A SECURE IMAGE STEGANOGRAPHY SYSTEM USING GENERATIVE MODEL AND ADVANCED ENCRYPTION TECHNIQUES

Sridharan S(Assistant Professor)[1], Chandru S[2], Madhavan S[3], Naveen R[4]

Department of Computer Science and Engineering, University College of Engineering, Thirukkuvalai.
(A constituent College of Anna University::Chennai and Approved by AICTE, New Delhi)

---***---

ABSTRACT - This work presents a novel method of image steganography in which the Glow model, an advanced generative model, is used to effectively insert a secret image within a cover image. Unlike traditional approaches that only hide text messages, our suggested solution significantly increases concealment ability by storing whole secret images. Furthermore, the system incorporates AES/Blowfish encryption techniques as an extra layer of protection to strengthen security measures and resilience against potential image-based attacks. This technique ensures that the hidden image is imperceptible to the human eye by using the Glow model to properly conceal it within the cover image. By encrypting the stego-image, the use of AES/Blowfish encryption improves security by reducing the possibility of unwanted access or manipulation. The effectiveness and resilience of the suggested system are proven by comprehensive testing and assessment. Effective steganography techniques, along with encryption techniques, provide a powerful solution to protect private image data in digital communication channels. All things considered, our study offers a thorough and practical method for image steganography, meeting the demand for increased security and concealment capability in modern data transfer settings.

***Key Words*:** Generative model, Glow model, Image steganography, AES encryption technique, Blowfish encryption technique.

1. INTRODUCTION

In the domain of digital communication security, image steganography is crucial for concealing sensitive information within images. However, existing systems encounter challenges such as vulnerability to attacks and limited concealment capabilities. This project introduces a novel approach that overcomes these limitations by utilizing the Glow model to conceal entire secret images within cover images. Additionally, AES/Blowfish encryption techniques are integrated to enhance security. This combination ensures imperceptibility of the hidden image and strengthens resilience against unauthorized access. The proposed system offers a significant advancement in image steganography, providing enhanced concealment capacity and robust security measures. The Glow model is a sophisticated technique used in image steganography, which is a technique for hiding secret information within images. Unlike traditional methods that can only hide text, the Glow model can hide entire images within other images. It works by carefully blending the secret image into the cover image, making it look like a normal picture to the human eye. In our project, we're using the Glow model to conceal secret images within cover images, making it difficult for anyone to detect that there's hidden information present. This advanced technique allows us to enhance the concealment capacity of our steganography system, making it more versatile and effective in securing sensitive data. By incorporating the Glow model into our project, we're able to take a significant step forward in the field of image steganography, offering a more robust and secure solution for concealing sensitive information within digital images. The AES (Advanced Encryption Standard) and Blowfish techniques are encryption methods used to enhance the security of digital data. They work by scrambling the information in such a way that only authorized users with the correct decryption key can access it. In our project, we're integrating AES/Blowfish encryption techniques as an additional layer of security to protect the concealed images within our steganography system. By encrypting the stego-image, we ensure that even if someone tries to access the information, they won't be able to decipher it without the proper decryption key. By incorporating AES/Blowfish encryption into our project, we're adding an extra level of protection to safeguard sensitive data from unauthorized access or tampering. This enhances the overall security of our steganography system, making it more robust and reliable for secure communication. This introduction sets the stage for exploring the methodology, advantages, and process flow of the proposed system, demonstrating its potential to address the evolving security needs in digital communication channels.

2. LITERATURE SURVEY

1) In recent decades, researchers have explored various methods of image steganography, focusing on concealing secret information within cover images. Early techniques involved modifying the least significant bit (LSB) of pixel values in cover images to embed secret data subtly.
2) To address the issue of image distortion caused by modification, adaptive steganographic approaches emerged. These methods introduced distortion functions to minimize additive distortion, resulting in less perceptible changes to the cover image.
3) Deep learning techniques have made significant strides in computer vision, prompting researchers to explore using neural networks to automatically learn distortion functions for steganography. These approaches have shown promise for further reducing image distortion.

4) Steganalysis, the detection of hidden information, relies on handcrafted or learned features to identify steganographic content within images. Conventional steganography often leaves detectable traces due to inherent image distortion, making detection challenging.
5) Generative steganography offers a novel approach by generating new images containing hidden information, potentially evading detection. This method involves creating realistic-looking images to conceal secret messages, thus enhancing stealthiness.
6) Secret messages are concealed in created texture images using generative steganographic techniques based on texture creation. However, these images may appear meaningless and raise suspicion during transmission.
7) Generative Adversarial Networks (GANs) provide a promising avenue for generative steganography, producing realistic-looking images that resemble authentic photographs. Researchers have explored leveraging GANs to transform secret messages into visually convincing stego-images.
8) By translating coded signals into noise, some researchers use GANs to produce stego-images. However, accurately extracting information from these images remains challenging, especially with high hiding payloads.
9) Current generative steganographic techniques have difficulty striking a compromise between extraction accuracy and concealment capability. Efforts are underway to develop schemes that achieve efficient and reversible transformation between secret messages and generated images while maintaining anti-detectability and imperceptibility.
10) The proposed system based on the Glow model and AES/Blowfish encryption techniques aims to address these challenges by achieving high-capacity information hiding and accurate information extraction simultaneously. And it achieves enough robustness against strong image attacks. For generative steganography, this method should preserve the necessary imperceptibility and anti-detectability.

3. EXISTING SYSTEM

In the existing image steganography system utilizing the Glow model, notable strengths include a high hiding capacity of up to 4 bits per pixel (bpp) and precise information extraction, achieving a 100% success rate. However, vulnerabilities to robust image attacks compromise its resilience. Moreover, limitations in the quality of generated images and discernible differences between real and synthesized images pose challenges for covert communication. Addressing these concerns is crucial for enhancing system effectiveness, necessitating further research to bolster resistance to attacks, improve image quality, and minimize visual disparities between authentic and synthesized images.

Disadvantages of the existing system:

- ➢ The existing system is not robust enough to strong image attacks.
- ➢ The created photographs have limited quality,
- ➢ The real and generated images differ visually.

4. PROPOSED SYSTEM

In the proposed image steganography system, a significant departure from convention is introduced by concealing a secret image within the cover image, employing the Glow model as the generative model. This innovative approach enhances the system's concealment capabilities, enabling the embedding of richer and more complex information within the cover image. Furthermore, to fortify the system against potential image attacks, an additional security layer is proposed. This advancement entails the integration of AES/Blowfish encryption techniques, serving as a robust defence mechanism to safeguard the concealed information from unauthorized access or detection. By combining the concealment of secret images with advanced encryption methods, the proposed system not only expands the scope of steganographic communication but also reinforces the security and resilience of the covert communication channel, making it more adept at protecting sensitive information from adversaries.

Advantages of the proposed system:

- ➢ Enhanced security: By incorporating AES/Blowfish encryption techniques as a security layer, the proposed system significantly improves its resilience against strong image attacks and unauthorized access.
- ➢ Increased concealment capacity: Unlike the existing system that hides only textual messages, the proposed system can conceal entire secret images within the cover images.

5. METHODOLOGY

5.1 GLOW MODEL

The GLOW (Generative Latent Optimization) model is a type of generative model used in machine learning, particularly for generating realistic images. It employs autoregressive modelling and invertible transformations to progressively generate high-quality images. GLOW utilizes a flow-based architecture, with coupling layers facilitating efficient data transformation. It learns a latent representation of images, enabling manipulation for tasks like image steganography. Through optimization, it minimizes the discrepancy between generated and real data distributions. GLOW's effectiveness lies in its ability to produce lifelike images while maintaining tractability and preserving information content, making it a powerful tool in image generation tasks.

GLOW model in steganography of images:

Generative Capability: Realistic visuals can be produced using the GLOW model. In steganography, this capability can be leveraged to generate cover images that serve as carriers for hidden information. These cover images should look natural and not arouse suspicion.

Information Embedding: The created cover images can have information embedded by the GLOW model. This embedding process involves modifying certain features or characteristics of the image in a subtle manner so that the hidden information is imperceptible to the human eye. This can be achieved through various techniques, such as modifying pixel values or introducing imperceptible perturbations.

Latent Space Manipulation: GLOW learns a latent representation of images, which captures essential features. In steganography, this latent space can be manipulated to encode hidden information. By carefully adjusting the latent variables, one can embed data in a manner that is difficult to detect without knowledge of the embedding process.

Security and Robustness: Since GLOW is capable of generating high-quality images, the embedded information is less likely to be detected by visual inspection or statistical analysis. Additionally, the use of invertible transformations and optimization techniques in GLOW ensures that the embedded information remains intact even after various transformations or attacks on the image.

Reconstruction and Extraction: At the receiver end, the embedded information can be extracted from the steganographic image using the GLOW model. By analyzing the differences between the generated image and the original cover image, one can retrieve the hidden data.

Overall, the GLOW model can be a valuable tool in image steganography, providing a means to embed information securely within images while maintaining their visual fidelity. Its generative capabilities, latent space manipulation, and robustness make it well-suited for this task.

Here is a brief description of the Glow model.

Let x be the variable of image space with a complex distribution px(x), and let z be the variable of latent space with the simple distribution pz(z), Specifically, a multivariate spherical Gaussian distribution. They have a representation

$$z \sim p_Z(z) \qquad (1)$$

$$x \sim p_X(x) \qquad (2)$$

When it is assumed that z is the same dimension as x and that its components, zd are independent.

The Glow model must learn an invertible mapping function fϴ throughout the training procedure in order to produce bijective mapping. The function can be represented by the following:

$$z = f_\theta(x) \qquad (3)$$

$$x = g_\theta(z) = f_\theta^{-1}(z) \qquad (4)$$

The relationship between the distributions px(x) and pz(z), according to the Jacobian determinant, can be expressed as

$$p_X(x) = p_Z(z)\left|det\frac{\partial z}{\partial x}\right| = p_Z(f_\theta(x))\left|det\frac{\partial f_\theta(x)}{\partial x}\right| \qquad (5)$$

Assume that a training dataset Pdata has a collection of image samples x and values for each element in the dataset fall between [0,1]. The following equation is used to estimate maximum log-likelihood in order to achieve the learning of fϴ, as per the previous equation:

$$\max_{\theta\in\Theta} \mathrm{E}_{x\sim P_{data}}[\log p_X(x)]$$
$$= \max_{\theta\in\Theta} \mathrm{E}_{x\sim P_{data}}\left[\log p_Z(f_\theta(x))\left|det\frac{\partial f_\theta(x)}{\partial x}\right|\right] \qquad (6)$$

Consequently, we may establish the bijective-mapping between image space and latent space by utilizing the learnt Glow model, which is comprised of a pair of mapping functions fϴ and gϴ as indicated in Eqns. (3) and (4).

5.2 AES METHODOLOGY

The AES algorithms come in three varieties: AES-128, AES192, and AES-256. This division is carried out according to the key utilized during the encryption and decryption process. The number of bits used determines the security level. Four separate byte-oriented transformations are used by the AES algorithm.

AES algorithm: Replace the byte, join the columns, move the row and add the round key.

Encryption

i. Substitute Byte:

The initial encryption stage, referred to as Sub-Bytes, entails byte-by-byte substitution during the forwarding process.

• The state array uses a 16x16 look-up table to find a replacement byte. Here, the idea of a Galois field is employed. The items in the table are completed using the multiplicative inverses. In order to disrupt the correlations formed at the bit level, bits are also jumbled.

$$x_{out} = A \cdot x' + c$$

ii. Shift Row:

This second encryption phase involves moving some rows in the array during the forward operation.

• The Shift-Rows transformation entails:

(a) not transposing the array's very first row.

(b) transposing the array's second row one byte to the left in a clockwise manner.

(c) by flipping the row labeled "3" (2 bytes to the left).

(d), in addition to moving the last row three bytes to the left.

•The main main objective of this transposing and shifting is to mingle up all byte sequence comprising the 128- bits structure.

iii. Mix Columns:

The third encryption phase, known as Mix Columns, randomly jumbles the bytes in each column throughout the forward process.

• Every single byte in a column is enlarged with the next byte, then doubled by that byte, tripled by the next byte, and finally by the next byte.

• Following ten processing rounds, each component of the ciphertext that depends on each component of plain text is generated by the shift-rows phase adjacent to the mix-column step.

iv. Add Round-Key:

Mix Columns is the fourth and last step in the encryption process. The process causes the bytes in each column to become disorganized.

• The 128-bit encryption key is expanded into a new 128-bit key for every round using AES key-expansion. The logic of the Key Expansion method is taken into consideration to ensure that any changes made to one bit of the key should impact the keys that are used in the rounds.

Decryption

i. Shift Row Inversion

• The analogous actions during decryption jumble the rows in exactly the opposite manner from how they were done during encryption.

• There are a few phases involved:

(a) The first row is left alone.

(b) The second row's right is increased by one byte.

(c) The third row's right is expanded by two bytes.

(d) Three bytes are transferred to the final row. These transpositions are certain to be circular.

ii. Inverse Substitute Byte:

This transformation is the reverse of byte substitution, as the name implies. This stage, which is often referred to as the reverse process of Substitute Byte, comprises specifically of operations where the converse of the Sbox is enforced on each byte. By using the affine transformation's inverse, we are able to obtain inverse substitution. Moreover, in Galois Field, this process is carried out concurrently with the multiplicative inverse.

iii. Add Round Key:

The round keys are added to each and every round, just like in the encryption process. The number of round-keys added is directly influenced by the type of AES being utilized.

iv. Columns with Inverse Mix:

Compared to the previous encryption technique, this is the exact opposite of how Mix Columns were modified. Every column must be controlled and seen as a 4-term polynomial during the state process. To process the state column by column, utilize inverse mix columns.

• Finally, these columns are analyzed and evaluated as polynomials over the Galois Field (2^8) and are multiplied modulo (x^4+1) with a fixed polynomial (x), given by:

$$a^{-1}(x) = \{0b\}x\text{^}3 + \{0d\}x\text{^}\ 2 + \{09\}x + \{0e\}$$

IMPLEMENTATION

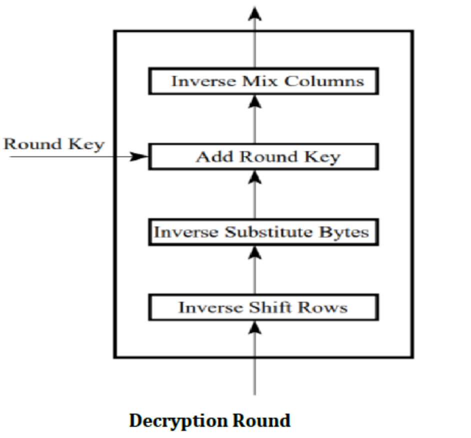

Figure 1: Block Diagram for steps involved in Encryption

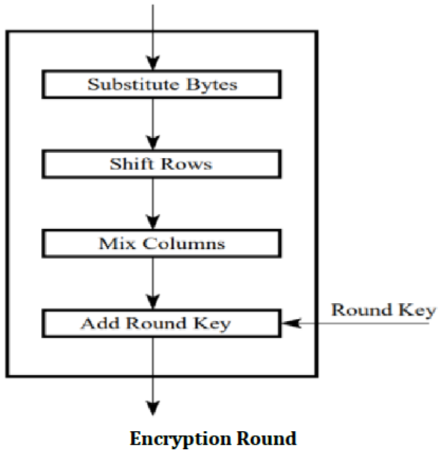

Figure 2: Block Diagram for steps involved in Decryption

There is a massive difference in the flow of the operations Decryption cannot be considered the precise reverse of encryption.

During encryption, rows are moved, columns are jumbled, and bytes are changed; in contrast, the decryption process carries out the same operations in reverse and a different sequence. The AES encryption and decryption technique is implemented using Python modules such as pycrypto, Numpy, hashlib, and cipher. For GUI implementations, Tkinter is utilized, along with AES and other protocols.

Images of various sizes were used as inputs, and a thorough examination of the performance was conducted. Figures 1 and 2 show the flow of the algorithm. In order to evaluate the efficacy of this model, we captured several photos with varying dimensions and sizes.

5.3 BLOWFISH METHODOLOGY

In 1993, Bruce Schneier created blowfish as a quick and cost-free substitute for the then-current encryption techniques. It has undergone a great deal of analysis since then, and its reputation as a powerful encryption method is gradually growing. The Blowfish algorithm has many advantages. It is suitable and efficient for hardware implementation and no license is required. The elementary operators of Blowfish algorithm include table lookup, addition and XOR. The table includes four S boxes and a P-array. Blowfish is a cipher based on Feistel rounds, and the design of the F-function used amounts to a simplification of the principles used in DES to provide the same security with greater speed and efficiency in software. It is recommended to use the 64-bit block cipher Blowfish instead of DES. 32-bit microprocessors can encrypt data using the quick Blowfish technique.

Bruce Schneier released the Blowfish block cipher in 1993. Schneier developed Blowfish to be a publicly available cryptographic algorithm with the potential to replace DES. Schneier also encouraged others to evaluate the performance and security of Blowfish. To date, the security of Blowfish has not been compromised. The 64-bit symmetric block cipher known as Blowfish has a variable-length key that can have a length of 32–448 bits (14 bytes). The technique was created to effectively and securely encrypt 64 bits of plaintext into 64 bits of cipher text. Table lookup, modulus, addition, and bitwise exclusive were the operations chosen for the technique, which aimed to reduce the amount of time needed for data encryption and decryption on 32-bit processors. A conscious attempt was made in designing the algorithm to keep the operations simple and easy to code while not compromising security. Blowfish uses a 16-round Feistel network for encryption and decryption, just like DES does. But unlike DES, which only alters the right 32-bits to become the left 32-bits for the next round, Blowfish affects both the left and right 32-bits of data throughout each round. Blowfish was used to perform a bitwise exclusive-or operation on the left 32 bits prior to the F function altering them or propagating it to the right 32 bits for the following round. In addition, Blowfish included a swap operation and two exclusive-or procedures that were to be carried out following the 16 rounds. The permutation function employed in DES is not the same as this operation.

Encryption Process:

Data image as a plaintext and the encryption key are two inputs of encryption process. In this case, original image data bit stream is divided into the block length of Blowfish algorithm.

Image header is excluded to encrypt and the start of the bitmap pixel or array begins right after the header of the file. The byte elements of the array are stored in row order from left to right with each row representing one scan line of the image and the rows of the image are encrypted from top to bottom.

Decryption Process:

The encrypted image is divided into the same block length of Blowfish algorithm from top to bottom.

The first block is entered to the decryption function and the same encryption key is used to decrypt the image but the application of sub keys is reversed. The process of decryption is continued with other blocks of the image from top to bottom.

The basic algorithm for Blowfish is illustrated as follows:

```
Divide X into two 32-bit halves XL and XR
For I = 1 to 16:
XL = XL Pi
XR = F (XL) XR
Swap XL and XR
End for
Swap XL and XR
XR = XR P17
XL = XL P18
Recombine XL and XR
Output X (64-bit data block: cipher text)
```

The identical procedure is used for decryption, with the requirement that the sub-keys Pi be supplied in reverse order. The nature of the Feistel network ensures that every half is swapped for the next round.

6. ARCHITECTURE DIAGRAM

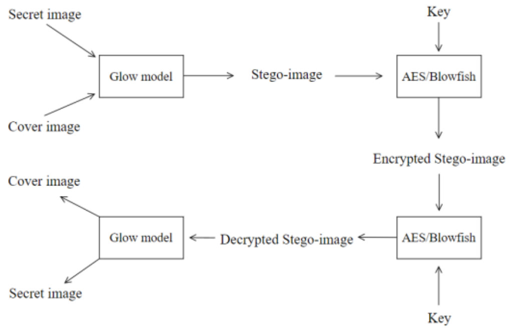

Fig -3: Architecture Diagram

7. RESULTS AND DISCUSSIONS

We present the results and discussions derived from the implementation and evaluation of our proposed Image Steganography system. Through a comprehensive analysis of experimental outcomes and a critical examination of their implications, we assess the effectiveness and limitations of the system. The section begins with an overview of the experimental setup and methodology, followed by a detailed presentation of the findings. Each result is accompanied by a discussion of its significance, drawing insights into the system's performance, security enhancements, and concealment capacity. Together, these results and discussions contribute to a deeper understanding of the proposed approach and its potential applications in the field of image steganography.

This image showcases the result of the first step in the process. It depicts the stego image generated by the GLOW model after embedding the secret image within the cover image.

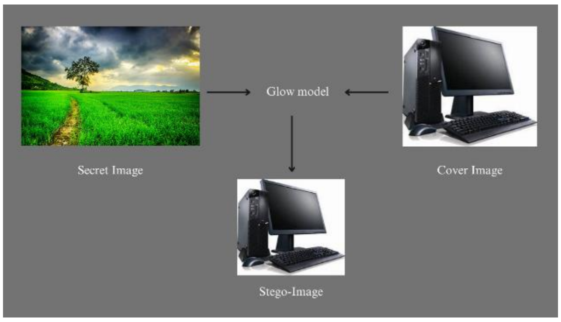

Figure 4: Creating Stego Images with the GLOW Model

This image illustrates the outcome of the second step, focusing on the encryption of the stego image generated in the previous step using AES or Blowfish algorithm.

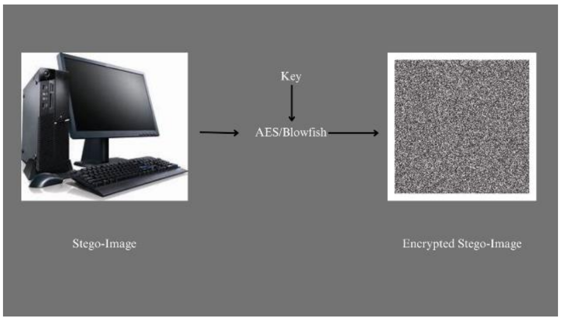

Figure 5: Encryption of Stego Image

This image represents the result of the third step, depicting the decryption process applied to the encrypted stego image using the same key.

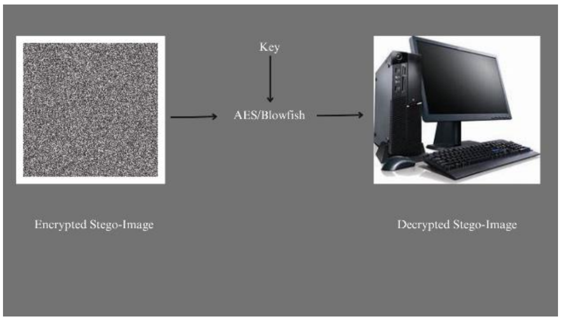

Figure 6: Decryption of Stego Image

This image showcases the final step of the process, where the decrypted stego image is inputted into the GLOW model to extract the secret image and the original cover image.

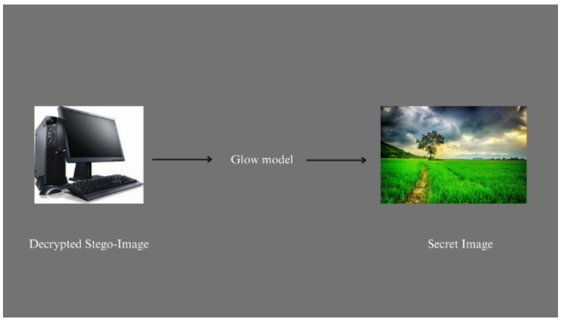

Figure 7: Decoding by GLOW Model

8. CONCLUSIONS

In this research, we proposed a novel approach to enhance the security and concealment capacity of image steganography systems. By integrating the Glow model for concealing secret images within cover images and incorporating AES/Blowfish encryption techniques as a security layer, our proposed system offers significant advancements over existing methods. Our proposed system addresses several limitations of existing systems, including vulnerability to strong image attacks, limited hiding capacity, and visual discrepancies between real and generated images. By concealing entire secret images within cover images, we increase the concealment capacity while maintaining high hiding capacity and accurate information extraction. Moreover, the integration of AES/Blowfish encryption techniques adds an additional layer of security, ensuring the confidentiality and integrity of the concealed information. This enhancement significantly improves the system's resilience against unauthorized access and strong image attacks, making it suitable for applications requiring robust and secure steganography solutions. Through a comprehensive process flow, we outlined the steps involved in our proposed system, demonstrating its effectiveness in concealing and extracting secret images seamlessly within cover images while maintaining security through encryption. Overall, our proposed image steganography system offers a promising solution for secure and efficient data concealment, with potential applications in various domains such as military communication, secure transmission of information, data storage, and digital forensics. Future research could explore further optimizations and extensions of the proposed system to enhance its performance and applicability in real-world scenarios.

9. REFERENCES

[1] "Using syndrome-trellis codes to minimize additive distortion in steganography," IEEE Trans. Inf. Forensics Secur., vol. 6, no. 3, pp. 920-935, Sep. 2011. Fridrich, T., and Filler, Judas.

[2] Soft, Z. Zhou, Y. Mu, and Q. J. Wu. "Coverless image steganography using partial-duplicate image retrieval," Comput., vol. 23, no. 13, pp. 4927–4938, 2019.

[3] Z. Zhou, H. Sun, R. Harit, X. Chen, and X. Sun, "Coverless image steganography without embedding," Proc. Int. Conf. Cloud Comput. Secur., 2015, pp. 123–132.

[4] K.-C. Wu and C.-M. Wang, "Steganography using reversible texture synthesis," IEEE Trans. Image Process., vol. 24, no. 1, pp. 130–139, Jan. 2015.

[5] In IEEE Trans. Image Process., vol. 28, no. 3, pp. 1482–1497, Mar. 2019, S. Li and X. Zhang, "Toward construction-based data hiding: From secrets to fingerprint images."

[6] Hu, D., Wang, L., Jiang, W., Zheng, S., and Li, B., "A novel image steganography method via deep convolutional generative adversarial networks," IEEE Access, vol. 6, pp. 38303–38314, 2018.

[7] Volume I of the 2014 International Journal of Engineering and Advanced Technology (IJEAT).

[8] Bruce Schneier. The Blowfish Encryption Algorithm Retrieved October 25, 2008, http:// www. schneier.com/ blowfish.html.

[9] Zhang, G. Fu, R. Ni, J. Liu, and X. Yang, "A generative method for steganography by cover synthesis with auxiliary semantics," Tsinghua Sci. Technol., vol. 25, no. 4, pp. 516–527, 2020.

[10]"Multi-object-recognition-based coverless image steganography," IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 7, pp. 2779–2791, Jul. 2021, Y. Luo, J. Qin, X. Xiang, and Y. Tan.

[11] W. Jiang, D. Hu, C. Yu, M. Li, and Z.-!. Zhao, "A new adversarial training-based steganography without embedding,"

[12] J. Li et al., “A generative steganography method based on WGANGP,” in Proc. Int. Conf. Artif. Intell. Secur., 2020, pp. 386–397.

[13] J. Mielikainen, In May 2006, IEEE Signal Process Lett., vol. 13, no. 5, pp. 285–287, "LSB matching revisited."

[14] "Content-adaptive steganography by minimizing statistical detectability," by V. Sedighi, R. Cogranne, and J. Fridrich In February 2016, IEEE Trans. Inf. Forensics Secur., vol. 11, no. 2, pp. 221–234.

[15] Dicing together joint distortion for adaptive steganography: W. Zhang, Z. Zhang, L. Zhang, H. Li, and N. Yu, IEEE Trans. Circuits Syst. Video Technol., vol. 27, no. 10, pp. 2274–2280, Oct. 2017.

[16] "A new payload partition strategy in color image steganography," X. Liao, Y. Yu, B. Li, Z. Li, and Z. Qin, IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 3, pp. 685–696, Mar. 2020.

[17] S. Baluja, "Images within images: concealing them," Published July 2020 in IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 7, pp. 1685–1697.

[18] Fully convolutional networks for semantic segmentation, J. Long, E. Shelhamer, and T. Darrell, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 3431–3440.

[19] M. Goljan and J. Fridrich, "On estimation ofSymp. Circuits Syst., 2008, pp. 3029–3032.

[20]The article "Steganalysis by subtractive pixel adjacency matrix" was published in June 2010 in the IEEE Transactions on Forensics and Security, volume 5, issue 2, pages 215–224. T. Pevny, P. Bas, and J. Fridrich wrote it.

[21] Rich models for steganalysis of digital images: J. Fridrich and J. Kodovsky, IEEE Trans. Inf. Forensics Secur., vol. 7, no. 3, pp. 868–882, Jun. 2012.

[22] "Learning and transferring representations for image steganalysis using convolutional neural network," Y. Qian, J. Dong, W. Wang, and T. Tan, Proc. IEEE Int. Conf. picture Process., 2016, pp. 2752–2756.

[23] "Ensemble of CNNs for steganalysis: An empirical study," by G. Xu, H.-Z. Wu, and Y. Q. Shi, was published in Proc. ACM Workshop Inf. Hiding Multimedia Security, 2016, pp. 103–107.

[24] "Robust steganography using texture synthesis," Z. Qian, H. Zhou, W. Zhang, and X. Zhang, Proc. Adv. Intell. Inf. Hiding Multimedia Signal Process, 2017, pp. 25–33.

[25] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1440–1448.

[26] "Improved training of wasserstein GANs," I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, arXiv:1704.00028, 2017.

[27] "Distributed representations of sentences and documents," by Q. Le and T. Mikolov, in Proc. Int. Conf.Mach. Learn., 2014, pp. 1188–1196.

[28] "Accelerating t-SNE using tree-based algorithms," by L. Van Der Maaten J. Mach. Learn. Res., vol. 15, no. 1, pp. 3221–3245, 2014.

[29] M. Imon, P. Goutam, and A. J. Jawahar, “Defeating steganography with multibit sterilization using pixel eccentricity,” IPSI BgD Trans. Adv. Res., 2015, Art. no. 25.

[30] J. Jing, X. Deng, M. Xu, J. Wang, and Z. Guan, “HiNet: Deep image hiding by invertible network,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 4733–4742.

[31] C. Yu, D. Hu, S. Zheng, W. Jiang, M. Li, and Z. Zhao, “An improved steganography without embedding based on attention GAN,” Peer-to-Peer Netw. Appl., vol. 14, no. 3, pp. 1446–1457, 2021.