IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

E-Security Vault Application using Image Steganography and Cryptography

Tejaswi Chaudhari, Manali Chaudhari, Shruti Gavali, Sumedh Pundkar
Computer Science and Technology, Usha Mittal Institute of Technology,
SNDT Women's University,
Juhu-Tara Road, Sir Vitthaldas Vidyavihar,
Santacruz(W), Mumbai, Maharashtra 400049

Abstract— **In a world where technology is becoming increasingly pervasive, it is more important than ever to safeguard our personal information. With the rise of mobile devices as the primary gateway to our digital lives, the risk of theft and compromised data has become a real concern. The safekeeping methods related to the storage of images, documents have evolved essentially in the last few decades, but the sensitive information in textual format (Passwords, Pass Codes, PIN, Address, Important messages) are still vulnerable. It is necessary to use an application to keep this textual information covered by a secure vault and out of sight. The security vault will be masked as an image gallery to reduce the likelihood of being recognized. The next stage in text data security will be the development of this application.**

Keywords— **Steganography, SteganoGAN, Cryptography, AES-GCM, GAN, Application, Security**

I. INTRODUCTION

In recent years, the widespread use of digital media has greatly impacted the way we send, receive, and store information. The ease and convenience that comes with this technology has transformed the way we work, learn, and interact with others. However, with these advancements also come new challenges, such as the growing threat of cyberattacks and intellectual property theft. As a result, there has been a greater focus on digital security and protecting sensitive information in our increasingly connected world.

For secure data storage and secure network communication, data concealing is used in many technical domains. One technique to hide data and protect sensitive information is steganography. Using Steganography for encryption has the benefit of making it impossible for anybody other than the media's owner to decrypt it without performing certain actions. In cryptography, we concentrate on the portion of encryption that renders the data impenetrable and challenging to decipher. On the other hand, steganography used in combination with encryption completely conceals any evidence of encryption. Images are the most often used file format for this approach among the many others that may be employed.

Several strategies are utilized, some simple and others complicated, to conceal the hidden message or information. In our project, to hide sensitive information from others, we are using image steganography. One key advantage of steganography is that it provides a layer of security through obscurity. Because the existence of hidden messages is not immediately apparent, unauthorized parties may not even be aware that there is anything to uncover. This makes it a powerful tool for covert communication, particularly in situations where conventional encryption may be suspect. To make it more secure, an additional layer of cryptography will be added before applying Steganography, which will protect our application from hazards like phone theft and system hacking. In the case of retrieval of the desired information, the owner can verify their identity and decrypt the information to access it.

Cryptography with Steganography acts as a two layer of security to the sensitive text data to be stored in the devices and hence providing greater security to the data.

The illusion of having digital media rather than directly stored text leads the user to believe that the sensitive information is just an ordinary image and is secure underneath.

II. LITERATURE REVIEW

The already developed models on various dataset have shown the improvements and drawbacks of each and every dataset in comparison to all the methods under StegoGAN.

The encoder-decoder design is primarily influenced

TABLE1

COMPARISON OF STEGANOGAN ON DIFFERENT DATASETS

Dataset	Image Size	Purpose	Accuracy
ImageNet	Arbitraryi	Computer Vision	1.2 bpp (bits per pixel)
MNIST	28*28*1	Image processing and computer vision	3 bpp
CelebA	178×218	Face attribute recognition, face detection, landmark localization	1.5 bpp
BossBase	512*512*1	Steganography & Steganalysis	0.6 bpp

by image steganography employing CNN models. Two inputs, the cover picture and the secret image, are fed into the encoder to make the stego image, which is then fed into the decoder to output the embedded secret image. The essential premise remains the same, except that different techniques have been experimented with on various structures. Large-scale The CelebFaces Attributes dataset, often known as the CelebA dataset, contains over 200K photos that are utilized for face recognition, detection, localization, and other face-related tasks. The dataset includes photographs from diverse sources, locations, background and postures and is best fit for steganography also. The Common Objects in Context (COCO) dataset was designed primarily for object recognition, segmentation, and picture captioning. This is another massive dataset containing photos from 80 different item categories. Another dataset that is utilised for computer vision and image processing is the Modified National Institute of Standards and Technology database (MNIST). The MNIST handwritten dataset includes a training and testing collection of photographs of handwritten numbers 0 through 9. The images in this collection are normalised, black and white, and have a size of 28 by 28 pixels. The training set has 60,000 photos, whereas the testing set contains 10,000 images.

ImageNet is a dataset that contains images from the WordNet hierarchy, with each node comprising more than 500 to 1000 images. ImageNet does not own the image and simply provides links or thumbnails to the original. The dataset is made up of photos of various sizes. The amount of photos, classes they belong to, backdrop, and image size may all be chosen from a large selection based on the requirements.

Table 1 demonstrates the given datasets on the image size, accuracy metric and the primary purpose in which the datasets are used. As the results conclude MNIST and CelebA are the provenly efficient pair of choice in the respective order to gain desired results. The work with datasets and GAN is collectively highlighted further in the paper.

III. AES GCM

An authenticated encryption technique that guarantees data integrity and secrecy is AES-GCM (Advanced Encryption Standard - Galois/Counter Mode). It is a block cipher mode of operation that provides data secrecy, data integrity, and authenticity in a single operation by combining the AES encryption method with the GCM authentication mode.

AES is a symmetric encryption technique that works with data blocks of a set length. It encrypts and decrypts the data using a secret key that is shared by the sender and the recipient. AES allows key sizes of 128, 192, or 256 bits and has a fixed block size of 128 bits[3].

The Galois field multiplication and Counter Mode (CTR) encryption are used in the authenticated encryption mode known as GCM to give both secrecy and authentication. The GCM mode creates a cryptographic tag for each data block by using a special initialization vector (IV) and a secret key. The tag is used to check the data's consistency and make sure that it wasn't altered while being transmitted. The AES GCM algorithm is used paired with steganography and firebase authentication to provide a three-layer security to our image vault.

IV. STEGOGAN

Steganography seeks to conceal a text behind an image that a sender provides in order to conceal a hidden message in transit. Steganography is implemented in our situation to hide secret data behind graphics on personal devices by embedding it in an application. The image in the device might be readily received by a second person, but with steganography, it is nearly impossible to tell that it is not a regular image and contains concealed text. (Fig 1.)

In contrast to traditional methodologies, the use of CNN in steganography has recently shown effectiveness in research. SteganoGAN is a technique that employs CNN to solve the vanishing gradient problem while also ensuring that the information concealed is lossless and secure.

StegoGAN is made up of two neural networks: a generator network and a discriminator network. The generator network accepts an input picture and a secret image and outputs an image that resembles the original input image but also has the secret image concealed within it. The discriminator network is taught to distinguish between the original input images and the generated output images.

The generator model in StegoGAN is a deep neural network that takes in two images as input: an input image and a secret image. The goal of the generator is to produce an output image that looks like the input image but also contains the secret image hidden within it.

The generator typically consists of several layers of convolutional and deconvolutional operations, as well as batch normalization and activation functions such as ReLU. The convolutional layers are used to extract features from the input image, while the deconvolutional layers are used to generate the output image.

The generator model in StegoGAN is typically trained using a loss function that encourages the output image to be as similar as possible to the input image while also ensuring that the secret image is effectively hidden within it. One common loss function used in StegoGAN is a combination of the mean squared error (MSE) loss and the adversarial loss.

The discriminator model in StegoGAN is a deep neural network that is trained to differentiate between the original input images and the generator's output images. The goal of the discriminator is to correctly classify each image as real (i.e., the original input image) or fake (i.e., the generator's output image). The discriminator typically consists of several layers of convolutional operations, followed by fully connected layers, and a final output layer that produces a binary classification of real or fake. The convolutional layers are used to extract features from the image, while the fully connected layers and the final output layer are used to make the classification. The discriminator model in StegoGAN is typically trained using a loss function that encourages it to correctly classify each image. One common loss function used in StegoGAN is the binary cross-entropy loss, which measures the difference between the discriminator's predicted output and the true label (i.e., real or fake).

During training, the discriminator is shown both the original input images and the generator's output images, and it learns to correctly classify each image. The generator is trained to produce output images that fool the discriminator into thinking they are real, which encourages it to effectively hide the secret information within the image.

Steganalysis is the process of detecting the presence of hidden information within an image that has been subjected to steganography. In the context of StegoGAN, steganalysis can be thought of as the process of detecting whether a given image contains a hidden secret image that was embedded using the StegoGAN model.

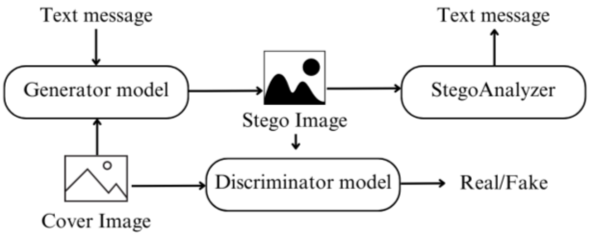

Fig 1. Workings of Steganography

Steganalysis in StegoGAN can be performed using two main functions:

A. Feature extraction

The first step in steganalysis is to extract features from the image that are indicative of the presence of hidden information. These features may include statistical properties of the image, such as the distribution of pixel values or the frequency of certain patterns. In StegoGAN, feature extraction can be performed using various techniques, such as convolutional neural networks (CNNs) or hand-crafted feature extractors.

B. Classification

Once the features have been extracted, the second step is to classify the image as either containing hidden information or not. This can be done using various machine learning algorithms, such as support vector machines (SVMs) or random forests, which are trained on a dataset of known images that either do or do not contain hidden information.

In our initial efforts to train generator and discriminator models we started with MNIST dataset as the low image size and the simplicity of data provides the opportunity to learn which network works better. Although, the observations were a bit contrary. Though the networks yield good results for MNIST, due to the drastic difference in complexity and image quality, the same networks fail to attain good results with CelebA as we proceed. As the image quality

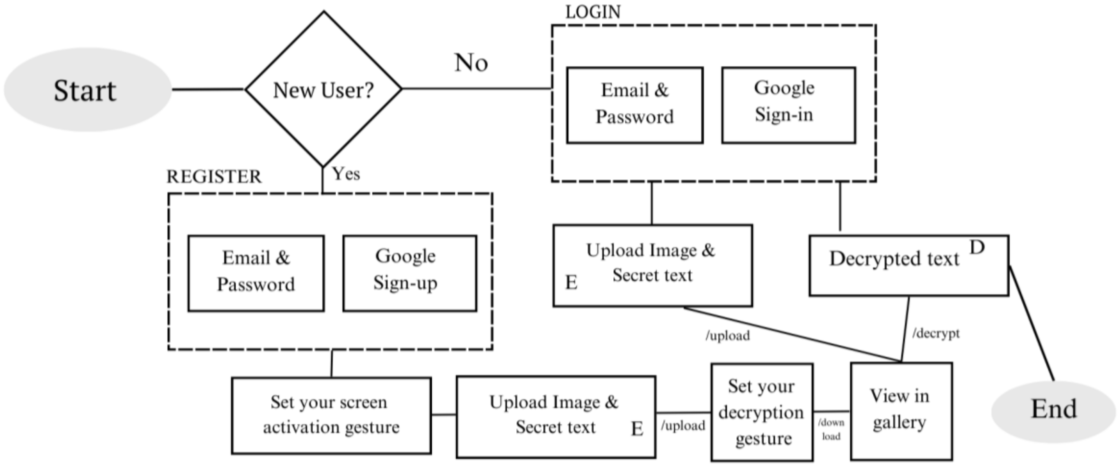

Fig 2. Architecture of the E-Security Vault

increases the time required to train the models increases exponentially giving below-average generated images for even 250 epochs. The progressive improvements have been observed but the learning rate is low, and the efforts were not time efficient for the project.

[2] A tool was created to serve the same purpose by MIT Data to AI Lab. The steganography network, based on a GAN architecture, embeds the secret message into the cover image while minimizing the perceptual distortion. The steganalysis network, also based on a GAN architecture, is trained to detect the presence of secret messages in images. The authors evaluate the performance of SteganoGAN on several benchmark datasets and compare it to existing steganography methods. The results show that SteganoGAN achieves higher capacity than existing methods while maintaining low detectability. The authors also demonstrate the robustness of SteganoGAN against various steganalysis attacks. In [2], Coco and Div2K datasets are used to train and evaluate our model. Coco performed better due to the diversity of surfaces and objects in the dataset image than Div2K, which offers less cluttered images.

The Steganogan distribution is available with pip for the use of the public. But the acceptable version of a few dependencies like torch is ==1.0.0 which is no more distributed by pip for use. To overcome this problem, an environment of conda has been created with a downgraded version of python=3.7.0 and the pytorch has been installed with the help of conda itself. The steganogan works well within the environment.

V. ARCHITECTURE

An API has been built in flask to combine the steganogan with AES GCM for the operational use under the conda environment. The endpoints of the API have been shown in Table 2.

TABLE 2
API ENDPOINTS

Endpoint	**Function**
/upload	To upload the cover image and secret text for encryption
/download	To download the encrypted image
/decrypt	To decrypt the secret text from encrypted image

Here the encryption takes place after we upload the image and text and the same function calls the download endpoint automatically to send the download link for the encrypted image as a response.

Flutter is used to build a cross-platform application with firebase authentication, and a good user interface to carry out the operations mentioned in above steps. The user enters the application (User Layer) and is faced with the authentication screen, the user will register or Log in, depending upon whether he is a new user or existing one. Authentication can be done with the Google Sign-in option as well. The data will be stored in the firebase (Database Layer) and an unique ID will be created for each user. The user will

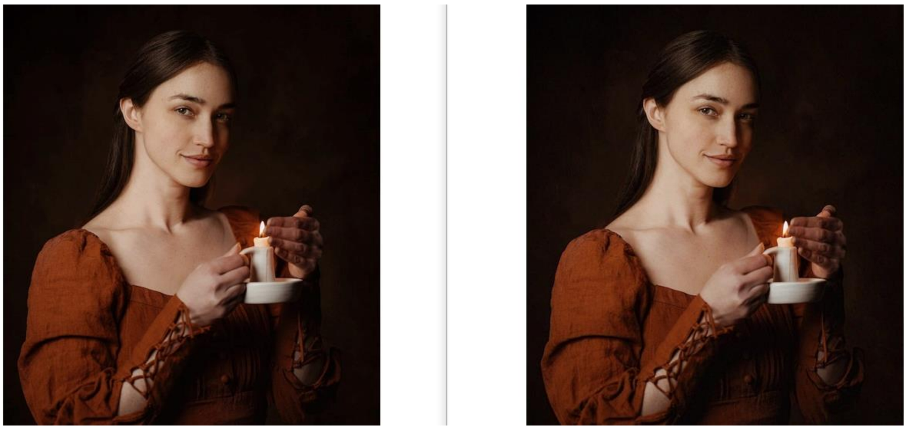

Fig 3. Comparison of the quality of input and output image

be prompted to set the activate screen gesture and decrypt gesture, and the information will be computed in the Logical Layer. As soon as the user uploads an image and the text message, the text message will be encrypted using AES GCM and then, the ciphertext and the image will be presented as inputs for GAN based steganography. A cover image will be generated and stored in the database. The text can be decrypted with the decryption gesture, and the user will receive his data back.

VI. CONCLUSION AND FUTURE WORK

The use of steganography along with cryptography has been a novel approach. The three-layer security results in an application which saves your textual data with secrecy and leaves almost no scope to vulnerability. The first layer of firebase ensures that the data is saved in a secured environment and provides the option of Google sign-in. The second layer of cryptography converts your textual data into a cipher which cannot be decrypted by anyone who is not aware of your authentication tag and secret key. The last layer concedes any operation performed on the data wisely and eliminates the traces of encryption from the root by hiding everything behind your cover image. No other than the owner can know nor decrypt the data. The existing tool used for steganography could be trained on CelebA dataset using transfer learning to yield finest resulTs.

The steganogan network can be optimized for further training with different dimensions of images. Although time is an essential factor of consideration while training, measures to overcome the problem of time efficiency can be taken by training the models on the virtual machines of higher capacity. The application can be converted into a progressive web app to serve the purpose of an equally efficient web version. The response time of the API calls can be minimized by hosting the API in a more stable VPS hosting services to help with the user experience and overcome the issues of network stability.

REFERENCES

[1] N. Subramanian, O. Elharrouss, S. Al-Maadeed and A. Bouridane, ”Image Steganography: A Review of the Recent Advances,” in IEEE Access, vol. 9, pp. 23409-23423, 2021, doi: 10.1109/ACCESS.2021.3053998.

[2] Zhang, Kevin Alex and Cuesta-Infante, Alfredo and Veeramachaneni, Kalyan. SteganoGAN: High Capacity Image Steganography with GANs. MIT EECS, January 2019.

[3] Nabihah Ahmad et al 2018 J. Phys.: Conf. Ser. 1019 012008

[4] Yang, Jianhua & Liu, Kai & Kang, Xiangui & Wong, Edward & Shi, Yun-Qing. (2018). Spatial Image Steganography Based on Generative Adversarial Network.

[5] Yang, Tianyu & Wu, Hanzhou & Yi, Biao & Feng, Guorui & Zhang, Xinpeng. (2022). Semantic-Preserving Linguistic Steganography by Pivot Translation and Semantic-Aware Bins Coding.

[6] S. Baluja, ‘Hiding Images in Plain Sight: Deep Steganography’, in Advances in Neural Information Processing Systems, 2017, vol. 30.

[7] Ahmed, Sarkar & Ahmed, Aram & Ahmed, O.H. & Kadir, Wrya. (2017). Combining steganography and cryptography on android platform to achieve a high-level security. Journal of Engineering and Applied Sciences. 12. 4448-4452.10.3923/jeasci.2017.4448.4452.

[8] H. Naito and Q. Zhao, "A New Steganography Method Based on Generative Adversarial Networks," 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), 2019, pp. 1-6, doi: 10.1109/ICAwST.2019.8923579.

[9] Xie, Z., Wang, S., & Shi, Y. Q. (2021). Progressive Steganography via Deep Generative Networks. IEEE Transactions on Circuits and Systems for Video Technology, 31(1), 55-65. doi: 10.1109/TCSVT.2020.2980433