CloakShare: End to End Secure Message Sharing Using Image Steganography

Siddharth Suresh
20BPS1042
VIT, Chennai

Kanishka Ghosh
20BPS1125
VIT, Chennai

Siddharth M
20BPS1007
VIT, Chennai

Prantik Dhara
20BPS1083
VIT, Chennai

***Abstract*--Steganography is a technique of hiding secret information within a cover medium such as an image, audio, or video, without altering the appearance or quality of the medium. In this paper, we propose a steganography method that combines encryption and image augmentation techniques to hide a secret message within an image. The proposed method involves encrypting the secret message using AES256. The cover image is augmented to prevent standard attacks and the Yb component of the image is used to hide the secret message using the Discrete Cosine Transform (DCT). Additionally, the bits of the message are randomly distributed in the image to make it difficult to detect the hidden message.**

The proposed steganography method provides a high level of security for the hidden message by using encryption, image augmentation, and DCT. The encrypted password and AES256 encryption ensure that the message is only accessible to authorized parties. The image augmentation and Yb component usage make it difficult for machine learning systems to detect the hidden message. The random distribution of bits adds an additional layer of security against detection. The proposed method has been tested on various images and shows promising results in terms of the quality of the steganographic image and the level of security provided.

I. INTRODUCTION

In today's digital age, ensuring the security of the information has become more necessary than ever. Steganography is one such technique used to conceal secret information within seemingly innocuous data. It is the art and science of hiding messages in such a way that only the sender and intended recipient know of the message's existence, while third-party observers remain oblivious to the transmission.

Steganography has been in use for ages and ages and has gone through many transformations as technology has advanced. The earliest known examples of steganography dates to ancient Greece, where they would hide messages on wax tablets covered with another layer of wax. In the medieval era, steganography evolved to include hiding messages within images, a techniquee involving a method that we are now going to explain in detail. Later, steganography was used during World War II to hide messages in music, where the length of pauses between notes corresponded to different letters to be decoded by the respective person to whom it is sent to.

With the rise of digital communication, steganography techniques have become more refined, and the applications have grown by leaps and bounds. Steganography has been used in digital watermarking, digital rights management, and data hiding. Digital watermarking is the practice of adding information that is name or logo to a brand such as images, videos, or audio recordings, to protect them from unauthorized use or mollified circulation. In digital rights management, steganography can be used to hide copyright information or authentication data in digital media to prevent unauthorized distribution. Data hiding can be used to hide encrypted data within other digital data, such as hiding a message within an image.

While steganography has several legitimate uses, it can also be used for malicious purposes that are criminal offences in many countries. Cybercriminals can use steganography to spread malware or other attacks that can destabilize an entire system and make it unfit for use. Terrorists and other criminal organizations can also use steganography to coordinate attacks and plan activities without being able to be identified for perpetrating the crime.

The use of steganography has raised concerns among security professionals who fear that this might be able to be used for disastrous reasons that any constructive reasons. Detecting steganography is challenging, and even advanced security software and firewalls may not be able to detect a hidden message in a digital file. As steganography techniques continue to advance, new methods for detection and finding must be done to escape the consequences arising out of the above-mentioned reasons.

The main aim of this research paper is to present a novel approach to secure file sharing using image steganography. In this project, we propose a desktop application that provides three-step security to ensure the confidentiality and integrity of the shared data. The approach ensures the confidentiality and integrity of the shared data while maintaining the visual quality of the image. The paper aims to contribute to the field of information security by presenting a secure and efficient approach to file sharing using image steganography.

II. LITERATURE REVIEW

Jui-Cheng Yen and Jim-In Guo a New Chaotic Key- Based Design for Image Encryption and Decryption[2000]

The authors present a new Chaotic Key-Based Design for Image Encryption and Decryption. The VLSI architecture for image encryption and decryption algorithm is proposed where bit-by-bit XOR or XNOR is used to predetermine keys for the chaotic binary sequence of the grey level of each pixel.
But One major limitation is that chaotic systems are deterministic, meaning that given the same initial conditions, they will always produce the same output. This makes them vulnerable to brute-force attacks, where an attacker can use

the same initial conditions to generate the same key and decrypt the encrypted image. Additionally, chaotic systems can also be vulnerable to other attacks, such as known-plaintext and chosen-plaintext attacks.

P. Radhadevi, P. Kalpana Secure Image Encryption Using AES[2012]

It presents a Modified AES Based Algorithm for Image Encryption. There are different methods for vector quantization (image protection technique) where the image is decomposed into vectors where encoding and decoding is done by vector by vector. Or by dividing the image into desired form into large number of shadows that guarantee the undetectable to illegal users

One limitation of vector quantization is that it can result in loss of information or image quality due to the quantization process. This can be particularly problematic for high-resolution images, as the compression process required to convert them into vectors or shadows can result in loss of important details.

Another limitation is that these methods can be vulnerable to attacks such as known-plaintext attacks or brute-force attacks. In a known-plaintext attack, an attacker has access to both the encrypted and unencrypted versions of an image and can use this information to try to decrypt the image. In a brute-force attack, an attacker tries every possible key until the correct one is found.

Seyed Hossein,Kamali, Reza Shakerian, Mohsen Rahmani A New Modified Version of Advanced Encryption Standard
(AES) Based Algorithm for Image Encryption[2010]

The authors proposed an enhanced model of AES to possess an appreciable Rangel of security and better range of image encryption. The modification process can be carried out by altering the Transformation of Shift Row. As the result shown, that the comparison has been made in between the original AES encryption algorithm and the modified algorithm which produces very good encryption results focusing towards the security against statistical attacks. One limitation is that modifying the AES algorithm in this way can make it more complex and difficult to implement, which may lead to slower encryption and decryption times. This can be a concern for applications that require fast image processing, such as video or real-time imaging. Another limitation is that modifying the AES algorithm in this way may make it less interoperable with other systems that rely on the original AES algorithm. This can be an issue if the modified algorithm is not widely adopted and supported by other software and hardware systems.

Zeghid, Medien, Mohsen Machhout, Lazhar Khriji A novel image encryption algorithm based on dynamic S boxes constructed by chaos[2016]

Key stream generators are used to enhance the output of AES for images which have decreased entropy. Image encryption is implemented by vector quantization. After encryption the image is transformed into shadows which are not understandable to intruders. One limitation is that the use of key stream generators may introduce additional complexity to the encryption process, which can increase the time and resources required for encryption and decryption. This can be a concern for applications that require fast image processing. Another limitation is that the use of vector quantization may result in a loss of image quality or fidelity, particularly if many shadows are used. This can be problematic for applications where preserving image quality is important, such as medical imaging or scientific research.

.Benrhouma, Oussama, Houcemeddine Hermassi, and Safya Belghith Security analysis and improvement of a partial encryption scheme [2013]

The paper proposes a novel pixel-based image encryption method that maintains important features of original images for privacy-preserving DNNs. A DNN model is trained with images encrypted by using the method with independent keys. One limitation is that the encryption method may introduce additional computational overhead, which can increase the time and resources required for training and inference of DNN models. This can be a concern for applications where real-time processing is required or where large datasets are being used. Another limitation is that the effectiveness of the encryption method may depend on the specific DNN model being used and the characteristics of the image data being processed. It is possible that some DNN models or image datasets may be more vulnerable to attacks or may not benefit as much from the encryption method.

Tatsuya Chuman, Hitoshi Kiya Block Scrambling Image Encryption Used in Combination with Data Augmentation for Privacy-Preserving DNNs[2021]

The paper proposes a novel learnable image encryption method for privacy-preserving deep neural networks (DNNs). The method that is laid down is carried out based on block scrambling used in tandem with data augmentation techniques. One limitation is that the encryption method may introduce additional computational overhead, which can increase the time and resources required for training and inference of DNN models. This can be a concern for applications where real-time processing is required or where large datasets are being used. Another limitation is that the effectiveness of the encryption method may depend on the specific DNN model being used and the characteristics of the image data being processed. It is possible that some DNN models or image datasets may be more vulnerable to attacks or may not benefit as much from the encryption method.

Warit Sirichotedumrong , Yuma Kinoshita,Hitoshi Kiya Pixel-Based Image Encryption Without Key Management for Privacy-Preserving Deep Neural Networks[2019]

The paper proposes a novel pixel-based image encryption method that maintains important features of original images for privacy-preserving DNNs. A DNN model is trained with images encrypted by using the method with independent keys. One limitation is that the encryption method may introduce additional computational overhead, which can increase the time and resources required for training and inference of DNN models. This can be a concern for applications where real-time processing is required or where large datasets are being used. Another limitation is that the effectiveness of the encryption method may depend on the specific DNN model being used and the characteristics of the image data being processed. It is possible that some DNN models or image datasets may be

more vulnerable to attacks or may not benefit as much from the encryption method.

Nada S. Mohammed, Areej M. Abduldaim Algebraic Decomposition Method for Zero Watermarking Technique in YCbCr Space [2022]

The paper uses the algebraic Hessemberge decomposition methods (HDM) as a transformation to extract the features of an image without using the popular transformation for building zero watermarking. It also highlights the advantages of HDM to convert image to another domain in YCbCr space. However, there are also limitations to this method. One limitation is that the effectiveness of the method may depend on the specific image data being used and the characteristics of the watermark being added. It is possible that some images or watermark types may not benefit as much from the HDM transformation or may be more susceptible to attacks. Another limitation is that the method may introduce additional computational overhead, which can increase the time and resources required for watermarking and processing of images. This can be a concern for applications where real-time processing is required or where large datasets are being used.

III. METHODOLOGY

3.1 Key Concepts Used:

Encryption: The text message is first encrypted using Advanced Encryption Standard (AES) with a key that is provided by the user. AES is a widely used encryption standard that provides strong security for the data.

DCT: Discrete Cosine Transform (DCT) is a mathematical technique that is used to convert an image from the spatial domain to the frequency domain. In this project, DCT is used to transform the image into its frequency domain.

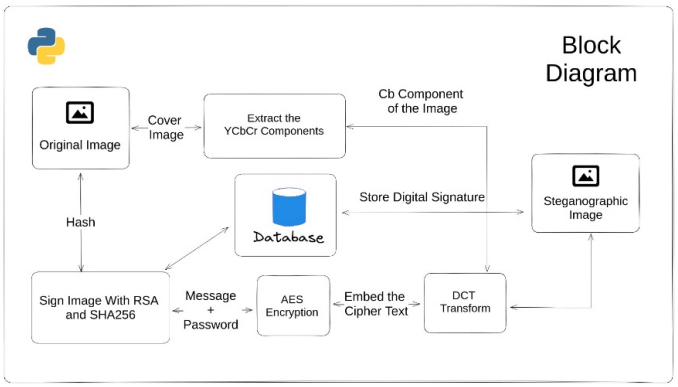
YCbCr: YCbCr is a color space used in digital image processing. It represents an image as three separate components: Y (luma), Cb (blue difference), and Cr (red difference). In this project, the encrypted text message is hidden in the Yb component of the YCbCr color space.

Steganography: The encrypted text message is embedded into the image using steganography. The Yb component is used because it is less sensitive to changes than the other components and can hide the encrypted message without significantly altering the appearance of the image.

Hashing: The steganographic image is hashed to create a unique digital fingerprint that can be used for verification purposes. The hash function used should be secure and produce a unique fingerprint for each image.

Desktop Application: A desktop application is used to provide a user-friendly interface for the project. The application should allow the user to input the text message and key and generate the steganographic image. It should also allow the user to extract the hidden message from the image using the key. The application should be designed to be easy to use and secure.

3.2 Block Diagram:



3.3 Modules:

Compression: Here we will compress the original message into a smaller size to make it easier to embed into the cover image.

Encryption: Here we will encrypt the compressed message using AES256 and a key derived from the SHA256 hash of the user-defined password.

Image Augmentation: Here we will modify the cover image to prevent detection by machine learning systems using black-box adversarial attacks.

Discrete Cosine Transform: Here we will use DCT to embed the encrypted message into the blue chroma component of the YCbCr components.

YCbCr Decomposition Here we will decompose the modified image into its YCbCr components to allow for efficient embedding of the encrypted message.

Colour Space Composition: Here we will combine the modified blue chroma component with the original luma and red-difference chroma components to create a new image in the YCbCr colour space.

Steganographic Image: Here we will transform the new YCbCr image back into the original colour space of the cover image to create the steganographic image.

3.4 Implementation:

The proposed methodology for secure file sharing using image steganography was implemented in a desktop application using Tauri, a toolkit for building native desktop applications. Rust was used for the backend and React and Vite were used for the frontend of the application. The desktop application provides options for secure file sharing using image steganography involves several steps that ensure the confidentiality, integrity, and authenticity of the shared data. The steps involved in the implementation of the methodology in the desktop application are as follows:

Step 1: Encryption

The first step is to encrypt the text message using Advanced Encryption Standard (AES) encryption. AES is a widely used encryption algorithm that ensures the confidentiality of the data by transforming the plaintext into ciphertext. In the desktop application, the user provides a key that is used to encrypt the message. The key can be any random string of

characters provided by the user, making it more difficult to decipher the message without the key.

Step 2: Steganography

The image used for steganography is usually a benign image that does not raise suspicion. The image should also be of high resolution to ensure that the quality of the image is not compromised after embedding the text message. In the desktop application, the image is converted from the RGB color space to the YCbCr color space. The Y component represents the brightness of the image, while the Cb and Cr components represent the chrominance. In the proposed steganography technique, the blue chroma component (Cb) is selected for hiding the secret message. This component is chosen because it is the least detectable by the human eye, making it an ideal location for hiding secret data. The below images demonstrate that.

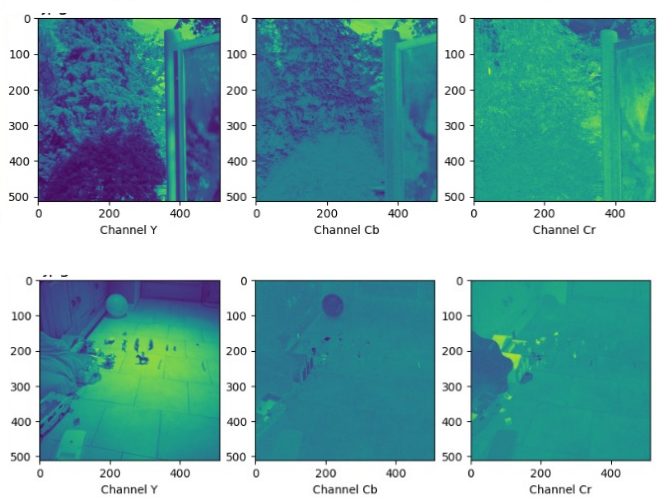


Fig: Y, Cb, Cr components of an image

Once the blue chroma component has been selected, it is ready for the next step in the process. After the message has been compressed, encrypted, and the cover image has been modified, the next step is to embed the message into the image. To do this, the ciphertext is divided into small blocks and converted into its frequency domain representation using the Discrete Cosine Transform (DCT). The blue chroma component is then divided into small blocks, and the DCT is applied to each block. Next, the transformed data from the ciphertext is inserted into the high-frequency components of the DCT coefficients of the blue chroma component. This is done in a randomized manner to prevent the hidden message from being easily detected. Finally, the modified blue chroma component is converted back into the YCbCr color space and combined with the other color components to create the steganographic image. Finally, the steganographic image is converted back to the original RGB color space to obtain the final output.

Step 3: Digital Signature

To ensure the authenticity and integrity of the shared data, both the original image and the steganographic image are hashed and digitally signed using RSA and SHA256. RSA is an asymmetric encryption algorithm used for digital signatures and encryption. SHA256 is a cryptographic hash function that generates a 256-bit hash value that uniquely identifies the input data. The hashed images are then stored in a database as a key-value pair, which can be compared with the image received by the receiver to verify the authenticity of the image.

Step 4: Decryption

To extract and decrypt the text message from the steganographic image, the reverse process is used. The Cb component is extracted from the YCbCr color space, and the frequency components are retrieved using DCT. The retrieved frequency components are then used to extract the encrypted text message. The encrypted text message is then decrypted using the same key that was used to encrypt it, using AES encryption.

In conclusion, the proposed methodology for secure file sharing using image steganography involves several steps that ensure the confidentiality, integrity, and authenticity of the shared data. The desktop application implements the methodology by encrypting the text message, embedding it in an image using steganography, and digitally signing and hashing the images for verification. The application provides a user-friendly interface for the entire process, making it easy for non-technical users to securely share files using image steganography.

IV. RESULTS

Below is the screenshot of the encryption page in the desktop application where a user can upload a cover image along with the secret message and the password.

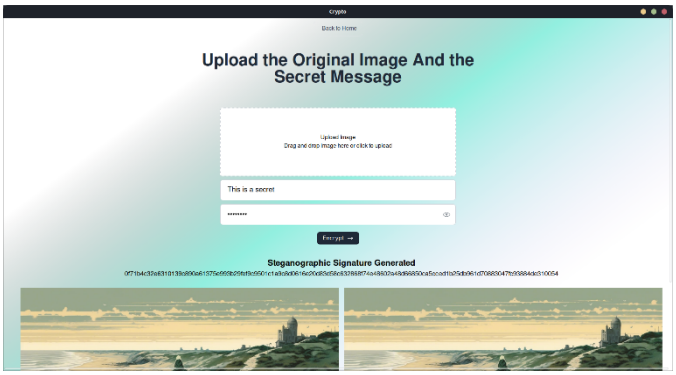


Fig: Encryption in Desktop Application

Fig: Decryption in Desktop Application

Shown above is the screenshot of the decryption page in the desktop application where u can upload a steganographic image along with the password and the hash digest generated to verify the integrity of the steganographic image and extract the hidden text from it.

The below graph represents the relationship between the size of text data stored vs the accuracy of the proposed model. As the size of data increases, the accuracy decreases minutely.

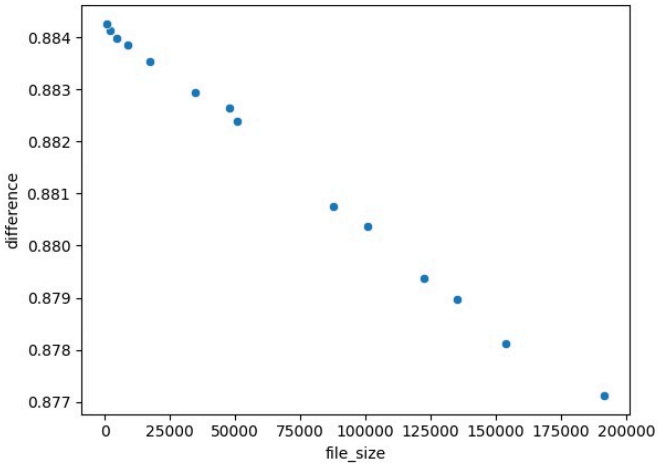


Fig: file size embedded into image vs similarity between original and steganographic image

Also, the maximum limit of text data that can be stored in an image depends on the actual size of the image along with the percentage of blue chroma component (Cb) present in the image. Based on experimentation, an image file with 300kB of size can store maximum of 145kB of text data. So, on average an image file can store text of size 48.3% of its own size.

We are using the dataset provided to the Research Prediction Competition, "ALASKA2 Image Steganalysis", to obtain the images encoded by the latest steganographic algorithms namely, JUNIWARD, UERD and JiMPOD and compared the results with the output images generated by our proposed system.

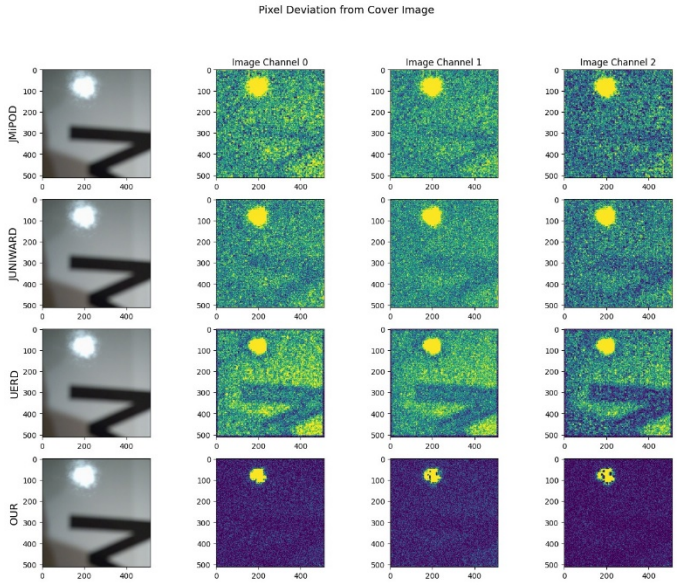


Fig: Deciphering the accuracy and reliability of the steganographic method

MSE (Mean Square Error) and PSNR (Peak Signal to Noise Ratio) are two commonly used metrics for assessing the quality of an image and determining the degree of difference between the original cover image and the modified steganographic image. MSE measures the average squared difference between each pixel of the cover image and its corresponding pixel in the steganographic image.

```
···   Proposed System
      MSE: 0.03862762451171875
      PSNR: (62.26182359768749+0j)
      JMiPOD
      MSE: 0.3160133361816406
      PSNR: (53.133749500600025+0j)
      JUNIWARD
      MSE: 0.2774658203125
      PSNR: (53.698708687746944+0j)
      UERD
      MSE: 0.6424980163574219
      PSNR: (50.05208569699373+0j)
```

The lower MSE value indicates that the steganographic image is closer to the cover image in terms of pixel values and the higher PSNR value indicates that the steganographic image has less noise and is closer to the original cover image.
So, we can conclude that the steganographic image produced by this system has a prominent level of quality and similarity to the original cover image.

V. FUTURE WORK

Security Analysis: While this technique provides robust and secure message hiding, further analysis is necessary to evaluate its resistance to various types of attacks, including brute-force attacks, statistical attacks, and machine learning-based attacks.

Optimization: The current technique uses image compression to reduce the file size of the message before embedding it in the image. However, further optimization can be done to reduce the size of the steganographic image without affecting the quality of the hidden message.

Capacity Improvement: The capacity of the current technique is limited by the size of the blue chroma component. Future work can explore ways to increase the capacity of the technique by using other color components or applying different embedding techniques.

VI. CONCLUSION

In this paper, we have presented a novel steganography technique that provides a secure and reliable way of hiding confidential information within images. The technique involves compressing the original message, encrypting it with SHA256 and AES256, and then embedding it within the blue chroma component of the cover image using the discrete cosine transform. To prevent detection, the cover image is augmented using black-box adversarial attacks, and the YCbCr components are combined with the modified blue chroma component to create the steganographic image.

Overall technique offers several advantages over traditional steganography methods. It is robust and can withstand various forms of image manipulation and compression, making it a reliable tool for secure communication. Additionally, the use of image augmentation and YCbCr decomposition ensures that the steganographic image is not easily detectable by machine learning systems or standard attacks.

REFERENCES

[1] Jui-Cheng Yen and Jim-In Guo, “A New Chaotic Key-Based Design for Image Encryption and Decryption”,2000.Schneier B, “Applied Cryptography”, John Wiley&Sons Publication, New York, 1994.

[2]P. Radhadevi, P. Kalpana , “Secure Image Encryption Using Aes”,2012 .

[3]Manoj. B, Manjula N Harihar, “Image Encryption and Decryption using AES” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-5, June 2012

[4]Zeghid, Medien, Mohsen Machhout, Lazhar Khriji, Adel Baganne, and Rached Tourki. "A Modified AES Based Algorithm for Image Encryption." International Journal of Computer Science & Engineering 1, no. 1 (2007).

[5]Benrhouma, Oussama, Houcemeddine Hermassi, and Safya Belghith. "Security analysis and improvement of a partial encryption scheme." Multimedia Tools and Applications (2013): 1-18

T. Chuman and H. Kiya, "Block Scrambling Image Encryption Used in Combination with Data Augmentation for Privacy-Preserving DNNs," 2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Penghu, Taiwan, 2021, pp. 1-2, doi: 10.1109/ICCE- TW52618.2021.9602969.

W. Sirichotedumrong, Y. Kinoshita and H. Kiya, "Pixel-Based Image Encryption Without Key Management for Privacy-Preserving Deep Neural Networks," in IEEE Access, vol. 7, pp. 177844-17782019, doi: 10.1109/ACCESS.2019.2959017.

Nada S. Mohammed* , Areej M. Abduldaim, Algebraic Decomposition Method for Zero Watermarking Technique in YCbCr Space.

[9] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," IEEE Transactions on Image Processing, vol. 13, no. 8, pp. 1147-1156, Aug. 2004.

[10] M. S. C. Cruz, S. S. Rodrigues, and P. M. Q. Aguiar, "Steganography based on image segmentation and improved pixel-value differencing," IEEE Transactions on Information Forensics and Security, vol. 14, no. 6, pp. 1518-1528, June 2019.

[11] T. Morkel, J. H. P. Eloff, and M. S. Olivier, "A survey of information hiding techniques," Journal of Information Hiding and Multimedia Signal Processing, vol. 1, no. 3, pp. 142-177, July 2010.

[12] S. S. Naik and V. Bhatnagar, "Steganography techniques: A review," International Journal of Computer Science and Information Security, vol. 7, no. 1, pp. 183-191, Jan. 2010.

[13]. Fridrich, "Steganography in digital images," Proceedings of the IEEE, vol. 88, no. 6, pp. 1062-1078, June 2000.

[14] C. Chiang and W. Chang, "Image steganography using AES encryption and LSB substitution methods," International Journal of Innovative Computing, Information and Control, vol. 8, no. 9, pp. 6501-6512, Sept. 2012.

[15] S. M. Al-Qershi and A. Zeki, "A survey of image steganography techniques," Journal of Network and Computer Applications, vol. 35, no. 5, pp. 1666-1691, Sept. 2012.

[16].J. Fridrich and M. Goljan, "Practical steganalysis of digital images—state of the art," Proceedings of the SPIE, vol. 5681, pp. 523-534, Mar. 2005.

[17] Singh, S., & Kapoor, K. (2018). Steganography Techniques: A Review. International Journal of Computer Science and Information Technologies, 9(1), 234-238.

[18] Wu, J., Peng, S., & Huang, L. (2018). A New Steganography Algorithm Based on YCbCr Color Space and DCT. Journal of Physics: Conference Series, 1006(1), 012118.

[19] Raval, H., & Doshi, P. (2018). Image Steganography: A Review of Techniques and Evaluations. International Journal of Advanced Research in Computer