Modification in Spatial, Extraction from Transform: A new approach for JPEG steganography

Morteza Darvish Morshedi Hosseini*, Mojtaba Mahdavi**
Department of Information Technology
University of Isfahan
Isfahan, Iran
*M.DarvishMorshedi@eng.ui.ac.ir, **M.Mahdavi@eng.ui.ac.ir

Abstract— **Steganography is a choice in order to have a secret communication and it can be achieved by embedding a message inside a carrier object. JPEG as a common image format is a good target for steganography. In JPEG steganography, due to compression artifacts, it is not possible to embed data in pixels. As a result, most of the prevalent steganography methods for JPEG tend to embed data in JPEG coefficients and thus, the modification – on the side of sender – and extraction of data – on the side of recipient – will be made in the same domain. This paper aims at introducing a new method for JPEG steganography in which, modification and extraction of data is performed in different domains. In the proposed method, some intentional changes in spatial domain is made on the side of sender. These changes develop a specific pattern in JPEG coefficients and the recipient should extract data from JPEG coefficients. Most of JPEG steganography methods use a key in order to decentralize modification artifacts and therefore avoid detection. In the proposed method, the need for steganography secret key is removed. The proposed method also opposes the common idea that fewer changes in JPEG coefficients lead to less detectability. The experimental results including comparison of the proposed method with one of the outstanding methods for JPEG steganography show that the proposed method alters more JPEG coefficients and it is yet less detectable.**

Keywords-MSET; Steganography; JPEG; spatial; DCT; keyless steganography; cover; stego

I. INTRODUCTION

Steganography is an offline way of secret communications, in which a digital object should be employed in order to carry secret data. Using steganography, by concealing the existence of communication another layer of security is added which is known as *undetectability* [1]. In steganography, the digital object that does not contain embedded data is named as cover and the digital object that includes embedded data is named as stego. It is conventional to assume a passive warden that monitors all suspicious objects. The quality of a steganography method is evaluated by answering to the question that how much detectable is the existence of secret message in the stego objects [2]. Therefore, the stego objects should be indistinguishable from the cover ones.

In Steganography of digital images, there has been a great interest on JPEG as it is the most common format for storing and transferring digital images [3]. The compression artifacts occur during JPEG formatting, prevent steganographers from embedding data in the pixels. Instead, steganographers tend to embed data in the JPEG coefficients. However, direct modification of these coefficients leads to distortion which is considered as a known effect. The produced distortion can be indicated through Cartesian calibration [4] of the image and calculation of certain features from the calibrated image (e.g., blockiness). A number of steganalysis feature sets such as CC-PEV[1] [5, 4] and CC-JRM [6] utilize this subject in order to detect stego objects.

In order to avoid the detection of JPEG stegos, two mainstreams are followed [7]: The first strategy attempts to embed data while preserving a defined model of cover; the weakness of this approach is in defining newer steganalysis models that are better than the model used in steganography [8]. In the second strategy, the aim is to minimize an embedding distortion function.

In this paper, a noble steganography scheme is introduced which has two important characteristics: (1) the modification and data extraction domains are different. At the best of authors' knowledge, separation between modification and data extraction domains is not studied in steganography of JPEG yet. (2) The need for steganography key is removed. Most of JPEG steganography methods use a steganography key to decentralize modifications in order to be less detectable. The other reason for existence of steganography key is to prevent unauthorized extraction of data. In the proposed method, the key is removed safely and it is assumed that the secrecy of the data is obtained before embedding using data encryption. The other assumption in the proposed method is that both sender and recipient know the length of data.

Rest of this paper is organized as follows: In section II, previous arts for JPEG steganography are reviewed. In section III, the proposed method is described. Experimental results are presented in section IV, including steganalysis of the proposed method and comparison results, using the state of art feature sets and classifier. Finally, the paper is concluded in section V.

[1] http://dde.binghamton.edu/download/feature_extractors/download/ccpev548.m

978-1-4673-7609-9/15/$31.00 ©2015 IEEE

Authorized licensed use limited to: Fachhochschule Nordwestschweiz. Downloaded on June 30,2020 at 11:42:28 UTC from IEEE Xplore. Restrictions apply.

II. PREVIOUS ARTS

Although there may not be any steganography method in which modification domain and data extraction domain can be separated, there are some prior works on JPEG steganography. The former steganography methods such as F5 [9] and its improved version, nsF5 [10], attempted to minimize changes needed in order to embed a specific length of data. In F5, a binary hamming code was used to achieve this aim which is known as *Matrix Embedding*. In nsF5, the *wet paper codes* [11] were utilized to solve the problem of becoming zero for some coefficients which happend in F5 embedding. In this way, the total embedding changes were reduced. The other method that was introduced as PQ [12] attempted to minimize embedding distortion caused by changing JPEG coefficients through utilizing both the rounding error that occurs during quantization step of JPEG compression – known as *Side-Information* – and *wet paper codes*. MME [13] was another method that used *Side-Information*. Unlike F5 with the objective of minimizing embedding changes, in MME the target was to minimize distortion, even though the embedding changes might be more than one.

The idea of considering images as a model that has constant and modifiable parts was put forward in [8] as MB steganography. In MB, by keeping the probability distribution of coefficients almost the same as cover, a model was generated. Then the message had to be modeled using an entropy decoder according to the built model. In [14] the utilization of BCH[2] coding in steganography was described, which calculated the location to be changed in two ways: Using matrix embedding and using generator polynomial *g(x)*. As both ways required exhaustive search to find proper solution for embedding, this method required high computational power. The improved version namely BCHopt [15] considered two consecutive overlapped blocks and then found a joint solutions for hiding data in the two blocks in a manner that the intersected area to be the same in both solutions.

Modification of JPEG coefficients in cover produces distortion. The former methods such as F5, nsF5 and BCH attempted to reduce the distortion by minimizing the changes needed for embedding data, but the newer methods such as J-UNIWARD [16] and UED [17] intend to hide data while minimizing an embedding-distortion function. The distortion function can be defined as a norm between cover and stego feature vectors. In such cases, minimizing the distortion leads to preservation of the cover model [7].

In this paper, a new steganography method is proposed which does not measure any distortion function while embedding. Therefore, it is compared with nsF5, one of the most powerful methods, which does not measure any distortion function, as well. The experimental results of implementation show that the proposed method leads to more changes in JPEG coefficients than nsF5, but it is still less detectable than nsF5. In the next section, the proposed method is described.

[2] Bose-Chaudhuri-Hocquenghem

III. PROPOSED METHOD

For JPEG steganography, the direct modification of pixels is not considered as a way of hiding data. This is due to the changes occurring in the pixel values after JPEG compression, even though the compression quality is 100%. However, changing pixel values can change the equivalent JPEG coefficients. The idea underlying the proposed method is to make indirect changes in JPEG coefficients by making direct changes in pixel values. In the opinion of authors, correct implementation of this strategy can lead to minimization of distortion in spatial domain, as well as to conservation of correlations between JPEG coefficients. Based on the mentioned idea, a steganographic method is proposed which is explained in the following.

A. Overall Approach

In the proposed method, by making minimum sufficient changes in each 8x8 blocks of spatial domain, one has to look for the first state that forms the required JPEG coefficients block. While the required pattern is not formed, changing the pixel values in spatial domain continues.

B. Implementation

Fig. 1 shows overall process of embedding data using the proposed method.

The proposed method embeds data in the blocks containing sufficient non-zero JPEG coefficients. In this way, each JPEG coefficients block may be better affected by the changes in spatial domain. The proper blocks for embedding are those blocks that do not contain more than a particular number of zero elements in their equivalent JPEG coefficients blocks. These blocks of spatial domain are named *White Blocks (WBs)* and the equivalent JPEG coefficients blocks related to them are termed *JPEG White Blocks (JWBs)*. On the other side, there exist *Black Blocks* which will not be used for embedding.

In order to embed an encrypted message using the proposed method, the first step is to count *White Blocks* of the image. Then, an important parameter named *Modulus* should be calculated according to Fig. 2.

le = Length of Encrypted Message (bit)
nwb = Number of existence White Blocks in the image

$$bitPerBlock = \frac{le}{nwb}$$

$$Modulus = \left\lceil 2^{bitPerBlock} \right\rceil$$

Figure 2. Pseudo-code for calculating Modulus

In the next step, the encrypted message should be converted into the base of *Modulus*. Afterward, the digits of resulted string (i.e., *secret data*) should be embedded in *White Blocks*. Starting from the first *White Block*, the embedder embeds each single digit in a single block. According to the decision of steganographer, embedding in the *WBs* can be made in an ordinary order or even before embedding a secret key can be used to make a pseudo-random permutation on the digits. However, in the following it is described that why in the proposed method not using a steganography key is not

Authorized licensed use limited to: Fachhochschule Nordwestschweiz. Downloaded on June 30,2020 at 11:42:28 UTC from IEEE Xplore. Restrictions apply.

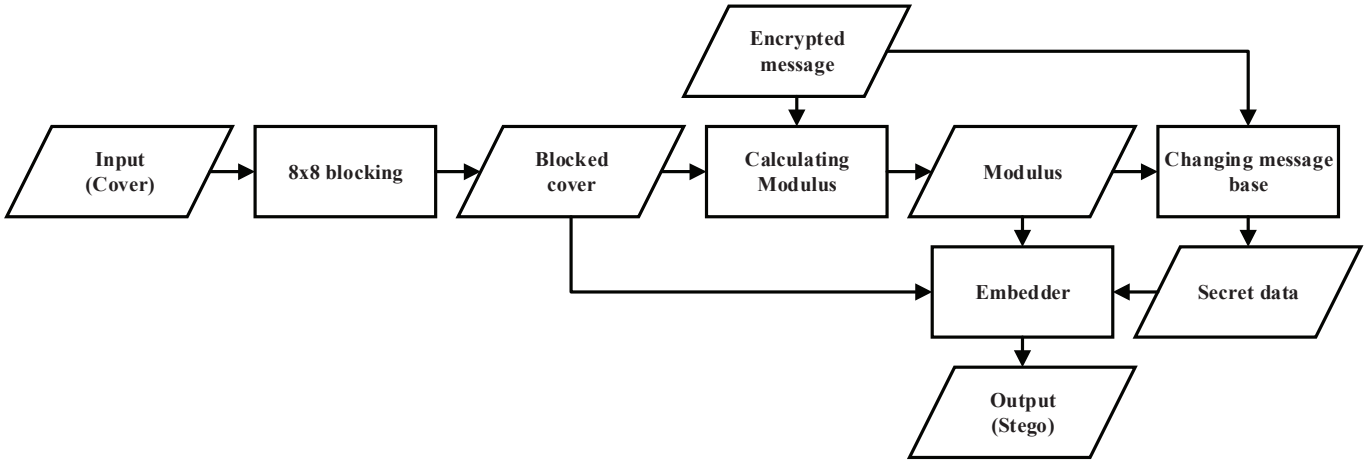

Figure 1. Overall embedding process using the proposed method

important. The process of ordinary embedding which can be related to *Embedder* part in Fig. 1, is presented in Fig. 3.

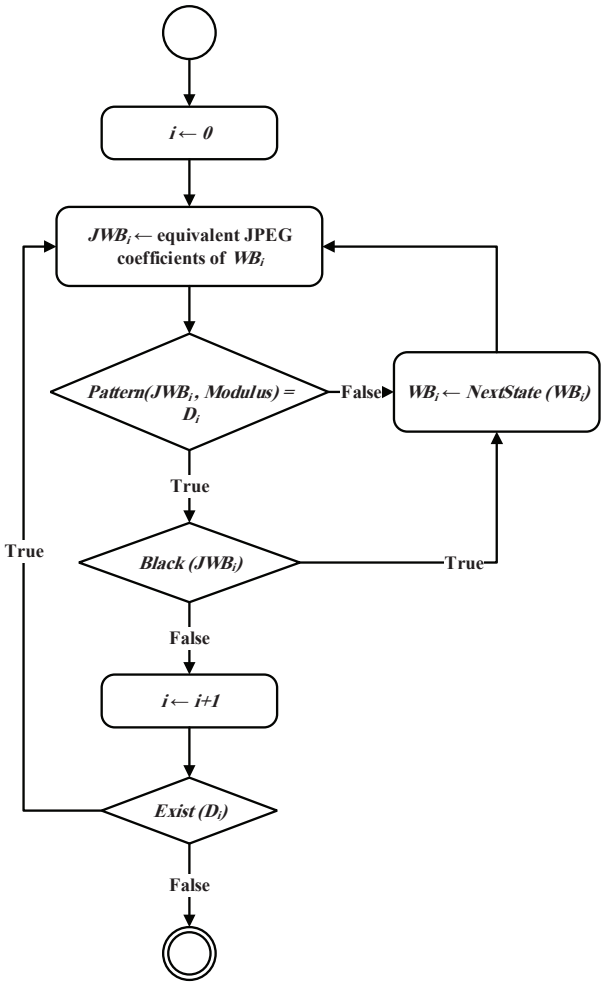

Figure 3. Ordinary embedding process related to Embedder

In Fig. 3, WB_i refers to the i^{th} *White Block* and JWB_i is its equivalent JPEG coefficients. Also D_i refers to i^{th} digit of *secret data* and *Pattern* is a function that returns a digit obtained from current block state. The pseudo-code for function *Pattern* is presented in Fig. 4.

```
Pattern(JWB , Modulus)
{
    H = hash(JWB)
    M = H mod Modulus
    return M
}
```

Figure 4. Pseudo-code of function Pattern

An important function in Fig. 3 is *NextState*. At first, this function resets the block to its original state; then, it selects a pixel from non-selected pixels in order to change the value by a particular number. Finally, it returns the JPEG coefficients of the changed block. Once all pixels have been checked and the required pattern is not found, the process continues by changing more than one pixel at a time. In these cases, the changes that are made in distinct locations are prior than those made in the identical locations.

It should be mentioned that changing the pixels does not necessarily lead to a change in JPEG coefficients. However, by continuing to change the pixels, finally the required pattern will be obtained.

The other important function in Fig. 3 is the *Black* function. This function verifies whether the current block is changed to a *Black Block* or not. This verification is important since changing a *White Block* to a *Black Block* during embedding leads to the corruption of data extraction on the side of recipient. Therefore, the patterns that convert the block to a *Black Block* are not acceptable.

In the proposed method, unlike the other steganography methods, not using a steganography secret key will not affect the security of the proposed method against steganalysis. This is because the encrypted message contains random bits as well

Authorized licensed use limited to: Fachhochschule Nordwestschweiz. Downloaded on June 30,2020 at 11:42:28 UTC from IEEE Xplore. Restrictions apply.

as the values obtained from the pattern of the blocks (due to hash function in Fig. 4). This is an important advantage of the proposed method over the other existent steganography methods.

It is suggested that according to maximum number of *White Block*s, some of them be used to show how many *White Block*s are used for embedding data. For example, a 512x512 image contains at most 4,096 *White Block*s, meaning that 4 digits in modulus of 10 is sufficient to represent the number of *White Block*s used. Therefore, quantity of 4 *White Block*s with the *Modulus=10* for them can be used to represent the length of *secret data*. However, the number of required blocks can be different according to selective *Modulus* for them.

C. Extranction of Secret Data

On the side of recipient, extraction of data is straightforward. The Embedding process does not convert any *White Block* to a *Black Block*, thus the number of *White blocks* in the cover and the stego are equal and the *White Blocks* of the stego are exactly in same locations that the covers exist. The recipient has to pick up JPEG coefficients plane from the JPEG file, then count number of *JPEG White blocks (JWBs)*. Provided that the length of data is known, the recipient can calculate *Modulus* according to Fig. 2. Afterward, the recipient should calculate the *Pattern* value of *JWBs* which are used for embedding *secret data* (Fig. 4). Joining the values together and converting the resulted string to a bit string will generate the encrypted message bits.

In the following, results of implementation are discussed including embedding time and its relation to *Modulus*, steganalysis of the proposed method by the state of art steganalysis feature sets, and comparison results.

IV. Experimental Results

In this section, the experimental results are presented. All of experiments were conducted on an Intel Core i7-4460 and 64-bit operation system. Main implementation was carried out using C#.NET. The image database used in this work was the original images of the third episode BOWS2 competition[3] which contains 10,000 512x512 8-bit grayscale images in 'pgm' format. All of the images where used for embedding. Using the proposed method, the embedding in each image was done as follows: first, the format of the input image was changed to BMP and a random bit string was produced as the encrypted message. Then using C# implementation, the produced bit string were embedded into the image. In the C# program, the value for changing pixels in order to generate new states of blocks was set to one. The output of C# program was a 2-D array of JPEG coefficients. Finally, an arbitrary 512x512 JPEG image was written in quality factor of 100% and using Matlab's 'imwrite'. Then using Matlab JPEG toolbox[4], the JPEG coefficients array of the image was replaced with the 2-D array obtained from the C# program. Based upon this procedure in steganalysis of proposed method, all that steganalysis finds is related to embedding changes and do not include compressor artifacts [16].

[3] http://bows2.ec-lille.fr/BOWS2OrigEp3.tgz

[4] http://dde.binghamton.edu/download/jpeg_toolbox.zip

In this implementation, the blocks that had more than eight non-zero JPEG coefficients were considered as *White Blocks*. In addition, MD5 was used as the hash function. Although length of embedding data in the proposed method can be an arbitrary number w.r.t the image size and context, in order to enable the comparison of the proposed method with the other JPEG steganography algorithms, some portions of non-zero AC coefficients (nzAC) were calculated as the length of embedding data. The embedding process was done for the relative data lengths of 0.1, 0.13, 0.15, 0.17 and 0.2 bit per nzAC. Accordingly, the length of embedding data in the proposed method (MSET) and nsF5 for each image was equal.

In the following, the results of embedding are discussed, and then the results of comparison with nsF5 method are presented.

A. Embedding Time Complexity

Although the image steganography is an offline process, the embedding time should be reasonable. In the following, the embedding time by the proposed method is discussed.

Fig. 5, 6 and 7 are related to 1000 randomly selected images from the total 10,000 images.

Fig. 5 shows that the length of data to be embedded has an exponential relationship with the parameter *Modulus*. Considering the relation between *Modulus*, number of *White Blocks* and the length of embedding data (Fig. 1), the relation between length of data and *Modulus* tends to be exponential. The scattered points in Fig. 5 are related to the images that have more *Black Blocks* than the typical images. In steganography by the proposed method, these images are not suitable for carrying embedded data and should be avoided.

The relation between *Modulus* and embedding time is presented in Fig. 6. Increment of embedding time is directly related to increment of Modulus (Fig. 6). As Modulus is related to the length of embedding data and number of *White Blocks* (Fig. 1), it is suggested that the images that have sufficient *White Blocks* be used. Therefore, the embedding time will be decreased. In addition, due to the fewer changes that will be

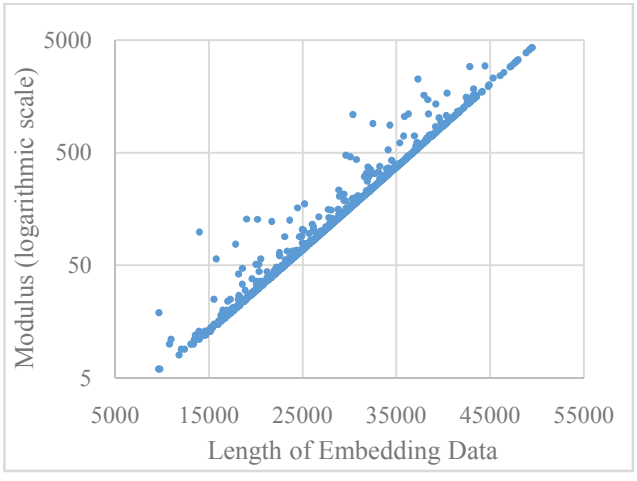

Figure 5. The relation between length of embedding data and *Modulus*, for 1000 images and output JPEG quality=100 (Vertical axies is logarithmic.)

Authorized licensed use limited to: Fachhochschule Nordwestschweiz. Downloaded on June 30,2020 at 11:42:28 UTC from IEEE Xplore. Restrictions apply.

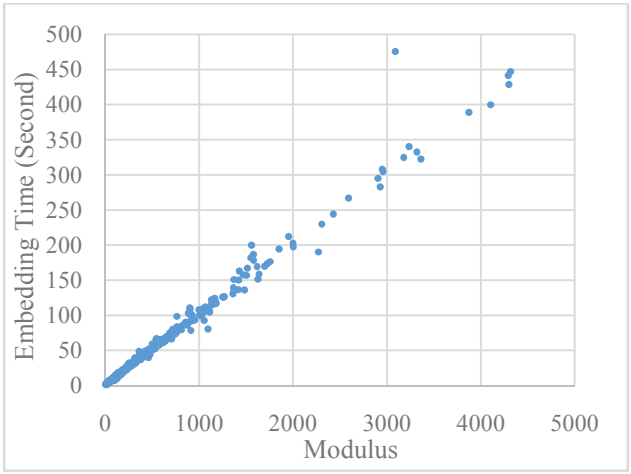

Figure 6. The relation between *Modulus* and embedding time, for 1000 images and output JPEG quality=100

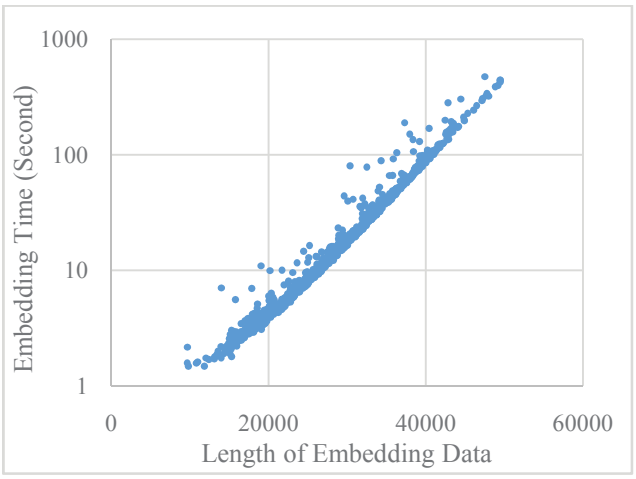

Figure 7. The relation between length of embedding data and embedding time, for 1000 images and output JPEG quality=100 (Vertical axies is logarithmic.)

required to form the pattern, the proposed method will be more secure.

Fig. 7 presents the overall relation between the length of data and the embedding time (which also can be concluded from Fig. 5 and 6). The embedding time has an exponential relation with the length of embedding data.

B. Steganalysis and Comparison to nsF5

nsF5 [10] is a JPEG steganography method which utilizes wet paper codes in order to minimize modification of JPEG coefficients. The reason of comparison MSET (proposed method) to nsF5 is that the proposed method does not use any distortion function, resembling nsF5. Therefore, this can be a fair evaluation for the proposed method. The other reason is to oppose the idea that nsF5 is designed based on it: 'fewer changes in JPEG coefficients lead to less detectability'.

A usual way for comparing the security of steganography methods is to compare the classification error for different embedded payloads by the methods. In order to compare the security of the proposed method with nsF5, the detection error of the two methods for particular payloads were obtained. In the first experiment, JPEG rich models (JRM) [5] [6] with dimension 22,510 – which is the state of art feature set for steganalysis in JPEG domain – was used, as well as ensemble classifier[6] – the state of the art classifier for steganalysis – [18]. The classifier was used with default settings in order to find optimum values for its parameters automatically. Half of the images were selected randomly, as the covers for training set and the remaining half were used as the covers for testing set. For each embedding payload rate, all the images in each set were embedded using the both methods. The resulted images were used as stegos. Therefore, for each classification there were 10,000 images in each one of training and testing sets. The testing error was calculated as (1).

$$Testing\ Error = \frac{False\ Alarm + Miss\ Detection}{Number\ of\ testing\ samples} \qquad (1)$$

In order to embed images using nsF5, Fridrich's nsF5 simulator[7] was used. For having a fair comparison and prevent steganalysis from detecting JPEG double compression which occurs in nsF5 simulator, the following operations were performed: first, arbitrary JPEG images with the quality of 100% were written. Then using Matlab JPEG toolbox, the coefficients planes of images were replaced with the ones obtained from C# implementation of DCT. These images were written as the covers. For constructing stegos, the same process was repeated except that before writing final images, some JPEG coefficients were changed using nsF5 simulator.

Fig. 8 shows the results of steganalysis using the proposed method (MSET) and nsF5.

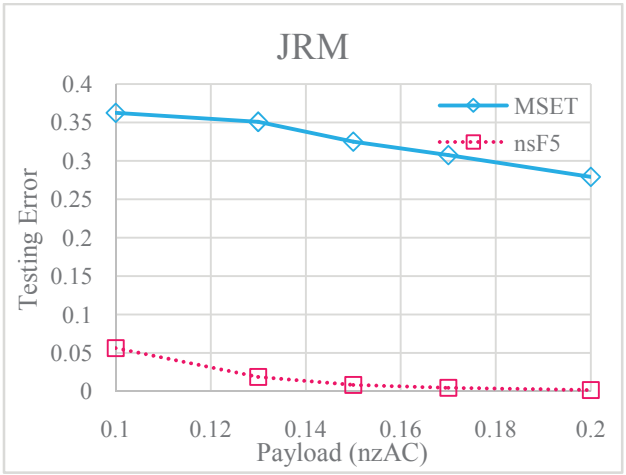

Figure 8. Comparision steganalysis of MSET and nsF5 using JRM feature set (JPEG Quality =100)

[5] http://dde.binghamton.edu/download/feature_extractors/download/cJRM.m

[6] http://dde.binghamton.edu/download/ensemble/ensemble_2.0.zip

[7] http://dde.binghamton.edu/download/nsf5simulator/nsf5_simulation.zip

Authorized licensed use limited to: Fachhochschule Nordwestschweiz. Downloaded on June 30,2020 at 11:42:28 UTC from IEEE Xplore. Restrictions apply.

Fig. 8 shows that the testing error related to MSET is always more than nsF5, which means in JPEG domain steganalysis, MSET is more secure. This is because MSET unlike nsF5, does not modify JPEG coefficients directly. It seems that modifying pixels will solely preserve correlations between resulted JPEG coefficients and thus the detection of stegos will be more difficult in comparison with the cases where the coefficients are modified directly.

Study on the images are embedded using MSET indicates that MSET leads to more changes in coefficients than nsF5. The mean numbers of changed coefficients using MSET and nsF5 for 1000 randomly selected images are presented in Table. 1. The results show that fewer changes in JPEG coefficients will not necessarily lead to less detectability. This contradicts the idea that nsF5 and some other methods are based on it.

The two columns in the right side of Table. 1 present PSNR[8] between the stegos and the compressed images without embedded data (JPG), and between the stegos and the original images (PGM).

A recent research upon the best steganalysis domain for JPEG images [19] indicates that the best steganalysis domain is not necessarily the domain in which modifications are made. Although it is shown that for nsF5 the best steganalysis domain is JPEG domain [19], the best steganalysis domain for the proposed method might not be JPEG domain. Therefore to make the comparison more fair, in the second experiment SRMQ1 features[9] [20] – a state of art spatial feature set with dimension 12,753 – were added to JRM features. The dimensionality of final feature set was 35,263. Fig. 9 shows steganalysis of MSET and nsF5 using SRMQ1+JRM.

According to Fig. 9, adding spatial domain features will help steganalysis detect MSET. It is because changes made in spatial domain in order to form the required pattern in coefficients also affect the values of the other pixels following JPEG decompression. The main reason is that MSET does not search for 'best matched pattern' with the minimum so called

TABLE I. AVERAGE CHANGES IN JPEG COEFFICIENTS AND PSNR

Payload (nzAC)	Method	Changes in JPEG Coefficients	PSNR (JPG)	PSNR (PGM)
0.1	MSET	26,637	56.6543	58.2297
	nsF5	2,396	62.5572	58.2260
0.13	MSET	28,748	56.4073	58.1082
	nsF5	3,313	61.3702	57.9816
0.15	MSET	30,494	56.2432	57.9882
	nsF5	3,968	60.7343	57.8173
0.17	MSET	32,148	56.0997	57.8696
	nsF5	4,651	60.1886	57.6534
0.2	MSET	34,153	55.9354	57.7275
	nsF5	5,737	59.4887	57.4084

[8] Peak Signal-to-Noise Ratio

[9] http://dde.binghamton.edu/download/feature_extractors/download/SRMQ1_windows_vc_2010_v1.1.zip

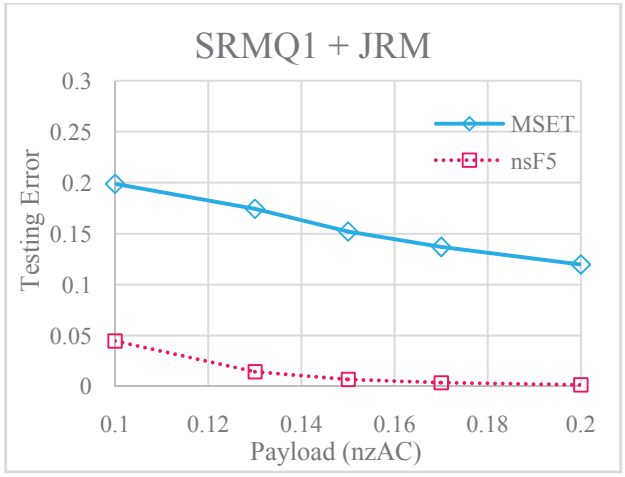

Figure 9. Comparison steganalysis of MSET and nsF5 using SRMQ1+JRM features (JPEG Quality =100)

"after-decompression distortion"; instead, it only searches for the 'first matched pattern'; however, MSET is clearly more secure than nsF5. Acquiring the 'best match' will make MSET more powerful and will be studied in the future works.

V. CONCLUSION AND FUTURE WORKS

As JPEG is the most common format for storage and transmitting images, it is a good target for steganography. Most of JPEG steganography methods embed data by applying direct modifications in JPEG coefficient. In this paper, a new JPEG steganography method was proposed in which the modifications are made in spatial domain, but the extraction of data is from JPEG domain. The proposed method is termed MSET and in this method, the data is embedded by applying intentional changes in spatial domain to make a specific pattern in equivalent JPEG coefficients. In MSET, the need for steganography key is removed. Although having a key in MSET is possible, it will not affect the steganalysis of MSET. In order to evaluate MSET, it was compared with nsF5; because neither of them uses a distortion function. The evaluation and comparison was done using two state of the art steganalysis feature sets (i.e., JRM with dimension 22,510 and its combination with SRMQ1, resulting in dimension 35,263) as well as the ensemble classifier which is deemed to be the most powerful classifier for steganalysis [18]. In each classification, 10,000 images were used for each of training and testing sets. The experimental results presented that MSET leads to more changes in JPEG coefficients than nsF5 and it is yet more secure than nsF5. It can be concluded that minimizing changes in JPEG coefficient will not necessarily lead to more secure steganographic methods.

In MSET, increasing the length of embedding data leads to exponential increment of time. The efforts to decrease embedding time will be placed in the future works. Experimental results also showed that considering spatial domain features in steganalysis of MSET result in more detectability of MSET. It is because MSET searches for the 'first match', not the 'best match'. The future works also will include the efforts to find the 'best match'. A proper solution

Authorized licensed use limited to: Fachhochschule Nordwestschweiz. Downloaded on June 30,2020 at 11:42:28 UTC from IEEE Xplore. Restrictions apply.

for finding the ‘best match’ probably will lead to the decrement of embedding time, as well.

REFERENCES

[1] A. Pfitzmann and M. Hansen, "A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management," 2010. [Online]. Available: http://dud.inf.tu-dresden.de/Anon_Terminology.shtml. [Accessed 28 April 2015].

[2] R. Böhme, Advanced statistical steganalysis, Springer, 2010, p. 13.

[3] L. Guo, J. Ni and Y. Q. Shi, "An efficient JPEG steganographic scheme using uniform embedding," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2012.

[4] J. Kodovský and J. Fridrich, "Calibration Revisited," in *11th ACM workshop on Multimedia and security*, New York, NY, 2009.

[5] T. Pevny and J. Fridrich, "Merging Markov and DCT features for multi-class JPEG steganalysis," in *Security, Steganography, and Watermarking of Multimedia Contents IX*, San Jose, 2007.

[6] J. Kodovský and J. Fridrich, "Steganalysis of JPEG images using rich models," in *Media Watermarking, Security, and Forensics*, Burlingame, California, USA, 2012.

[7] T. Filler, J. Judas and J. Fridrich, "Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes," *IEEE Transactions on Information Forensics and Security,* vol. 6, no. 3, pp. 920-935, 2011.

[8] P. Sallee, "Model-Based Steganography," in *Digital Watermarking*, 2004.

[9] A. Westfeld, "F5—A Steganographic Algorithm," *Information Hiding,* vol. 2137, pp. 289-302, 2001.

[10] J. Fridrich, T. Pevný and J. Kodovský, "Statistically undetectable jpeg steganography: dead ends challenges, and opportunities," in *9th workshop on Multimedia & security*, New York, NY, 2007.

[11] J. Fridrich, M. Goljan and D. Soukal, "Wet paper codes with improved embedding efficiency," in *Security, Steganography, and Watermarking of Multimedia Contents VIII*, San Jose, CA, 2006.

[12] J. Fridrich, M. Goljan and D. Soukal, "Perturbed quantization steganography with Wet Paper Codes," *Multimedia Systems,* vol. 11, no. 2, pp. 98-107, 14 October 2005.

[13] Y. Kim, Z. Duric and D. Richards, "Modified Matrix Encoding Technique for Minimal Distortion Steganography," *Information Hiding,* vol. 4437, pp. 314-327, 2007.

[14] D. Schönfeld and A. Winkler, "Embedding with Syndrome Coding Based on BCH Codes," in *8th workshop on Multimedia and security*, New York, NY, USA, 2006.

[15] V. Sachnev and H. J. Kim, "Modified BCH data hiding scheme for JPEG steganography," *EURASIP Journal on Advances in Signal Processing,* vol. 1, pp. 1-10, 2012.

[16] V. Holub , J. Fridrich and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security,* vol. 1, pp. 1-13, January 2014.

[17] L. Guo, J. Ni and Y. Q. Shi, "Uniform Embedding for Efficient JPEG Steganography," *IEEE Transactions on Information Forensics and Security,* vol. 9, no. 5, pp. 814-825, 2014.

[18] J. Kodovský and J. Fridrich, "Ensemble Classifiers for Steganalysis of Digital Media," *IEEE Transactions on Information Forensics and Security,* vol. 7, no. 2, pp. 432 - 444, 2012.

[19] V. Holub and J. Fridrich, "Challenging the Doctrines of JPEG Steganography," in *SPIE 9028, Media Watermarking, Security, and Forensics*, San Francisco, California, USA, 2014.

[20] J. Fridrich and J. Kodovsky, "Rich Models for Steganalysis of Digital Images," *Information Forensics and Security,* vol. 7, no. 3, pp. 868-882, 2012.

Authorized licensed use limited to: Fachhochschule Nordwestschweiz. Downloaded on June 30,2020 at 11:42:28 UTC from IEEE Xplore. Restrictions apply.