to Aeneas the Tactician. Steganography has been widely used, including in recent historical times and the present day. In early days to hide a message, had two choices: have the messenger memorize it, or hide it on the messenger. Modern Steganography entered the world in 1985. During World War II, null ciphers (unencrypted message) were used to hide secret messages.

III . MECHANISM OF STEGANOGRAPHY

The steganography methods can be classified based on cover media used to hide the sensitive data. We can classify the steganography methods based on the steganography carrier such as image, video, audio and text in HTML, doc or XML format. There are some most popular data formats used are .bmp, .doc, .gif, .jpeg, .mp3, .txt and .wav.

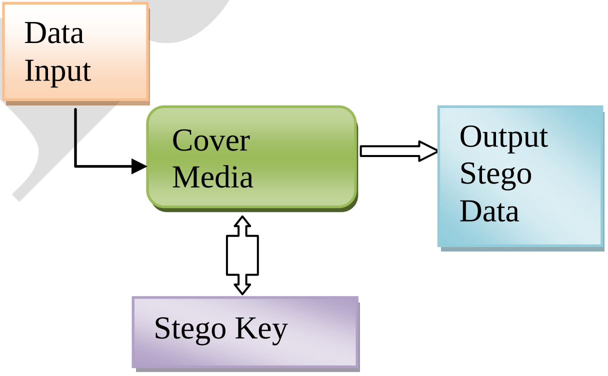


Figure.1 Basic Mechanism of Steganography

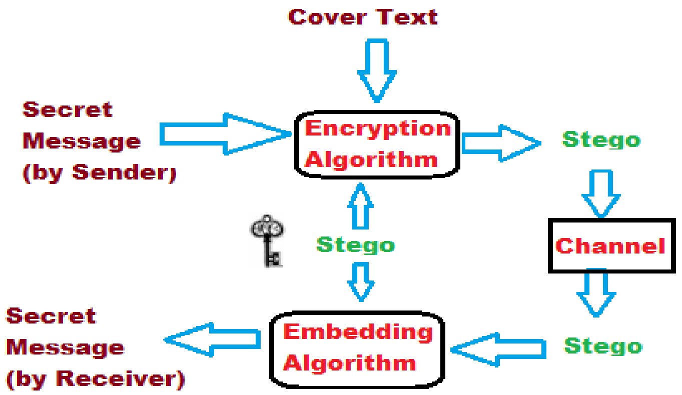


Figure.2 Text Steganography Mechanism

The basic mechanism or functionality of the steganography method is shown in figure 1. It consists of Data Input, Cover media, stego key and Output stego data. *Data Input* component takes the data to be hide using steganography method. *Cover media* is what kind of cover is used to hide the sensitive data. It may be either an image, video, audio or text media. *Stego key* is a key to encode or embed the data in cover media. *Output Stego Data* is an output of embedded or encoded sensitive data in cover media.

We can classify the Steganography methods based on cover media as follows:

1. Text Steganography
2. Image Steganography
3. Audio Steganography
4. Video Steganography
5. Protocol Steganography

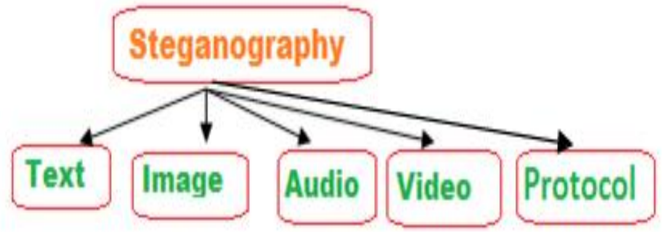


Figure.3 Different types of Steganography

Applications

1. Military and intelligence agencies
2. Criminals
3. Law enforcement and counter intelligence agencies
4. Online free speech on the net, including anonymous remailers and Web proxies
5. Digital elections and digital cash

6. Marketers use email forgery techniques
7. Finger prints & forensic
8. Telecommunication
9. Digital communications and storage
10. Medical images

IV. PROBLEM DEFINITION

Text Steganography can involve anything from changing the formatting of an existing text, to changing words within a text, to generating random character sequences or using context-free grammars to generate readable texts. Storing text file require less memory and its faster as well as easier communication makes it preferable to other types of steganographic methods. Unperceivable changes can be made to an image or an audio file, but, in text files, even an additional letter or punctuation can be marked by a casual reader.

Text Steganography can be done in any of the methods:
1. Replacing the format.
2. Replacing the meaning

Text Steganography can be broadly classified into three types:
1. Format based
2. Random and Statistical generation
3. Linguistic methods

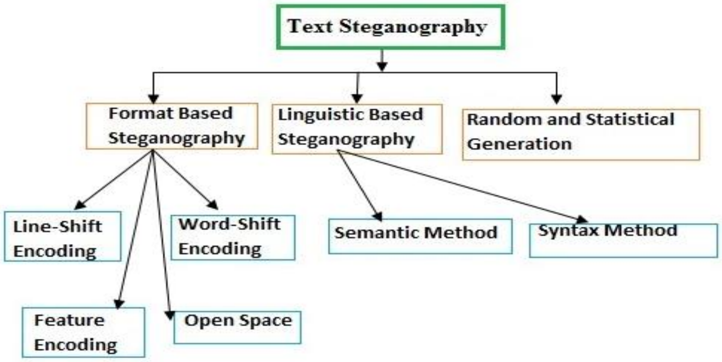


Figure.3 Different types of Text Steganography

A. Format Based Steganography

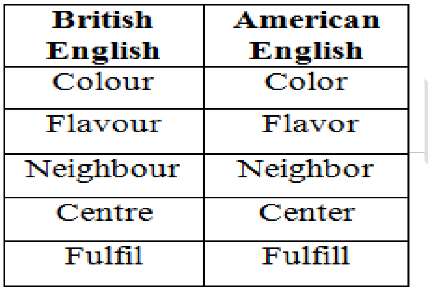
It uses the physical formatting of text as a space in which to hide in format ion. It generally modifies existing text in order to hide the steganographic text.

Any of the following:
1. Insertion of spaces
2. Non-displayed characters
3. Misspellings
4. Resizing of fonts.

It might fool people or readers who ignore occasional misspellings, but can often be easily detected by a computer. A computer might not recognize font resizing as a problem, particularly if it is only concentrating on text contents within a document for which text-resizing however, a people might detect strange font sizes almost immediately.

1. Character shifting

In this method, selecting the random sequence of characters for sending and receiving the message. However, it must appear to be random to anyone who intercepts the message. The words spelled in British and American English are arranged in separate columns [4].



British English	American English
Colour	Color
Flavour	Flavor
Neighbour	Neighbor
Centre	Center
Fulfil	Fulfill

Table.1 Difference between British & American English

2. Line-Shift

The locations of the text lines are vertically shifted to encode the secret information. In this method, alters the lines of text document by vertically shifting the position of the locations of text lines and may be applied to both the page image and the file format [3]. The codeword reassigned for a certain document specifies the text lines that will be moved in that document. We may have h “0” for a line shifted up and h “1” for a line shifted down. But also we may have h “-1” for a line shifted up, h “0” for an unmoved line and h “+1” for a line shifted down.

The length of each codeword that can be hidden is reduced, comparing to the technique that shifted every line, but the number can still be large.
For example, having a page with 40 lines, that is 2^{20} = 1,048,576 distinct codewords per page.

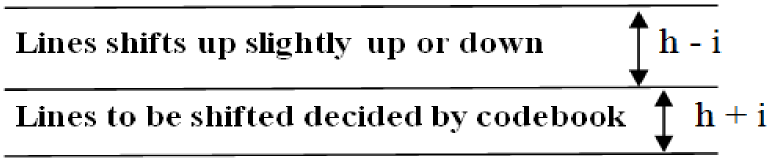

Lines shifts up slightly up or down
h - i
Lines to be shifted decided by codebook
h + i

Figure.4 Line shifting technique

Algorithm

Step 1: Start
Step 2: The encoder shift the lines up or down wanting to embed into the file.
Step 3: The decoder measures the distance between each pair of two neighboring lines.
a) Decoder measure the distance between the *baselines* of

adjacent lines (Baseline is a logical line on which the characters of a line sit).

b) Decoder measures the distance between the *centroids* of two adjacent lines (Centroid is the center of mass of a certain text line).

Step 4: if text lines *i-1* and *i+1* are not shifted and the line *i* is shifted either up or down. Then the distance between the baselines of two adjacent lines is constant.

Step 5: If hi-1 >h1 then line I is shifted down
If hi-1<h1 then the line I is shifted up
Otherwise uncertain

Step 6: Calculate the position of centroids

$$ci = \frac{\sum_{f=ti}^{bi}[j.n(j)]}{\sum_{f=ti}^{bi} n(j)}$$

where *i=1..N*, is the current line, *N* is the number of lines on the page, *ti* and *bi* are top and bottom limits of the line *i*, *n* is a function that counts how many pixels are ON (*f(k,j)=1*, *k =0…W*

Step 7: Calculate the distance between the centroids
If hi-1 – t i-1 > h1 - t1 line is shifted down
Otherwise line is shifted up

Step 8: Stop

But, in this technique, the distances can be discovering by using special instruments of distance assessment and necessary changes can be introduced to destroy the hidden information. If the text is retyped or character recognition programs are used, the hidden information would get destroyed. This method hides information by shifting the text lines to some degree to represent binary bits of secret information.

3. Word Shift

The locations of words within text lines are shifted horizontally to encode the document. In these method altering the document text shifting words horizontally and changing the distance between adjacent words space. These methods are useful for where the distance between words is varying. Variable word spacing is commonly used to distribute white space when justifying a file.

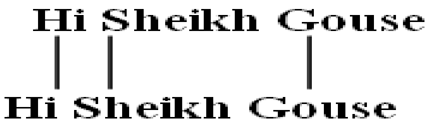


Figure.5 Word shifting

Algorithm

Step 1: Start

Step 2: The encoder determines if a line has sufficient number of words to encode; short lines are no encoded.

Step 3: On each encodable text line found is applied the differential encoding technique for this scheme.

Example: The second, fourth, sixth, etc. word from the left margin is displaced. The first and the last word on each line are unshifted to maintain the column justification.

Step 4: Suppose the *ith* word in text is shifted and the original centroids' positions are w*i-1*, w*i* and w*i-1* and the modified centroids' positions are w*i*-1 ' , w*i* ' and w*i*+1' then the algorithm calculates:

Dleft = wi – wi-1 D'left = w'i – w'i-1
Dright = wi+1 – wi D'right = w'i+1 – w'i

Step 5: If D'left - Dleft < D'right - Dright then the word i was shifted left
If D'left - Dleft > D'right - Dright then the word i was shifted right.

Step 6: After the process of shifting words is finished, the document is distributed.

Step 7: Stop.

But if someone was aware of these algorithms distances, then they can compare the present text with the algorithm and extract the hidden information by using the difference. The text image can be also closely studied to identify the changed distances. Although this method is very time consuming, there is a high probability of finding information hidden in the text. Retyping of the text or using OCR programs destroys the hidden information.

4. Feature Coding

Certain text features are altered to encode the secret information. Possible modifications are the extending or shortening of the upward vertical end lines of letters such as "d" or "h". With these the image is examined for chosen text features and those features are altered, depending on the mark inserted. Such features may be the vertical lines of the letters *b*, *d*, *h*, *k*, etc. The length of those lines may be modified in a way that is imperceptible to the ordinary readers. The character heights within a given font may also be changed. There are also techniques that change the words themselves substituting them with synonyms. Usually there are two pairs of synonyms and using one or the other synonym is equivalent with embedding a "0" or a "1". The two parties involved must share the synonymous pairs. The difference between those two techniques is that the first one can be used for embedding copyright information, but the second one only hides information, being adequate in the prisoners' problem. In the former all documents will have the same content, but some characters will be modified, in the latter two documents having different marks embedded will be different.

5. Open space methods

This method works because to a casual reader one extra space at the end of line or an extra space between two words does not prompt abnormality. However, open space methods are only useful with ASCII format the three methods of using white space to encode data [2].

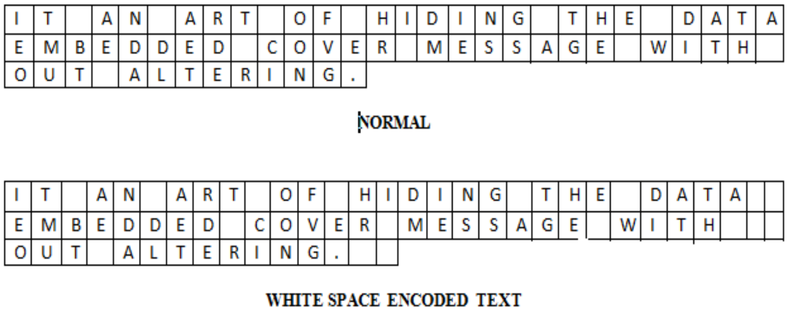
5.1 Inter-sentence spacing

This method encodes a "0" by adding a single space after a period in English prose. Adding two spaces would encode a "1". This method works, but needs a large amount of data to hide only little information. Some word processing tools automatically correct the spaces between

sentences. The first method encodes a binary message into a text by placing either one or two spaces after each terminating character, e.g., a period for English prose, a semicolon for C-code, etc.
A single space encodes a “0,” while two spaces encode a “1.” This method has a number of inherent problems. It is inefficient, requires a very few bits to encode. Finally, inconsistent use of white space is not transparent.

5.2 Line spacing

In this method, white space to encode data is to insert spaces at the end of lines. It allows a predetermined number of spaces at the end of each line. Two spaces encode 1 bit per line, 4 encode 2, 8 encode 3, etc., dramatically increasing the amount of information we can encode over the previous method Additional advantages of this method are that it can be done with any text, and it will go unnoticed by readers, since this additional white space is peripheral to the text. A problem unique to this method is that the hidden data cannot be retrieved from hard copy.
NORMAL WHITE SPACE ENCODED TEXT



I T A N A R T O F H I D I N G T H E D A T A
E M B E D D E D C O V E R M E S S A G E W I T H
O U T A L T E R I N G .

NORMAL

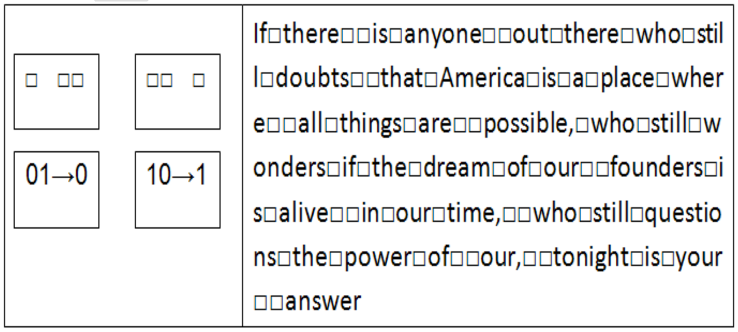
I T A N A R T O F H I D I N G T H E D A T A
E M B E D D E D C O V E R M E S S A G E W I T H
O U T A L T E R I N G .

WHITE SPACE ENCODED TEXT

Figure.3 End line spacing

5.3 Inter word spacing

In this method white space to encode data involves right-justification of text. Data are encoded by controlling where the extra spaces are placed. One space between words is interpreted as a “0.” Two spaces are interpreted as a “1.” It employs a Manchester-like encoding method i.e., interpreting “01” as a “ 1” and “10” as a “0.” The bit strings “00” and “11” are null. For example, the encoded message “1000101101” is reduced to “001,” while “110011” is a null string.



If□there□□is□anyone□□out□there□who□stil
l□doubts□□that□America□is□a□place□wher
e□□all□things□are□□possible,□who□still□w
onders□if□the□dream□of□our□□founders□i
s□alive□□in□our□time,□□who□still□questio
ns□the□power□of□□our,□□tonight□is□your
□□answer

Figure.6 Manchester encoding

B. LINGUISTIC STEGANOGRAPHY

It is concerned with making changes to a cover text in order to embed information, in such a way that the changes do not result in ungrammatical or unnatural text. Most of its methods use either lexical (semantic) or syntactic transformations or combination of both. The synonym substitution is the popular lexical steganography method. It substitutes the original word with another word that possesses mostly the same meaning as the original word. The syntactic methods transform the grammatical style of the original sentences.

1. Lexical Steganography

A word could be replaced by its synonym and the choice of word to be chosen from the list of synonyms would depend upon secret bits. It used synonym replacement by using a word dictionary to get synonym. The secret text to be hidden is first compressed by Huffman Compression. It is based on word substitution over an IRC channel. The generation of the word substitution table is based on a session key and used synonyms from a public thesaurus.

Example: Hyderabad is a nice big city now, suppose list of synonyms for nice is {nice, wonderful, great, and decent}.

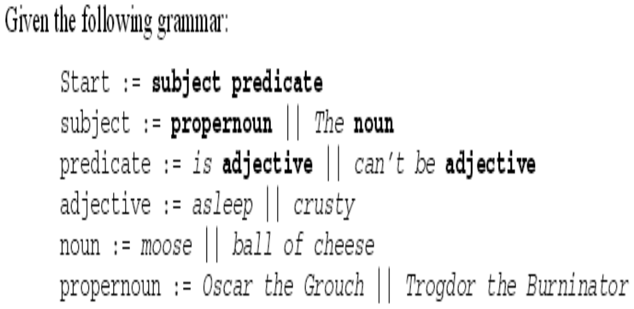
Word	Code
decent	00
great	01
nice	10
wonderful	11

Table 2. Binary code

Depending upon the input secret bits appropriate synonym for nice will be selected and put in the stego text. So, the possible stego texts could be:
a) Hyderabad is a nice big city.
b) Hyderabad is a wonderful big city.
c) Hyderabad is a decent big city.
d) Hyderabad is a great big city.
This method produces better quality text than syntactical techniques. It is hard to find presence of hidden & replacements are critical part .In the above mentioned synonym replacement approach, some words can have more than one sense. (Noun —bank‖ has two senses a long pile or heap or an institution for receiving, lending, and safeguarding

2. Syntactic Steganography

CFG create tree structure which can be used for concealing the bits where left branch represents '0' and right branch corresponds to '1'. A grammar in GNF can also be used where the first choice in a production represents bit 0 and the second choice represents bit 1.This method has some drawbacks. First, a small grammar will lead to lot of text repetition. Secondly, although the text is syntactically flawless, but there is a lack of semantic structure. These approaches make use of Context Free Grammars (CFG) to build syntactically correct sentences. There is another famous algorithm, NICETEXT , also based on CFG. It examined two highly predictable and reasonably common grammatical phenomena in English that can be used in data hiding, the swapping of complementisers and relativisers, which rely on a well-established technology: syntactic parsing. The other people explored the morph syntactic tools for text watermarking and developed a syntax-based natural language watermarking scheme in [5]. The unmarked text is first transformed into a syntactic tree diagram in which the syntactic hierarchies and the functional dependencies are coded. The watermarking software then operates on the sentences in syntax tree format and executes binary changes under control of Wordnet to avoid semantic drops.



Given the following grammar:

```
Start := subject predicate
subject := propernoun || The noun
predicate := is adjective || can't be adjective
adjective := asleep || crusty
noun := moose || ball of cheese
propernoun := Oscar the Grouch || Trogdor the Burninator
```

Figure.7 Syntactic grammar

C. RANDOM AND STATISTICAL GENERATION

In order to avoid comparison with a known plaintext, steganographers often resort to generating their own cover texts [7]. One method is concealing information in random looking sequence of characters. In another method, the statistical properties of word length and letter frequencies are used in order to create words which will appear to have same statistical properties as actual words in the given language [2, 3].

V. OTHER TEXT STEGANOGRAPHY TECHNIQUES

A. White Steg

This technique uses white spaces for hiding a secret message. There are three methods of hiding data using white spaces. In Inter Sentence Spacing, it place single space to hide bit 0 and two spaces to hide bit 1 at the end of each terminating character [9]. In End of Line Spaces, fixed number of spaces is inserted at the end of each line. For example, two spaces to encode one bit per line, four spaces to encode two bits and so on. In Inter Word Spacing technique, one space after a word represents bit 0 and two spaces after a word represents bit 1. But, inconsistent use of white space is not transparent [9].

B. SNOW Steganography

It exploits the Steganographic Nature Of Whitespace(SNOW). Locating trailing whitespace in text is like finding a polar bear in a *snow*storm. he encoding scheme used by **snow** relies on the fact that spaces and tabs (known as *whitespace*), when appearing at the end of lines, are invisible when displayed in pretty well all text viewing programs. This allows messages to be hidden in

ASCII text without affecting the text's visual representation. And since trailing spaces and tabs occasionally occur naturally, their existence should not be sufficient to immediately alert an observer who stumbles across them.

C. Spam Text

HTML and XML files can also be used to hide bits. If there are different starting and closing tags, bit 0 is interpreted and if single tag is used for starting and closing it, then bit 1 is interpreted [12,15]. In another technique, bit 0 is represented by a lack of space in a tag and bit 1 is represented by placing a space inside a tag [15].

D. SMS-Texting

SMS-Texting language is a combination of abbreviated words used in SMS [8]. We can hide binary data by using full form of word or its abbreviated form. A codebook is made which contains words and their corresponding abbreviated forms. To hide bit 0, full form of the word is used and to hide bit 1, abbreviated form of word is used [8].

E. SSCE (Secret Steganographic Code for Embedding)

This technique first encrypts a message using SSCE table and then embeds the cipher text in a cover file by inserting articles a or an with the non specific nouns in English language using a certain mapping technique [5]. The embedding positions are encrypted using the same SSCE table and saved in another file which is transmitted to the receiver securely along with the stego file.

F. Word Mapping

This technique encrypts a secret message using genetic operator crossover and then embeds the resulting cipher text, taking two bits at a time, in a cover file by inserting blank spaces between words of even or odd length using a certain mapping technique [12]. The embedding positions are saved in another file and transmitted to the receiver along with the stego object.

G. MS Word Document

In this technique, text segments in a document are degenerated, mimicking to be the work of an author with inferior writing skills, with secret message being embedded in the choice of degenerations which are then revised with changes being tracked. Data embedding is disguised such that the stego document appears to be the product of collaborative writing [13].

H. Cricket Match Scorecard

In this method, data is hidden in a cricket match scorecard by pre-appending a meaningless zero before a number to represent bit 1 and leaving the number as it is to represent bit 0 [7,17].

I. CSS (Cascading Style Sheet)

This technique encrypts a message using RSA public key cryptosystem and cipher text is then embedded in a Cascading Style Sheet (CSS) by using End of Line on each CSS style properties, exactly after a semicolon. A space after a semicolon embeds bit 0 and a tab after a semicolon embeds bit 1 [5,18].

VI. TEXT STEGANOGRAPHY TOOLS

Within the text steganographic area we survey a total of 15 tools, which included freeware, shareware, open source and commercial license products

Text Steganographic Tools	Plain Text	Other	Source code	License	Production
PGPn123		Yes		Shareware	Yes
Nicetext	Yes		Yes	Open Source	Yes
Snow	Yes		Yes	Open Source	Yes
Texto	Yes		Yes	Open Source	Yes
Sam's Big Play Maker	Yes		Yes	Open Source	Yes
Steganosaurus	Yes		Yes	Open Source	Yes
FFEncode	Yes			Open Source	Yes
Mimic	Yes			Open Source	Yes
wbStego	Yes	HTML, PDF	Yes	Open Source	Yes
Spam Mimic	Yes			Not specified	Yes
Secret Space	Yes			Not specified	
WitnesSoft	Yes	Yes		Not specified	Yes
MergeStreams		Hides excel file in word		Freeware	Yes
Steganos	Yes	HTML		Commercial	Yes

Table.3 Different steganography tools

VII. CONCLUSION

As steganography becomes more widely used in computing there are issues that need to be resolved. A wide variety of different techniques are discussed in present paper with their advantages and disadvantages. Many of currently used techniques are not robust enough to prevent detection and removal of embedded data. The use of benchmarking to evaluate techniques should become more common and more standard definition of robustness is required to help overcome this problem.

VIII. FUTURE SCOPE

Data hidden in text has a variety of applications, including copyright verification, authentication, and annotation. Making copyright information inseparable from the text is one way for publishers to protect their products in an era of increasing electronic distribution. Annotation can be used for tamper protection. For example, if a cryptographic hash of the paper is encoded into the paper, it is a simple matter to determine whether or not the file has been changed. Verification is among the tasks that could easily be performed by a server, which in this case would return the judgment "authentic" or "unauthentic" as appropriate. One of the possible uses of text-based steganography is reconstruction of printed document. This is done because if somehow the document is torn out then the important information will be lost. Extracting the information from the torn part of the document and recreate the document can help in regaining the lost information. Other uses of data hiding in text involve embedding instructions for an autonomous program in a text. For example, a mail server can be programmed to check for hidden messages when transmitting an electronic message. The message is rejected or approved depending on whether or not any hidden data are found. In this way a company running its own mail server can keep confidential documents from being inadvertently exported.

XI ACKNOWLEDGMENT

The authors would like to thank the Research & Development Laboratory in MLReddy Institute of Technology for providing software's. They would also like to thank acknowledge extensive comments from the reviewers and the anonymous associate editor.

X. REFERENCES

[1]. Bender, W., Butera, W., Gruhl, D., Hwang, R., Paiz, F.J. and Pogreb, S., (2000). *Applications for Data Hiding. IBM Systems Journal, 39 (3&4): 547-568.*

[2]. Petitcolas, F.A.P., (2000). *"Introduction to Information Hiding". In: Katzenbeisser, S and Petitcolas, F.A.P (ed.) (2000) Information hiding Techniques for Steganography and Digital Watermarking. Norwood: Artech House, INC.*

[3]. Wayner, P. (2002). *Disappearing Cryptography*. *2nd ed. USA: Morgan Kaufmann Publishers.*

[4]. Johnson and Katzenbeisser, (2000). *"A survey of Steganographic techniques". In: Katzenbeisser, S and Petitcolas, F.A.P (ed.) (2000) Information hiding Techniques for Steganography and Digital Watermarking. Norwood: Artech House, INC.*

[5]. S.H. Low, N.F. Maxemchuk, J.T. Brassil, and L. O'Gorman, "*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), 2-6 April 1995, vol.2, pp. 853 - 860.

[6]. W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "*Techniques for data hiding*", IBM Systems Journal, Vol. 35, Issues 3&4, 1996, pp. 313-336.

[7]. Richard Popa, "*An Analysis of Steganographic Techniques*", The Politehnica University of Timisoara, Faculty of Automatics and Computers, Department of computer science and Software Engineering. 1998.
[8]. T. Moerland, "*Steganography and Steganalysis*", May 15, 2003, www.liacs.nl/home/tmoerlan/privtech.pdf, last visited: 1 May 2006.

[9]. An Analysis of Steganographic Techniques by Richard Popa

[10]. Y. Kim, K. Moon, and I. Oh, "*A Text Watermarking*-Algorithm based on Word Classification and Inter- *Word Space Statistics*", Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR'03), 2003, pp. 775–779

[11]. M. Niimi, S. Minewaki, H. Noda, and E. Kawaguchi, "*A Framework of Text-based Steganography Using SD-Form Semantics Model*", Pacific Rim Workshop on Digital Steganography 2003, Kyushu Institute of Technology, Kitakyushu, Japan, July 3-4, 2003.

[12]. K. Rabah, "*Steganography-The Art of Hiding Data*", Information Technology Journal, vol. 3, Issue 3, pp. 245-269, 2004.

[13]. K. Bennett, "*Linguistic Steganography: Survey, Analysis, and Robustness Concerns for Hiding Information in Text*", Purdue University, CERIAS Tech Report 2004-13.

[14]. Jibran Ahmed Memon, Kamaran Khowaja, Hameedullah Kazi, "*Evaluation of Steganography for Urdu/Arabic Text*", Journal of Theoretical and Applied Information Technology, 2005 JATIT.

[15]. M.Hassan Shirali-Shahreza, Mohammad Shirali- Shareza, "*A New Approach to Persian/Arabic Text Steganography*", Proceedings of the 5th IEEE/ACIS International Conference on computer and Information Science, 2006 IEEE.

[16]. Adnan Abdul-Aziz Gutub and Manal Mahammad Fatami, "A Noval Arabic Text Steganography Method Using Letter Points and Extensions", Proceedings of world academy of science, engineering and technology, Vol. 21 May 2007 ISSN 130-6884.

[17]. Simmons, G. J., (1984). *The Prisoners' Problem and the Subliminal Channel. Proceedings of CRYPTO83- Advances in Cryptology, August 22-24. 1984. pp. 51.67.*

[18]. Kurak, C. and McHugh, J., (1992). A cautionary note on image downgrading. *Proceedings of the Eighth Annual Computer Security Applications Conference. 30 Nov-4 Dec 1992 pp. 153-159.*

[19]. Johnson, N. F. and Jajodia, S., (1998). *Exploring Steganography: Seeing the Unseen. IEEE Computer, 31 (2): 26-34, Feb 1998.*

[20]. Johnson Neil F., Zoran Duric, Sushil Jajodia, Information Hiding, Steganography and Watermarking - Attacks and Countermeasures, Kluwer Academic Publishers,2001.

[21]. Heckbert Paul, Colour Image Quantization for Frame Buffer Display. In Proceedings of SIGGRAPH 82, 1982.

AUTHORS DETAILS

1. Abhishek Koluguri , currently pursing B.Tech CSE final year in the Department of Computer Science and Engineering, MLRIT, Andhra Pradesh, India. His research interesting areas Data Mining, Network Security & Computer Networks.

2. Sheikh Gouse B.Tech, M.Tech CSE. He is currently working in the Department of Computer Science and Engineering, MLR Institute of Technology, Hyderabad, Andhra Pradesh, India. He is having 7.6 years of teaching experience**.** He is Certified in Oracle 9i: SQL & Java SE 6. His research interesting areas Programming (C&DS, C++, and JAVA), Data Mining, Software Engineering, Network Security & Computer Networks.

3. Dr. P. Bhaskara Reddy, B.E (ECE), M.Tech., Ph.D., F.I.S.E.E., MCSI, MISTE. He is presently the Director MLR Institute of Technology. He has 25 years of Teaching, Research and Administrative experience in reputed Engineering Colleges and Industry. He received Bharath Jyothi award in 2003 and Rastraprathiba award in 2004.He has acquired various positions from Asst. Professor to Principal. He published 9 laboratory manuals & 50 Research papers at National and International Level on Education, Electronics Communication, I.T., Computer Networks and E-Commerce. He guided 5 Research Scholars for their Doctorates.