Securing graph steganography by using Shamir’s secret sharing

A N Sari*
Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Bandung, Jalan Gegerkalong Hilir, Ciwaruga, Bandung 40012, Indonesia.

*aprianti.nanda@polban.ac.id

Abstract. Nowadays, information exchange becomes popular along with the development of information technology. To protect their confidential information, people usually use steganography because it hides the presence of the secret information in a media file called as cover. In 2008, Desoky et.al. proposed a method namely graphstega or graph steganography that conceals message as data points in a graph. However, this method is vulnerable since the value represented by the graph is not realistic and this will raise suspicion. To overcome the problem, this paper presents a novel approach by exploiting a graph built based on real data as the cover. To securing the position of the secret message, Shamir’s secret sharing scheme is used. The experiment showed that this method has better security than graphstega does.

1. Introduction
To avoid noise in a steganographic method, Desoky and Younis proposed a method called as Noiseless Steganography or Nostega [1]. This method works by producing the cover itself based on the given secret message. One of the covers in Nostega is a graph, it called as graphstega or graph steganography. Graphstega conceals the message by camouflaging it as data points in a graph. However, since the data point in the graph cover is built by the secret message, then there is a probability that the data point represents a peculiar value in some circumstances [2,3]. This condition might not satisfy imperceptibility requirement in steganographic scheme [4,5].

This paper presents a novel approach by exploiting a real graph (a graph built based on real data) as the cover. Moreover, to determine the position of the secret message in the graph cover, Shamir’s secret sharing is used since it is a simple and elegant secret sharing scheme [6,7,8].

2. Method
In this section, graphstega and Shamir’s secret sharing scheme are explained. These two preliminary studies are the foundation of the proposed method.

2.1. Graphstega
Graph steganography or graphstega is a method to conceal secret messages as data points in a graph. Graphstega works as follows, suppose there are two parties Alice and Bob. They agreed to communicate securely by concealing message as data points in a graph. The subject of the graph is chosen based on their profession. As an example, Alice wants to send a secret message “this is my secret”. Then, she converts it into binary based on ASCII code of each character as follows 011101000110100001101001
011100110010000001101001110110010000000110010110110111001100111100100100000111001101100101011100011011100100110010101110100

The next step is slicing the binary into groups of n bits. As an example, n is 5, such that the sliced binary as follows.

| Binary | Integer |
|--------|---------|
| 01110  | 7       |
| 10001  | 9       |
| 10100  | 26      |
| 00110  | 14      |
| 10010  | 18      |
| 11100  | 30      |
| 11001  | 25      |
| 00000  | 0       |
| 01101  | 17      |
| 11001  | 29      |
| 00100  | 8       |
| 00001  | 1       |
| 11001  | 29      |
| 10110  | 22      |
| 01010  | 10      |
| 11000  | 24      |
| 10111  | 27      |
| 10010  | 18      |
| 01100  | 14      |
| 01100  | 14      |
| 11010  | 26      |
| 00000  | 0       |

Each group binary is converted into integer and generate a graph based on the converted integer. Since Alice is a human resource manager in her company, then the suitable title of her graph cover is about wage compression rate. The generated graph cover is seen in Figure 1.

![Graph cover](image)

**Figure 1.** Graph cover.

After receive the graph cover, Bob will convert the data point into binary and convert it into text to read the secret message.

2.2. Shamir secret sharing scheme

In 1979, Adi Shamir proposed a secret sharing scheme that divides data or secret D into n pieces called shares $D_1, D_2, ..., D_n$ and give the shares to each participant uniquely [9]. To reconstruct the secret, at least need any $k$ combination of $n$ shares or can be described as $(k, n)$ scheme. The detailed share generation is described as follow.

- Step 1. Determine the secret.
- Step 2. Determine the $(k, n)$ scheme.
- Step 3. Determine the polynomial which has degree of $k - 1$.
- Step 4. By using the determined polynomial, picks any $x$ values except $x = 0$ for $n$ times. Then calculate the $y$ value of each $x$ based on the polynomial in the previous step.
- Step 5. Each pair of $x$ and $y$ from the previous step will distribute to $n$ participants uniquely.

To reveal the secret, at least $k$ shares are known. The steps to reveal the secret is by using Lagrange formula with given $k$ points as shown in (1).

$$f(x) = \sum_{i=1}^{k} y_i \prod_{j=1, j\neq i}^{k} \frac{x - x_j}{x_i - x_j}$$ (1)
2.3. Proposed methods
In this section, the proposed method is discussed. The proposed method is using realistic graph (graph that built based on existing data) as the cover. To determine where the message is camouflaged, Shamir’s secret sharing is used.

2.3.1. Message concealing. To conceal the message, there are three main processes described as follows.

- Convert plain text to binary. For instance, the secret message will be hidden is “Use my secret key”, then the binary representation is 010101011100110110101110101011100011011010000000111011101001100101100011011011100100001110110101110110101110001101101011101111001.

- Plaintext code generation. The graph used as cover is shown in Error! Reference source not found. that taken from legal institution [10]. If there are floating data, use floor function to convert it to integer. Hence the plotted data \( A = \{13, 8, 9, 13, 10, 7, 13, 32, 9, 5, 2, 3, 9, 14, 3, 2, 2, 4, 9, 15, 22, 4, 33, 2, 3, 5, 9, 2, 2, 3, 2, 2, 1, 3\} \). Since the maximum value of plotted data is 33, then it needs 6 bits to represent it in binary. Therefore, the binary representation of plain text in the previous step is sliced into groups of 6 bits as follows.

\[
\begin{align*}
010101 & \quad 010111 & \quad 001101 & \quad 100101 & \quad 001000 & \quad 000110 & \quad 110101 & \quad 111001 & \quad 001000 & \quad 000111 & \quad 001101 & \quad 100101 & \quad 011000 & \quad 110111 & \quad 010010 & \quad 101101 & \quad 011011 & \quad 100100
\end{align*}
\]

Each group is converted into integer \( B = \{21, 23, 13, 37, 8, 6, 53, 57, 8, 7, 13, 37, 24, 55, 9, 37, 29, 2, 1, 43, 25, 23, 36\} \). Then calculated the plain text code \( C \) by perform XOR operation between \( A \) and \( B \) based on (2)

\[
C = \{a_i \mod n \oplus b_i, ..., a_m \mod n \oplus b_m \}
\]

Where \( n = |A|, m = |B| \) and \( i \) satisfied \( 0 \leq i \leq m \). Hence \( C = \{24, 31, 4, 40, 2, 1, 56, 25, 1, 2, 15, 38, 17, 57, 10, 39, 31, 6, 8, 36, 15, 19, 5\} \).

![Figure 2. Graph cover used in proposed method.](image-url)
Share generation. Each element in \( C \) is treated as secret to generate the share by using Shamir’s secret sharing scheme. In this example, the (3, 3) scheme is used. The result of share generation is shown in Table 1.

### Table 1. Share generation result.

| \( C_i \) | Polynomial | Generated shares |
|---------|------------|------------------|
| 24      | \( f(x) = 21x^2 + 52x + 24 \) | (46, 89), (67, 9), (52, 23) |
| 31      | \( f(x) = 38x^2 + 72x + 31 \) | (26, 18), (16, 3), (53, 95) |
| 4       | \( f(x) = 5x^2 + 95x + 4 \) | (72, 40), (1, 3), (25, 50) |
| 40      | \( f(x) = 48x^2 + 46x + 40 \) | (76, 4), (71, 46), (18, 58) |
| 2       | \( f(x) = x^2 + 70x + 2 \) | (39, 11), (34, 3), (85, 47) |
| 1       | \( f(x) = 99x^2 + 38x + 1 \) | (81, 57), (87, 87), (30, 48) |
| 56      | \( f(x) = 96x^2 + 23x + 56 \) | (82, 36), (80, 95), (74, 32) |
| 25      | \( f(x) = 97x^2 + 87x + 25 \) | (53, 66), (42, 57), (1, 7) |
| 1       | \( f(x) = 44x^2 + 1 \) | (97, 99), (65, 61), (53, 74) |
| 2       | \( f(x) = 79x^2 + 19x + 2 \) | (13, 66), (29, 29), (78, 47) |
| 15      | \( f(x) = 18x^2 + 4x + 15 \) | (56, 26), (34, 52), (31, 65) |
| 38      | \( f(x) = 96x^2 + 49x + 38 \) | (13, 30), (17, 32), (45, 97) |
| 17      | \( f(x) = 74x^2 + 39x + 17 \) | (90, 58), (17, 48), (35, 21) |
| 57      | \( f(x) = 2x^2 + 59x + 57 \) | (43, 20), (20, 17), (87, 27) |
| 10      | \( f(x) = 18x^2 + 78x + 10 \) | (36, 88), (21, 92), (82, 77) |
| 39      | \( f(x) = 48x^2 + 46x + 39 \) | (77, 20), (36, 71), (58, 54) |
| 31      | \( f(x) = 55x^2 + 65x + 31 \) | (1, 50), (62, 48), (98, 28) |
| 6       | \( f(x) = 10x^2 + 34x + 6 \) | (31, 65), (38, 83), (22, 39) |
| 8       | \( f(x) = 63x^2 + 39x + 8 \) | (54, 83), (92, 13), (19, 60) |
| 36      | \( f(x) = 65x^2 + 13x + 36 \) | (17, 54), (37, 16), (84, 16) |
| 15      | \( f(x) = 27x^2 + 94x + 15 \) | (85, 70), (99, 36), (98, 77) |
| 19      | \( f(x) = 16x^2 + 3x + 19 \) | (23, 68), (2, 89), (63, 82) |
| 5       | \( f(x) = 26x^2 + 43x + 5 \) | (49, 100), (16, 77), (22, 1) |

After the concealing process was done, the sender transmits the graph cover in public channel while the shares from Table 1 is transmitted through private channel.

2.3.2. Message revealing. In the receiver side, the message revealing process is conducted to obtain the secret message. This process is described as follows.

- Plain text code reconstructions. The shares that transmitted by sender through a private channel is used to reconstruct the plain text code by using Lagrange formula (1). This step is similar with shares reconstruction process in section 2.2. Thus, the plain text code after the reconstruction process is \( C = \{24, 31, 4, 40, 2, 1, 56, 25, 1, 2, 15, 38, 17, 57, 10, 39, 31, 6, 8, 36, 15, 19, 5\} \).

- Message extraction. The integer value of plotted data from graph cover \( A = \{13, 8, 9, 13, 10, 7, 13, 32, 9, 5, 2, 3, 9, 14, 3, 2, 2, 4, 9, 15, 22, 4, 33, 2, 3, 5, 9, 2, 2, 3, 2, 2, 1, 3\} \) and \( C \) from the previous step is used to obtain the \( B \) by using (3). Thus \( B = \{21, 23, 13, 37, 8, 6, 53, 57, 8, 7, 13, 37, 24, 55, 9, 37, 29, 2, 1, 43, 25, 23, 36\} \)

\[
B = \{a_i \mod n \oplus c_i \ldots , a_m \mod n \oplus b_m \}
\]  

(3)

Since the maximum value of plotted data has 6 bits to represent, then convert \( B \) into 6-bits binary as follows

010101 010111 001101 100101 001000 000110 110101 111001 001000
000111 001101 100101 011000 110111 001001 100101 011101 000010
000001 100101 011001 010111 100100
Each group of 6-bit are concatenated and converted into characters based on ASCII code. And the resulted secret message was revealed as “Use my secret key”.

3. Results and discussion
The experiments are done to evaluate the suspicion and security level of the proposed method.

3.1. Suspicion analysis
The strategy to analyze suspicion level of graphstega and proposed method is by comparing the graph cover in both methods [3]. In graphstega, the resulted graph cover is shown in Figure 1 will raise suspicion. This phenomenon happens because how could be possible a wage compression rates are almost zero in Malawi, Philippines, and Australia. This unnatural data will raise suspicion and further investigation might have conducted by the third parties. Meanwhile, the proposed method was avoided suspicion of the third parties since the graph cover (see Figure 2) was built based on existing data.

3.2. Security analysis
The strategy to analyze security of graphstega and proposed method is by comparing the probability of the third parties guess the secret message with given graph cover in both methods [3]. In graphstega, the third parties are always success guess the secret message since they just convert the plotted value of the graph cover in binary and then convert the binary to characters. In the proposed method, the probability of guessing secret message is depend on the length of the message as seen in (4).

\[
P(\text{guessing}) = \left( \frac{1}{2^n} \right)^{\left\lfloor \frac{m \times 8}{n} \right\rfloor}
\]

Since the secret message is plotted between 0 and \(2^n\) where \(n\) is the length of binary representation of maximum value in graph cover, then the probability of guessing secret message is depends on number of plain text code or can be calculated by \(\left\lfloor \frac{m \times 8}{n} \right\rfloor\) where \(m\) is how many characters in the secret message.

The comparison of probability successful guess in both methods is shown in Figure 3.

4. Conclusion
Based on the analysis and discussion in previous section, it can be concluded that hide the presence of secret message by using a realistic graph as cover has better performance against suspicion than graphstega does. Moreover, the proposed method also has better performance in security by calculated the probability of successful guessing secret message by third parties if they only know the graph cover.
Acknowledgment
This research was supported by Politeknik Negeri Bandung. I would express my gratitude to my fellow in Jurusan Teknik Komputer dan Informatika Politeknik Negeri Bandung especially Trisna Gelar for useful discussions.

References
[1] Desoky A and Younis M 2008 Graphstega: graph steganography methodology Journal of Digital Forensic Practice 2(1) 27-36
[2] Desoky A 2010 Comprehensive linguistic steganography survey International Journal of Information and Computer Security 4(2) 164-197
[3] Sari A N and Barmawi A M 2016 Modified Graphstega Based on Chart Scale Modification In 2016 European Intelligence and Security Informatics Conference (EISIC) 156-159
[4] Amirtharajan R and Rayappan J B B 2013 Steganography-time to time: A review Res. J. Inform. Technol 5 53-66
[5] Rakhi S G 2013 A review on steganography methods International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering 2(10)
[6] Tompa M and Woll H 1989 How to share a secret with cheaters journal of Cryptology 1(3) 133-138
[7] Ulutas M, Ulutas G and Nabiyev V V 2011 Medical image security and EPR hiding using Shamir's secret sharing scheme Journal of Systems and Software 84(3) 341-353
[8] Sarrna K S, Larnkuche H S and Urnamaheswari S 2013 A Review of Secret Sharing Schemes Research Journal of Information Technology 5 67-72
[9] Shamir A 1979 How to share a secret Communications of the ACM 22(11) 612-613
[10] The World Bank n.d. Wage Compression Rate [online] retrieved from https://databank.worldbank.org/data/download/catalog/wage_compression_data.xls