RGB-Based Text Steganography Using Randomized Embedding Positions for Secure Message Hiding

Baharudin, O, B.[1*], Amran, A.[2], and Abd. Razak, R.[3]
[1,2,3]*School of Computing, Universiti Utara Malaysia, Sintok, Malaysia*
*Contact: bahaosman@uum.edu.my

Abstract— **Text steganography remains a significant challenge in information security due to the limited redundancy available in textual data. This paper introduces a lightweight text-based steganography method that uses RGB triplets—derived from ASCII codes—to embed secret messages into a cover text at randomized positions. A reversible digit-mapping rule converts ASCII values into RGB triplets, which are stored in a stego-key alongside the corresponding embedding positions and sequence indices. The technique preserves cover text readability, resists pattern-based detection, and is computationally efficient. Experimental evaluation demonstrates the method's feasibility and suitability for contexts where covert communication through textual channels is preferred. Future work may integrate zero-width character embedding, automated key generation, and natural-language–driven steganographic enhancements.**

Keywords: Data Analytics, Sentiment Analysis, Predictive Analytics

I. INTRODUCTION

Steganography conceals the existence of a message rather than merely encrypting its content. Although image- and audio-based steganography have been widely studied, text-based methods face greater challenges due to minimal structural redundancy and susceptibility to visible distortion. Recent surveys highlight that text steganography remains an evolving research domain, with significant limitations in embedding capacity, robustness against attacks, and dependence on linguistic features [1].

The rise of text-centric communication platforms, including emails, messaging apps, and online forums, increases the relevance of secure text steganography, particularly in environments where transmitting images or audio may attract unwanted attention. Existing methods, such as whitespace manipulation, linguistic substitution, and invisible Unicode characters, often struggle with reliability, limited payload capacity, or detectability [2], [3]. Linguistic-based approaches, while preserving fluency, have low capacity and are sensitive to contextual or domain shifts, making them unsuitable for embedding larger messages.

To address these challenges, this study proposes a randomized RGB-based embedding strategy derived from ASCII codes, which maps message characters to Red, Green, Blue (RGB) values and embeds them at non-sequential positions in the cover text. The proposed approach maintains the visual integrity of the cover text, avoids linguistic modifications, and is language-independent. By leveraging a pseudorandom number generator (PRNG) for position selection, the method ensures both security and reproducibility while significantly improving embedding capacity compared with conventional text steganography techniques. It conceals the existence of a message rather than merely encrypting its content. Although image- and audio-based steganography have been widely studied, text-based methods face greater challenges due to minimal structural redundancy and susceptibility to visible distortion. Recent surveys highlight that text steganography remains an evolving research domain with significant limitations in capacity, robustness, and linguistic dependency [1].

II. RELATED WORK

The following paragraph provides a review of the related work that is relevant to this study, highlighting previous approaches, their limitations, and how they inform the development of the proposed text steganography method.

A. Format-based techniques

Methods such as whitespace manipulation, indentation changes, and font variation attempt to hide patterns in formatting. These methods are vulnerable to automatic text reformatting and are easily disrupted during document conversion [2]. Example: A hidden binary message can be encoded by inserting one or two spaces at the end of each line. However, if the document is copied into a plain-text editor, the extra spaces are removed, destroying the embedded data.

B. Zero-width and invisible character techniques

Invisible Unicode characters have been shown to increase payload while remaining imperceptible [3]. More recent studies demonstrate strong imperceptibility and high capacity using zero-width characters for embedding [4]. Zero-width characters can only embed a small amount of data per cover text. Example: A typical sentence of 100 characters can accommodate only

around 50–100 bits if one zero-width character is used per character. This is much lower than other steganography methods such as whitespace, formatting-based, or multimedia embedding.For large messages, the technique becomes impractical, as multiple zero-width characters may make the text suspicious or increase file size unnecessarily.

C. Linguistic and generative steganography

Modern approaches leverage synonym substitution, syntactic rewriting, and even language models. Recent work using BERT-based autoregressive generation and semantic-aware coding attempts to preserve fluency while embedding data [5], [6]. Knowledge-graph-guided generative steganography has been explored to maintain semantic coherence [7]. One of the limitation of this technique is the embedding capacity is relatively low, as only limited modifications (e.g., synonyms or syntactic changes) can be applied without affecting naturalness, making it impractical for large messages. For example, consider embedding 1 KB message (~1024 bytes) using synonym substitution, you would need ~8,000 words (assuming 1–2 bits per word), which is extremely long and impractical for normal text. This illustrates the low embedding capacity of linguistic steganography, highlighting why it is unsuitable for large messages.

D. Colour- and metadata-based embedding

RGB-based embedding schemes have been applied in text and images. Osman et al. proposed using dark colour coding combined with mathematical encoding to hide secret data [8]. These methods inspired the use of numeric colour components as carriers for text-based embedding. However special symbol, and numbers is not allowed by using this technique.

III. METHODOLOGY

The proposed technique for secure message embedding involves three primary steps:

Step 1: ASCII-to-RGB Conversion

Each character of the secret message is first converted into its corresponding ASCII value (e.g., M = 077, E = 069). This numerical representation is then mapped to an RGB triplet, effectively transforming textual data into colour codes. This conversion ensures that each character is represented uniquely as a combination of Red, Green, and Blue values. For example

$$\text{Character E} = \text{ASCII } 0\ 6\ 9 \rightarrow \text{RGB } (0, 6, 9)$$

In this study, the phrase “MEET YOU AT TEN” was employed as the hidden message. The encoded message was then processed to generate the following results, which are presented in the table below for clarity and analysis.

TABLE I
HIDDEN MESSAGE REPRESENTATION IN RGB COLOR

Sequence	Char	ASCII	Digits (R G B)
1	M	77	0 7 7
2	E	69	0 6 9
3	E	69	0 6 9
4	T	84	0 8 4
5	(space)	32	0 3 2
6	Y	89	0 8 9
7	O	79	0 7 9
8	U	85	0 8 5
9	(space)	32	0 3 2
10	A	65	0 6 5
11	T	84	0 8 4
12	(space)	32	0 3 2
13	T	84	0 8 4
14	E	69	0 6 9
15	N	78	0 7 8

Step 2: Randomized Embedding of RGB Codes in the Cover Text

The generated RGB triplets are embedded into randomly selected positions within the cover text using PRNG algorithm. A PRNG is a mathematical algorithm that produces numbers that look random, but are actually fully deterministic The randomness of the embedding process enhances security by making it difficult for an attacker to predict the location of the hidden data. The positions of these embedded triplets are stored in a stego-key, which acts as a guide for extracting and reconstructing the hidden message.

The following are a random number generated by PRNG using a seed key (2025)

53, 2, 59, 27, 39, 12, 44, 7, 36, 61, 14, 48, 25, 31, 10

For this study, the cover text selected for embedding the hidden message is presented below.

“Digital communication has become a routine part of daily life, allowing people to exchange messages quickly and easily. As information travels across various platforms, concerns about privacy and data protection continue to grow. To address these challenges, researchers often explore subtle techniques that allow important details to be hidden within ordinary text. When crafted properly, a cover text appears completely natural and does not draw attention to itself. It maintains a smooth flow, consistent tone, and coherent structure so that readers notice nothing unusual. At the same time, the text provides enough length and variability to support the embedding of coded information at predefined positions. These positions are usually generated by a shared key or a pseudorandom number sequence. Anyone with the correct key can later retrieve the embedded values and reconstruct the hidden message. This balance between readability and concealment makes text-based embedding a practical and useful method.”

Step 3 : Decoding Process

During the extraction phase, the stego-key is used to identify the exact positions of the embedded RGB triplets within the cover text. The RGB values are then converted back to their corresponding ASCII values, allowing the original secret message to be accurately reconstructed. The reliance on the stego-key ensures that only authorized users with the correct key can retrieve the hidden message. The algorithm for the decoding process as below.

1. Extracting RGB values based on stored positions
2. Converting RGB values back to ASCII digits
3. Ordering characters by sequence index

Finally, ASCII values are mapped back to characters to reconstruct the original secret message.

Overall, this method provides a secure and efficient mechanism for concealing textual information within another text medium, combining the unpredictability of randomized embedding with the precision of key-based reconstruction.

IV. RESULT

The results of the proposed technique indicate that the generated stego text exhibits characteristics highly consistent with the original stego text. In other words, the application of the embedding algorithm does not introduce noticeable changes in the structure, readability, or appearance of the text, maintaining the natural flow and coherence of the original content. This outcome demonstrates that the proposed method effectively conceals the hidden message while preserving the integrity of the cover text. As illustrated in the example below, the stego text produced by the proposed technique is indistinguishable from the original cover text, with no observable differences in structure, readability, or appearance to the human reader.

"Digital communication has become a routine part of daily life, allowing people to exchange messages quickly and easily. As information travels across various platforms, concerns about privacy and data protection continue to grow. To address these challenges, researchers often explore subtle techniques that allow important details to be hidden within ordinary text. When crafted properly, a cover text appears completely natural and does not draw attention to itself. It maintains a smooth flow, consistent tone, and coherent structure so that readers notice nothing unusual. At the same time, the text provides enough length and variability to support the embedding of coded information at predefined positions. These positions are usually generated by a shared key or a pseudorandom number sequence. Anyone with the correct key can later retrieve the embedded values and reconstruct the hidden message. This balance between readability and concealment makes text-based embedding a practical and useful method."

Pseudorandom number generators (PRNGs) can produce sequences of numbers that are useful for determining the positions in which a message can be embedded. Table II presents five sets of random numbers generated using a seed value of 7777 with Python's deterministic PRNG. All numbers were produced within the range of 1 to 150. In particular, the values in Set 2 correspond to PRNG calls numbered 16 through 30, ensuring reproducibility and deterministic placement of the hidden message.

TABLE II
VARIOUS SET OF PRNG RANDOM NUMBER

Set 1	71, 141, 82, 5, 52, 46, 50, 67, 94, 133, 24, 139, 135, 108, 144
Set 2	7, 29, 48, 99, 68, 121, 31, 146, 112, 131, 6, 16, 128, 42, 27
Set 3	114, 104, 62, 128, 94, 57, 117, 73, 81, 132, 100, 149, 42, 64, 5
Set 4	144, 45, 19, 133, 103, 14, 61, 13, 112, 35, 49, 40, 23, 75, 116
Set 5	80, 135, 38, 121, 144, 56, 28, 11, 108, 59, 10, 83, 94, 101, 33

Table III presents a comparison of the embedding capacity of the proposed technique with of other existing techniques.

TABLE III
VARIOUS SET OF PRNG RANDOM NUMBER

Technique	**Bits per char**	**1011 chars (with spaces)**	**Notes**
Format-based	~1 bit per space/line	~148 bits (18.5 bytes)	Fragile
Zero-width	~1 bit per word boundary	~148 bits (18.5 bytes)	Moderate
Linguistic	~1 bit per substitution	~90–180 bits (11–22 bytes)	Higher but semantic drift
RGB (full 24-bit)	24 bits	24,264 bits (3,033 bytes)	Very high capacity

Using **RGB color encoding** massively increases capacity compared to format-based, zero-width, or linguistic methods. Even with restricted levels (0–8), this technique can embed **over 1 KB of hidden data** in a single 1,000-character text, far beyond what spacing or synonym tricks allow.

V. CONCLUSION

This study proposed a randomized RGB-based text steganography method that combines ASCII digit mapping with non-sequential embedding positions to securely conceal messages within text. The technique preserves the visual integrity and readability of the cover text while enhancing security through deterministic yet randomized placement of hidden data. Compared with conventional linguistic or model-based steganography approaches, it provides a lightweight, reproducible, and practical solution that balances embedding capacity and imperceptibility. Experimental results confirm that

the generated stego texts are visually indistinguishable from the original cover texts, demonstrating the method's effectiveness in maintaining text fidelity and resisting detection. The use of a pseudorandom number generator ensures reproducibility and controlled randomness, while the RGB-based encoding allows precise embedding at the character level. Future research may explore the potential directions include investigating machine learning–based steganalysis resistance to make the embedding less detectable by AI-based detection tools, developing dynamic RGB encoding schemes that adapt to different character distributions in the cover text, and exploring cross-platform text compatibility to ensure the hidden message remains intact across various text editors, web applications, and file formats. Additionally, research could examine hybrid embedding strategies that combine RGB-based encoding with other text steganography techniques, such as whitespace manipulation or synonym substitution, to further enhance capacity and security

REFERENCES

[1] M. A. Majeed, R. Sulaiman, Z. Shukur, and M. K. Hasan, "A review on text steganography techniques," Mathematics, vol. 9, no. 21, 2021.

[2] A. A. Idress and Y. H. Ismael, "Proposed method for text steganography," JMCER, 2024.

[3] M. Abood, "Using invisible characters to hide text within text," Int. J. Sci. Res. Sci. Eng. Technol., 2025.

[4] S. Ishtiaq et al., "Enhancing imperceptibility: Zero-width character-based text steganography for message privacy," Acta Informatica Pragensia, 2025.

[5] X. Zheng and H. Wu, "Autoregressive linguistic steganography based on BERT and consistency coding," arXiv, 2022.

[6] T. Yang, H. Wu, B. Yi et al., "Semantic-preserving linguistic steganography via pivot translation and semantic-aware coding," arXiv, 2022.

[7] Z. Yang et al., "Graph-Stega: Semantic controllable steganographic text generation guided by knowledge graph," arXiv, 2020.

[8] B. Osman, N. I. Yahya, K. M. Zaini, and A. Abdullah, "Text steganography using the second quotient remainder theorem and dark colour schemes," JCIA, 2023.

[9] Al Azzam, "Text steganography via ASCII bit-mapping," JASET, 2023.

[10] Irfan et al., "Linguistic steganography methods: A comprehensive analysis," IRJIET, 2023.

[11] M. Wildan and W. M. Ashari, "Text data security using LCG and CBC with steganography technique," JAIC, 2024.

[12] A. I. Al-Hasan et al., "A new steganography method for hiding text into RGB image," IJACSA, 2023.

[13] H. Muhammad et al., "Knowledge security through text steganography: A review," JISTM, 2024.

[14] Review Group, "Randomization strategies in steganography," CMC, 2024.

[15] Anonymous, "Text Steganography Techniques: A Review," IRJIET, 2023.