Multi-layer architecture for efficient steganalysis of Undermp3cover in multi-encoder scenario

Hamzeh Ghasemzadeh
Michigan State University
Email: ghasemza@msu.edu

Abstract:

Mp3 is a very popular audio format and hence it can be a good host for carrying hidden messages. In fact, different steganography methods have been proposed for mp3. But, only steganalysis of mp3stego has been conducted in the literature. In this paper we mention some limitations of mp3stego and argue that Undermp3cover (Ump3c) does not have them. Then, steganalysis of Ump3c is conducted based on measuring mutual information between global gain and other fields of mp3 bit stream. Furthermore, we show that different mp3 encoders have quite different performances and therefore we propose a novel multi-layer architecture for its steganalysis. In this manner, the first layer detects the encoder and the second layer performs the steganalysis job. One of advantages of this architecture is that feature extraction and feature selection can be optimized per encoder. We show this architecture outperforms the conventional single-layer methods. Comparing results of the proposed method with other works show an improvement of 20.4% in accuracy of multi-encoder scenario.

1. Introduction:

Secret communications has always been of interest to both individuals and states. This need has been addressed differently throughout the history. Existing methods can be classified into cryptography and steganography. Cryptography turns the message into unintelligible data and steganography conceals the existence of the message by hiding it inside another data. There are some shortcomings associated with cryptography alone. For example, it cannot prevent traffic analysis [1] and important information regarding pattern of communications, its durations, intended recipient and much more can be acquired. Also, exchange of encrypted messages could rise a red flag and make the adversary even more suspicious. Steganography can address these by adding an extra layer of protection. Steganalysis is the opposite of steganography and it tries to break steganography and to detect the presence of hidden messages. Also, it provides insight into steganography and help to develop better steganography methods. Audios are common multimedia signals and hence they are good candidates for cover. In fact different audio steganography methods have been proposed [2]. Also, steganalysis community has presented effective analysis for them. Those methods can be divided into non-compressed and compressed domain. Both of these methods were reviewed recently and comparative studies between them were conducted [3].

First, audio quality metrics (AQM) were used for differentiating between covers and stegos [4]. Another work argued that AQM are not suitable for steganalysis and instead used Hausdorff distance [5]. Work of [6] showed that features extracted from second order derivative of signals are more significance. Ghasemzadeh et al. argued that by definition human auditory system (HAS) should be insensitive to steganography noise and therefore models of HAS are not suitable for steganalysis [7]. They proposed a new scale that had the inverse frequency resolution of HAS for feature extraction. Finally, effects of different data hiding algorithms on different bit-planes were investigated [8]. Based on this observation a universal stego-based calibration was proposed.

In the compressed domain most works are devoted to analysis of mp3stego algorithm [3]. Mp3stego hides the message during compression itself [9]. Westfeld showed that the variance of block size of stego and cover are different [10]. Later, an ultra-lightweight system for real time applications such as integrating steganalysis into intrusion detection systems [11] was proposed. Yan et al. noticed that bit reservoirs in stegos and covers have different characteristics [12]. Work of [13] argued that taking difference between quantization step of consecutive granules improves significance of steganalysis features. Another work showed that quantization step is a band limited signal and therefore low pass filtering may be used for calibration [14]. Additionally, it was shown that different mp3 encoders have divergent behaviors which may affect results of steganalysis. To address this, steganalysis features were augmented with encoder classification features. Detecting traces of data hiding in the modified cosine transform (MDCT) is another approach. Generalized Gaussian distribution and statistical measurements of second order derivative of MDCTs were used in [15]. Finally, difference of absolute values of MDCTs across different channels were exploited for steganalysis [16].

Mp3stego has found a lot of attention from steganalysis community but it has two major shortcomings. First, cover should be in non-compressed format. Regarding the problem with using de-compressed signal as cover [17] applicability of mp3stego is only limited to never-compressed signals. Second, mp3stego is designed on top of 8Hz encoder and this information benefits steganalysis [14]. For example, if we determine an mp3 files has not been encoded with 8Hz, we could conclude that it has not been embedded with mp3stego. On the other hand Ump3c works directly with mp3 samples and therefore, it does not have
problems of mp3stego. Unfortunately, steganalysis of Ump3c has not been investigated properly. One of reasons is that the existing implementation is written for Linux and running it on windows is challenging. To the best of our knowledge, the only work on steganalysis of Ump3c is a modified version of regular-singular (RS) method [18]. In the RS method samples of signal are grouped and then effect of applying a flipping function on noise of each group is studied [19].

There are some important aspects of Ump3c that existing paper has not addressed. For example, RS steganalysis was proposed for image and it may not be as efficient for audio signals. Also, there are some issues with work of [18] which may affect it generalization. First, only 200 files were used for evaluation which limits external validity of the work. Second, embedded message were text files. It is known that distribution of bits 1 and 0 are not the same in text files. This change distribution of the signal and could bias result of steganalysis. Third, the work did not consider the existence of different encoders which could play a major role in the performance of the system.

This paper tries to fill these gaps. First, analysis of Ump3c is conducted and two problems with existing implementation is solved. Second, a new set of features based on mutual information (MI) between global gain (GG) and other fields of mp3 is proposed. Potency of these features are further improved by proposing the multiple re-embedding calibration. Third, effect of different encoders on distribution of GG is investigated and it is shown that performance of steganalysis varies between encoders. Finally, to account for different encoders a multi-layer system is proposed. This system has the unique advantage that features extraction and selection could be optimized for each encoder. Through different simulations we show that this novel architecture outperforms the conventional single layer systems.

The rest of this paper is organized as follows. Section 2 introduces different fields of mp3 bit stream and presents analysis of Ump3c. Section 3 is devoted to the proposed method and its analysis. Simulation results are shown in section 4. After discussing the proposed method in section 5, the paper concludes in section 6.

2. Mp3 bit stream and Ump3c

2.1. Mp3 bit stream and its analysis

Mp3 algorithm is among the most popular audio formats that provides high quality sound for a compression rate of 90%. Mp3 achieves this through a combination of different techniques including perceptual coding, non-uniform quantization, and Huffman coding. The compression algorithm works as follows [20]. Signal is framed into chunks of 1152 samples. After transforming each frame into frequency domain, its psycho-acoustic model is constructed. Later, this model is used to determine which parts of the signal are inaudible and how much quantization noise is tolerable in each frequency region. Separately, each frame is divided in two equal granules with 576 samples. Then, each granule is transformed into frequency domain through polyphase filter bank and MDCT operation. Now, MDCT coefficients are quantized in a nested loop. In this fashion, the inner loop adjusts the value of quantization step and makes sure that the existing bit budget is enough for storing the quantized MDCTs. On the other hand, the outer loop compares the quantization noise with psycho-acoustic model of current frame and makes sure that it is below the masking threshold and hence inaudible. Finally, quantized MDCTs with their corresponding side information (SI) are written in the bit stream, where, SI is the part of mp3 bit stream that stores parameters of encoder for each granule and there it is vital for correct decoding of mp3. SI has different fields and the relevant ones are described as follows.

One of the most important part of SI is GG and it stores information regarding the value of quantization step of each granule. Decoder uses this information for dequantization of samples and reconstruction of uncompressed signal.

Typically, audio signals have a wide dynamic range covering both fast transient and smooth portions. Mp3 standard has accounted this phenomenon by defining different block types. Psycho-acoustic model of each frame determines the degree of its stationarity and selects the suitable block type accordingly [20]. In this fashion, the best trade-off between time and frequency domain can be achieved. Each granule has a set of fields in SI that determines which block types were used during its encoding.

After quantization, MDCTs are encoded into bits with Huffman tables. Mp3 standard has defined different tables for different frequency regions. In fact there are three table select fields per granule in SI. Our analysis have shown that the first field is more informative so this work only uses the first table select field.

Typically, frames of audio signals have different complexities. That is, some frames may need fewer bits for reconstruction whereas other may need more. Mp3 standard benefits from a mechanism known as bit reservoir to store extra bits from encoding of simple frames for more complex ones. In this fashion, overall bit rate is held constant but length of each individual frame could vary. Part2_3_length is the field that stores length of each granule.

2.2 Analysis of Ump3c data hiding algorithm
Ump3c is a data hiding algorithm that works directly with mp3 bit stream [21], therefore, it doesn’t have limitations associated with mp3stego. Data hiding is achieved through LSB embedding in the GG of SI. It also uses the parameter of bit spacing for selecting which granules should be selected for data embedding. For example, if bit spacing=2 every second granule is used for data hiding. Maximum capacity of Ump3c is 1bit/granule for bit spacing=1. There are about 38.28 frames in every second of mp3, so, the maximum capacity of Ump3c is 76.56 bit/s.

Our investigation showed some problems with current implementation of Ump3c that could bias results of steganalysis. First, message is not encrypted before embedding, this could improve steganalysis of non-uniform messages. In order to show it, an experiment was conducted. 100 covers were selected randomly and then they were embedded with three different types of random messages. Probability of bit zero was 25%, 50%, and 75% in the first, second, and third case. Then, probability of GG of each case was calculated separately. Fig. 1 shows the result.

![Fig 1. Distribution of global gain of covers vs. stegos](image)

Referring to fig. 1 it is evident that GG of stegos embedded with message type 1 and 3 have quite different distributions than covers. More specifically, probability of odd values are higher (lower) than even values in category 1 (category 3). Also, it is quite evident that in category 2 which correspond to uniform random message, distributions of GG of cover and stego are quite alike, which makes its steganalysis real challenging. It is known that bit stream of encryption systems have uniform distribution, so encrypting messages prior to embedding would make steganalysis harder and add to security of Ump3c.

Second, Ump3c select granules consecutively. This leads to following characteristic, no matter what embedding capacity is used, granules of first frames are always embedded with message. Steganalysis system could use this information and only extract features from specific frames. Previously, we showed that this characteristic could bias results of steganalysis system [14].

Based on these arguments and in order to address the problem of running Ump3c in the windows, Ump3c was simulated in the Matlab. This new version XOR the message with a pseudo-random sequence and also selects embedding granules randomly. It is noteworthy that XORing the message with a pseudo-random sequence is not totally secure from cryptography point of view, but it is lightweight, already available in Matlab, and output sequence has a uniform distribution. This new implementation was used for the rest of simulations.

3. The proposed method

3.1 Mutual information

GG is one of the most important parameters of mp3 and in fact when inner loop increases value of GG many other parts of bit stream change as well. More specifically, using larger value of GG decreases the number of bits that are assigned to the granule. Also, it can affect indices of Huffman tables that are used, characteristics of bit reservoir, and MDCTs. In Ump3c only GG is changed (very subtly) and all other parts of bit stream are kept intact. Therefore, statistical features from other parts of mp3 does not have any steganalysis relevance. Also, referring to fig. 1 we see that for uniform messages GG of covers and stegos have the same distribution. So, statistical measures of GG may not produce powerful features. We hypothesize that there would be some deviations between joint probability of GG and other parts of bit stream. But, if joint probability is used with 15 bins per each variable and four different fields of bit stream are used the total number of features would add up to 900. In order to tackle this, we have used another approach for capturing traces of embedding.
Mutual information (MI) is a useful metric for measuring the amount of information that we can acquire about a random variable \(X\), by observing another random variable \(Y\). Let \(p(x,y)\) denote the joint probability of variables \(X\) and \(Y\) and \(p(x)\) and \(p(y)\) denote their marginal distributions. MI between \(X\) and \(Y\) is denoted by \(I(X;Y)\) and for discrete random variables is defined as:

\[
I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \cdot \log \left( \frac{p(x,y)}{p(x)p(y)} \right)
\]

(1)

**3.2 The proposed features**

Based on arguments of previous section we have used MI between GG and other fields of mp3 bit stream to detect trace of hidden messages. Let \(G_{i,j}\) denote GG of granule \(i (i \in \{1,2\})\) of frame \(j (1 \leq j \leq N)\). Also, let \(g_k, b_k, p_k, t_k\) denote GG, block type, part_2_3_length, and table select of mp3 bit stream when they are arranged consecutively \((1 \leq k \leq 2N)\). That is:

\[
g_k = [G_{1,1}, G_{2,1}, G_{1,2}, G_{2,2}, ..., G_{1,N}, G_{2,N}]
\]

(2)

Our proposed features can be categorized into three classes. First set contains self-MI of GG and they include:

\[
F_1 = [I(G_{1,1;N}; G_{2,1;N}), I(G_{2,1;N}; G_{1,1;N}), I(G_{1,1;N}; G_{1,2;N}), I(G_{2,1;N}; G_{2,2;N})]
\]

(3)

Second set of features re MI between GG and other fields of mp3 and they include:

\[
F_2 = [I(G_{1,1;N}; b_{1;2N}), I(G_{1,1;N}; p_{1;2N}), I(G_{1,1;N}; t_{1;2N})]
\]

(4)

Our analysis showed that tail of distribution of GG in some stegos are different from their covers. In order to capture those instances, third set of features were calculated as maximum and minimum of \(G_{1,1;N}\) and \(G_{2,1;N}\).

**3.3 Multiple re-embedding calibration**

Finding a set of features that are independent from the content of signal and only reflect the presence/absence of hidden messages is very hard. Steganalysis methods address this through calibration. Estimating the cover through noise removal [4], down-sampling [22], low-pass filtering [14], and desynchronizing jpeg blocks [23] are some of those approaches. Another possibility is, re-embedding which estimates characteristics of stegos [8]. We extend the idea of re-embedding and propose multiple re-embedding for feature calibration.

Let \(x(t), A_{em}, m(t)\), and \(\mathcal{F}\) denote a signal, the embedding algorithm that we are trying to detect, a random message, and the feature extraction procedure. The re-embedded calibrated features are defined as difference between features of signals \(x(t)\) and its re-embedded version with the message \(m(t)\) [8]. The calibrated features vector \(\mathbf{F}\) is:

\[
\mathbf{F} = \mathcal{F}(x(t)) - \mathcal{F}(A_{em}(x(t), m(t)))
\]

(5)

Because \(m(t)\) is a random signal, \(\mathbf{F}\) is a random vector which has a certain distribution. Multiple re-embedded calibrated features are defined as statistical moments of distribution of vector \(\mathbf{F}\). In this work we used mean and std of \(\mathbf{F}\). Assuming \(\mathbf{F}\) has \(l\) features and it is re-embedded \(n\)-times, figure 2 shows the proposed feature extraction method.

![Multiple re-embedding calibrated features](image)

**3.4 Effect of different encoders**

Previous works have shown that different fields of mp3 have dissimilar distributions [14, 24] and work of [14] showed that almost all aspects of different encoders have quite divergent behaviors. The work was able to classify different encoders with only 4 features. Regarding steganalysis of Ump3c, variation between GG of different encoders could play a major role in the performance of system. Fig. 3 shows distribution of GG of some different encoders.
Based on fig 3, we see different encoders have dissimilar GG distributions. As we show later in the simulation, if this characteristic is not addressed properly, performance of the system could degrade significantly. One possible solution is to add encoder classification features to steganalysis features (single-layer approach) [14]. We argue that for steganalysis of Ump3c this is not a good choice.

First, we need to find \( p(x,y) \) before calculation of \( I(X;Y) \). Referring to fig 3, GG has dissimilar distributions for different encoders. Because in the single-layer approach all the files should follow the exact same routine for feature extraction we should find a binning mechanism that is suitable for all encoders. If we use large number of bins we could measure subtle changes in \( p(x,y) \) of all encoders but we needs much more samples to fill the bins. On the other hand, if we use low number of bins fewer samples are needed but we lose precision. Second, due to divergent behaviors of different encoders it is quite possible that different subsets of features are optimum for each encoder. But, in the single-layer approach we should find a sub-set of features that works fine for all encoders which it may not be optimum for all encoders.

In order to address these problems, we propose a novel multi-layer architecture where, in the first layer encoders are classified and then in the second layer the actual steganalysis is carried out. The nice thing about this approach is that, steganalysis features could vary for different encoders. Furthermore, feature selection can be done per each encoder and hence the optimum sub-set of features could be used. Fig. 4 shows the proposed multi-layer architecture.

4. Experimental results

4.1 Experiment setup

For generating the database we used 27 audio disks. In this manner we made sure that our excerpts were never compressed before. After splitting all audio tracks into 30 seconds clips, we arrived at 2249 samples. To construct our covers, all samples were then compressed with ten widely used mp3 encoders including 8Hz, Audition, Blade, Fastenc, Gogo, Jetaudio, L3ENC, Lame, Plugger, and Xing. Then, each cover was embedded at 100%, 75%, and 50% of the maximum embedding capacity with different random messages. Therefore, the final database consisted 89960 files.
For encoder classification we used the 4 features proposed in [14]. Also, in multi-layer method (fig. 4) support vector machine (SVM) in one against one (One-One) arrangement with linear kernel was used to achieve the best performance [14]. All of other classification were implemented with binary SVM with radial basis function (RBF) kernel. All tests were carried out with 10-folds cross validation. Also, previous works have shown that normalizing features improves performance of classification [25]. Therefore, for each feature, values of mean ($\mu$) and std ($\sigma$) over all training samples were calculated and then features from both training and testing sets were normalized according to (6):

$$\bar{x}_k = (x_k - \mu_k) / \sigma_k$$

Previous works have compared performance of different feature selections and have shown that genetic algorithm (GA) achieves the best results [25]. Therefore, best feature were selected with help of GA. Details of our GA are as follows. Every generation had 200 individuals and the fitness function was accuracy of classifier. Two point cross-over [26], random replacement, and tournament selection were used for cross-over, mutation, and selection operations, respectively. Finally, in all of simulations steganalysis features are combination of sets 1 and 2, unless otherwise specified.

### 4.2 optimizing bins of histogram

Finding joint probability is the prerequisite of feature sets 1 and 2. Our analysis showed that using different bins for GG, affects significance of these sets considerably. Two different cases are compared. In method 1, maximum and minimum of data were extracted and then their distance was divided into 30 equal bins. In the method 2 we had a bin for every possible value of GG. Then, difference between distributions of GG of covers and stegos were measured and bins with difference larger than $10^{-5}$ were used for feature extraction. Accuracy of these method are compared in table II.

| Table I. Performance of different binning strategies |
|------------------------------------------------------|
| **Encoders** | **Method 1 (Embedding Capacity)** | **Method 2 (Embedding Capacity)** |
| | 100 | 75 | 50 | | 100 | 75 | 50 |
| 8Hz | 87.6 | 81.8 | 73.9 | | 90.4 | 85.9 | 78 |
| Audition | 85.9 | 81.3 | 73.1 | | 88.4 | 83.9 | 75.5 |
| Blade | 82.9 | 78.8 | 70.4 | | 86.8 | 81.6 | 74 |
| Fastenc | 83.3 | 79 | 71.4 | | 85.3 | 79.7 | 72.7 |
| Gogo | 81.2 | 76.5 | 68.3 | | 85.9 | 80.2 | 73.5 |
| Jetaudio | 62.9 | 59.7 | 56.4 | | 63.3 | 60.1 | 56.9 |
| L3ENC | 63.2 | 60.6 | 56.9 | | 66.6 | 62.8 | 60.2 |
| Lame | 54.5 | 54.9 | 53.4 | | 63.1 | 60.4 | 57.4 |
| Plugger | 95.6 | 93 | 86.1 | | 91.8 | 87.5 | 81.4 |
| Xing | 65.5 | 63 | 59 | | 67.1 | 62.5 | 59.9 |

Table I shows that improvement as high as 8.6% is achieved when method 2 is used. Therefore, for the rest of simulation method 2 is used.

### 4.3 Optimizing parameters of calibration

The proposed calibration method has two important parameters of re-embedding capacity and re-embedding order ($n$). A simulation was conducted to measure effect of different re-embedding capacities on performance of the system. Table II reflects the results.

| Table II. Effect of re-embedding capacity ($C_k$) on accuracy |
|-------------------------------------------------------------|
| **Encoders** | **Re-embedding capacity** |
| | $C_k = 100$ | $C_k = 75$ | $C_k = 50$ |
| 8Hz | 90.4 | 85.9 | 87.9 | 83.3 |
| Audition | 88.4 | 83.9 | 85.1 | 79.9 |
| Blade | 86.8 | 81.6 | 83.3 | 78 |
| Fastenc | 85.3 | 79.7 | 82.9 | 77.5 |
| Gogo | 85.9 | 80.2 | 84.6 | 78.5 |
| Jetaudio | 63.3 | 60.1 | 61.8 | 57.8 |
| L3ENC | 66.6 | 62.8 | 66 | 61.6 |
| Lame | 63.1 | 60.4 | 62.5 | 58.5 |
| Plugger | 91.8 | 87.5 | 90.2 | 84.8 |
| Xing | 67.1 | 62.5 | 65 | 61.4 |
Calculating mean accuracy of each category leads to 76.7%, 75.7%, and 74.1% for $C_R$ equal to 100%, 75%, and 50%, respectively. We see re-embedding with capacity of 100% leads to more discriminative features. Therefore, for the rest of simulations $C_R=100\%$ is used.

Effect of different re-embedding orders was measures. For a better presentation, average accuracy of all encoders for $1 \leq n \leq 10$ is plotted in fig. 5.

Based on fig. 5 average accuracy and re-embedding order are directly correlated. But, using higher re-embedding order leads to increase in the complexity of feature extraction. Therefore, we used order of 10 for the rest of simulations. Finally, our further investigation showed that the improvement is dissimilar for different encoders. For example Plugger had the highest improvement (7%) and L3enc had the lowest (1%).

### 4.4 Calibrated vs. non-calibrated features

Effectiveness of calibration was tested. To that end, accuracy of classifier with different feature sets was measured. We tested three different cases. Non-calibrated case which contained feature sets 1 and 2. Calibrated case which contained calibrated version of feature sets 1 and 2. Finally the extended case was constructed with augmenting calibrated case with feature set 3. Table III reflects accuracy of these cases.

| Encoders | 100 | 75 | 100 | 75 | 100 | 75 |
|----------|-----|----|-----|----|-----|----|
| 8Hz      | 89.7| 85.5| 94.3| 90.8| 94.5| 90.9|
| Audition | 80.5| 75.4| 90.5| 85.7| 92.4| 87.5|
| Blade    | 86.7| 82  | 91.8| 87.5| 91.7| 87.2|
| Fastenc  | 78.1| 74  | 88.2| 84.3| 90.5| 86.1|
| Gogo     | 85.8| 80  | 91.1| 86.3| 92.3| 88  |
| Jetaudio | 64.9| 61.4| 67  | 63.1| 74.3| 69.8|
| L3ENC    | 61.2| 58.9| 67.4| 63.6| 67.7| 62.7|
| Lame     | 57.6| 55.5| 65.8| 61.9| 93.6| 88.3|
| Plugger  | 99.5| 98.8| 97.7| 95.7| 98.2| 96.4|
| Xing     | 68.1| 63.7| 69.1| 66  | 69.9| 65.7|

Referring to table III, we see improvement as large as 10.3% can be achieved when calibration is used. Furthermore, when extended set is used, some encoders (ex. Lame) shows large improvements. Therefore, for the rest of simulations extended feature set is used.

### 4.5 Comparison with previous methods

To the best of our knowledge the only work on steganalysis of Ump3c is RS method [18]. We used the estimated message length of that work as steganalysis features. Furthermore, both mp3stego and Ump3c change values of GG. Therefore, we analyzed methods published on steganalysis of mp3stego and found that methods of differential quantization step (DQS) [13] and calibrated quantization step (Cal-QS) [14] are also based on statistical analysis of GG. Therefore, they can be applied to Ump3c. Table IV compares accuracy of the proposed method with these works for different embedding capacities in the single-encoder scenario.
Table IV shows advantage of the proposed method over existing works. For example, comparing results of Plugger shows that improvement as large as 41.7% is achieved when the proposed method is used.

### 4.6 Performance in multi-encoder scenario

Previous sections assumed single-encoder scenario. That is, the system was trained and tested separately for each encoder. But, in the real world mp3 files are from a mixture of different encoders. In order to test this scenario a new database was generated where for each file one of encoders was selected randomly. Three different scenarios for the proposed method were implemented and test. In the first scenario, extended feature set were used in a single layer architecture. As we discussed in section 3.4 in single layer scenario we cannot optimize binning and feature selection for each encoder. Therefore, we used union of bins of all encoders. Second scenario was basically the first scenario but steganalysis features were augmented with encoder classification features form [14]. Finally, the third scenario was implementation of multi-layer architecture as presented in fig. 4. Comparison between sensitivity (Se.) and specificity (Sp.) of these scenario with previous works are presented in table V.

### Table V. PERFORMANCE IN MULTI-ENCODER SCENARIO

| Method   | Embedding Capacity | 100 | 75  | 50  | 100 | 75  | 50  | 100 | 75  | 50  |
|----------|--------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|          |                    | Sp. | Se. | Sp. | Se. | Sp. | Se. | Sp. | Se. | Sp. |
| RS       |                    | 46.1| 73.2| 40.9| 74.9| 44.4| 66.5|
| DQS      |                    | 47.2| 80  | 44.3| 77.2| 41.2| 73.2|
| Cal-QS   |                    | 52.3| 67.5| 46.8| 66.3| 44.7| 60.1|
| Scenario I|                  | 68.8| 82.5| 64.2| 80  | 56.7| 74.8|
| Scenario II|              | 72.5| 84.1| 68.2| 79.9| 58.6| 75.5|
| Scenario III|              | 82.4| 85.6| 79  | 79.7| 72.4| 73.7|

Finally, receiver operating characteristic (ROC) of these methods in multi-encoder scenario are presented in fig. 5.
5. Discussion

Referring to tables I through IV shows that accuracy of different encoders are quite dissimilar. From steganography point of view, we could say some encoders are more secure for data hiding. This is an interesting observation, because previous works have not investigated the effect of encoders on steganalysis. In order to track source of these differences we ran a set of analyzes on correlation between different parts of mp3 bit stream for different encoders. In one analysis we measured mean of correlation coefficient between GG and its delayed version over all covers for each encoder. Then we sorted different encoders based on two different criteria. In the first case, they were sorted from lowest to highest value of average of correlation coefficient. In the second case, they were sorted according to accuracy of steganalysis for calibrated feature (table III). Comparing two cases showed comparable orders. Therefore, dissimilar accuracy of different encoders is due to their intrinsic behavior. That is, encoders that have more predictable outcomes are easier to detect and vice versa.

Comparing results of table IV shows that accuracy of the proposed method is much higher than DQS which is the best existing method. This, shows that different fields of mp3 bit stream are highly dependent on each other. Therefore, if this information is exploited efficiently discriminative property of steganalysis features is improved significantly.

Referring to table V and fig. 5 another strong point of the proposed method becomes evident. Specifically, existing works have very low specificity in multi-encoder scenario. More valuable piece of information is inferred when we compare performance of three different proposed scenarios. Comparing results of scenarios I and II shows that when encoder features are added in single layer architecture performance doesn’t improve a lot. On the other hand, when results of scenarios I and III are compared improvement as large as 15.7% is achieved. Therefore, we conclude that the multi-layer architecture is more powerful than single-layer systems. It is noteworthy that application of the proposed multi-layer structure is not limited to this work. For example, better steganalysis system could be constructed in this manner where the first layer classifies the content (music vs. speech, different genres of music, different signal complexities, and etc.) and next layers do the steganalysis part. Also, a universal steganalysis system could be constructed in this way, where each layer uses a different set of features and detects trace of only certain embedding algorithms. Finally, the proposed multi-layer structure is a general framework and is applicable to other classification tasks. That is, the first layer categorizes the data (ex. gender in speech or lighting condition in image) and next layers do the actual job (ex. speech recognition in speech or face recognition in image).

Finally, complexity of the proposed multi-layer structure with single layer structure is compared. For multi-layer structure three different multi-class strategies were used, one against one SVM (One-One), one against all SVM (One-All), and tree classifier. We used the number of classifiers (Nc), training time, testing time, and accuracy for capacity of 100% for this purpose. The results for 10-fold cross validation are reported in table VI.
| Method            | Nc | T_Train | T_Test | Acc. |
|-------------------|----|---------|--------|------|
| Single-Layer      | 1  | 5.29    | 0.14   | 78.98 |
| Multi-One-One     | 55 | 47.3    | 3.62   | 83.59 |
| Multi-One-All     | 20 | 144     | 4.69   | 83.12 |
| Multi-Tree        | 11 | 2.40    | 1.39   | 83.56 |

Based on results of table VI we see that running times of One-One and tree multi-layer structures are quite acceptable. Therefore, the proposed method does not add any limitations in terms of running time.

6. Conclusion

Steganalysis of Ump3c in multi-encoder scenario was conducted in this paper. Analyzing embedding mechanism of Ump3c showed that only global gains of bit stream are changed. Therefore, we hypothesized by measuring joint probability between global gain and other parts of mp3 bit stream, suitable features could be extracted. In order to extract relationship between different parts of bit stream and keep steganalysis features to a minimum, we instead used mutual information. To further improve significance of features we introduced multiple re-embedding calibration. Finally, we proposed a novel multi-layer structure for tackling the problem of multi-encoder scenario. This new structure has the important advantage that feature extraction and feature selection can be optimized for each individual encoder.

References:

[1] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail," in Security and Privacy (SP), 2012 IEEE Symposium on, 2012, pp. 332-346.
[2] F. Djebbar, B. Ayad, K. A. Meraim, and H. Hamam, "Comparative study of digital audio steganography techniques," EURASIP Journal on Audio, Speech, and Music Processing, vol. 2012, p. 25, 2012.
[3] H. Ghasemzadeh and M. H. Kayvanrad, "A Comprehensive Review of Audio Steganalysis Methods," "arXiv preprint arXiv:1701.05611" , 2017.
[4] H. Ozer, I. Avci, B. Sankur, and N. D. Memon, "Steganalysis of audio based on audio quality metrics," in Proceedings of SPIE, 2003, pp. 55-66.
[5] Y. Liu, K. Chiung, C. Corbett, R. Archibald, B. Mukherjee, and D. Ghosal, "A novel audio steganalysis based on high-order statistics of a distortion measure with hausdorff distance," Lecture Notes in Computer Science, vol. 5222, p. 487, 2008.
[6] Q. Liu, A. H. Sung, and M. Qiao, "Temporal derivative-based spectrum and mel-cepstrum audio steganalysis," IEEE Transactions on Information Forensics and Security, vol. 4, pp. 359-368, 2009.
[7] H. Ghasemzadeh, M. T. Khass, and M. K. Arjmandi, "Audio steganalysis based on reversed psychoacoustic model of human hearing," Digital signal processing, vol. 51, pp. 133-141, 2016.
[8] H. Ghasemzadeh and M. K. Arjmandi, "Universal Audio Steganalysis Based on Calibration and Reversed Frequency Resolution of Human Auditory System," IET Signal Processing, 2017.
[9] F. Petitcolas. (1998, 09/18/2017). "DTE Security." (Online). https://sourceforge.net/projects/ump3c/.
[10] A. Westfeld, "Detecting low embedding rates," in International Workshop on Information Hiding, 2002, pp. 324-339.
[11] D. Yan, and R. Wang, "Detection of MP3Stego exploiting recompression calibration-based feature," Multimedia tools and applications, vol. 72, pp. 865-878, 2014.
[12] D. Yan, R. Wang, X. Yu, and J. Zhu, "Steganalysis for MP3Stego using differential statistics of quantization step," Digital Signal Processing, vol. 23, pp. 1181-1185, 2013.
[13] D. Yan, R. Wang, X. Yu, and J. Zhu, "Steganalysis for MP3Stego using differential statistics of quantization step," Digital Signal Processing, vol. 23, pp. 1181-1185, 2013.
[14] H. Ghasemzadeh, "Calibrated steganalysis of mp3stego in multi-encoder scenario," Under review (preprint available at arXiv:1709.08084), 2017.
[15] M. Qiao, A. H. Sung, and Q. Liu, "Feature mining and intelligent computing for MP3 steganalysis," in Bioinformatics, Systems Biology and Intelligent Computing, 2009. IJCB '09. International Joint Conference on, 2009, pp. 627-630.
[16] C. Jin, R. Wang, and D. Yan, "Steganalysis of MP3Stego with low embedding-rate using Markov feature," Multimedia Tools and Applications, vol. 76, pp. 6143-6158, 2017.
[17] J. Fridrich, M. Goljan, and R. Du, "Steganalysis based on JPEG compatibility," in Proc. SPIE, 2001, pp. 275-280.
[18] C. Jin, R. Wang, D. Yan, and X. Yu, "Steganalysis of UnderMP3Cover," J. Comput. Inf. Syst., vol. 8, pp. 10459-10468, 2012.
[19] J. Fridrich, M. Goljan, and R. Du, "Detecting LSB steganography in color, and gray-scale images," IEEE multimedia, vol. 8, pp. 22-28, 2001.
[20] R. Raisi, "The theory behind MP3," MP3 Tech, 2002.
[21] P. C. Under mp3 cover, Online available at https://sourceforge.net/projects/ump3c/.
[22] A. D. Ker, "Steganalysis of LSB matching in grayscale images," IEEE signal processing letters, vol. 12, pp. 441-444, 2005.
[23] J. Fridrich, M. Goljan, and D. Hogea, "Attacking the outguess," in Proceedings of the ACM Workshop on Multimedia and Security, 2002.
[24] R. Böhme and A. Westfeld, "Statistical characteristics of MP3 encoders for steganalysis," in Proceedings of the 2004 workshop on Multimedia and security, 2004, pp. 25-34.
[25] H. Ghasemzadeh, M. T. Khass, M. K. Arjmandi, and M. Pooyan, "Detection of vocal disorders based on phase space parameters and Lyapunov spectrum," biomedical signal processing and control, vol. 22, pp. 135-145, 2015.
[26] H. Ghasemzadeh, "A metaheuristic approach for solving jigsaw puzzles," in Intelligent Systems (ICIS), 2014 Iranian Conference on, 2014, pp. 1-6.