A New Approach of Graph Realization for Data Hiding using Huffman Encoding

Fatema Akhter, Member, IEEE
Jatiya Kabi Kazi Nazrul Islam University, Bangladesh

Md. Selim Al Mamun
Okayama University, Japan

***Abstract*—The rapid advancement of technology has changed the way of our living. Sharing information becomes inevitable in everyday life. However, it encounters many security issues when dealing with secret or private information. The transmission of such sophisticated information has become highly important and received much attention. Therefore, various techniques have been exercised for security of information. Graph steganography is a way of hiding information by translating it to plotted data in a graph. Because of numerous usages of graphs in everyday life, the transmission can take place without drawing any attention. In this paper, we propose a new graph realization technique for steganography that looks as if innocent and imperceptible to present day steganalytic attacks and the hidden message can only be read by its respective recipient. The secret message is first translated to prefix codes using huffman encoding. Then the prefix code for separate word in the message is plotted in a graph. The proposed technique offers high embedding capacity and imperceptibility due to prefix presentation and word by word encoding of the message. The experimental outcomes show strong resistance towards steganalytic attacks in contrast to other approaches.**

***Keywords*—Data hiding; Graph steganography; Huffman encoding; Steganalytic attack**

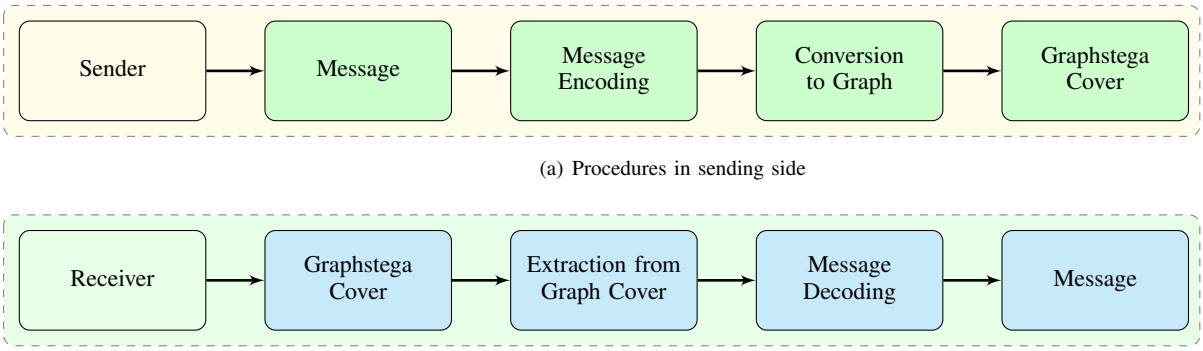
I. Introduction

In recent years, electronic communication has become an integral part of everyday life. Be it email, audio or video, people exchange information mutually through electronic medium. The security of information has become essential as the transmission on public communication channel increases. This is mandatory to preserve the integrity and security of information that are being transmitted over public communication channel. Several methods and techniques have been studied in order to achieve the security of information. Unfortunately, they are still in research to enhance the security. Methods like Steganography [1]–[4] and cryptography [5]–[7] are commonly used for the security of information. However, in last two decades, steganography perhaps got much more attention than any other method. Steganography is the art of passing information in such a way that the existence of the message cannot be detected by intruders [8]. The cover can be any digital medium like image, audio or video.

Among numerous strategies of information security, steganography using graph has drawned a variety of interest of researchers as it can avoid noise within the cover [9]–[11]. The method avoids the noise in cover by plotting the information as facts in a graph for the secured transmission in contrast to other approaches. People use graph in daily life which makes it harmless and risk free to attract attention and preclude numerous attacks. Any length of message can be translated to graph-data retaining the integrity of the information while transmission in public channel. Hiding message in graph is simple and straightforward. This does not require any special overhead. Consequently, it becomes a popular subject of studies in information security. The superiority of graph over other medium is tabulated in Table I. Hiding message in graph is a exceptionally new idea in the field of information security. The concept will shine with time due to interest of many researchers in this subject. In [12], the authors presented a technique for integration of message in graph that uses vertex-coloring method. In [13], the authors proposed a technique using integer wavelet transform (IWT) along with graceful graph to offer a secure and random image steganography with high imperceptibility. In [14], the authors presented a technique where the message is camouflaged as plotted data in graph. In [15], the authors presented a technique using hamiltonian graph for the security of the message in public communication.

In this paper, we study popular strategies of information hiding in graph to grasp the ideas. Then, we propose a new graph realization technique from the message. In this work, we use huffman encoding method to generate prefix code for every character in the message. The prefix codes are considered as the binary representation of the characters. Then, we classify the prefix code of character by the way of word within the message. The group of binary prefix code is then converted to its equivalent decimal value. We introduced two constants α and β that present the decimal value of a white space and scaling factor of decimal value of a word respectively. We add α to every word value to make it different from the white space. β is multiplied to the resultant value for diversion. Finally, we draw bar plot using these values in an excel file. To the best of our knowledge, similar technique for information hiding has not been addressed in any existing work. We investigated the proposed technique on various steganalytic attacks. Experimental outcomes show that the proposed technique is more secure as compared to other information hiding methods against various parameters such as embedding capacity, security and fidelity.

We prepare the remaining of the paper as follows: Section II presents the initial studies and provisions for the proposed technique. Section III presents the proposed technique for graph steganography.Section IV presents the experimental results and discussions.Finally, we conclude this paper in section V with some future works.



(a) Procedures in sending side

(b) Procedures in receiving side

Fig. 1. General procedures in graph steganography: a) procedures in encoding in sender side and b) procedures in decoding in receiver side.

TABLE I. Different Steganography Methods

Parameters	Steganography Methods		
	graph	image	audio
Noise	×	○	○
Distortion	×	○	○
Message size limit	×	○	○
Complexity	×	○	○
Less cost	○	×	×
Cover conversion	○	×	×
Traceable	×	○	○

II. Preliminary Study

In this section, we present some preliminary studies for graph steganography. Specifically, we describe simple graph steganography and prefix code generation using huffman encoding algorithm.

A. Simple Graph Steganography

In graph steganography [16]–[18], messages are converted to facts or records to plot them in a graph. The generated graph looks simple that we frequently use in daily life. The presentation of message using graph can be carried out to a wide variety of domains of steganography where the cover needed to be noiseless. Unlike other steganography methods, graph steganography does not conceal facts in any digital medium like photograph, audio or video. This is referred to as noiseless steganography as it does not introduce noises within the cover while hiding information. A simple graph steganography interprets the message and converts to compatible data that can be plotted in a graph. Finally, the generated graph is transferred to the recipients of the message. Fig. 1 shows the procedures in a conventional graph steganography. The secret message to be transmitted is referred to as plaintext. In sending side, the plaintext is directly converted to facts in a graph. The titles and legends are given that look meaningful to facts in graph. The receiving side follows the methods in reverse order.

B. Huffman Encoding

Huffman encoding [19] is a method of generating an optimal prefix code for a string. The method assigns variable-length bit string to every character in the string that unambiguously represents that character. The variable-length bit string is called binary prefix codes throughout this paper. The method minimizes the number of bits required to represent a standard string composed of these characters. The method counts the frequency of each character in a string and generates minimal prefix code for each character. The characters with higher frequency have fewer bits than the characters of lower frequency. An encoding tree is generated by utilizing a priority queue where nodes with lower frequency are assigned higher priority. The procedure of huffman encoding is given below:

① Create a leaf node for each symbol and add it to the priority queue.
② While there is more than one node in the queue:
   a) Remove the node of highest priority (lowest probability) twice to get two nodes.
   b) Create a new internal node with these two nodes as children and with the probability equal to the sum of the two probabilities of these two nodes.
   c) Include the newly created node to the queue.
③ The node that remains in the queue, make it root of the tree. This completes the generation of tree.
④ Generate the prefix code by traversing the tree from root to leaves putting a zero (0) if every time a lefthand branch is taken and a one (1) if the right hand branch is taken.
⑤ The resultant 0 and 1 in the path from root to it its leaf is the prefix code for the symbol at the leaf.

III. Proposed Graph Steganography Method

The proposed graph steganography approach is composed of two methods:

- **Encoding**: In this method, the message is hidden into an excel graph.
- **Decoding**: In this method, the message is retrieved to its original form.

A. Proposed Encoding Method

Fig. 2 shows the steps of proposed encoding method for graph steganography. The method first interprets the message and converts every character in the message to its equivalent binary prefix code. Prefix code is generated using huffman encoding presented in II-B. The prefix codes are grouped by the word within the message. The prefix codes of characters are concatenated as they appear in the word and accumulated prefix code for each word is derived. The binary prefix code for

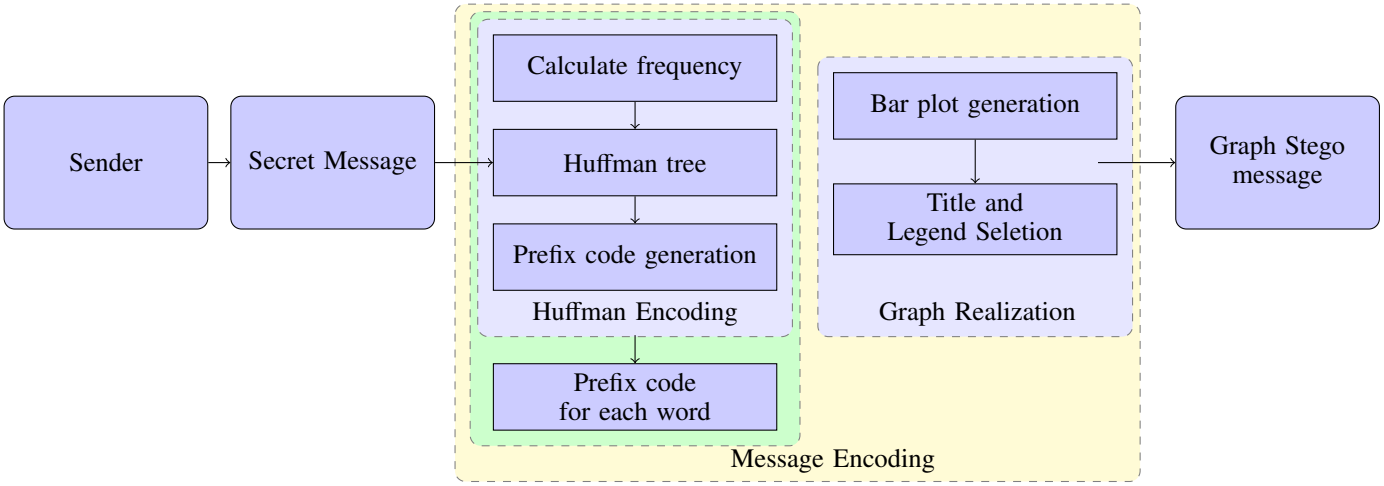


Fig. 2. Procedures of the proposed message encoding algorithm for the proposed graph steganography approach.

each word is then converted to its decimal value. The decimal value of white space is represented by a constant α. To avoid ambiguity, α is added to every decimal value of word. To avoid redundancy and to escape statistical attacks, a scaling factor β is introduced. β is multiplied to the resultant decimal value of each word. The resultant value becomes distinctive from the original prefix presentation of the message. These decimal values are then plotted as graph in an excel file. The steps of the proposed encoding method is given below:

Proposed Encoding Method

① Take the input message, M and read every character L_i within the message. Here L_i presents the ith character in message, M.
② Count the frequency $n(L_i)$ for character L_i and $n(SP)$ for white space. Here $n(L_i)$ and $n(SP)$ present the frequency of character L_i and white space respectively.
③ Call huffman encoding method on character set $\mathcal{L}$ to assign prefix code P_i for the character L_i in the message M, where $\mathcal{L} = \{L_i : \forall i, L_i \in M\}$.
④ Concatenate the prefix code P_i of all characters within the message to generate the prefix code stream, P, where $\mathcal{P} = P_1 \cdot P_2 \cdot P_3 \cdots P_m$.
⑤ The prefix code stream P is classified by the words W_i in the message M, where W_i presents the prefix code of the ith word in the message and $P = W_1 \cup W_2 \cup \cdots$.
⑥ For each word, W_i within message M, do the following:
   a) Obtain binary value B_i for each word W_i in the message. The binary B_i is converted to its equivalent decimal, D_i.
   b) Add the white space value α to each decimal D_i to obtain $D_i + \alpha$.
   c) Finally, multiply $D_i + \alpha$ by the scaling factor β to obtain $(D_i + \alpha) \times \beta$.
⑦ Multiply the white space value, α by the scaling factor, β to obtain $\alpha \times \beta$.
⑧ Plot each word $(D_i + \alpha) \times \beta$ and white space $\alpha \times \beta$ in a excel file to generate the graph..

B. Proposed Decoding Method

Fig. 3 shows the steps of proposed decoding method for graph steganography. The decoding method retrieves the original message from the received graph. The decoding method follows the reverse procedures of the encoding procedures. The received graph incorporates indices for word and space in x-axis where the corresponding decimal values in y-axis. The graph is first interpreted and the decimal values for word and spaces are extracted from the graph as they appear in the message. Then, every extracted decimal value is divided by β and the resultant value is compared to α to check whether it is a white space or character. The result is considered as a space if it equals to α, in any other case, the value is taken into consideration for a word within the message. The white space value α is deducted from the result to obtain the decimal presentation of every word in the message. The resultant decimal is converted to the prefix code using decimal-to-binary conversation. This is the prefix code of a word in the message. The prefix codes for all the characters in the message are derived from the huffman tree presentation Fig. 6. The steps of the proposed decoding method are given below:

Proposed Decoding Method

① Take the graph G as input and interpret all the decimal values, G_i in the graph. Here G_i is the y-axis value ith index in x-axis.
② For each decimal value, G_i in graph G , repeat the remaining steps.
③ Divide decimal value G_i by β to obtain R_i where β is the scaling factor and $R_i = \frac{G_i}{\beta}$.
④ Compare the resultant, R_i with α. Here α is the decimal value of white space. If $R_i = \alpha$, output R_i as the white space value.
⑤ Otherwise, if $R_i \geq \alpha$, do the following:
   a) Output R_i as the decimal value of the ith word in the message.
   b) Calculate $D_i = R_i - \alpha$. Here D_i is the decimal value of ith word in the message.

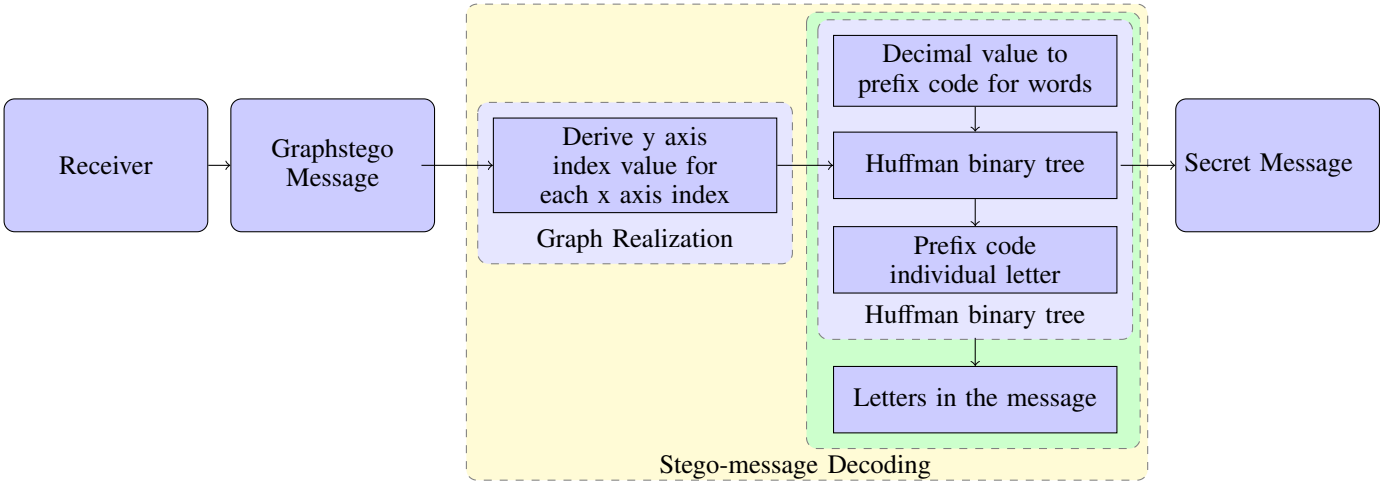


Fig. 3. Procedures of the proposed stego-message decoding algorithm for the proposed graph steganography approach.

c) Calculate the binary value of the ith word, B_i from the decimal valueD_i.
d) Traverse the tree according to binary prefix code B_i to find the ith word W_i in the message.

⑥ Otherwise, show decoding error message.

C. Solution Example

To illustrate the proposed algorithm for graph steganography, an example of the proposed algorithm is described here. Let the message be *"it is my war to win"*. In the first step, the algorithm counts the frequency of individual letter in the message.

Letter	*a*	*m*	*n*	*o*	*r*	*s*	*y*	*t*	*w*	*i*
Frequency	1	1	1	1	1	1	1	2	2	3

In the second step, the list is sorted according to the frequency of the letter. For the tie break, the list is sorted alphabetically. In this phase *huffman encoding* is applied on the sorted list. This transforms two lowest elements to leaves and creates a parent node with a frequency that is the sum of frequencies of two lowest elements as shown in Fig. 4. The two-lowest frequency letters get replaced by their parent node with frequency $2 : *$ in the list. The list becomes as follows:

Letter	*n*	*o*	*r*	*s*	*y*	$*$	*t*	*w*	*i*
Frequency	1	1	1	1	1	2	2	2	3

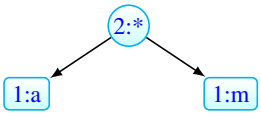


Fig. 4. Generation of prefix code using huffman encoding: step one.

Again, a parent node is created with the sum of the frequencies of two lowest elements in the list, as shown in Fig. 5. Two letters of lowest frequencies get replaced by their parent node with frequency $2 : *$ in the list and sorted as follows:

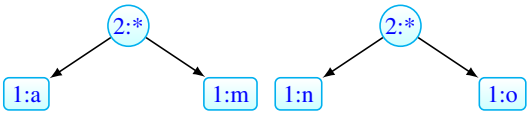


Fig. 5. Generation of prefix code using huffman encoding: step two.

Letter	*r*	*s*	*y*	$*$	$*$	*t*	*w*	*i*
Frequency	1	1	1	2	2	2	2	3

The process is repeated until there is only one element left in the list as shown in Fig. 6. This element becomes the root of the huffman binary tree. In the third step, the prefix code for each letter in the message is generated by traversing the huffman tree from the root to its leaves. A zero (0) and one (1) are embedded in the prefix code while traversing left branch and right branch respectively. This step ends with the generation of prefix code for every letter in the message. The generated prefix code for each letter is shown below:

Letter	*i*	*y*	*a*	*m*	*n*	*o*	*r*	*s*	*t*	*w*
Prefix code	00	010	0110	0111	1000	1001	1010	1011	110	111

In the fourth step, the prefix code for a word is constructed by concatenating the prefix codes of letters in the word. The generated prefix codes for the words in the message *it is my war to win*, are shown below:

Word	*it*	*is*	*my*	*war*	*to*	*win*
Prefix code	110	1011	111010	11101101010	1101001	111001000

In the fifth step, the prefix code of a word is considered as binary presentation and is converted to its equivalent decimal value. The resultant decimal value for each word in the message is shown below:

Word	*it*	*is*	*my*	*war*	*to*	*win*
Prefix code	110	1011	111010	11101101010	1101001	111001000
Decimal	6	11	58	1898	105	456

If the value for the white space is considered as $\alpha = 500$

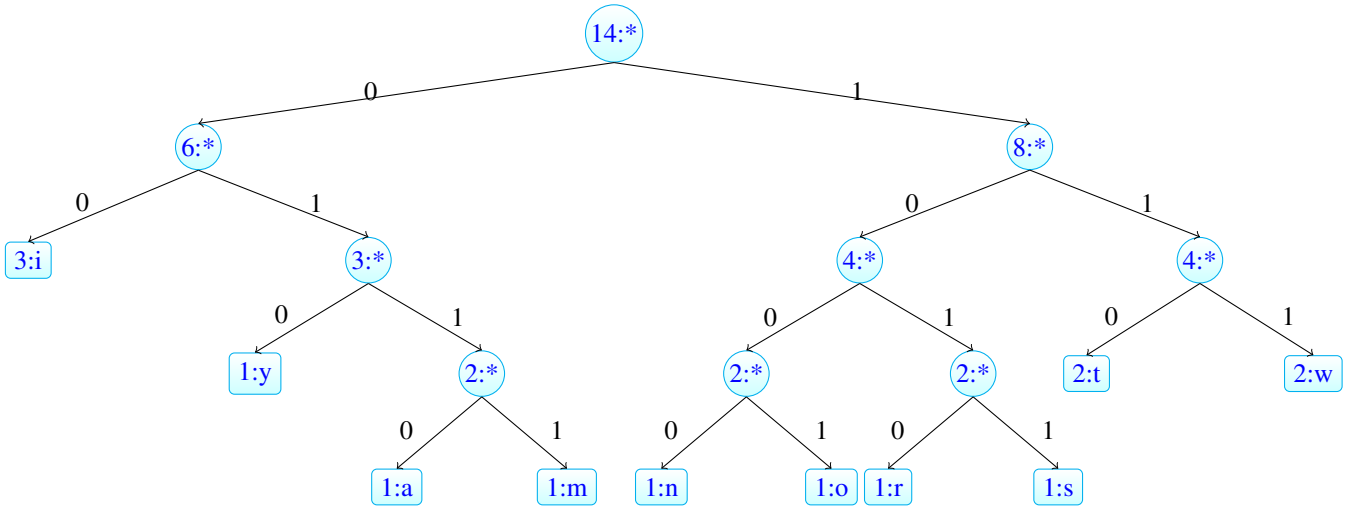


Fig. 6. Generation of prefix code using huffman encoding for the given message ***it is my war to win.***

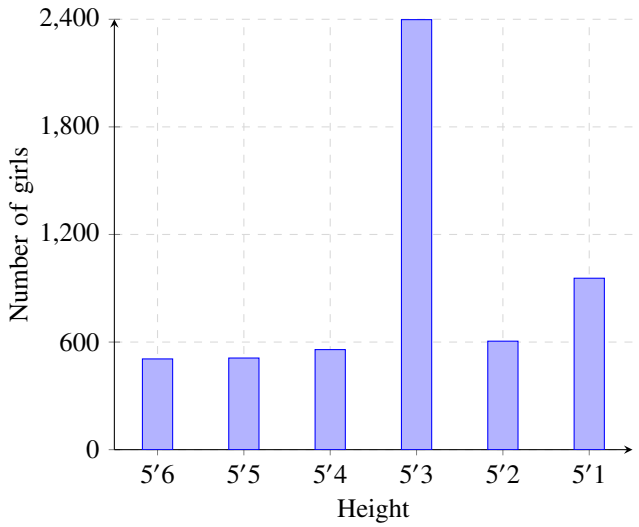


Fig. 7. A stego-graph for the message *it is my war to win* using proposed algorithm.

and the value for the scaling factor $\beta = 1$, then for every word, value of $(D_j + \alpha)\beta$ is computed which presents the plot data in the graph. The plot data for each word in the message becomes as below:

Word	*it*	*is*	*my*	*war*	*to*	*win*
Plot data	506	511	558	2398	605	956

Finally, the solution value for the message *it is my war to win* is plotted in a graph as shown in Fig. 7

IV. Experimental Results and Discussions

In this section, we present experimental results of the proposed graph steganography technique and compare with existing works. We additionally describe the durability of the proposed technique against steganalytic attacks. The software and hardware configuration of this work are summarized in Table II.

TABLE II. Hardware and Software Summary.

Grpah Plot	Microsoft Excel 2007
IDE	Microsoft Visual Studio 2012
CPU	Intel Core i5
Memory	4 GB
Operating System	Windows 7

A. Resultant Graph

Fig. 7 shows the stego-graph encoded from the message *it is my war to win*. This graph displays some inconsistent peaks. Therefore, the cover *Girls with height above* 5′ of an institution is chosen, because it suits well with the disparity of the number distribution. In an institution, it is more likely that girls with average height will be in majority than of others. And it matches perfectly with the graph-data.

B. Evaluation by Comparison

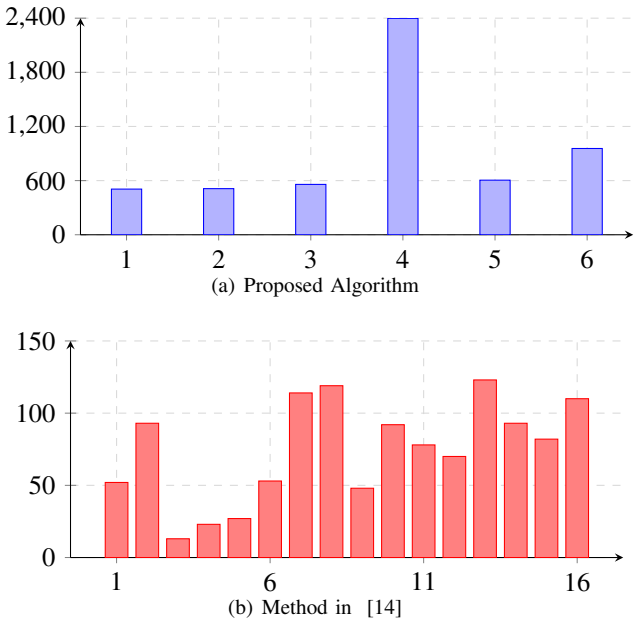
In this section, we evaluate the proposed graph steganography technique by comparing the results of the proposed technique with the existing method in [14]. The secret message that is considered for the purpose of comparison is:

- it is my war to win

Fig. 8 shows the comparison between the proposed algorithm and [14]. Both the graphs are generated from the message *it is my war to win*. Several points are worth noting from the comparison:

- Embedding Capacity
- Randomness of Data

The most eye-catching difference between the graphs is the number of bars. The graph generated using proposed algorithm has less bars than that of [14]. This clearly indicates the higher embedding capacity of the proposed algorithm than [14]. The proposed algorithm embeds word by word on contrary to the existing letter by letter approach [14]. Therefore, the proposed algorithm performs better in embedding long messages. Experimenting with long messages using [14] may produce disastrous results, because there will be numerous



(a) Proposed Algorithm

(b) Method in [14]

Fig. 8. Comparison of the proposed algorithm with [14] for the message **it is my war to win**

TABLE III. COMPARISON BETWEEN THE PROPOSED AND METHOD [14]

Parameters	Comparison	
	Proposed	[14]
Embedding unit	Word	Letter
Embedding value	Prefix	ASCII
Embedding capacity	High	Low
Security caution	α and β	Nothing special
Graph data	Word and space	Character
Letter based attack	Resistant	Vulnerable

data, difficult for handling. That is, difficult to represent as graph-data and choose a cover.

The next significant difference is the randomness of data. According to Fig. 8, the graph-data generated using proposed algorithm has a higher disparity than that of [14]. The difference is rooted in the methods chosen by each approach. Proposed algorithm uses prefix codes generated by huffman encoding. Huffman encoding tries to eliminate redundancy by assigning less bits to the high frequent letters. While in the binary conversion of ASCII values, there is no such scope. So, the graph-data produced by [14] has less randomness than the proposed algorithm. The comparison results are summarized in Table III.

C. Effectiveness of Proposed Graph Steganography Technique

Effectiveness of the proposed approach may be evaluated considering the three basic aspects:

- Payload: Amount of information that can be hidden in a graph.
- Security: Impossibility of attack to detect hidden information in stego-graph.
- Fidelity: Inability of human eyes to distinguish between stego-graph and original graph.

Although there is no literal binding in the amount of information that can be embedded using graph steganography, proposed word by word message embedding approach outplays existing letter by letter graph steganography [14]. As the name suggests, proposed word by word embedding approach has significantly higher embedding capacity compared to [14]. Thus proposed approach serves better the first basic principle of steganography.

For ensuring security, proposed approach consults several experts. Firstly, it chooses graph as the cover that has an innocent look. Secondly, it consults the word by word embedding which is a very new technique in graph steganography. This can avert steganalytic attacks in contrast to character by character method. This is where the proposed technique makes the main difference from [14]. Thirdly, introduction of white space value α and scaling factor β increase the randomness to the produced graph-data and makes it strongly durable against attackers. Final two steps make the proposed approach far apart from [14]. Graph steganography can be imperceptible if it chooses appropriate cover or subject of graph relevant to graph-data. Otherwise, it may raise suspicion. From this respect, both proposed and existing [14] have same performance since both have the same advantage.

D. Resistance Against Traffic Analysis

Traffic analysis is a popular steganalytic attack. It works on the principle of analyzing any conversation sample between the sender and receiver that is publicly available or can be derived using any tool. That is, the intruder may intercept any publicly shared content from website or they may keep track of website visitors etc. The attack has three steps. First, interception of data. Second, checking if the data is meaningful or not. Third, verifying the meaning against the relation between communicating parties. Generally this attack performs best against cryptographic data. Because ciphers generally mean nothing and looks conspicuous, thus are easily distinguished by step two. Sometimes, it is also effective against image steganography, text steganography, audio steganography etc.

But when this attack is used against graph steganography, step three is checked before step two. That is, cover verification is the first priority, then comes data analysis. For example, if a customs officer sends data on prediction of tomorrow's temperature, it stinks of something fishy going on. But, if a meteorologist forecasts tomorrow's weather, then it is a normal phenomena. Still, the temperature has to be in the normal range, otherwise it will be subject to further analysis. Now, for proposed approach, we have used prefix codes which is almost next to impossible to find. To make it more difficult, we have taken advantage of space value α and scaling factor β. So, while implementing proposed method, cautions should be taken choosing the appropriate cover type. Thus, proposed method can play smoothly with traffic analysis attack as long as the cover is appropriate.

E. Resistance Against Statistical Analysis

Statistical analysis is another popular attack used both in cryptography and steganography. It benefits from the statistical behavior of a language in Cryptography. Against graph steganography, it is exercised only if found guilty in Traffic

analysis. Statistical analysis may take two forms against graph steganography. In one case, it may count occurrences of same decimals and try to interpret the frequency to something meaningful like any pattern in real life. Another approach may consult ASCII table and convert the decimals to letters if the decimals are in the range $0 - 255$. There is no room for suspicion of word values. Because even if they do, there is no direct method of conversion for word values.

V. Conclusion and Future Work

In this paper, we propose a new graph realization technique for information hiding by presenting information to facts in a graph. In the proposed graph steganography technique, we take a message as input and assign prefix code to every character within the message using huffman encoding method. We concatenate all the prefix code in the message which are classified later according to the words in the message. Then we convert the prefix codes for all the words to obtain their equivalent decimal values. Finally, we plot these decimal values as a bar plot in an excel file. To verify the effectiveness of the proposed technique, we investigated the traffic analysis attack and the statistical analysis attack on the resultant graph. Finally, we compare the experimental outcomes with existing works. The results show the superiority of the proposed technique over existing techniques in terms of embedding capacity, security and strong resistance against steganalytic attacks such as traffic analysis attack and statistical analysis attack. In future, we want to further evaluate the proposed technique on various sizes of messages and investigate on other steganalytic attacks.

Acknowledgment

We are immensely grateful to all the anonymous referees for their constructive feedback and critical suggestions on an earlier version of the manuscript that helped us significantly to improve the quality of the work.

References

[1] F. Akhter, *A novel approach for image steganography in spatial domain*, Global Journal of Computer Science and Technology, vol. 13, no. 7-F, 2013.

[2] A. Anees, A. M. Siddiqui, J. Ahmed and I. Hussain, *A technique for digital steganography using chaotic maps*, Nonlinear Dynamics, Springer Netherlands, vol. 75, no. 4, pp. 807-816, October 2013.

[3] H. Tian, J. Liu and S. Li, *Improving security of quantization-index-modulation steganography in low bit-rate speech streams*, Multimedia systems, Springer Berlin Heidelberg, vol. 20, no. 2, pp. 143-154, February 2013.

[4] Y. Y. Tsai, *An adaptive steganographic algorithm for 3D polygonal models using vertex decimation*, Multimedia Tools and Applications, Springer US, vol. 69, no. 3, pp. 859-876, June 2012.

[5] B. Schneier, *Applied cryptography protocols, algorithm and source code in C*, Wiley India Edition, 2nd edition, 2007.

[6] H. Delfs and K. Helmut, *Symmetric-Key Cryptography*, Introduction to Cryptography. Springer, Berlin Heidelberg, pp.11-48, 2015.

[7] S. Garg, Y. Ishai, E. Kushilevitz, R. Ostrovsky,and A. Sahai *Cryptography with One-Way Communication*, In Annual Cryptology Conference, pp. 191-208, Springer, Berlin Heidelberg, Aug. 2015.

[8] C. Cachin, *An information theoretic model for steganography*, Proc. 2nd Inform. Hiding Workshop, vol. 1525, pp. 306-318, 1998.

[9] D. Martin and A. M. Barmawi, *List Steganography Based on Syllable Patterns*, PhD diss., Telkom University, 2015.

[10] A. Desoky, *Noiseless Steganography: The Key to Covert Communications*, CRC Press LLC, 2012.

[11] A. Desoky, *Nostega: A novel noiseless steganography paradigm*, J. of Digital Forensic Practice, vol. 2, no. 3, pp. 132-139, July 2008.

[12] S. M. Douiri, O. Medeni and S. Elbernoussi, *New steganography scheme using graphs product*, in Interactive Collaborative Learning (ICL) IEEE Int. Conf., pp. 525-528, December 2014.

[13] V. Thanikaiselvan, P. Arulmozhivarman, S. Subashanthini and R. Amirtharajan, *A graph theory practice on transformed image: A random image steganography*, The Scientific World J., October 2013, online available: http://www.hindawi.com/journals/tswj/2013/464107/

[14] A. Desoky and M. Younis, *Graphstega: graph steganography methodology*, Journal of Digital Forensic Practice, vol. 2, no. 1, pp. 27-36, March 2008.

[15] S. K. Muttoo and V. Kumar, *Hamiltonian graph approach to steganography*, Int. J. of Electronic Security and Digital Forensics, vol. 3, no. 4 , pp. 311-332. February 2010.

[16] F. Akhter, *A secured word by word Graph Steganography using Huffman encoding*, International Conference on Computer Communication and Informatics (ICCCI). IEEE, 2016.

[17] H. Wu, H. Wang, H. Zhao and X. Yu, *Multi-layer assignment steganography using graph-theoretic approach*, Multimedia Tools and Applications, Springer US, vol. 74, no. 18, pp. 8171-8196, May 2014.

[18] S. Hetzl and P. Mutzel, *A graph-theoretic approach to steganography*, In IFIP International Conference on Communications and Multimedia Security, pp. 119-128, Springer, Berlin Heidelberg, September 2005.

[19] D. A. Huffman, *A method for the construction of minimum-redundancy codes*, Resonance, vol. 11, no. 2, pp. 91-99, February 2006.

Fatema Akhter received her B.Sc. (Engg.) degree in Computer Science and Engineering from Jatiya Kabi Kazi Nazrul University, Trishal, Mymensingh-2220, Bangladesh in 2016. She is currently working toward the MS degree in Computer science and Engineering. Her general research interests are in the area of Cryptography and Network Security, Public Key Cryptosystem, Image Steganography, Quantum Cryptography and Anonymous Credential System. Her current research focuses on Pseudo Random Sequence, Elliptic Curve Cryptography and Noiseless Steganography. She is a student member of IEEE, e-mail: fatema.kumu02@gmail.com.

Md. Selim Al Mamun received his B.Sc. and MS degree in Computer Science and Engineering from University of Dhaka, Dhaka-1000, Bangladesh in 2007 and 2009 respectively. He joined as a faculty member at the Department of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul University, Trishal, Mymensingh-2223, Bangladesh in 2011. Since 2014, he has been working as a PhD student at the Graduate School of Natural Science and Technology, Okayama University, Japan. His research interests include Wireless Local Area Networks (WLANs), Cryptography and Network Security and Quantum Computing. He is a member of IEICE, e-mail: mamun0013@s.okayama-u.ac.jp.