SteGSIGN: A Digital Image Authentication Model using Steganography and AES Encryption

Ajmal Basheer
UG Student
Department of Computer Science & Engineering
Amal Jyothi College of Engineering Institute Kottayam, Kerala

Joe N Sabu
UG Student
Department of Computer Science & Engineering
Amal Jyothi College of Engineering Institute Kottayam, Kerala

Dona Reetha Dominic
UG Student
Department of Computer Science & Engineering
Amal Jyothi College of Engineering Institute Kottayam, Kerala

Jomy Jose
UG Student
Department of Computer Science & Engineering
Amal Jyothi College of Engineering Institute Kottayam, Kerala

Syam Gopi
Assistant Professor
Department of Computer Science & Engineering
Amal Jyothi College of Engineering Institute Kottayam, Kerala

Abstract

Nowadays people use social networking websites like WhatsApp, Facebook and Instagram to share their life moments and events via images and videos. These popular Social apps also present many risks. At present, these social networking websites are not bound by stringent privacy and copyright laws. Images once shared on such networks can be easily downloaded by miscreants. Recently there have been several reports of private pictures of people being shared and spread in such sites. Currently tracking such photos is a hideous task. Validation of the ownership of these images is practically impossible due to high degree of replication. The authorities would have to check the server logs to find out the culprits in such cases. This paper proposes a digital image encryption system based on Steganography (using LSB 1 bit technique) combined with AES encryption. Using this technique, secret data related to the owner of the image and other users who handle the image gets embedded within the image without affecting the visual quality of the image. These details can only be retrieved by authorized party, by decoding the image. Experimental details prove that the visual quality of the images is not affected even after repeatedly encoding the image.

Keywords: Steganography, LSB, Authentication, AES Algorithm, Security

I. INTRODUCTION

Rapid growth in social networking sites has resulted in public image sharing to be one of the most popular communication medium. Almost all social networking sites provide facility for sharing and uploading images. Authentication of these images is impossible due to high degree of replication. This causes many data thefts such as illegal editing and modification of image by cyber miscreants.

Digital Signature is a technique that binds entity/person to digital data. This technique cannot be used for public image sharing over internet as there is no individual verification between sender and receiver.

This paper presents a new authentication technique for digital data, using Steganography. It is a method of concealing data within another data, as by embedding text in a carrier medium. Carrier medium could be any files such as videos, images etc. In our proposed method we have used LSB 1-bit steganography mechanism. Images of all formats can be used as carrier files that are converted into PNG format.

In this Method whenever an image is uploaded, the details of the uploader is digitally embedded within the image using Steganography. The embedded data is encrypted using AES encryption to prevent unauthorized users from accessing the embedded data.

II. LITERATURE SURVEY

In the field of Plain Text Encryption Using AES [1], Madhumita Panda and Atul Nag [1] have examined the performance of the three basic Symmetric Key algorithms -AES, Blowfish and Salsa20. The evaluations were based on execution time, memory required for the implementation and throughput between two operating systems. Based on the obtained simulation results, the conclusions were made that AES and Salsa20 are much preferred over the Blowfish, for basic plain text data encryption.

Congfu Xu, Yafang Chen and Kevin Chiew, proposed [2] a new model which is based on the Base64 encoding system of image files and the n-gram technique which is basically used for feature extraction. They tried to extract features of an image with n-gram technique, which was implemented by transforming the normal images into Base64 presentation. Using these features they trained an SVM (support vector machine) which represented efficiency and effectiveness for detecting the spam images from legitimate one. Experimental results show that this approach, in comparison with the existing methods for feature extraction, achieve very high performance for the classification of image spams, based on some measures like precision, accuracy, and recall. Moreover, the approach takes less running time for classifying the image spam when compared with other methods.

The paper by Nidhi Grover and A.K. Mohapatra [3] proposes a digital image authentication system which uses an adaptive steganography (Edge detection with variable LSB as the embedding technique). In the proposed methodology, the information related to user's login credentials will be embedded in the original image using steganography by not affecting the overall visual quality of the original image. The information hidden in that image can be decoded only by the authorized party. This will validate the ownership and source of the uploaded image.

Min Wu and Bede Liu, June, 2003, proposed [4] a new model for embedding data into binary images that may include scanned text, figures or signatures. This model manipulates the 'flippable' pixels in order to enforce specific block based relationship so that to embed a significant amount of information without causing noticeable flows. They have performed a Shuffling before the embedding process to make even the embedding capacity from block to block. The hidden information can be then extracted without the use of the original image.

III. PROPOSED METHOD

In this system, the image being uploaded is the Cover Image. The data being embedded is the User ID of the uploading user and the timestamp at which the image is being uploaded. The data is encrypted using standard AES Encryption algorithm. The resultant encrypted string is converted to Base64 and embedded within the image using LSB 1-bit technique to generate the Stego Image which is then uploaded.

While uploading the image the image is first decoded to check the presence of any previously embedded data. If found, then the original owner of the image could be identified from the embedded data and alerted about the illegitimate usage of his image. This feature can be utilized by social networking platforms to alert users about possible fake accounts.

Messenger applications like WhatsApp, Hike, Telegram, etc provide users the facility to forward images. There have been several instances where such platforms were misused to share private and controversial images of individuals. Finding out the culprits was always a complicated task. The authorities need to analyze the user logs from the Servers to find out the origin of the image in such cases. The proposed method could be implemented in such platforms where the information about each user who forwards the image can be embedded within the image. So just by decoding a single image, the authorities can identify the miscreant.

A. Encryption:

Embedding of the user information during the upload is performed in the encryption section. When a user gives upload instruction, the following steps are executed.

1) The image is passed as an argument to the encryption module. The passed image is first decoded for previously embedded information.
2) If the image contains no previously stored information i.e., the image is being uploaded for the first time, then the following steps are executed.
    1) The user id of the user, user name and the current timestamp is stored into a string variable.
    2) This string is then encrypted using AES encryption.
    3) This encrypted string is then again encrypted using the base64 encryption technique.
    4) Then this string is encoded into the selected image by using the LSB 1-bit encryption technique.
    5) After the execution of the encoding section, the image is stored or saved into the server directory with the filename of the image as the current time stamp so that to make it much easier to identify the uploaded image.
3) Now if the image contains previously stored data, ie the image is being uploaded not for the first time, then the following steps are carried out.
    1) The user name of the logged in user and the current time stamp is appended to the previously stored information and stored into a string variable.
    2) Now, from the previously stored information, the user id of the user who has uploaded the image for the first time is taken and stored into another variable.
    3) The user id of the currently logged in use and the value of the variable is compared.
    4) If they differ then, the original user is provided with a notification that another user has misused his/her image.
    5) The string with previously stored data, username and current timestamp is then encrypted using AES encryption.
    6) This encrypted string is then again encrypted using the Base64 encryption technique.
    7) Then this string is encoded into the selected image by using the LSB 1-bit encryption technique.

8) After the execution of the encoding section, the image is stored or saved into the server directory with the filename of the image as the current time stamp so that to make it much easier to identify the uploaded image.

B. Decryption

Decoding of the embedded signature from the image is performed in the decryption section. Decode module runs as an independent module and only by the authorized personals.

1) The image is passed as an argument to the decryption module. The passed image is decoded for previously embedded information.
2) The LSB 1-bit decoding is performed over the past image to get the encrypted string, which has been formerly embedded in the image.
3) Then Base64 decryption is executed in the obtained string.
4) Now AES decryption is performed over this string to obtain the original information about the user/users who has used this image
5) This string is then parsed to obtain the details like, user information, image forward path, original user and subsequent miscreants.

IV. RESULTS

Any regular picture, being uploaded by the user can be used as a Cover Image for embedding the data. Here let a user "abc@xyz.com" upload an image, which is shown in Figure 1.

Fig. 1: Image being Uploaded.

The signature to be embedded is created by combining the User Id and the timestamp at which the image is being uploaded. It is then encrypted using AES encryption and the output string is Base64 encoded.

```
Plain  String          : 12-abc@xyz.com:2016:05:02:22:13:54
AES Encrypted String :
□□□/□□#□9й□□□Mнwg□F□p$、p□8□□□,4SI□ч□@□巽□
Base64 Encoded String : rpbEL7SgI9A506WQ2fuhTdSpd2eDRoRwJMu0cL84AuupCyw0U0niz5cOQK/mm6nx
```

Fig. 2: Encrpyted Signature

Fig. 3: Stego Image

This Base64 encoded String is embedded within the image using LSB 1-bit Steganography. The Stego image generated is visually identical to the image being uploaded.

The embedded data can be retrieved from the Stego Image using the Decoder module, which is shown in Figure 4. The key used during the Encryption algorithm is required to decode the Signature.

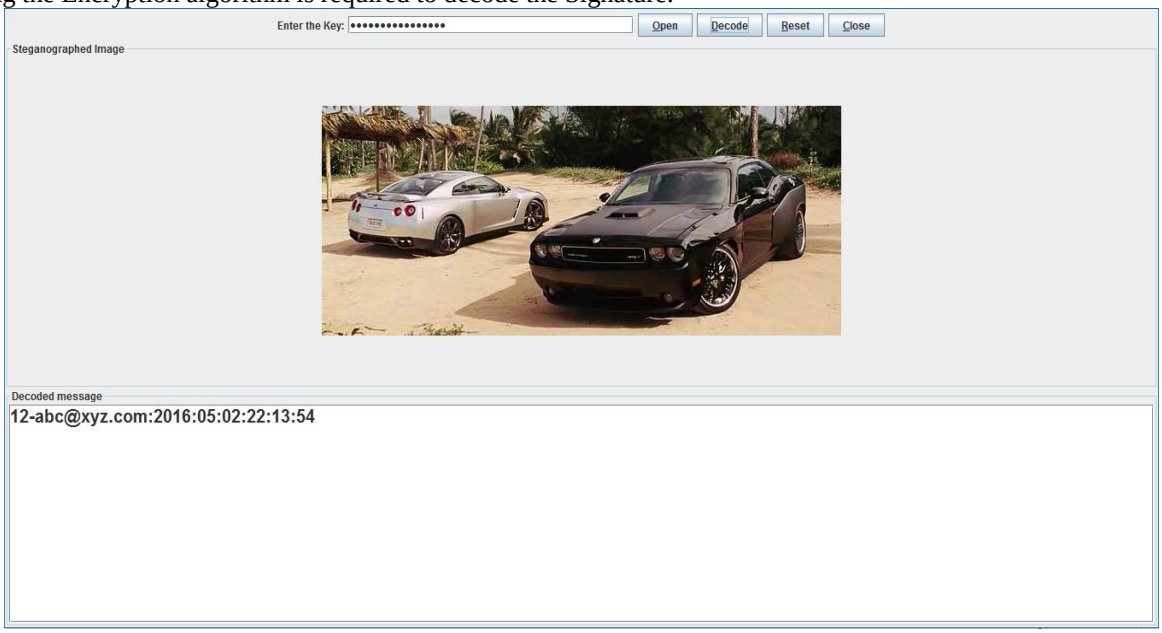

Fig. 4: Decoding the Stego image.

If a user, for eg "pqr@xyz.com" tries to download and then upload the same Stego image then the credentials of the new user will get appended o the previous embedded data., which is shown in Figure 5.

```
Plain  String          : 12-abc@xyz.com:2016:05:02:22:13:54//pqr@xyz.com:2016:05:02:22:15:53
AES Encrypted String :
□□□/□□#□9и□□□Mнwg□F□p$、p□8□□f□□□N□□□□rC□2□D□□&□r□□9?□a□fs□□□5M□[□□□□\□□N□□*
Base64 Encoded String : rpbEL7SgI9A506WQ2fuhTdSpd2eDRoRwJMu0cL84AuuhZr7r607Mxt3nckOpMqdEs/smoXIbhDk/
vGGbZnO00dM1TRpbA4aGB1zuAU6m2yo=
```

Fig. 5: New Signature

From the previous signature present in the image it could be identified that "*abc@xy.com*" is the real owner of the image and the application ca notify the user about the usage of his image by another user. Figure 6 shows the decoded Signature obtained from a previously uploaded Stego image.

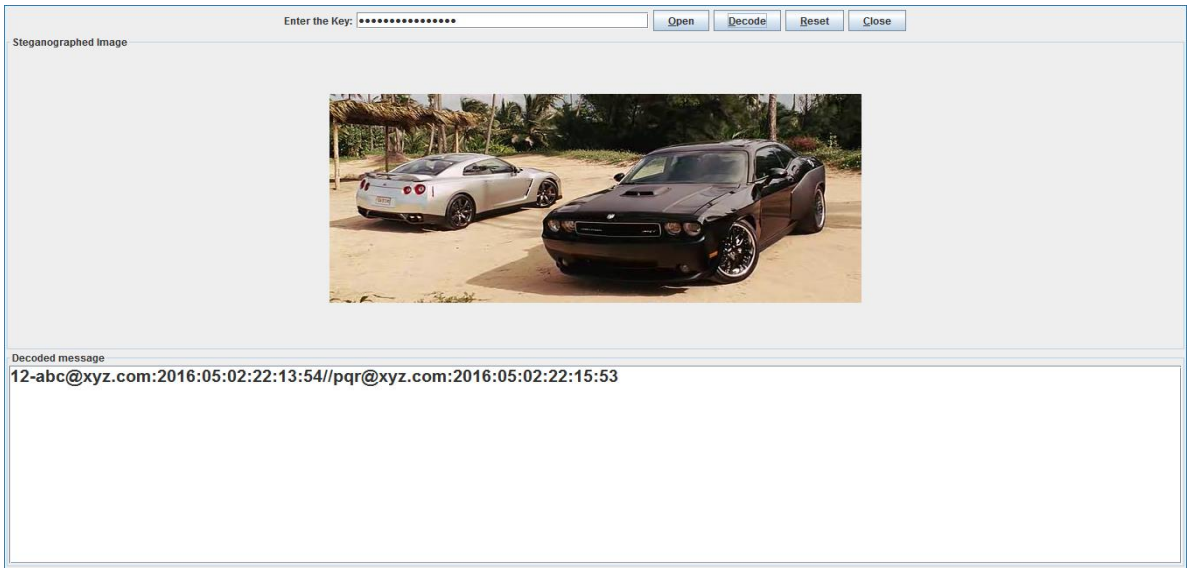

Fig. 6: Decoding a previously uploaded Stego Image

V. CONCLUSION

In this paper we have presented a digital image Authentication technique for the images being uploaded on social networking websites. The image that uploaded by the LSB steganography and AES algorithm hides the person's email id and current timestamp. The experimental details prove that the visual quality of the images is not affected even after repeatedly encoding the image. Secret data is not included in every pixel of the image.

REFERENCES

[1] Madhumita Panda and Atul Nag, "Plain Text Encryption Using AES, DES and SALSA20 by Java Based Bouncy Castle API on Windows and Linux" , 2015 Second International Conference on Advances in Computing and Communication Engineering

[2] Congfu Xu, Yafang Chen and Kevin Chiew , "An Approach to Image Spam Filtering Based on Base64 Encoding and N-Gram Feature Extraction", 2010 22nd International Conference on Tools with Artificial Intelligence

[3] Nidhi Grover and A.K. Mohapatra , " Digital Image Authentication Model Based on Edge Adaptive Steganography", 2013 Second International Conference on Advanced Computing, Networking and Security

[4] Min Wu, Member, IEEE, and Bede Liu, Fellow, IEEE, "Data Hiding in Binary Image for Authentication and Annotation", IEEE Trans. Image Processing, volume 6, Issue 4, Aug. 2004 Page(s): 528 – 538