Indiscernible Communication through ASCII Text Document/File (Communication in Veil)

Khan Farhan Rafat and Muhammad Sher

**Department of Computer science, International Islamic University
Islamabad, 44000, Pakistan**

Abstract

Internet has embarked comforting impact on people's lives across the globe as personal tasks such as cash transactions, mails, fax, data storage or retrieval etc. are now preferably being done electronically using personal computers/cell phones/TABs etc. This, however, has also elevated issues vis-à-vis availability, legitimacy, and renunciation of information causing discomfort in information security field because the number of cases including copyright infringement, heavy financial loss on account of e-theft of credit cards, impersonation through hacking of social IDs etc. are on increase, and calls for their immediate remedy. Steganography – the art and science of oblivious communication – has emerged as an exciting research area for expert and naïve academics now days to mitigate aforesaid security issues and is the prima facie of our research that focuses on evolving secure ASCII text-cover centric data hiding scheme for stealth communication.

Keywords: *ASCII Text Steganography, Covert Channel, Eccentric way of writing, Stealth Communication.*

1. Introduction

Take the situation of a person watching commercials on TV or adds in a magazine / NEWS paper where all of a sudden he/she gets pushed to buy that advertised item or get to that particular piece of interest without any compulsion or force or even without being asked to do so. This form of communication is referred to as oblivious communication and is induced in our lives to an extent that we take it as for granted and hardly realize its existence. However, when the intent is to hide personal data or some form of information from others, or to deliver it safely to another end without being noticed, the modus operandi is referred to as information hiding [1]. Steganography, a branch of information hiding, is not a new subject and dates back to 400 B.C. where Greeks pioneered the art [2] that has now been evolved into a science with digitization and the introduction of personal computers. Figure 1, derived from [3], illustrates on types of information hiding.

Earlier examples of steganography include wooden tablets, covered with wax, that carry hidden (engraved) messages and were retrieved by melting the wax [4]; an insurgent desirous to upheaval against the Persian king shaved the head of his trusted slave, tattooed a secret message on it, waited for his hair to grow and sent him to his allied group of trusties in that territory where the slave's head was once again shaved to retrieve embossed message [5]. WORLD WAR - II saw use of invisible ink and micro dot techniques that highlights on the significance of use of Steganography [6].

Modern Steganography is more centered on exploiting human auditory-visual System (HAVS)'s limitations [7] where text, images and multimedia audio/video contents have become preferred means of stealth communication [8]. In fact Steganography emerged as an alibi in regions where Government / Corporate etc. has imposed ban on cryptography for public usage [9].

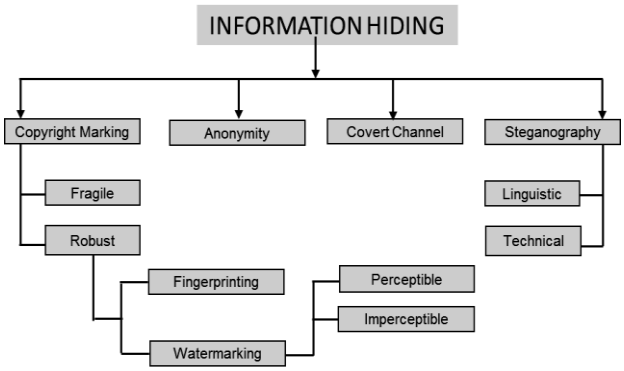

Fig. 1. Classifying Information Hiding.

ASCII Text-Cover centric steganography uses "text" as medium for covert communication. The distinguishing characteristic of text format from that of image, audio or video is that text files are viewed, saved and retrieved in a form analogous to what human eye perceives. Image, audio and video file formats, on the other end, have associated Meta data that dictates the manner in which that information is to be presented before end user. Further, changing a single bit of ASCII text character code results in another text code that has a dubious impact on the viewer e.g., misspelled word. It is because of the difficulty linked with ASCII text character codes that makes evolution of text – cover centered steganography a challenging task. Following briefly summarizes some text-cover based steganography schemes:

1.1 Text Content's Manipulation

In [10] authors demonstrated variety of unconventional alternates for hiding secret information in ENGLISH text through deliberate modifications including syntax errors, use of acronyms, articulating document format etc. as shown below:

a. Using intentional typographical mistake – writing "there" in place of "there"
b. Preferring "yr" rather than "your" and "TC" in place of "Take Care"
c. Formatting Text by inserting additional carriage returns or separating text into irregular paragraphs, or by adjusting line or word spacing.
d. Through annotating text e.g., :) expresses pun
e. Using multi lingual text – "we always commit the same mistakes again, and 'je ne regrette rien'!"

1.2 Use of Blank/Space Character

Authors in [11] have suggested scheme that hides secret binary bit 1 using a single space while secret binary bit 0 depicts a double space. Following example illustrates the concept: paper must simply type your text into it.

.

Example. Let 00101110 be the secret message bits and let the text cover selected for bit hiding purpose be "*A quick brown fox jumps over the lazy dog*". Going by analogy of suggested scheme secret binary message bit 0 will be replaced by double spaces in place of single white space character while no extra space is inserted for secret binary bit 1. After bit embedding process gets completed the resultant Stego Object takes the form as follows, where "▮" represents a double space:

A▮quick▮brown fox▮jumps over the lazy▮dog.

1.3 Paper Plot

This paper is planned as follows: Section 2 expounds on evaluation criteria while Section 3 elaborates on our proposed solution. Quantified test results and graphical outputs are illustrated in Section 4. Theoretical conception is given in Section 5. Section 6 highlights advantages and limitations of our proposed solution. Section 7 finalizes ongoing proceeding.

2. Evaluation Criteria

As stated earlier, steganography manipulates human's HAVS limitations, hence the best evaluation parameter towards gauging any steganographic system after system's security is its imperceptibility. However, it is apparent from [12] that Cachin's [13] notion of perfect security (given below) may only exact on the similarity/difference between cover text and stego object rather than describing system's security:

$$D(P_c \| P_s) \leftarrow \varepsilon \leftarrow \sum_{q \in Q} P_c(q) \log_2 \frac{Pc(q)}{Ps(q)} \quad \ldots\ldots\ldots (1)$$

, where Pc and Pq denotes probability distribution of Cover and Stego Object, It is obvious that $\frac{Pc(q)}{Ps(q)} = 0$ for $\frac{0}{Ps(q)}$ *and* ∞ $\frac{Pc(q)}{0}$

Hence, to achieve information theoretic security we opted for the model proposed by [14] as illustrated in Figure 2. Quantified test results are tabulated using mean, variance and standard deviation. Additionally, to graphically illustrate bit embedding effects on text cover, probability distribution plots of cover text and stego object are contrasted using MiniTab16 [15].

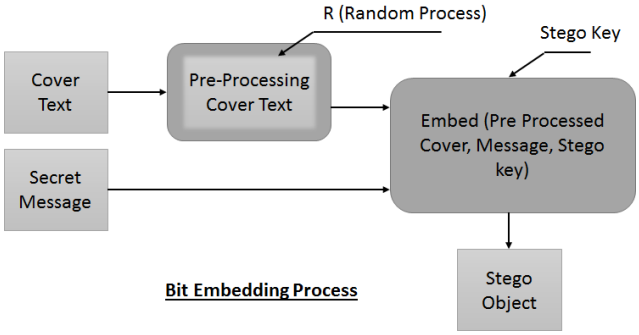

Fig. 2. Information Theoretically Secure Steganography Model.

3. Proposed Steganographic Scheme for ASCII Text Files

Instant inference on limitations of some existing methods for text steganography (Sect. 1 refers), less space manipulation scheme, include *perceptibility* (where erroneous words can catch viewer's attention), *commencement of bit embedding from beginning of text that continues in the same direction till end of secret bits*, and *no indication regarding type* (format) *of data and its length that is embedded inside cover text*. Further, in the *absence of stego key*, a careful study of just few words/sentences may also reveal bit embedding algorithm. Based on above findings we wanted our proposed scheme to:

a. Be in accordance with the evaluation criteria (Sect. 2 refers).
b. Adhere to Kerchoff's principle.

To meet our desired objectives we opted for the following (as explained in subsequent section):

- Use of symmetric stego key steganography.
- Reign in influence of stego key on bit embedding algorithm where 256-bit stego key length is preferred.
- Pre-processing of cover text before commencement of bit embedding process.

- To ensure 100% perceptibility we opted for insertion technique.

3.1 Design Considerations

With reference to peculiarity associated with ASCII text files i.e., contents are saved, and retrieved in manner in which these appear before human eye, we analysed ASCII text codes and found two such characters with codes *141* & *157* that even when made part of text file remains imperceptible (we shall refer to these ASCII codes as '***Stealth.Char***') upon its retrieval. Visual Basic 6 used as tool for embedding, saving, retrieving and extracting *Stealth.Char* from ASCII text file where access mode for read/write operation was *binary*.

The constraint linked with *Stealth.Char*(s), however, is that we can either interpret *141* as binary bit '0' and *157* as '1' or *vice versa* when hiding secret message bits which in absence of stego key can easily be comprehended by knowing bit embedding algorithm. Hence, to achieve variation in its interpretation during bit embedding we arranged Stealth.Char(s) as a 2 x 3 table and assigned these fixed values as shown in column two of the Table 1 leaving the third column as blank to be populated afresh with every new message for subsequent interpretation during bit embedding process.

Table 2: Quantified Test results

Stealth.Char	Assigned bit value	Key dependent interpretation
141	0	
157	1	

Next we added stego key byte values and reduced the result to modulo 32, the outcome of which was a pointer to the byte in stego key who's last bit i.e. 0/1 would decide either to retain or swap the prefixed bit pattern for interpreting Stealth.Chars. Figure 3 illustrates the said steps.

3.2 Processing the Stego Key

Keeping in view the salient characteristic of HASH function where a single bit change in input induces a change in at least half of the total output bits, the 32-bit stego key serves as input to SHA-2 HASH algorithm [16] and corresponding 256 bits stored/retained by counting number of ones' (binary bit 1). If *number of binary bit 1* ≤ (*bits in secret message* + 64) then processes of bit embedding/extraction may commence. In case of otherwise, SHA-2 output serves as feedback (input) till the desired condition for binary bit 1 is met. For multiple iterations output from SHA-2 gets concatenated with previous generated HASH.

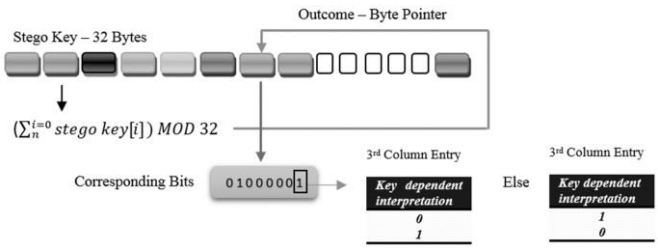

Fig. 3. Key dependent Stealth.Char interpretation in terms of binary bits.

3.3 Modus operandi

With preliminary work done, bit embedding and extraction processes are performed as under:

1. ***Bit-Embedding****:* Bit embedding commences by:

a. Encrypting secret message with stego key through XoR operation. The stego key is repeatedly used in case message length exceeds stego key length.

b. Translating secret message (*which is now encrypted*) into equivalent bits and storing its type (i.e. extension) and length in an eight byte header with four bytes reserved for each entry. The eight byte header is also translated into equivalent bits and attached as prefix to secret message bits.

c. SHA-2 generated HASH bits are traversed from left to right and the 'space' corresponding to binary bit 1 is marked for embedding encrypted bits.

d. In place of starting bit embedding from the beginning of ASCII cover text file, a random starting point for the said purpose is obtained using following equation:

$$\left(\left(\sum_{i=0}^{n} \text{stego key} * (i+1)\right) \text{MOD } 65537\right) \text{MOD Total\_Spaces} + 1 \quad \ldots (2)$$

If the result is 1 the process of bit embedding continues normally otherwise it gets completed in two steps. In first step bit embedding commences from the point of insertion till end of encrypted message bits. In step two, bit embedding starts from the beginning of cover text file and continues till the point of commencement (Equation 2 refers) is reached or where the encrypted message bits get exhausted as the case may be.

e. Cover text is iterated taking encrypted bits in sequence. On finding the 'space' marked for bit embedding, Stealth.Char from column 3 of Table 1 corresponding to encrypted bit replaces the Stealth.Char attached with that 'space' during pre-processing stage. The process terminates when all encrypted bits gets embedded in cover text.

2. ***Bit-Extraction****:* Bit extraction commences by processing stego key (as in Sec. 3.2) followed by:

a. SHA-2 generated HASH bits are traversed from left to right and the 'spaces' in stego object corresponding to

binary bit 1 are marked as location containing hidden (encrypted) bit.

b. Point for traversing stego object for extracting hidden bits is obtained as given in para *1 (d)* above.
c. The Stealth.Char(s) attached with each of the marked 'spaces' are replaced by the corresponding bits given in column 3 of Table 1.
d. First 32 bits thus extracted gives hidden message type while the next 32 its length.
e. Remaining extracted bits are XoR-ed with the stego key to get decrypted output.
f. The output (hidden message) is then translated into equivalent bytes and saved in appropriate file format.

4. Test Results

In one of the experiments the perceptibility of stego object contrasted with cover (extracted from http://en.wikipedia.org/wiki/Steganography) and pre-processed cover text is illustrated vide Figures 4 – 6, Jaro-Winkler distance [17][18] computed equated to **0.9563** while similarity/difference observed between the three is shown by plotting probability distribution graphs using MiniTab 16 for which their mean, variance and standard deviation were calculated as shown in Table 2 and illustrated in Figures 8 - 10 respectively. We experimented with 75 ASCII cover text files of varied lengths and found the results close to those as exemplified.

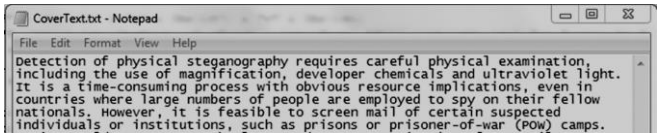

Fig. 4. ASCII Cover Text.

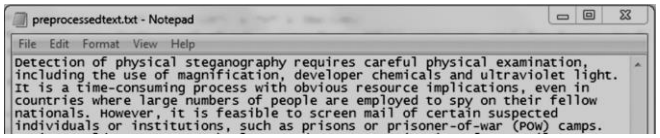

Fig. 5. Pre-Processed ASCII Cover Text.

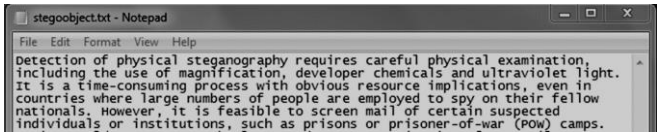

Fig. 6. Stego Object.

Figure 7 shows directory listing for the aforementioned three files where each is shown to have a file length of 3KB:

Fig. 7. Directory Listing for Cover, Pre-Processed Text and Stego Object.

Table 2: Quantified Test Results

COMPUTATION	COVER TEXT	PRE-PROCESSED COVER TEXT	STEGO OBJECT
Mean	0.092388	0.099824	0.099765
Variance	0.881894	1.142325	1.136536
STD	0.029696	0.033798	0.033712

5. Theoretical Conception

For security to prevail the uncertainty about bit embedding must not get revealed merely on the basis of knowledge about cover text and stego object. Undoubtedly if 'Eve/Wendy' can spot differences between cover text and stego object then they can also break the system. This, however, can only be possible when the differences are caused by bit-embedding alone. Hence, security in steganography can be achieved through arbitrary selection of cover text and then pre-processing it via some hard to predict random process before applying bit-embedding over it under the control of stego key but without compromising on its perceptibility.

6. Advantages and Limitations

1. Advantages of our proposed scheme include:

   i. Imperceptibility of stego object.
   ii. Information Theoretic Security.
   iii. Hidden message's type and length known before commencement of bit-extraction process.
   iv. Key dependent arbitrary starting point to initiate bit embedding.

2. Some of the limitations are:

   i. Increased stego object file size *equivalent to number of spaces in cover file*.
   ii. Opening the stego object via applications other than Windows 'Notepad' may result in unintelligible text.

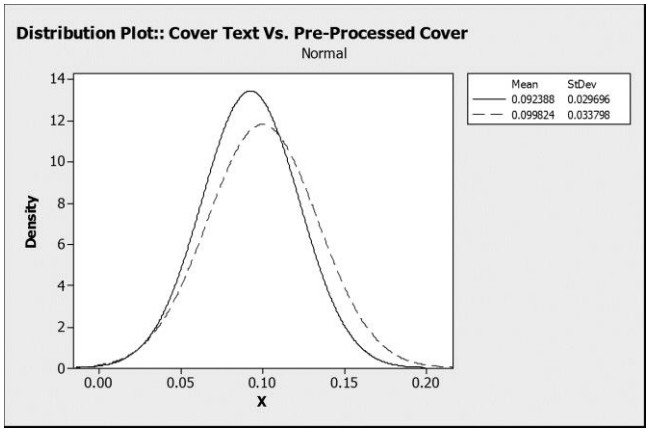

Fig. 8. Contrasting Probability Distribution Plots of Cover Text and Pre-Processed Cover Text.

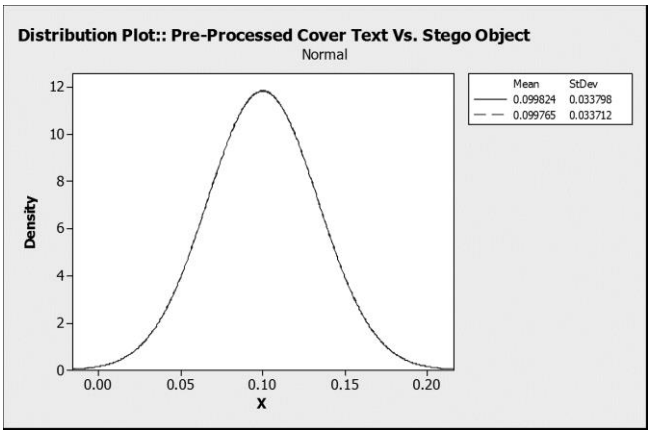

Fig. 9. Contrasting Probability Distribution Plots of Pre-Processed Cover Text and Stego Object.

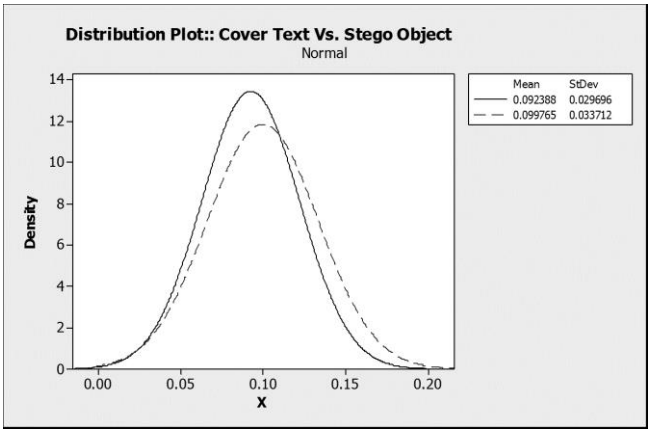

Fig. 10. Contrasting Probability Distribution Plots of Cover Text and Stego Object.

7. Conclusion

Less printing cost, high transmission efficacy, low resource occupancy and completeness in its semantics are some salient attributes that has made ASCII text documents the most commonly opted type of media in communication. However, lack of redundant information as well as non-alterability of alphabets while writing a character or word has made ASCII text document a difficult but challenging choice as Carrier/Cover for the purpose of information hiding in contrast to other media such as video, audio, image etc. having meta data that is/can be easily manipulated for the said purpose.

This research presented a secure steganographic scheme by inserting indiscernible characters corresponding to secret message bits in ASCII text document/file and is also in accordance with Kerchoff's principle.

References

[1] Pfitzmann, B., "Information Hiding Terminology," Proc. of First Internet Workshop on Information Hiding, pp. 347-350, Cambridge, UK, 1996.

[2] Stefan Katzenbeisser and Fabien A.P. Petitcolas, Introduction to information hiding. In Information Hiding: Techniques for Steganography and Digital Watermarking, Artech House. 1-14, Boston: 2000.

[3] Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. Kuhn, Information Hiding - A Survey,Proceedings of the IEEE, special issue on protection of multimedia content, 87(7):1062-1078, July 1999.

[4] Herodotus, The Histories, Penguin Books, London, 1996. Translated by Aubrey de Selincout

[5] N. Provos and P. Honeyman, "Hide and Seek: An Introduction to Steganography", IEEE Security & Privacy Magazine, Vol. 1, issue 3, pp. 32-44, June. 2003.

[6] N.F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen," Computer, vol. 31, no. 2, pp.26–34, 1998.

[7] Der-Chyuan Lou and Chia-Hung Sung. "A steganographic scheme for secure communications based on the chaos and Euler Theorem", IEEE Transactions on Multimedia, Volume 6 Issue 3, pp. 501-509, June 2004.

[8] Miroslav Goljan and Andreas Westfeld. "Secure Steganography in Multimedia Content", EURASIP Journal on Information Security, 2009:257131 doi:10.1155/2009/257131.

[9] Jenny Shearer and Peter Gutmann. Government, "Cryptography, and the Right to Privacy." Journal of Universal Computer Science (J.UCS), Volume 2, No.3, p.113, March 1996.

[10] Mercan Topkara, Umut Topkara, Mikhail J. Atallah. "Information Hiding through Errors: A Confusing Approach", 2007. Internet: http://umut.topkara.org/papers/ToToAt_SPIE07.pdf, [July 12, 2012]

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

[12] Khan Farhan Rafat and M.Sher, "On the Limits of Perfect Security for Steganographic System", International Journal of Computer Science Issues, Vol. 4, July, 2013. www.ijcsi.org. **Accepted – *Under Publication***

[13] Cachin, C. "An Information-Theoratic Model for Steganography," Proceedings of the Second International Workshop on Information Hiding, vol. 1525 pp. 306-318, Lecture Notes in Computer Science, Springer, 1998.

[14] J.Z Llner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G.Wicke, G.Wolf. "Modeling the security of steganographic systems," Proc. 2nd Workshop on Information Hiding, pp. 345-355, LNCS 1525, Springer-Verlag, Portland, 1998.

[15] Minitab 16. http://www.facebook.com/Minitab [June 6, 2012]

[16] T Hansen - 2006, "Secure Hash Algorithms (SHA and HMAC-SHA)," http://tools.ietf.org/html/rfc4634 [june 2012]

[17] Jaro, M. A., "Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florid," Journal of the American Statistical Association 84:414–420, 1989.

[18] Winkler, W. E., "The state of record linkage and current research problems," Statistics of Income Division, Internal Revenue Service Publication R99/04, 1999. http://www.census.gov/srd/www/byname.html.

KHAN FARHAN RAFAT is a Ph.D. Scholar at International Islamic University, Islamabad – Pakistan. He did his MCS from Gomal University, D.I.K. followed by MS in Telecommunication Engineering from UMT, Lahore – Pakistan. A veteran of information security having an experience of almost about 24 years has worked in varied roles in areas that are not limited only to programming, evaluation & analysis of Software/Hardware based security modules, and formulating security policies.

Professor Dr. Muhammad Sher is Dean Faculty of Basic and Applied Sciences at International Islamic University, Islamabad – Pakistan. He received B.Sc. degree from Islamia University Bahawalpur and M.Sc. degree from Quaid-e-Azam University, Islamabad, Pakistan. He did Ph.D. in Computer Science and Electrical Engineering from TU Berlin, Germany His area of research is Next Generation Networks Security. An eminent Scholar who has a number of research publications to his credit.