STEGO IMAGE CORRUPTION DETECTION AND SECRET TEXT RECOVERY IN STEGANOGRAPHY COMMUNICATION

[1]JAGAN RAJ J, [2]C. KAVITHA, [3]K. SAKTHIVEL

[1]Research Scholar, Periyar University, Department of Computer Science, India

[2]Assistant Professor, Thiruvalluvar Govt. Arts College, Department of Computer Science, India

[3]Professor, K. S. Rangasamy College of Technology, Department of CSE, India

E-mail: [1]jae.jaganraj@gmail.com, [2]kavithachellapan@gmail.com, [3]sakthivelk@ksrct.ac.in

ABSTRACT

In this proposed work, an effort has been made to use multiple image files for steganography encoding alongside with the potential of secret text recovery in the event of any image corruption during the stego image transit. Proposed algorithm is effective on the safety factors of secret image, since the embedded checksum will validate for any unauthorized users or intruders, who plan to corrupt the image in any aspect. If any of the stego image underwent any steganalysis or Man in the Middle attack, then this proposed algorithm can effectively identify the potential corruption. The proposed multi cover image steganography model enables the receiver to send the secret text in more secured way and has the ability to detect the corruption in secret message. This solution is not only detecting the possible cover image corruptions but it also withstands one stego image corruption and has the capability to recover the original secret text even after one stego image corruption during the transmission of the secret message. This proposed work will increase the security of secret text that are being sent using steganography methods.

Keywords: *Image Steganography, Multi-Image Steganography, Data Corruption Detection*

1. INTRODUCTION

With the advent of the Internet, information security has been one of the most critical considerations in IT and communication. Cryptography has been created as a security tool and various different method of encrypting and decrypting data have been developed to keep the message confidential. Unfortunately, the contents of a message are often not adequate to keep it confidential, and the message will still need to be kept secret. Steganography is the method used for this. The art and science of hidden communications is steganography.

Steganography is a science of hiding a secret document, image or video inside the other file, message, image or video files, which continues to be an exceptionally flexible and powerful way of hiding or cover up information in plain sight. There are a few methods of hiding data using steganography techniques. The most common method is to embed data into a digital image file. We all recognize that digital images imply that there are a lot of megabytes of pixel values. It allows the room in the digital file for someone to embed the secret steganographic data. A good programmer can modify the least significant bits (LSB) of any media file format by using steganographic tool or program and embed the needed content in the digital image. This is achieved by hiding information in other information and also by hiding the presence of the information transmitted.

The word Steganography derives from the Greek term 'stego' which refers to 'cover' and 'graphic' which means 'writing' and refers to 'covered writing.' The information is concealed entirely in pictures in the image steganography. Image processing is one of the key fields for multimedia applications and we know that almost anywhere in the world these applications can be found.

2. STEGANOGRAPHY USAGE

The objective of steganography is to hide and deceive. It is a type of clandestine interaction, which may require some special technique to conceal messages. Since it doesn't require manipulating data, it's not a form of cryptography. Instead, it is a method of hiding of information, which can be implemented in sneaky ways. Where cryptography is a science that allows anonymity to a considerable degree, steganography is a discipline that allows secrecy as well as deception. Until now, the ultimate function in cryptography was always to maintain the confidentiality between the sender and the target recipient.

However, nowadays steganographic methods are more and more used in addition to cryptography to complement the secret data with more protective layers. The benefit of using steganography over encryption alone is that it does not turn your focus as an object of investigation on the planned hidden message. Clearly readable encrypted texts, however unbreakable, can be of concern to countries in which encryption is unlawfully happening and incriminate in themselves.

The very first documented usage of steganography could be dated directly to 440 B.C. in Greece, when Herodotus mentions two examples in his history[1]. Histiaeus sent a secret message to his vassal, Aristagoras, by shaving the head of his most loyal servant marking the secret message to his scalp, and then sending it on his way as soon as his hair had resurfaced, with the instruction, "When thou art come to Miletus, bid Aristagoras shave thy head, and look thereon." Wax tablets were also commonly used as reusable writing surfaces, occasionally shorthand also used for this communication. In addition, Demaratus sent a notification about the impending assault on Greece by writing it directly to the wooden back of the wax tablet before adding the beeswax surface. Wax tablets were in common use then as reusable writing surfaces, sometimes used for shorthand.

The second instance[2] also originated from Herodotus, saying that it was important for a soldier named Demeratus to send Sparta a message that Xerxes was preparing to conquer Greece. Back then, text printed on wax-covered tablets was the writing tool. The wax was extracted from the tablet by Demeratus, the hidden letter was written on the underlying wood, the wax tablet was retrieved to pose as a blank tablet, and the text was eventually sent without being identified. Invisible inks, based on the natural substances such as fruit juices and milk, were used by the Romans. By heating the secret text, this was done, thereby exposing its contents.

3. TYPES OF STEGANOGRAPHY

Based on the type of cover file being used in the steganography technique, various types of steganography methods as shown in figure 1. For example, A steganography technique can be called as image steganography, if it uses a digital image for storing the secret text and so on so forth for other steganography types.

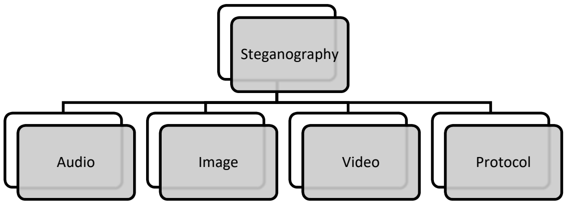

Figure 1: Types of Steganography

3.1 Text Steganography

The cover file being used for the steganography process would be in text format and the secret message found in the cover file predominantly in text type. For text steganography, the embedding method is determined by the number of letters, white spaces and capital letters like how it is used in the Morse code for radio communication.

3.2. Image Steganography

By taking the cover object as the image file, image steganography is a method used to hide a secret message. In this steganography, graphical visual images are commonly used as a cover source, and this cover file allows the user to embed a huge amount of secret data bits. The main value of image steganography is that the cover image does not draw an intruder's eye during the transmission or when it is posted online for the communication to the other party.

3.2 Audio Steganography

Audio steganography is a tool used to imperceptibly relay confidential information by altering an audio signal digitally. It's the science of concealing any text or audio information in a cover file. Embedding secret signals in optical sound is a more complex process as it requires using the cover file's decoding algorithm to manipulate the bits and store information inside it.

Several variants algorithms available for embedding data into digital audio have been developed by domain experts. As opposed to other cover signals, audio signals are used as covers more often; this is because of their bigger size, so that the cover file can hide more data. Moreover, digital audio signals are more redundant and have a high data transmission rate which makes them acceptable for coverage files.

3.3 Video Steganography

Video steganography has also become a major research area with various data hiding technologies, and has been a promising technique. A video is a collection of frames, and each frame is an image. So if we pull out all the frames from a video, we can use this method to store our data using LSB steganography and stitch those frames back into a video with the secret message. This is not only about the protection of hidden data being transferred in covert communications, but video files still provide a large volume of data stream to utilize.

Based on the embedded position of the secret message, video steganography is broken down into three groups: intra-embedding, pre-embedding and post-embedding[3]. Intra-embedding methods are categorized according to the video encoding phases, such as intra-prediction, motion vectors, pixel interpolation, transformation coefficients.

3.4 Protocol/Network Steganography

The Steganography using Protocol/Network is a modern data hiding solution that has become popular in recent days. For hiding information in this steganography, the TCP/IP (Transmission Control Protocol/Internet Protocol) suite network layer protocol is used and not confined solely to network protocols. Covert channels in the network layer of the OSI architecture are used for data hiding. Covert canals circumvent the authentication protocols of the network system. The aim is either to steal data or to exchange secret messages over a network by using the network protocol.

Network steganography is a family of approaches for modifying the configuration of packet forwarding and hybrid methods of data within the headers of network protocols and pay loading fields of packets. TCP (Transmission Control Protocol), IPv4 (Internet Protocol version 4), NFS (Network File Sharing), CIFS (Common Internet File System) etc., are examples of protocols used in protocol steganography.

4. STEGANOGRAPHY PHASES

In order to finish the task of secret message exchange from sender to receiver, each Steganography algorithm must move through three phases as given in figure 2 and figure 3. An intruder can also establish attacks over the stego images, if any image causes suspicion, when it is getting transmitted over the network or posted in the social media. Figure 3 shows where the possible attacks or stego image corruption can happen in the steganography cycle.

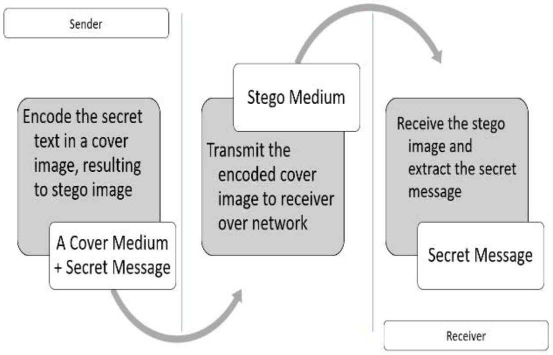
Sender
Encode the secret text in a cover image, resulting to stego image
A Cover Medium + Secret Message
Stego Medium
Transmit the encoded cover image to receiver over network
Receive the stego image and extract the secret message
Secret Message
Receiver

Figure 2. Phases of Steganography

4.1 Sender

The sender's primary objective is to embed the secret message in the stego-medium and relay it through the contact channel. As discussed, cover file format can be any one of the file types from image, audio, video etc.,

4.2 Communication channel

A physical or wireless medium that carries an encoded cover image, which is called as stego-image, with a hidden message in the network or some other delivery medium. This file is also sometimes shared over the social media postings. The embedding technique should be sufficiently powerful to protect the secret message for any possible interference.

4.3 Receiver

This is the last step in which the cover medium is identified and processed in this steganography method to decide if the secret text transmitted via the transmission channel is present.

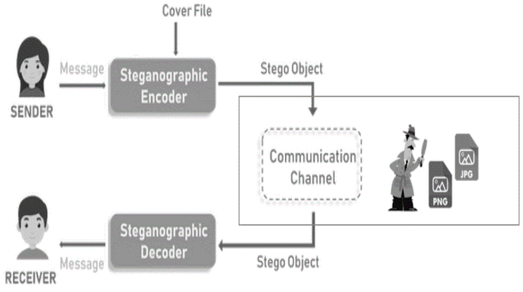

Figure 3. LSB Steganography communication flow

5. RELATED WORKS

A succinct review based on the study of these papers related to our work is as follows. J. Homg et al. [4] and M. A. Hameed rt al [5] described several image steganography techniques in spatial domain. Along with existing techniques like LSB, layout management schemes and replacing only l's or only zero's, some more methods like replacing intermediate bit, raster scan principle, color-based data hiding and shape-based data hiding are also proposed. M. C. Kasapbasi et al. [6] and S. D. R. I. Moses [7] developed an improved method for image-based steganography using LSB technique. All these techniques are primarily focused on the LSB steganography optimization and the steganography operation happens in one cover image. Though a high-capacity focus is made on the research contributions in [4] to [7], these algorithms lack the ability to withstand the Steganalysis or man in the middle (MiM) attack on stego images during transmission with the intention of disrupting the transmission or acquiring the secret text that is in transit. During the transmission there are some additional metadata is embedded as for better security[8].

6. EXISTING STEGANOGRAPHY MODEL

A typical Steganography system consists of following elements as discussed earlier in this article. Functions and usage of these elements are depicted in the figure 4.

- Cover Object (C)
- Secret Message/Text (M)
- Stego Object (S)

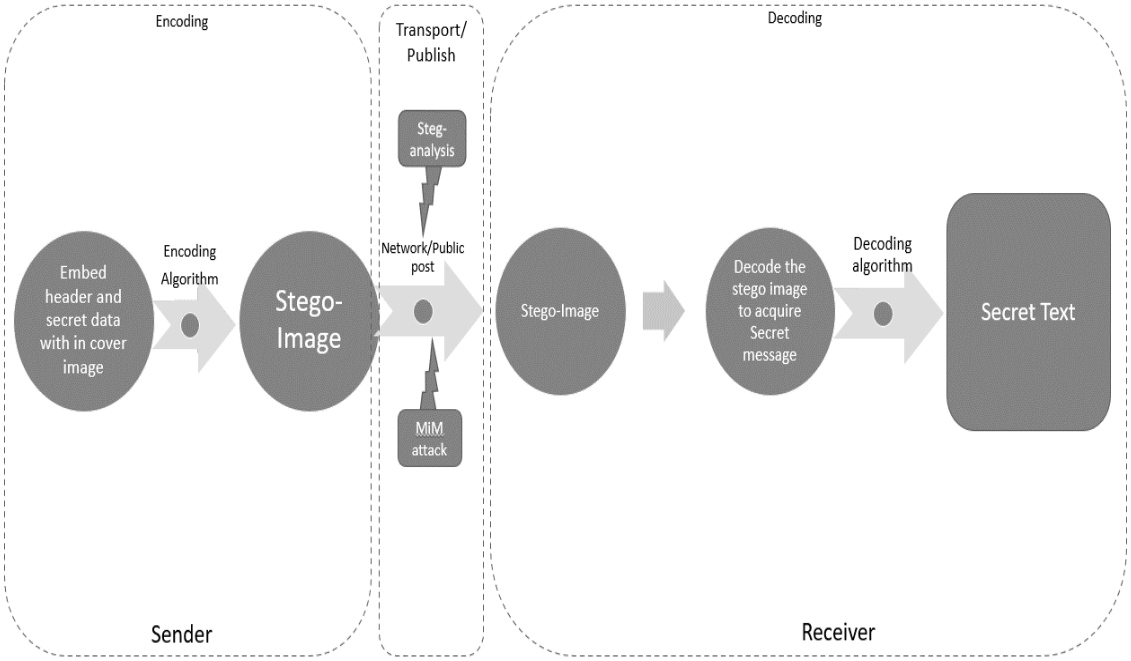

Figure 4. LSB Steganography model

6.1. Cover Object

The Steganography cover-objects are those where coded messages are concealed. Any digital files like photos, audio, writing and pictures may be a cover object. An image file is most commonly used over object to hide data. Most of the times the cover image stays as single file in a steganography cycle.

6.2. Secret Text/Message

T The true vital element is the secret message in steganography, which must be hidden in the cover file. It is necessary to not allow noticeable deterioration of quality of the cover file after embedding the secret message. This will unnecessarily attract the attention and create distrust during its transmission.

6.3. Stego Object

The cover file entity is called as "stego object" after storing the secret data. This entity is further passed to the receiving end via public social media post or sent by e-mail servers to finish the process.

7. PROPOSED STEGANOGRAPHY MODEL

The proposed multi cover image steganography model as shown in figure 5 enables the receiver to send the secret text in more secured way and has the ability to detect the corruption in secret message. This solution is not only detecting the possible cover image corruptions but it also withstands one stego image corruption and has the capability to recover the original secret text even after one stego image corruption during the transmission of the secret message. There are several chances for a stego file to get corrupted during the transmission. Incidents like MiM attack or any network transmission issue can cause this corruption and which might lead to lose the ability to extract message. This algorithm adds metadata on the secret data being transferred in more than one stego images and maximizes the security of hidden message.This proposed work will increase the security of secret text that are being sent using steganography methods.

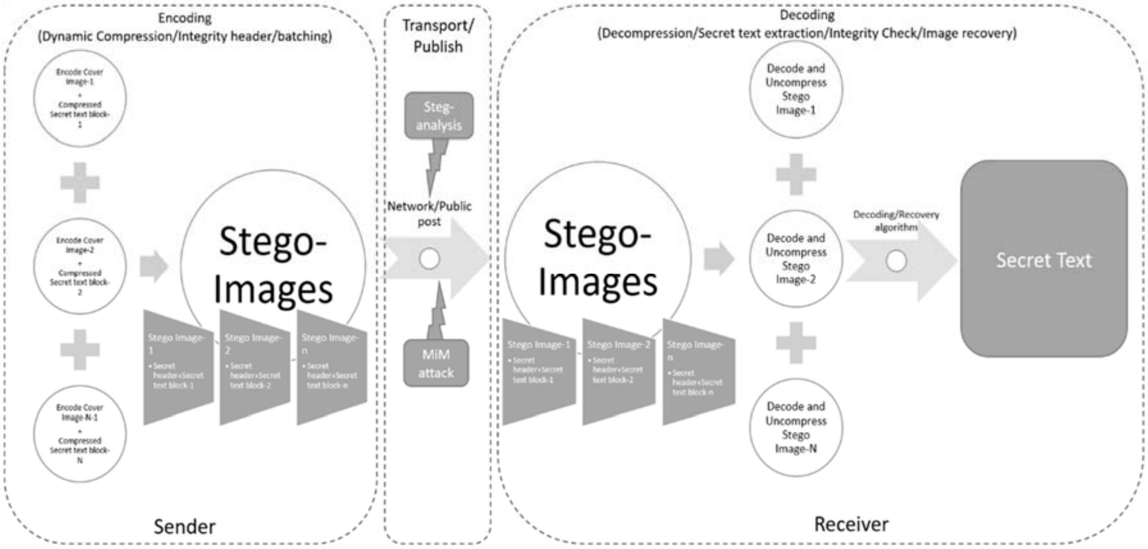
Encoding
(Dynamic Compression/Integrity header/batching)
Encode Cover Image-1 + Compressed Secret text block-1
Encode Cover Image-2 + Compressed Secret text block-2
Encode Cover Image-N-1 + Compressed Secret text block-N
Stego-Images
Stego Image-1 • Secret header+Secret text block-1
Stego Image-2 • Secret header+Secret text block-2
Stego Image-n • Secret header+Secret text block-n
Sender
Transport/ Publish
Steg-analysis
Network/Public post
MiM attack
Decoding
(Decompression/Secret text extraction/Integrity Check/Image recovery)
Stego-Images
Stego Image-1 • Secret header+Secret text block-1
Stego Image-2 • Secret header+Secret text block-2
Stego Image-n • Secret header+Secret text block-n
Decode and Uncompress Stego Image-1
Decode and Uncompress Stego Image-2
Decode and Uncompress Stego Image-N
Decoding/Recovery algorithm
Secret Text
Receiver

Figure 5. Proposed Steganography model for corruption detection and recovery

7.1. Embedding Algorithm

As part of secret message embedding in given cover images, first the embedding algorithm (refer algorithm 1) reads the height (X) and width (Y) of the cover image. The embedding algorithm then calculates the other meta data needed for the secret data embedding, verification and recovery in case of corruption detection in the receiving end. The metadata includes the md5 (message digest 5) hash, total number of cover images used, order of the current stego image, start and end LSB bit of the secret data existence in the given cover image. The given secret message is then divided into equal number of chunks as number cover images available for the embedding minus one.

The last stego image will be storing the XOR (refer table 1 for sample XOR value calculation when using multiple stego images using the proposed algorithm in the steganography model) of all the equally divided chunks and stores the final output in its LSBs. MD5 being calculated is for each chunk of the secret message, which will be used in the receiving end for corruption determination. The collected meta data gets transferred alongside with the secret message.

The primary inputs to the algorithm are cover image list (C), secret message (M) and number of least significant bit (k) that must be used during the embedding process. The embedding process includes traversing of each color pixel's RGB bits for the given 'k' bits.

7.2. Extraction Algorithm

Extraction algorithm is similar to compression procedure, but the steps are orderly reversed to obtain original secret message from stego-images (refer algorithm 2). After reading the metadata from each of the stego image, the images are orderly extracted for the secret message (M). During the extraction process the metadata and the actual secret message are separately stored. An important entity in the metadata at the receiving end is the MD5 hash value for each of this message chunk.

The extracted message chunks will again undergo the MD5 message digest hash value calculation and this value should match the value in metadata (MD), if not XOR calculation of all the equivalent bits of other stego images (except the corrupted stego image) is calculation and the original bits are recovered as shown in the table 2. The table 2 shows the recovering of original secret text on the event of stego image-3 corruption.

Algorithm-1: Secret data embedding algorithm
Input: Cover images, Message
Output: Stego images list

```
procedure LSB_embed (C, M, k)
    Read cover images, C
    Read secret message, M
    $X \leftarrow$ Height of the cover image, C
    $Y \leftarrow$ Width of the cover image, C
    $W \leftarrow$ Number of words in the message, M
    $L \leftarrow$ Number of characters in the message, M
    $E \leftarrow$ Equally divided secret message based on $n(c)$
    $T \leftarrow$ String vector of words from the secret message, M
    $S \leftarrow$ Secret message vector in binary form, T
    Initialize, $S \leftarrow [\,]$
    for $w \leftarrow 1$ to $W$ in steps of 1 do
        $S_{[w]} \leftarrow binary((T_{[w]}))$
    end for
    $H \leftarrow binary\ (StegoHeader(S))$
    $S \leftarrow H + S$
    for $e \leftarrow 1$ to $E$ in steps of 1 do
        for $i \leftarrow 1$ to $Y$ in steps of 1 do
            for $j \leftarrow 1$ to $X$ in steps of 1 do
                for $x \leftarrow 1$ to 8 in steps of 1 do
                $rb = resetFromNthBit(k)_b$
                $C = \sum_{l=(8-k)}^{8} (c_{[i][j][x]} \& rb)$
                $C = \sum_{l=(8-k)}^{8} (c_{[i][j][x]} \mid s_{[c++]})$
                end for
            end for
        end for
    end for
    return stego-image list C, the secret text embedded cover images
end procedure
```

Algorithm-2: Multi cover image extraction algorithm
Input: Stego image
Output: Secret message

```
procedure_eLSB_extract(C)
    Read Stego images, C
    $H \leftarrow$ extracted header from stego images, C
    $X \leftarrow$ Height of the cover image from header, H
    $Y \leftarrow$ Width of the cover image from header, H
    $MD \leftarrow$ List structure of MD5 metadata from each stego image, $C_i$
    $k \leftarrow$ number of LSB used from header data, H
    $l \leftarrow$ Length of secret message in bytes from header data, H
    $E \leftarrow Equally\ divided\ secret\ message\ based\ on\ n(c), from\ header\ data, H$
    $Initialize, h \leftarrow 64$
    $for\ i \leftarrow 1\ to\ Y\ in\ steps\ of\ 1\ do$
        $for\ e \leftarrow 1\ to\ E\ in\ steps\ of\ 1\ do$
            $for\ j \leftarrow 1\ to\ X\ in\ steps\ of\ 1\ do$
                $for\ x \leftarrow 1\ to\ 8\ in\ steps\ of\ 1\ do$
                    $if\ bytes \leq h$
                        $bytes \leftarrow bytes + 1$
                        $continue\ next\ iteration\ in\ i\ loop;$
                    $end\ if$
                    $T = \sum_{l=(8-k)}^{8} (c_{[i][j][x]} \gg k)$
                    $S_{[i]} = S_{[i]} + T$
                $end\ for$
            $end\ for$
        $end\ for$
    $end\ for$
    $corruption \leftarrow 0$
    $corruptedindex \leftarrow 0$
    $for\ x \leftarrow 1\ to\ E\ in\ steps\ of\ 1\ do$
        $if\ messagedigest(S_{[e]} = MD_{[i]})$
            $no\ corruption;$
        $else$
            $corruption \leftarrow 1$
            $corruptedindex \leftarrow x$
        $end\ if$
    $end\ for$
    $if\ messagedigest(corruption = 1)$
        $T = call\ recoverImageXOR(S, corruptedindex);$
    $else$
        $no\ corruption;$
    return T, the secret message
end procedure
```

8. EXPERIMENTAL RESULTS

Using the above proposed algorithm, a sample “hello world” can be encoded in five different cover images as mentioned below in Table 1 and the same can be sent over the network to the receiver to complete the communication cycle. If all the MD5 checksum values are matching for each message chunk, then it means that no corruption took place during the transmission and the data sent by the sender is intact and hence, no recovery is needed in that case.

Table 1. Sample text interpretation in the proposed algorithm using five cover images

Secret Text	ASCII	Binary Equivalent	Cover Image-1	Cover Image-2	Cover Image-3	Cover Image-4	Cover Image-5 (RS)
h	104	01101000	01	10	10	00	**01**
e	101	01100101	01	10	01	10	**00**
l	108	01101100	01	10	11	10	**10**
l	108	01101100	01	10	11	10	**10**

o	111	01101111	01	10	11	11	**11**
w	119	01110111	01	11	01	11	**00**
o	111	01101111	01	10	11	11	**11**
r	114	01110010	01	11	00	01	**11**
l	108	01101100	01	10	11	10	**10**
d	100	01100100	01	10	01	10	**00**

On the contrary, let us say that the stego image-4 from stego images set underwent an attack and the message digest (MD5) mismatches with the content. On this contradiction, receiver will be able to identify that there was a corruption and recover the original text from other stego images as depicted below on Table 2. The extraction algorithm flags when there is a mismatch in the MD5 and regenerated the original secret text using the recoverImageXOR() function call, which takes the all stego images and the index number of corrupted image as input. This function collects LSB values of all the secret text embedded bits from non-corrupted stego images and generates the XOR to find the original value as calculated in (1).

$$I_r = (I_1 \oplus I_2 \oplus \dots I_n) \qquad (1)$$

Table 2. Secret text recovery on the loss of Image-3 due to steganalysis attack

Secret Text	**ASCII**	**Binary Equivalent**	**Image-1** **(I₁)**	**Image-2** **(I₂)**	**Image-3** **(I₃)**	**Image-4** **(I₄)**	**Image-5(RS)**	**Recovered Image-3** $I_3 = I_1 \oplus I_2 \oplus I_4)$
h	104	01101000	01	10	~~**10**~~	00	01	**10**
e	101	01100101	01	10	~~**01**~~	10	00	**01**
l	108	01101100	01	10	~~**11**~~	10	10	**11**
l	108	01101100	01	10	~~**11**~~	10	10	**11**
o	111	01101111	01	10	~~**11**~~	11	11	**11**
w	119	01110111	01	11	~~**01**~~	11	00	**01**
o	111	01101111	01	10	~~**11**~~	11	11	**11**
r	114	01110010	01	11	~~**00**~~	01	11	**00**
l	108	01101100	01	10	~~**11**~~	10	10	**11**
d	100	01100100	01	10	~~**01**~~	10	00	**01**

The comparison of results with data transfer without corruption/intruder's intervention are shown in Table 3 and Table 4 with their corresponding checksums for a message chunk. The first entry shows the uncorrupted message hash, which matches in both on the receiving end and the checksum sent over the header, which was part of the metadata as seen in the algorithm-1. Any mismatch means in this checksum value means that the stego file went through an attack during the transmission. The second entry in the table 3 shows the mismatched MD5 checksum value.

Table 3. Comparison of checksum for the steganographed image, which transferred with corruption because of image color change

Image stage	**Checksum value(md5)** **(D)**	**Secret Text** **(ST)**
After Encoding	e64d69492b 460cd25dbb 42f970409f 23	This is a secret text, which is hidden in an image file using steganography and

		having embedded checksum in it
After Decoding	e1c2e6f45c57978c86a78df764295972	Secret text got corrupted as the message digest (MD5) are not identical

Table 4 compares the proposed model with the existing LSB steganography model and lists the advantages of the proposed method. The security aspect of the secret message in addressed here with message corruption detection and the solution also adds the ability to recover the secret text, when there is a corruption.

Table 4. Qualitative Comparison of proposed methodology

Parameters	**LSB methods (Existing)**	**Multi-image model Method (Proposed)**
Secret Text Recovery on image corruption	No	Yes
MD5 hash on Stego files	No	Yes
Capability to detect MiM (Man in the Middle) attack	No	Yes
Digest size used(md5)	0 bit	128 bits
Robustness	Less data loss	No Data loss
Integrity Check at receiving end	No	Yes

The existing steganography methods are based on single cover files. Whereas the novelty of this method enables the multi-image steganography with the provision of detecting the secret text or cover image corruption as shown in Table 3. Table 4 compares the proposed model with the existing LSB steganography model and lists the advantages of the proposed algorithm.

The proposed method can perform the integrity check of the cover file and when the embedded checksum and newly calculated checksums are different, it means that the secret data sent by the sender is not received as it is and went through some attacks during the transmission. This work not only detects the possible data corruptions but also provides a way to recover the secret text using XOR calculation of remaining cover files. This technique is never tried in any of the past steganography methods and newly introduced technique in this algorithm.

9. CONCLUSIONS

The proposed novel technique is effective on protecting secret message compared to existing LSB steganography algorithms. As there are no known previous attempts made to detect secret data corruption or recover the corrupted secret data in the existing works, this embedded checksum method would detect or validate for any unauthorized users or intruders corrupted the picture in any aspect. If any of the stego image underwent any steganalysis or MiM (Man-in-the-Middle) attack, then this proposed algorithm can recover the content of one stego image using other intact stego images received at the receiving end. Even if the attacker found the algorithm used for steganography in the stego picture by steganalysis and altered the content of the hidden document, the tampering can be found by calculating and comparing the checksum at the recipient end. Novelty of this approach, is the security of secret message is preserved and the model withstand a stego-image attack. This work is not only detecting the attack but has the ability to recover the lost stego image content in the receiving end during a steganography communication. This particular work will increase the security of secret text being transferred in the network using steganography methods. The future work could be improving the algorithm to enable the algorithm for multiple cover image corruption recovery.

10. REFERENCES

[1] Petitcolas, FAP, Anderson RJ, Kuhn MG (1999). "Information Hiding: A survey", *Proceedings of the IEEE*. 87 (7): 1062–78. doi:10.1109/5.771065.

[2] Dunbar, B., "A detailed look at steganographic techniques and their use in an open-systems environment", *Sans InfoSec Reading Room* (2002)

[3] Y. Liu, S. Liu, Y. Wang, H. Zhao, S. Liu "Video steganography: a review", *Neurocomputing,* 335 (2019), pp. 238-250 (2019)

[4] J. Horng, C. Chang and G. Li, "Steganography Using Quotient Value Differencing and LSB

Substitution for AMBTC Compressed Images", *IEEE Access*, vol. 8, pp. 129347-129358, 2020, doi: 10.1109/ACCESS.2020.3009232.

[5] M. A. Hameed, M. Hassaballah, S. Aly and A. I. Awad, "An Adaptive Image Steganography Method Based on Histogram of Oriented Gradient and PVD-LSB Techniques", *IEEE Access*, vol. 7, pp. 185189-185204, 2019, doi: 10.1109/ACCESS.2019.2960254.

[6] M. C. Kasapbaşi, "A New Chaotic Image Steganography Technique Based on Huffman Compression of Turkish Texts and Fractal Encryption with Post-Quantum Security", *IEEE Access*, vol. 7, pp. 148495-148510, 2019, doi: 10.1109/ACCESS.2019.2946807.

[7] Setiadi De Rosal Ignatius Moses, "Payload Enhancement on Least Significant Bit Image Steganography Using Edge Area Dilation", *Polish Academy of Sciences, Committee of Electronics and Telecommunication*, Vol. 65, No. 2, pp. 287-292, 2019, DOI 10.24425/ijet.2019.126312

[8] Jagan Raj J and Prasath S, "Validating Data Integrity in Steganographed Images using Embedded Checksum Technique", *IJCA Proceedings on National Conference on Research Issues in Image Analysis and Mining Intelligence NCRIIAMI* 2015(1):5-8, June 2015

[9] J. R. Jayapandiyan, C. Kavitha and K. Sakthivel, "Enhanced Least Significant Bit Replacement Algorithm in Spatial Domain of Steganography Using Character Sequence Optimization,", *IEEE Access*, vol. 8, pp. 136537-136545, 2020, doi: 10.1109/ACCESS.2020.3009234.

[10] Jayapandiyan, Jagan Raj & C., Kavitha & Sakthivel, K. "Optimal Secret Text Compression Technique for Steganographic Encoding by Dynamic Ranking Algorithm", *Journal of Physics: Conference Series*. 1427. 012005, (2020), 10.1088/1742-6596/1427/1/012005.

[11] Jayapandiyan, Jagan Raj & C., Kavitha, "Coalesced Technique in Steganographic Images Using Encoded Conversion for Augmented Security", *International Journal of Innovative Technology and Creative Engineering*, Vol.7, No.4, April 2017

[12] Jayapandiyan, Jagan Raj & C., Kavitha, "Enhancing the data security and data integrity in steganographed images by store bit randomization", *International Journal of Innovative Technology and Creative Engineering*, Vol.5, No.12, Dec 2015

[13] S. Prasad and A. K. Pal, ''Logistic map-based image Steganography scheme using combined LSB and PVD for security enhancement'', *Emerging Technologies in Data Mining and Information Security*, vol. 814, P. Dutta, J. Mandal, A. Bhattacharya, and S. Dutta, Eds. Singapore: Springer, 2018

[14] T. Sudhakar, S. S. Venugopalaswamy Sriraman and N. Venkateswaran, "Synthesis and Evaluation of Improved Reference Matrix Models for High Capacity Image Steganography," *2020 International Conference on Artificial Intelligence and Signal Processing (AISP)*, 2020, pp. 1-6, doi: 10.1109/AISP48273.2020.9073531.

[15] Zhang, L., Chen, D., "The large capacity embedding algorithm for H.264/AVC intra-prediction mode video steganography based on linear block code over Z4", *Multimed Tools Appl* 79, 12659–12677 (2020). https://doi.org/10.1007/s11042-019-08528-7

[16] C. Lee, J. Shen, S. Agrawal, Y. Wang and Y. Lee, "Data Hiding Method Based on 3D Magic Cube,", *IEEE Access*, vol. 8, pp. 39445-39453, 2020, doi: 10.1109/ACCESS.2020.2975385.

[17] Chen, Bolin, W. Luo and Peijia Zheng. "Enhancing Steganography via Stego Post-processing by Reducing Image Residual Difference.", *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security* (2019): n. pag.

[18] S. Chen, C. Chang and Q. Liao, "Fidelity Preserved Data Hiding in Encrypted Images Based on Homomorphism and Matrix Embedding,", *IEEE Access*, vol. 8, pp. 22345-22356, 2020, doi: 10.1109/ACCESS.2020.2968577.

[19] S. S. Yadahalli, S. Rege and R. Sonkusare, "Implementation and analysis of image steganography using Least Significant Bit and Discrete Wavelet Transform techniques", *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, COIMBATORE, India, 2020, pp. 1325-1330, doi: 10.1109/ICCES48766.2020.9137887.

[20] Akram, Waseem & Kumar, Puneet. "A Novel Approach for Implementing the Bit Inversion Technique to Increase the PSNR of Stego Images and to Remove its Issues", *International Journal of Applied Engineering Research.* 13, 2018

[21] D. Samidha and D. Agrawa, "Random Image Steganography in Spatial Domain", *IEEE International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System(ICEVENT),* pp. 1–3, 2013.

[22] A. Singh and H. Singh, "An Improved LSB Based Image Steganography Technique for RGB Color Images", *IEEE International Conference on Electrical, Computer and Communication technologies*, pp. 1-4, 2015.

[23] Boritz, J. "IS Practitioners' Views on Core Concepts of Information Integrity", *International Journal of Accounting Information Systems.* Elsevier. Retrieved 12 August 2011.

[24] B. Pfitzmann, "Information hiding terminology", *Proc. ACM 1st Int. Workshop Inf. Hiding,* Cambridge, U.K. Berlin, Germany: SpringerVerlag, 1996, pp. 347–350.

[25] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen", *Computer,* vol. 31, no. 2, pp. 26–34, Feb. 1998.

[26] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography", *IEEE Security Privacy*, vol. 1, no. 3, pp. 32–44, May 2003.

[27] E. Lin and E. Delp, "A review of data hiding in digital images", *Proc. PICS,* Savannah, GA, USA, 1999, pp. 274–278.

[28] M. M. Sadek, A. S. Khalifa, and M. G. Mostafa, "Video steganography: A comprehensive review", *Multimedia Tools Appl.,* vol. 74, no. 17, pp. 7063–7094, 2015.

[29] F. Djebbar, B. Ayad, K. A. Meraim, and H. Hamam, "Comparative study of digital audio steganography techniques", *EURASIP J. Audio, Speech, Music Process.,* vol. 25, p. 25, Oct. 2012.

[30] H. Singh, P. K. Singh, and K. Saroha, "A survey on text-based steganography", *Proc. 3rd Nat. Conf.,* New Delhi, India, 2009, pp. 1–6.

[31] Tanmay Sinha Roy, "Image Steganography Using Lsb Bit-Plane Substitution", *International Research Journal of Engineering and Technology (IRJET),* vol: 3 issue: 12, Dec.2016