International Journal of the Physical Sciences Vol. 6(36), pp. 8130 - 8142, 30 December, 2011
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.1232
ISSN 1992 - 1950 ©2011 Academic Journals

Full Length Research Paper

An enhanced embedding method using inter-sentence, inter-word, end-of-line and inter-paragraph spacing

Lip Yee Por[1]*, Kok Onn Chee[1], Tan Fong Ang[1] and Delina Beh[2]

[1]Faculty of Computer Science and Information Technology, University of Malaya, Lembah Pantai, 50603 Kuala Lumpur, Malaysia.
[2]Malaysian Institute of Information Technology, Universiti Kuala Lumpur, 50250, Kuala Lumpur, Malaysia.

Accepted 18 October, 2011

This paper discusses a novel embedding mechanism which uses hybrid methods of inter-sentence, inter-word, end-of-line and inter-paragraph spacing to embed the secret message. From the test cases analysis, the proposed system has proven that it is able to hide larger capacity compared to other current steganographic tools such as wbStego4open, SNOW and Spacemimic. From the experimental results, the proposed method is able to improve the embedding capacity approximately 1 to 2 times compared to other steganographic tools by using plain text as payload. In terms of using image files as payload, the proposed method is also able to reduce the stego-text produced from 0.1 to 1 time as compared to other steganographic tools when the payload size is less than 100 kB.

Key words: Inter-sentence, inter-word, end-of-line, inter-paragraph, steganography, information hiding.

INTRODUCTION

The hiding of information technique, which aims to transfer information secretly and to establish a hidden relationship between the message and its counterpart, has long been of a great interest to researchers (Hmood et al., 2010). In this context, steganography is a type of information hiding technique which is able to conceal a secret data in a medium (that is, text, image, audio and video) and then the stego-object will be sent via a public communication channel without raising any suspicion (Katzenbeisser and Petitcolas, 2000; Frank, 2007). Figure 1 shows the steganography classification in which it can be categorised into technical steganography and linguistic steganography (Johnson and Jajodia, 1998; Kessler, 2004). Technical steganography is a steganographic method that uses scientific methods such as invisible ink, microdots, hollow heels and other size-reduction methods to conceal a secret message (Kessler, 2004; Bauer, 2007). On the other hand, linguistic steganography uses language as the cover medium in order to hide the secret message (Kipper, 2003; Benette, 2004; Gutub and Fattani, 2007). According to Kessler (2004), linguistic steganography can be subdivided into semagrams and open codes categories. In open codes category, message is hidden in an innocent carrier medium which is not obvious to any adversary (Kessler, 2004). Open codes can be divided into jargon code and covered cipher classifications. Jargon code utilises terminology that is only understood by a specific group or profession to hide a secret message whereas covered cipher hides a secret message via a cover medium in plain sight. According to Kessler (2004), covered cipher can be classified into null cipher and grille cipher. Null cipher hides a secret message according to some predefined rules whereas a grille cipher utilises a template that is used as a cover for the secret message (Kessler, 2004). Since the main focus of this paper is on text semagrams, the details of the open codes category will not be discussed further.

According to Kessler (2004), semagrams employ symbols or signs to hide information. Visual semagrams and text semagrams are the subdivisions of semagrams. Visual semagrams use unsuspicious objects such as doodles or the item's positioning on a desk or website to transmit a message. However, text semagrams modify

*Corresponding author. E-mail: porlip@um.edu.my.

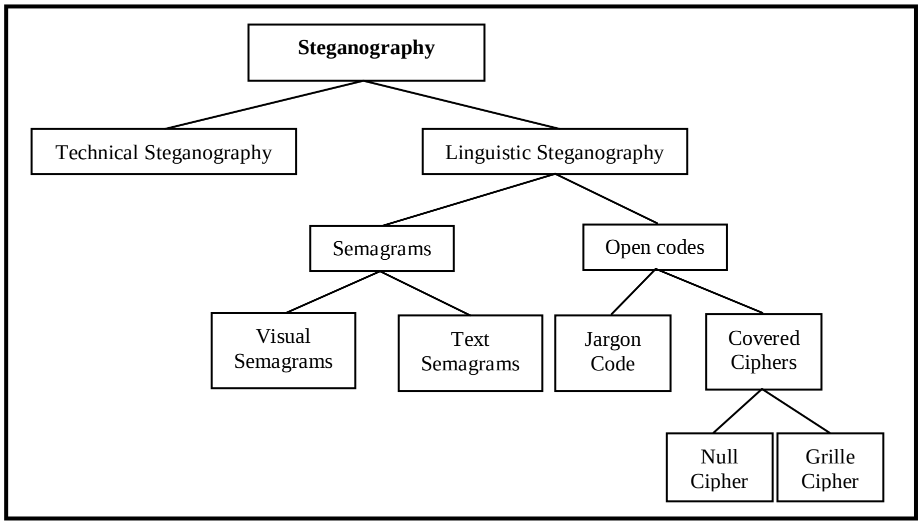


Figure 1. Classification of steganography techniques [adapted from (Kessler, 2004)].

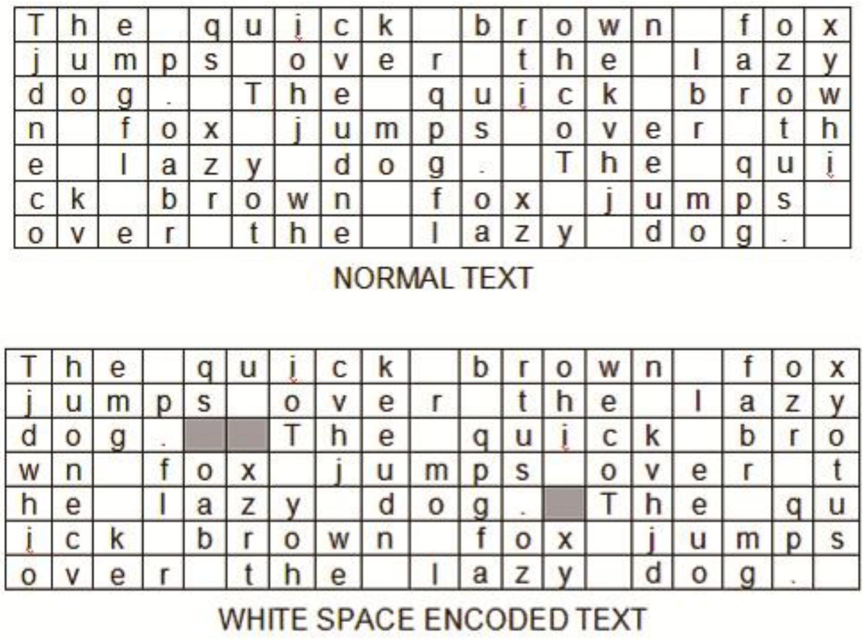
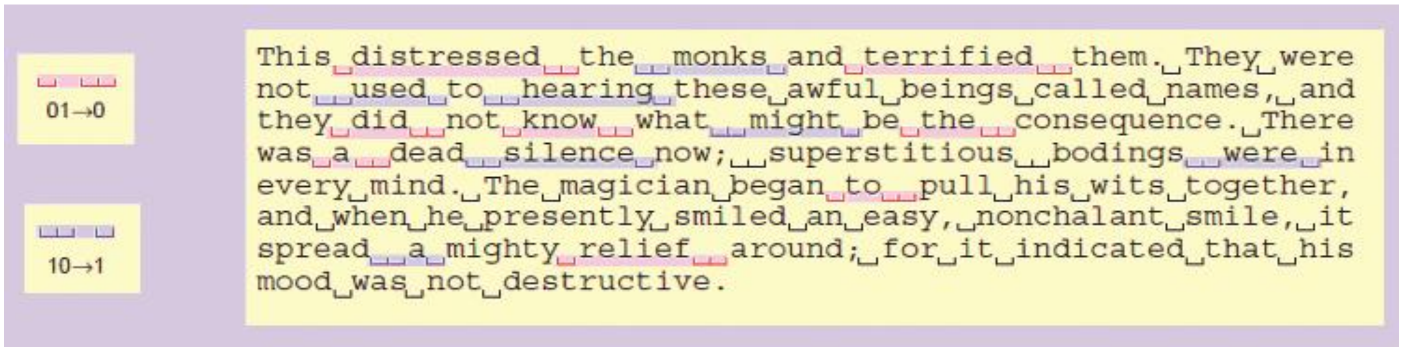


Figure 2. Inter-sentence spacing method [adapted from (Bender et al., 1996)].

the appearance of the cover-text (carrier medium) such as adding extra spaces to hide secret messages. Open space method proposed by Bender et al. (1996) is the earlier instance of text semagrams. Open space method encodes the secret message by manipulating the white space (unused space on a printed page) in the text (Bender et al., 1996). There are three methods proposed by Bender et al. (1996) to conceal a secret message and they are inter-sentence spacing, inter-word spacing in justified text, and end-of-line spaces methods. Other semagram approaches can be found in Roy and Manasmita. (2011), Wu and Liu. (2002), Liu and Tsai. (2007), Khairullah (2009), and Chotikakamthorn (1998).

Critical review

According to Bender et al. (1996), in order to hide the binary formed secret message into a text passage, the inter-sentence spacing method inserts one or two spaces after every terminating character such as a period in English text, a semicolon for C-code and etc. Figure 2



This distressed the monks and terrified them. They were not used to hearing these awful beings called names, and they did not know what might be the consequence. There was a dead silence now; superstitious bodings were in every mind. The magician began to pull his wits together, and when he presently smiled an easy, nonchalant smile, it spread a mighty relief around; for it indicated that his mood was not destructive.

Figure 3. Inter-word spacing in justified text method [adapted from (Bender et al., 1996)].

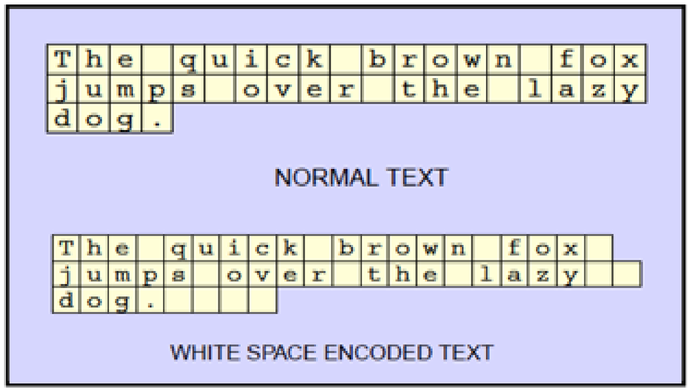

The quick brown fox jumps over the lazy dog.
NORMAL TEXT
The quick brown fox jumps over the lazy dog.
WHITE SPACE ENCODED TEXT

Figure 4. End-of-line spacing method [adapted from (Bender et al., 1996)].

shows the example of the inter-sentence spacing method. During the encoding, a single space represents "0" bit whereas two spaces are used to represent "1" bit.

In inter-word spacing method, the secret message is embedded using the spaces between words. The inter-word spacing method utilises a single space between words to represent "0" bit and two spaces to represent "1" bit. Besides, Manchester-like encoding method is used to determine the inter-word spaces when concealing secret message bits. In Manchester encoding, "01" and "10" is define as "1" bit and "0" bit respectively while "00" and "11" are not to be used for concealing any data (Figure 3).

In end-of-line spacing method, the secret message is hidden at the end of each line by inserting extra spaces. Unlike the inter-sentence and inter-word spacing methods, the end-of-line spacing allows two spaces to conceal one bit of secret message, or four spaces to conceal two bits of secret message, or eight spaces to conceal three bits of secret message and etc (Figure 4).

In 2008, Por and Delina (2008) proposed an inter-paragraph spacing method. The inter-paragraph spacing method conceals the secret message by inserting spaces between two newlines or an empty line between paragraphs (Figure 5). According to the authors, the proposed method is able to increase the embedding capacity compared to the method proposed by Bender et al. (1996). Apart from semagram and open codes, Natural Language Processing (NLP) is a linguistic technique for designing information hiding system using natural language text (Topkara et al., 2006; Bergmair, 2004). Some of the common NLP-based data embedding are synonym substitution (Topkara et al. 2006), context free grammar (Wayner, 2002) and synonymous paraphrasing (Calvo and Bolshakov, 2004).

Problem statement

In general, most of the data hidden methods using open space method are suffering from the limitation of the size. All the aforementioned methods require a relatively large cover-text in order to be sufficient for hiding only a few bits of the secret message. Consequently, the afore-mentioned methods are not convincing and incapable to conceal a large size secret message.

Objective of research

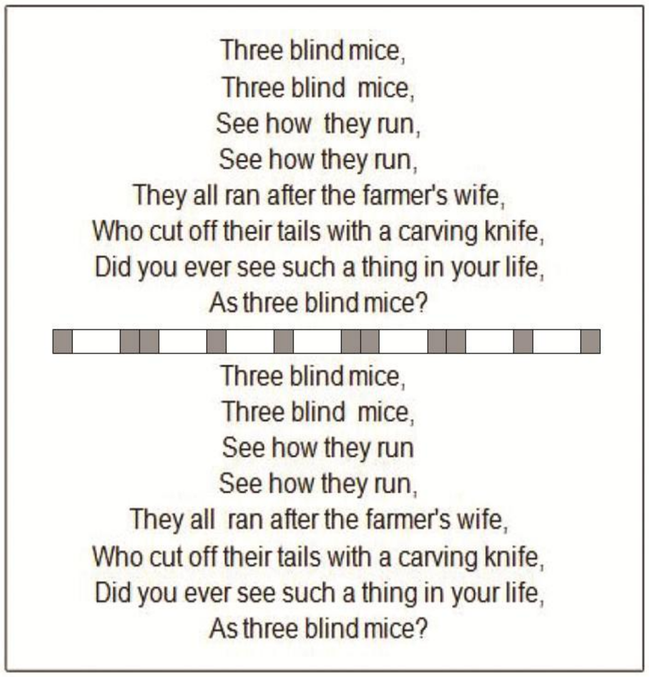
Therefore, an enhanced embedding method of utilising various types of white spaces is proposed to: (1) Reduces the large volume of stego-text produced, and (2) Improve the security of the embedding processes when hiding secret message.

MATERIALS AND PROPOSED METHOD

In general, two processes such as Steg (encoding) and DeSteg (decoding) are proposed. Figure 6 shows the encoding process of the proposed system.

Steg process

In the Steg process, the sender is prompted to browse for the file (secret data) that is going to be embedded. If the selected file does



Three blind mice,
Three blind mice,
See how they run,
See how they run,
They all ran after the farmer's wife,
Who cut off their tails with a carving knife,
Did you ever see such a thing in your life,
As three blind mice?

Three blind mice,
Three blind mice,
See how they run
See how they run,
They all ran after the farmer's wife,
Who cut off their tails with a carving knife,
Did you ever see such a thing in your life,
As three blind mice?

Figure 5. Inter- paragraph spacing method [adapted from (Por et al., 2008)].

not exist, a warning message will notify the sender. Then, the sender will be prompted to select another file. After a file is selected, the sender can add two optional features to the secret data, which are encryption and also selection of his own cover-text. If the secret data is successfully identified, the sender is required to choose whether he/she wants to use (option 1) the auto content generation feature or (option 2) the content which is able to be customised by the user to embed the identified secret data. Once the option has been chosen, the secret data will be compressed using deflate compression and then the bytes data will be converted into string using the base64 encoding.

$$x = \frac{4A}{3} \qquad (1)$$

A is the capacity of the secret data (byte), and x is the capacity of the secret data after base64 encoding (exclude the secret data file extension).

The total capacity for the identified secret data is denoted as

$$x + \alpha \qquad (2)$$

where α is the extension of the original file.

If the user chooses option 1: The system will identify the minimum capacity of the white-spaces required to accommodate the ($x+\alpha$) bytes of the secret data using the following formula:

$$y_1 \geq x + \alpha \qquad (3)$$

y_1 is the capacity of the white-spaces that is identified from a predefined cover-text.

Although the previously proposed formula is able to optimise the size of a generated text file, it may raise suspicion when an incomplete word or sentence is produced. Example:

… the Han Dynasty when incompetent eunuchs deceived the emperor and banished good officials. The government had become extremely corrupt on all levels, leading to widespread deterioration of the empire. During the reign of the penultimate Han emperor, Emperor Ling, the Yellow Turban

In order to alleviate the aforementioned issue, the proposed formula has been modified as follows:

$$y_1 + \beta \geq x + \alpha \qquad (4)$$

β is the minimum capacity of extra text required to construct a full sentence. Example

… the Han Dynasty when incompetent eunuchs deceived the emperor and banished good officials. The government had become extremely corrupt on all levels, leading to widespread deterioration of the empire. During the reign of the penultimate Han emperor, Emperor Ling, the Yellow Turban **Rebellion broke out under the leadership of Zhang Jiao, who allegedly practiced Taoist wizardry.**

If the user chooses option 2, the system will prompt a graphical user interface (GUI) and the user is required to input his/her character/string/text. The system will then identify the capacity of the white-spaces produced by the user based on his/her customised character/string/text. If the customised character/ string/text is insufficient to accommodate the ($x+\alpha$) bytes of the secret data, the customised character/string/text is then to be duplicated based on the following algorithm:

$$ny_2 + (n-1) \geq x + \alpha \text{ ; } n = 1, 2, 3 \ldots n \qquad (5)$$

y_2 is the capacity of the white-spaces produced by the customised character/string/text, and n is the minimum number of duplication required for [$ny_2 \; + (n-1)$] to accommodate the ($x+\alpha$) bytes of secret data.

After the cover text (generated/customised text file) has been generated, the user then is able to choose whether he/she wants to use (option A) encryption or (option B) without encryption to embed the secret data. If the user chooses option A, the system will then encrypt the secret data ($x+\alpha$) with a user-defined public key. The encrypted secret data ($x+\alpha+8$) will then undergo the base64 encoding. After the encoding process, the capacity of the secret data becomes $\frac{4(x+\alpha+8)}{3}$ bytes. For the user who selects the auto content generation feature, the minimum capacity of the white-spaces required to accommodate the $\frac{4(x+\alpha+8)}{3}$ bytes of secret data can be identified using the following formula:

$$z_1 \geq \frac{4(x+\alpha+8)}{3} \qquad (6)$$

z_1 is the capacity of the white-spaces that identified from a predefined cover-text; as mentioned previously, in order to alleviate the suspicion arisen due to the incomplete word or sentence produced, the proposed formula has been modified as follows:

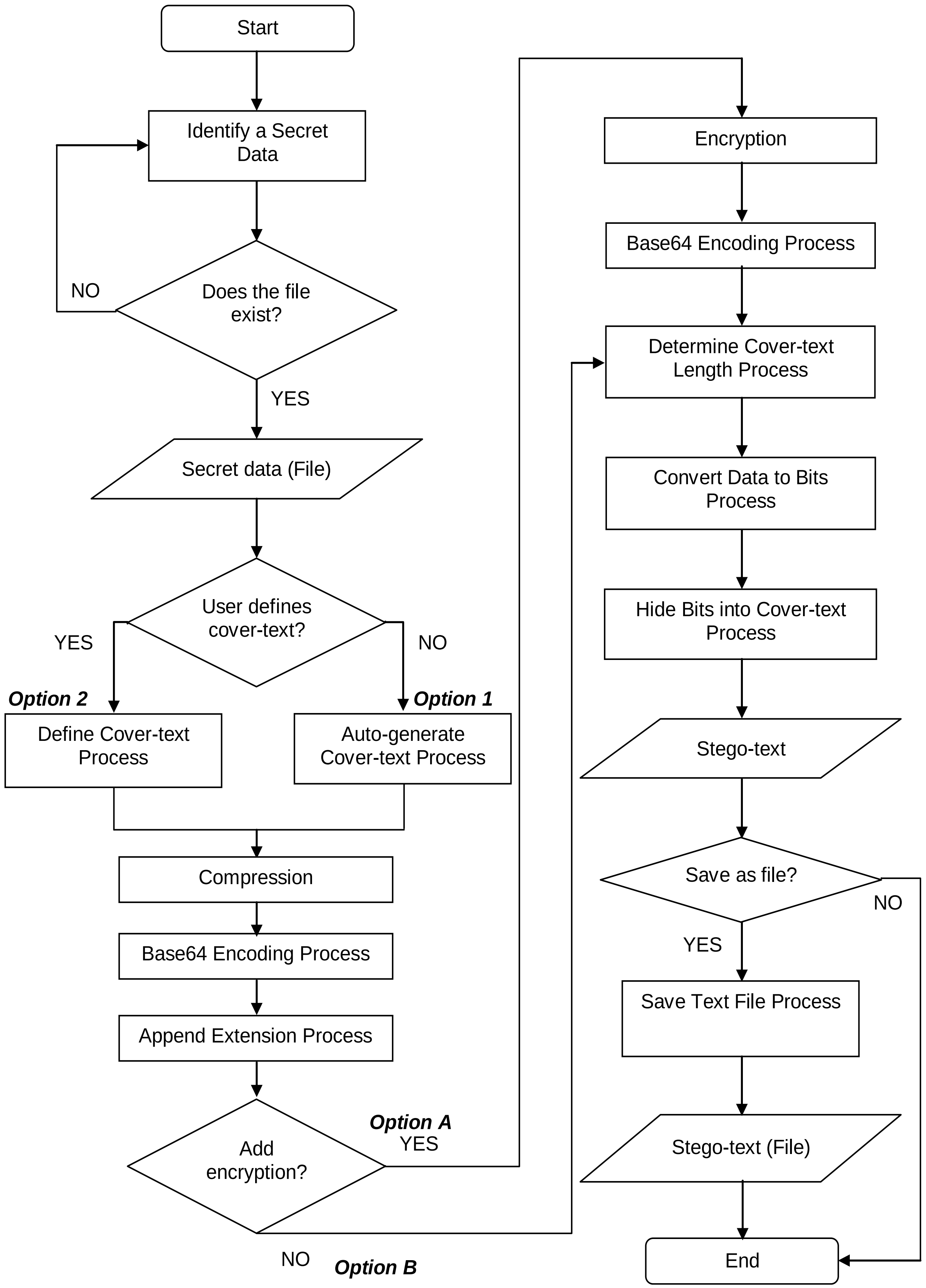


Figure 6. Encoding process.

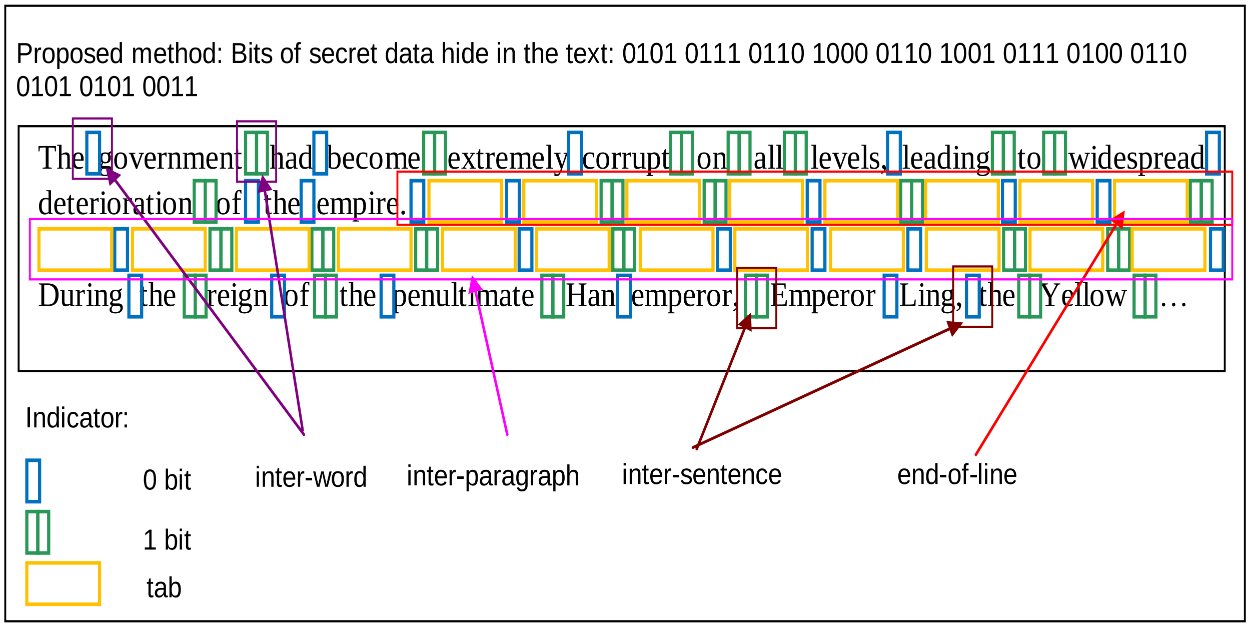

Proposed method: Bits of secret data hide in the text: 0101 0111 0110 1000 0110 1001 0111 0100 0110 0101 0101 0011
The government had become extremely corrupt on all levels, leading to widespread deterioration of the empire.
During the reign of the penultimate Han emperor, Emperor Ling, the Yellow …
Indicator:
0 bit
1 bit
tab
inter-word
inter-paragraph
inter-sentence
end-of-line

Figure 7. Proposed method.

$$z_1 + \beta \geq \frac{4(x+\alpha+8)}{3} \qquad (7)$$

For the user who uses customised character/string/text, the minimum capacity of the white-spaces required to accommodate the $\frac{4(x+\alpha+8)}{3}$ bytes of secret data which can be identified using the following formula:

$$nz_2 + (n-1) \geq \frac{4(x+\alpha+8)}{3}\text{ ; } n = 1, 2, 3 \ldots n \qquad (8)$$

z_2 is the capacity of the white-spaces produced by the customised character/string/text, and n is the minimum number of duplication required for $[nz_2 + (n-1)]$ to accommodate the $\frac{4(x+\alpha+8)}{3}$ bytes of secret data.

If the user chooses option B, the capacity of the white-spaces required to accommodate the secret data is similar as (4) and (5) for auto content generation and customised character/string/text features respectively.

After that, the secret data will be converted into bit-wise dataset before hiding into the cover-text. The bit-wise dataset is a stream of bits (string with bits value of "1" and "0") that is converted from the previous process and inserted into the white-spaces in the cover-text. The system will encode a "0" bit in a single space, an extra space will be appended to encode a "1" bit (encode "0" bit in a single space; encode "1" bit in double spaces).

In order to fully utilise the white-spaces in a document, "tab" will be appended at the end of the sentence or between paragraphs to act as invisible "word" or NULL value. Therefore, more bits can be embedded at the end of the sentence or between the paragraphs (Figure 7).

Once the secret data has been embedded into the cover-text, a stego-text (secret data + cover-text) will be produced. Next, the sender will be prompted to save the stego-text. The sender has an option whether to save the stego-text. Otherwise, the sender can choose to return to the main page or exit the application.

DeSteg process

Figure 8 shows the encoding process of the proposed system. In DeSteg process, the receiver is prompted to browse for the text file (stego data) that is going to be recovered. If the selected text file does not exist, a warning message will notify the receiver. The receiver will be prompted to select another text file. If the stego-text is successfully identified, the stego-text is loaded into the system and undergo the decoding process. All the spaces in the text will be gathered and calculated. One space represents "0" bit and two spaces represent "1" bit. Therefore, all the "0" bit and "1" bit will be combined into a bit-wise dataset. The bit-wise dataset will be converted into strings (a stream of characters) in the convert bits to string process.

If the strings (bytes) were encrypted in the Steg process (option A), the receiver is required to enter the password for decrypt the strings. Else, the receiver is not able to retrieve the secret data. The strings (bytes) will be decoded with Base64 decoding process.

$$x = \frac{3B}{4} \qquad (9)$$

B is the capacity of the strings (bytes), and *x* is the capacity of the bytes data after base64 decoding (include the secret data file extension)

The system will then decrypt the bytes data (x) with a receiver-provided public key. The decrypted bytes data ($x-8$) is going to be detached into bytes data ($x-\alpha-8$) and the extension of the original secret data (α) in Detach Extension Process. After that, the bytes data will go through a second Base64 decoding process and the capacity of the bytes data will become $\frac{3(x-\alpha-8)}{4}$ bytes.

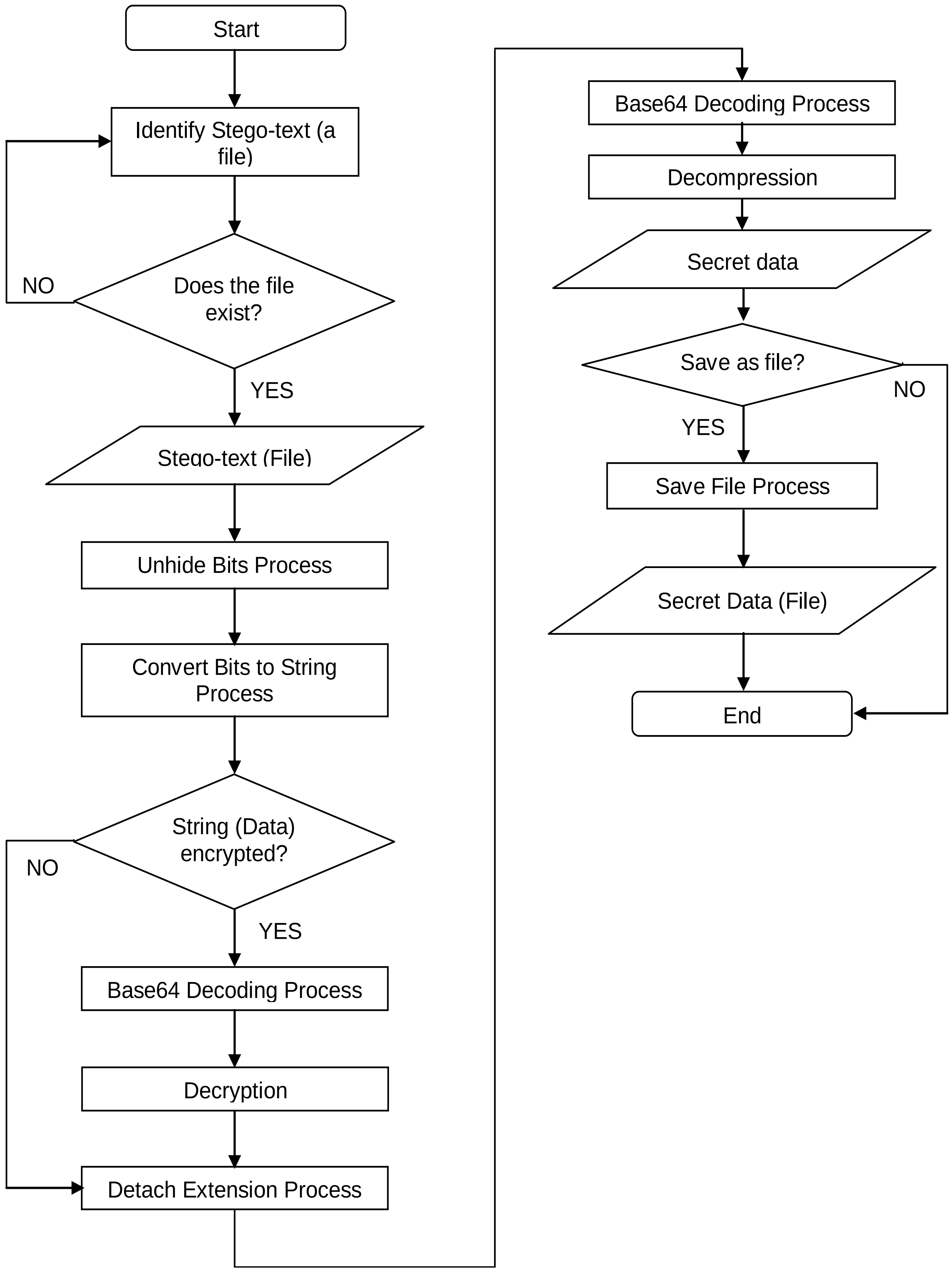


Figure 8. Decoding process.

Table 1. Capacity comparison using plain text as payload.

Payload file size (kB)	Stego-text file size (kB)			
	Proposed method	wbStego4open	SNOW	Spacemimic
1	34	47	45	40
2	61	92	80	80
4	113	183	134	154
8	209	362	266	307
16	394	722	527	616
32	723	1437	1020	1227
64	1344	2871	1998	2446
128	2631	5743	4150	4904
256	5212	11493	8709	9797
512	10273	22992	17383	19607
1024	20405	45979	34731	−

If the strings (bytes) were not encrypted (option B), detach extension process will be directly proceeded. The extension of the original secret data (α) will be detached. The remaining bytes data ($x-\alpha$) will then be decoded in Base64 decoding process. The capacity of the bytes data will become $\frac{3(x-\alpha)}{4}$ bytes. The bytes data is the compressed secret data (in bytes format).

Finally, the secret data will be decompressed using deflate compression and the receiver will be prompted to save the secret data. The receiver has an option whether to save the secret data. Otherwise, the receiver can choose to return to the main page or exit the application.

RESULTS AND DISCUSSION

Here, several test cases are carried out to compare the proposed method with several existing steganographic tools such as SNOW (Kwan, 2009), wbStego4open (wbStego, 2004) and Spacemimic (Spacemimic, 2010) in terms of capacity aspect (the stego-text produced). SNOW is a steganographic tool that utilises the end-of-line spacing method to conceal secret messages by appending white space characters at the end of text lines. To differentiate the encoded bits during the encoding process, tab space characters are used. wbStego4open is another instance of steganographic tool that utilises the hybrid methods of inter-sentence spacing and inter-word spacing methods to embed secret messages. Spacemimic adopts the end-of-line spacing method and inter-paragraph spacing method to embed secret message.

Test case 1: Capacity comparison by using plain text as payload

In this test case, different payload size which solely contains of plain text (.txt) with the range from 1 kilobyte to 1024 kilobytes is used to perform the process of embedding. The payload that larger than 1024 kilobytes is not included in this test case because the embedding efficiency increased exponentially for the selected steganographic tools and Spacemimic fail to embed any data when the payload size is equivalent or greater than 1024 kilobytes. Table 1 shows that the proposed system is able to produce smaller stego-text compared to wbStego4open, SNOW, and Spacemimic. In general, wbStego4open produces the largest stego-text followed by Spacemimic then SNOW. Averagely, the stego-text produced by the proposed method is approximately 1 time smaller than the other steganographic tools when using small file size and approximately 2 times smaller when using larger file size. From this test case, the results also indirectly indicated that the proposed system is able to hide more plain text format information compared to the aforementioned steganographic tools. Moreover, the proposal of adopting Base64 encoding is able to work as an alternative method to encode the payload although this method increases 33.3% of the original bits.

Test case 2: Capacity comparison by using multimedia files as payload

In this test case, different length of video files (with the length from 10 to 60 s) and different dimension of image files (with the dimension from 8×8 pixels (p) to 1024×1024p) are used to perform the process of embedding. SNOW and Spacemimic are excluded in this test case because both of them can only embed plain text. Appendix (Figures 1 and 2) show that wbStego4open is able to produce smaller stego-text compared to the proposed system for both multimedia files in most of the cases. On the average, the stego-text produced by the proposed system is approximately 18% larger than wbStego4open via video as payload. In terms of using image as payload, the proposed system produces approximately 1 time smaller stego-text when using small file dimension (8p × 8p, 16p × 16p and 32p × 32p) compared to wbStego4open. This advantage is minimized when the file dimension increases (approximately 31 and 14% smaller file size produced when using file dimension 64p × 64p and 128p × 128p respectively). Therefore, these results indicated that the proposed system is more capable on hiding the jpeg image when the payload size is less than 100 kB.

From this test case, we concluded that the adopting Base64 encoding is not suitable to be used to encode video and large image files although the proposed system is able to hide more plain text format information compared to the other existing steganographic tools mentioned previously. The main reason why the Base64 encoding is not suitable to be used in such environment is because it increases 33.3% of the original bits after using.

Test case 3: Automated vs. self-defined text as cover

We have also conducted an experiment and analysis on the stego-text produced using the automated and self-defined text as the cover (Appendix (Figure 3)). From the experimental result, the embedding capacity drops to about 29% averagely when using self-defined text. Thus, we concluded that the capacity of the stego-text produced is dependent on the cover used. Moreover, the embedding capacity produced by the proposed method is higher compared to other steganographic tools (except when using smaller payload size which is less than 16 kB). Thus, we concluded that the proposed method is efficient in embedding larger payload compared to other steganographic tools when using self-defined text.

Conclusion

In this paper, a proposed system that utilises the hybrid methods of inter-sentence, inter-word, end-of-line and inter-paragraph spacing methods to embed the secret message are presented. The proposal of adopting Base64 encoding has proven to be able to work as an alternative method to encode the plain text payload although it increases 33.3% of the original bits. The screenshot of the embedding and decoding processes can be obtained as Appendix (Figures 4 and 5). From the test cases analysis, the proposed system is able to hide larger capacity compared to other steganographic tools such as wbStego4open, SNOW and Spacemimic. In future, we will focus on the robustness of the stego-text produced. Research on improving the tamper-resistance of the stego-text and preserving the secret data using watermarking technique or other relevant techniques will be our next investigation target. We will also look into the error correction on the text transmission via the Internet using information hiding method and encryption methods for extra security protection. Besides, the integration of information hiding method and authentication method (Yee and Kiah, 2010) is also an interesting domain which we will focus on.

ACKNOWLEDGEMENTS

This study was supported by Ministry of Science, Technology and Innovation (MOSTI) Science Fund - 01-01-03-SF0456, and University Malaysia Research Grant (UMRG) - RG012/09ICT.

REFERENCES

Bauer FL (2007). Decrypted secrets-methods and maxims of cryptology. Springer, Heidelberg.

Bender W, Gruhl D, Morimoto N, Lu A (1996). Techniques for data hiding. IBM Syst. J., 35: 313-335.

Bennett R, Phipps R, Strange A, Grey P (2004). Environmental and human health impacts of growing genetically modified herbicide-tolerant sugar beet: a life-cycle assessment. Plant Biotechnol. J., 2(4): 273–278.

Bergmair R (2004). Towards linguistic steganography: A systematic investigation of approaches, systems, and issues. Master’s thesis, University of Derby.

Calvo H, Bolshakov IA (2004). Using selectional preferences for extending a synonymous paraphrasing method in steganography. In Proc. of CIC2004: XIII Congreso Internacional de Computacion, pp. 231-242.

Chotikakamthorn N (1998). Electronic document data hiding technique using inter-character space. In Proc. of the IEEE Asia-Pacific Conf. on Circuits and Systems, pp. 419-422.

Frank YS (2007). Digital watermarking and steganography: Fundamentals and techniques. CRC Press, Inc., Boca Raton, FL, USA.

Gutub AAA, Fattani MM (2007). A novel arabic text steganography method using points and extensions. In Proc. of Int. Conf. on Computer, Information and Systems Science and Engineering (ICCISSE*)*, pp.28-31.

Hmood AK, Kasirun ZM, Jalab HA, Zaidan AA, Zaidan BB, Alam GM (2010). On the accuracy of hiding information metrics: Counterfeit protection for education and important certificates. Int. J. Phys. Sci., 5 (7): 1054-1062.

Johnson NF, Jajodia S (1998). Exploring steganography: Seeing the unseen. Computer, 31: 2634.

Katzenbeisser S, Petitcolas FAP (2000). Information hiding techniques for steganography and digital watermarking. Artech House, Boston, USA.

Kessler GC (2004). An overview of steganography for the computer forensics examiner. Forensic Science Communications, 6: 3.

Khairullah M (2009). A novel text steganography system using font color of the invisible characters in microsoft word documents. In Proc. of Int. Conf. on Computer and Electrical Engineering (ICCEE), 1: 482-484.

Kipper G (2003). Investigator’s guide to steganography. Computer Security Innovations, Herndon, Virginia, USA, pp.1-240.

Kwan M (2009). The Snow Home Page, http://www.darkside.com.au/snow/.

Liu TY, Tsai WH (2007). A new steganography method for data hiding Microsoft word documents by a change tracking technique. IEEE Trans. on Information Forensics and Security, 2(1): 24-30.

Por LY, Delina B (2008). Information hiding: A new approach in text steganography. In Li et al. (eds) WSEAS: Advances on applied computer and applied computational science, Elect. Comput. Eng., pp. 689-695.

Por LY, Ang TF, Delina B (2008). WhiteSteg: A new scheme in information hiding using text steganography. WSEAS Trans. on Computers, 7: 735-745.
Roy S, Manasmita M (2011). A novel approach to format based text steganography. In Proc. of Int. Conf. on Communication, Computing & Security (ICCCS). ACM, New York, NY, USA.
Spacemimic (2010). http://www.spammimic.com/index.shtml.
Topkara U, Topkara M, Grothoff C, Grothoff K (2006). The hiding virtues of ambiguity: Quantifiably resilient watermarking of natural language text through synonym substitutions. In Proc. of the 8th Workshop on Multimedia and Security, pp.164-174.
Wayner P (2002). Disappearing cryptography information hiding: Steganography & watermarking. Morgan Kaufmann, Los Altos, USA.
wbStego (2004). http://wbstego.wbailer.com
Wu M, Liu B (2002). Multimedia Data Hiding. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
Yee PL, Kiah MLM (2010). Shoulder surfing resistance using penup event and neighbouring connectivity manipulation. Malaysian J. Comput. Sci., 23: 121-140.

APPENDIX

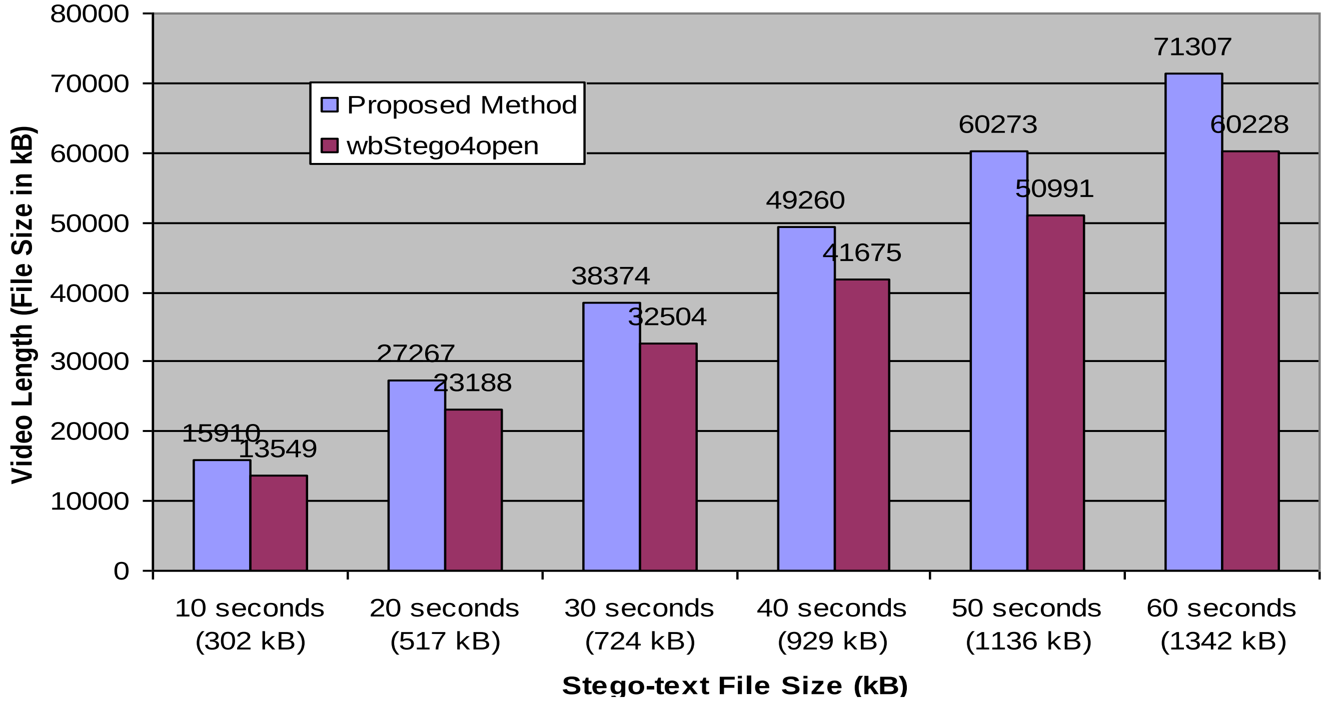


Figure 1. Capacity comparison using video file as payload.

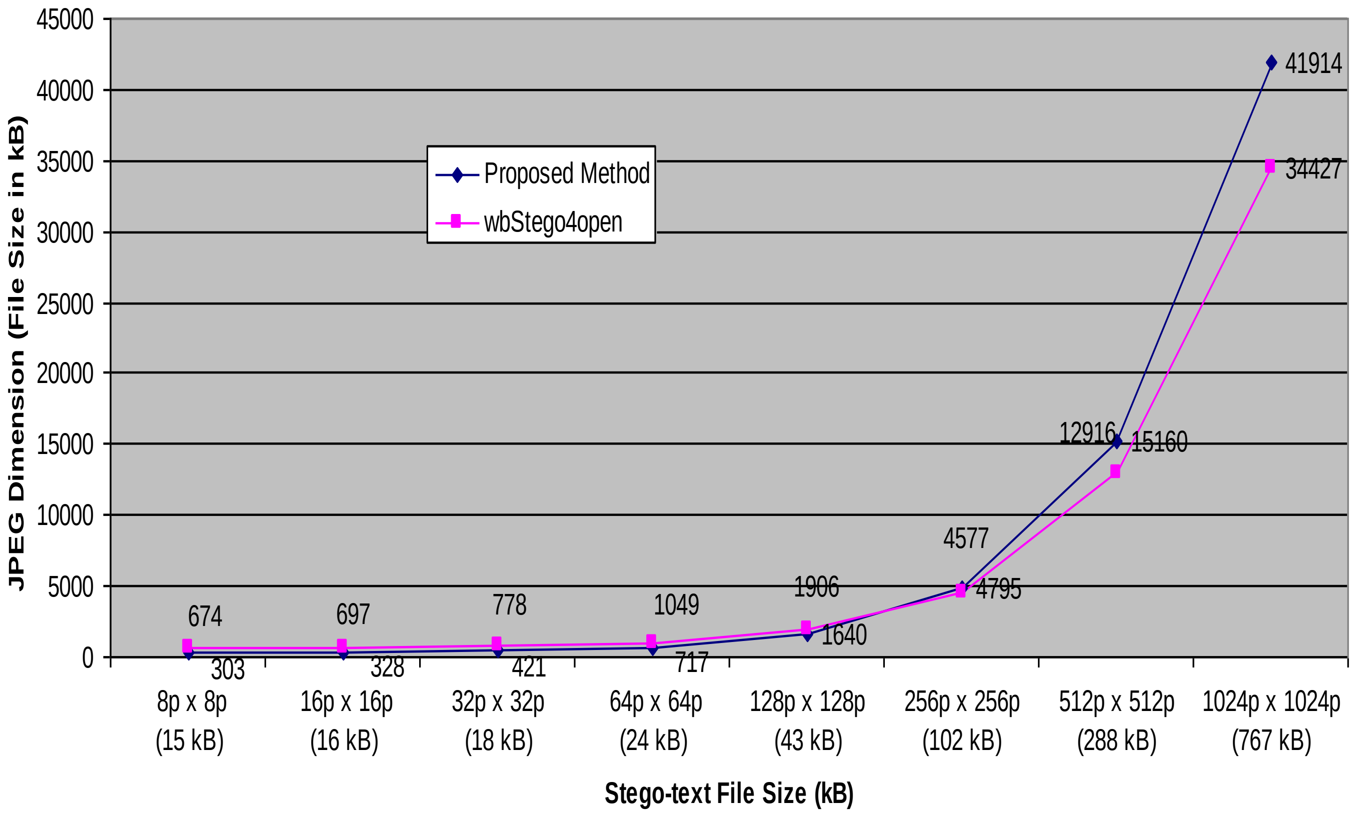


Figure 2. Capacity comparison using JPEG file as payload.

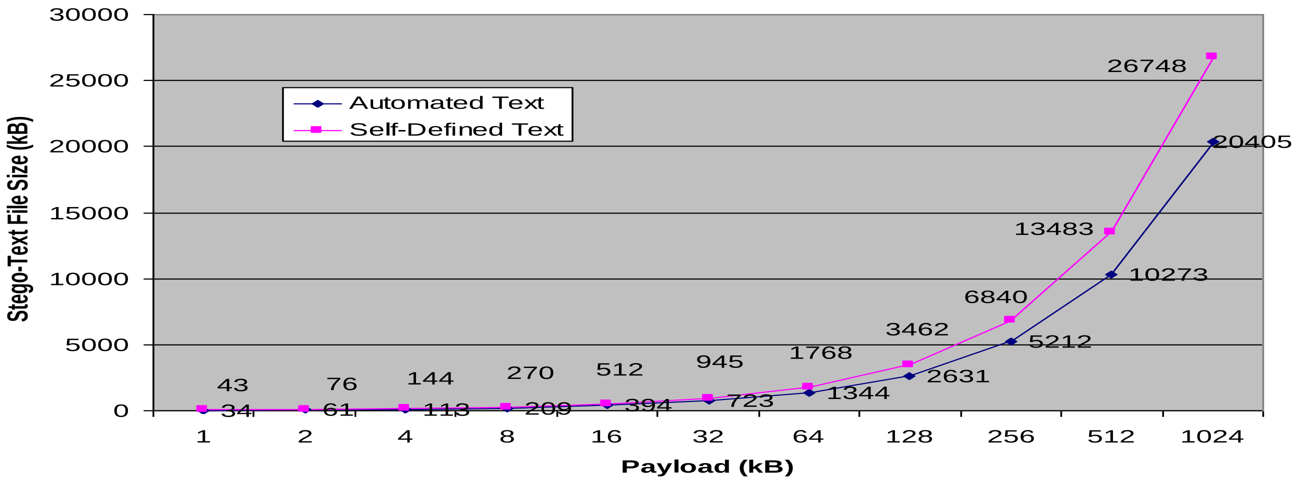
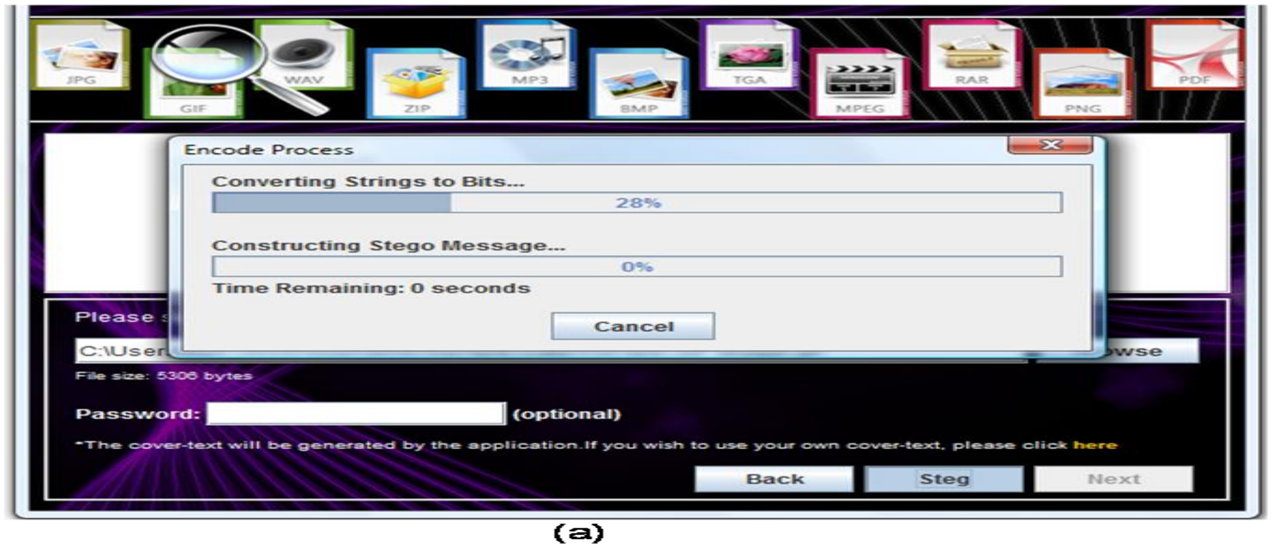


Figure 3. Automated vs. self-defined text.



Encode Process

Converting Strings to Bits...

28%

Constructing Stego Message...

0%

Time Remaining: 0 seconds

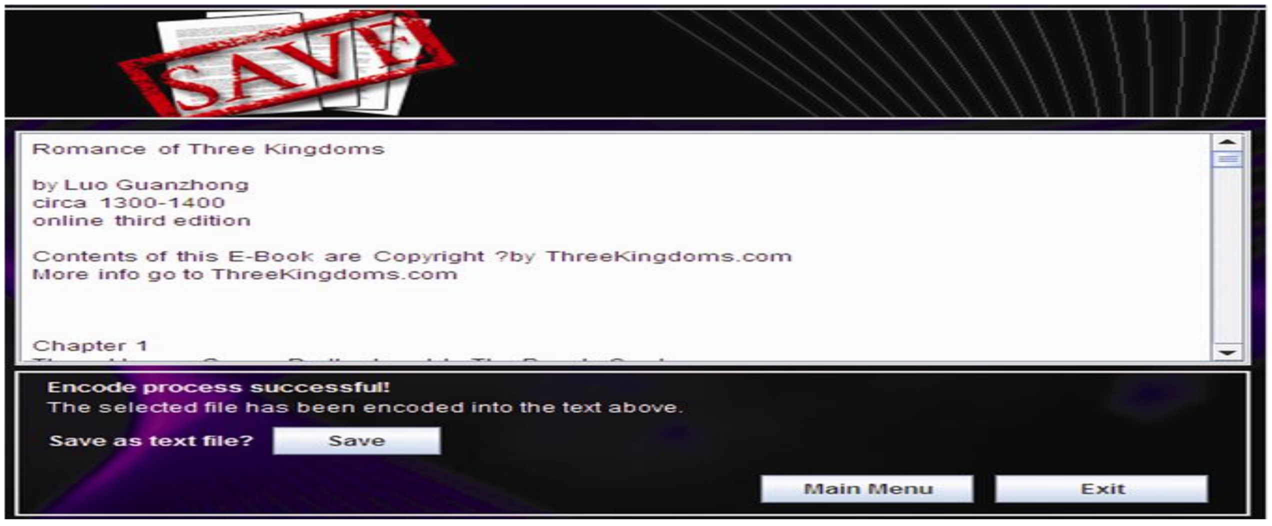
Please s

C:\User

File size: 5306 bytes

Password: (optional)

*The cover-text will be generated by the application.If you wish to use your own cover-text, please click here



Romance of Three Kingdoms

by Luo Guanzhong
circa 1300-1400
online third edition

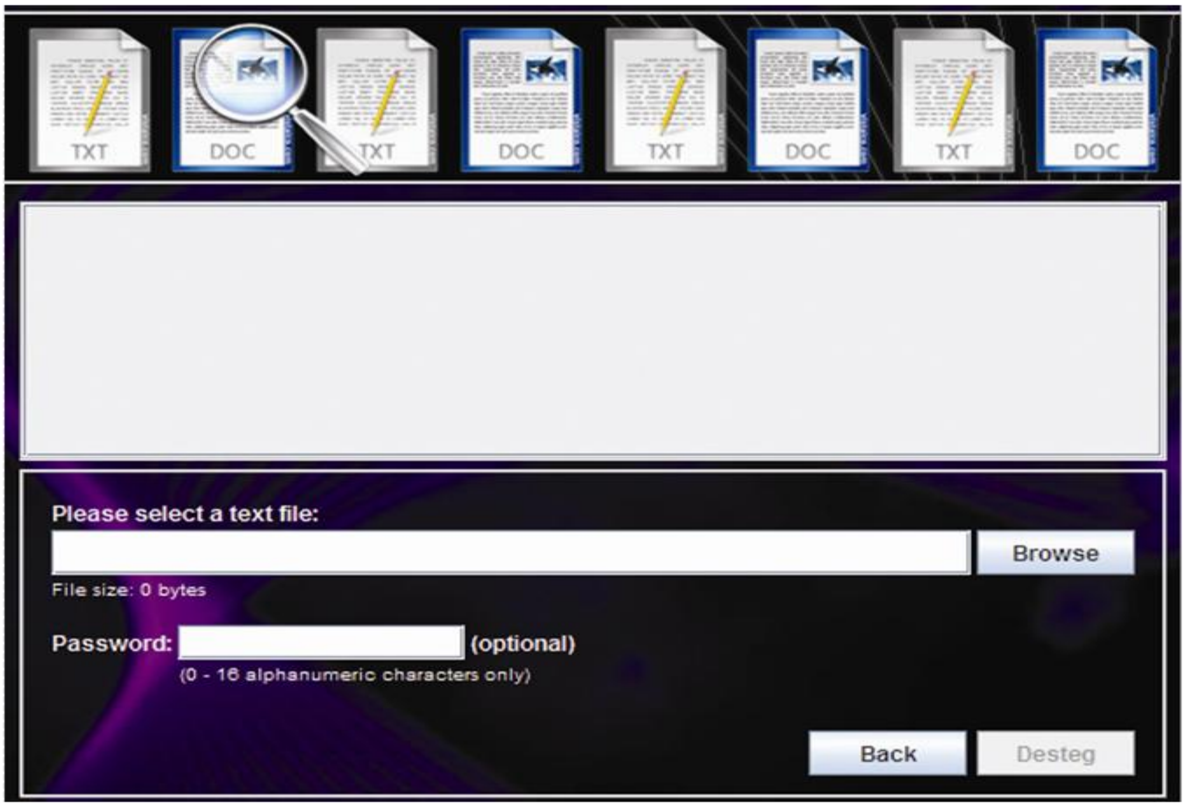
Contents of this E-Book are Copyright ?by ThreeKingdoms.com
More info go to ThreeKingdoms.com

Chapter 1

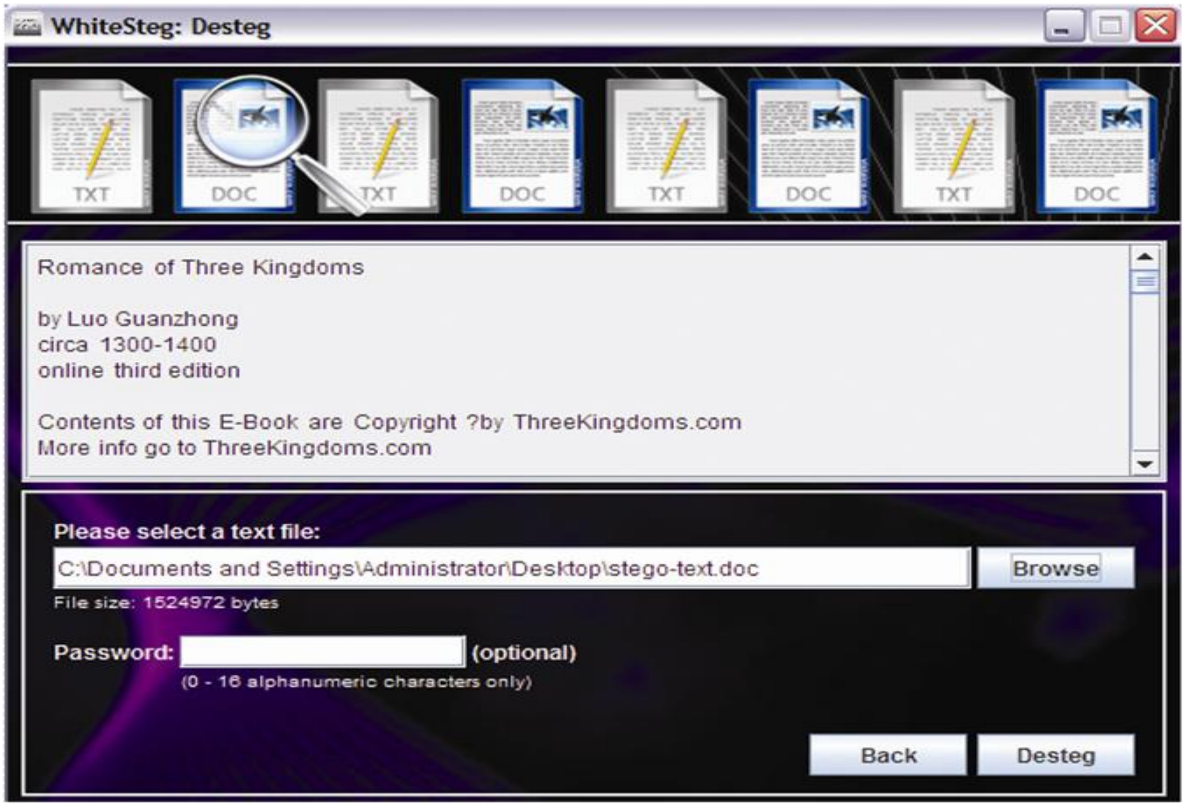
Encode process successful!
The selected file has been encoded into the text above.

Save as text file?

Figure 4. Encoding process (*Steg*) continue; (a) transforming the secret data to bits; (b) The secret data is embedded in the cover text.



(a)



Romance of Three Kingdoms

by Luo Guanzhong
circa 1300-1400
online third edition

Contents of this E-Book are Copyright ?by ThreeKingdoms.com
More info go to ThreeKingdoms.com

Please select a text file: C:\Documents and Settings\Administrator\Desktop\stego-text.doc

File size: 1524972 bytes

Password: (optional)

(0 - 16 alphanumeric characters only)

(b)

Figure 5. Decoding process (*DeSteg*); (a) encoding process GUI, and (b) identify the stego-text.