Steganalysis of Compressed Speech to Detect Covert VoIP Channels

Yongfeng Huang [1], Shanyu Tang [2], Chunlan Bao [1], Yau Jim Yip [3]

[1] Department of Electronic Engineering, Tsinghua University, Beijing 100084

[2] London Metropolitan University, London N7 8DB, UK

[3] University of Huddersfield, Huddersfield HD1 3DH, UK

Abstract

A network covert channel is a passage along which information leaks across the network in violation of security policy in a completely undetectable manner. This paper reveals our findings in analysing the principle of G.723.1 codec that there are 'unused' bits in G.723.1 encoded audio frames, which can be used to embed secret messages. A novel steganalysis method that employs the second detection and regression analysis is suggested in this study. The proposed method can detect the hidden message embedded in a compressed VoIP speech, but also accurately estimate the embedded message length. The method is based on the second statistics, i.e. doing a second steganography (embedding information in a sampled speech at an embedding rate followed by embedding another information at a different level of data embedding) in order to estimate the hidden message length. Experimental results have proven the effectiveness of the steganalysis method for detecting the covert channel in the compressed VoIP speech.

Keywords: Second steganography, Steganalysis, VoIP, Compressed speech

1. Introduction

Steganography is the art and science of hiding the very presence of covert communication by embedding secret messages in innocent-looking electronic signals such as digital images, video and audio. To achieve covert communication, stego-signals, which are signals containing secret messages, should be indistinguishable from cover signals not containing any secret message. On the contrary, steganalysis deals with the technique used to distinguish between stego-signals and cover signals [1].

Steganalysis is the science of detecting messages hidden using steganography. The goal of steganalysis is to distinguish stego objects (containing a secret message) from cover objects with little or no knowledge of steganographic algorithms. The simplest method to detect steganographically-encoded packages/files is to compare them to known originals. Comparing the package against the original file will yield the differences caused by encoding the payload – and, thus, the payload can be extracted. Nowadays, steganalysis becomes increasingly important in computer forensics, for tracking and screening documents/audios/videos that are suspect of criminal and terrorism activities, and for information security to prevent leakage of unauthorized data.

There has been quite some effort to study the steganalysis of digital images, and recent publications are [2][3][4][5][6]. In contrast to image steganography and steganalysis， audio steganography and steganalysis are largely unexplored. Westfeld and Pfitzmann proposed a steganalysis method for Least Significant Bit (LSB) based embedding and also addressed the steganalysis of the MP3 steganography algorithm [7].

Voice over IP (VoIP) enables the digitalisation, compression and transmission of analogue audio signals from a sender to a receiver using IP packets. As the size of the used network and the distance

between the communicating parties have little relevance to transmission, VoIP is used for worldwide telephony such as Skype. VoIP streams are dynamic chunks of a series of packets that consist of IP headers, UDP headers, RTP headers, and numbers of audio frames. Those headers and frames have a number of unused fields, providing plausible covert channels and thus giving scope for steganography.

With the upsurge of Voice over IP applications available for commercial use in recent years, VoIP becomes one of the most interesting cover media for information hiding. Several steganography methods have been suggested in the literature [8][9][10][11][12][13], and some of which are based on streaming media and their network protocols such as VoIP or IP, which are used to form network covert channels. The network covert channel is a passage along which information leaks across the network in violation of security policy in a completely undetectable manner.

Although some research work had managed to detect network protocols based covert channels [14][15][16][17][18][19], so far there are still few steganalysis methods available for the compressed VoIP speech. This is the reason that led us to propose this work in the first place. In this paper how steganalysis can be performed in VoIP applications and operational aspects are discussed. Furthermore, the paper focuses on introducing a novel steganalysis method for the low bit rate speech codec such as G.723.1 widely used in VoIP communications.

Given the wealth of statistic and information-theoretic tools, several approaches may be used to analyse problems like VoIP covert communications. One could study the capacity of the covert channel, and then analyse the probability of detection as a function of the embedding rate, which is defined as the ratio of the secret message length to the stego VoIP stream length. The other can measure the entropy of the covert channel and compare that to the entropy of a system without

embedding. Our approach to the problem is to utilise a statistical test in combination with doing a second steganography (i.e. embedding information in a sampled speech at an embedding rate followed by embedding another information at a different level of data embedding) so as to estimate the embedded message length.

The rest of this paper is organized as follows. In Section 2 the principle of the speech codec used in VoIP, such as G.723.1, is analysed. Section 3 details a new steganalysis method for compressed VoIP speech streams. The evaluation of the proposed steganalysis method is presented in Section 4. Finally, it ends with conclusions in Section 5.

2. VoIP Applications with Covert Channels

In general, the ITU-G.723.1 speech codec is widely used in VoIP communications for compressing the speech or audio signal component of streaming media. Close analysis of the principle of the G.723.1 codec shows the G.723.1 encoded frame (a short chunk of the speech signal) is composed of a number of speech parameters since the codec is based on Analysis-by-Synthesis (AbS) coding, one of the vocal code models. Differing from the G.711 codec, the G.723.1 codec has two bit rates associated with it, 5.3 kbps and 6.3 kbps. This study focuses on the high bit rate (i.e. 6.3 kb/s) because it gives better voice quality. The 6.3 kbps codec adopts Multi-pulse Maximum Likelihood Quantization (MP-MLQ) excitation, which is different from Algebraic Code Excited Linear Prediction (ACELP) used by the 5.3 kbps codec.

The G.723.1 encoder is performed on a frame-by-frame basis, and it operates on frames of 240 audio samples each, based on Pulse-code Modulation (PCM). First of all, each frame is filtered by a

high pass filter to remove the DC component and is then divided into four subframes of 60 samples each. A 10th order Linear Predictive Coding (LPC) filter is computed using the unprocessed input signal for every subframe, and the last subframe is quantized using a Predictive Split Vector Quantizer (PSVQ). For every two subframes (120 samples), the weighted speech signal is used to compute the open loop pitch period. A harmonic noise shaping filter is then constructed using the open loop pitch period computed previously, and a closed loop pitch predictor is constructed according to the impulse response created by the noise shaping filter. Finally, both the pitch period and the differential value are transmitted to the decoder and the non-periodic component of the excitation is approximated. After completion of these operations, all speech parameters such as LPC, Pulse sign and Pulse position and etc., are obtained. The bit allocation of the 6.3kb/s coding algorithm is listed in Table 1. There are no LPC parameters for subframes and each speech frame has a LPC value of 24 bits.

Table 1: The bit allocation of the 6.3kb/s coding algorithm

Parameters	Subframe 0	Subframe 1	Subframe 2	Subframe 3	Subtotal (bits)
Adaptive codebook lags (Olp / Aclg)	7	2	7	2	18
LPC index (Lsf)	-	-	-	-	24
Grid index (Grid)	1	1	1	1	4
All the gains combined (Mamp)	12	12	12	12	48
Pulse positions (Ppos)	20	18	20	18	73
Pulse signs	6	5	6	5	22
Total					189

Experiments were carried out to estimate the Capability of Noise Tolerance (CNT) for each parameter of the G.723.1 codec. First, a speech frame was compressed and encoded by G.723.1 encoder to form bit streams. The least significant bits of one parameter of the bit streams were substituted and then decoded to output a stego-speech. Similar experiments were repeated for the other parameters, and so the difference signal-to-noise-ratio (DSNR) between the original speech and the stego-speech was determined for each parameter.

All the speech parameters were sorted into three levels in terms of their DSNR values, and the results are listed in Table 2. The CNT of the parameter is defined as 'Level 1' if its DSNR is less than 1dB. A detailed analysis of the experimental results reveals that there is much difference in CNT between different speech parameters. For example, the parameter, Ppos, has higher Capability of Noise Tolerance ('level 1') than the other parameters such as Olp with 'level 3' in some bits.

Table 2: Capability of Noise Tolerance (CNT) of G.723.1 speech parameters

Number of bit	Olp (s1)	Lsf (s2)	Aclg (s3)	Grid (s4)	Mamp (s5)	Ppos (s6)
7		Level 3				Level 1
6	Level 3	Level 3				Level 1
5	Level 3	Level 3				Level 1
4	Level 3	Level 2			Level 3	Level 1
3	Level 2	Level 2			Level 3	Level 1
2	Level 2	Level 2			Level 2	Level 1
1	Level 1	Level 1	Level 2		Level 2	Level 1
0	Level 1	Level 1	Level 2	Level 1	Level 1	Level 1

The goal of VoIP steganography is to embed a secret binary message in the compressed VoIP speech streams that consist of a series of packets with each carrying a certain number of audio frames. An effective LSB steganographic algorithm for VoIP communications using G.723.1 and G.729a codecs was suggested in our previous papers [20][21]. According to the steganographic algorithm, the Least Significant Bits (LSBs) of some parameters of the G.723.1 codec can be replaced with secret messages. The LSBs of parameters are those bits whose CNT levels are identified as 'level 1' in Table 2. In other words, the cover objects used for embedding secret messages are the LSBs of the parameters such as Olp, Lsf, Grid, Mamp, and Ppos, denoted by $S = \{s1, s2, s4, s5, s6\}$, where $s1$ denotes Olp, $s2$ denotes Lsf, $s4$ denotes Grid, $s5$ denotes Mamp, and $s6$ denotes Ppos.

Steganalysis, the official countermeasure to steganography, is the science of detecting and often decoding the hidden information within the cover medium. In contrast to the LSB based steganographic algorithm, the steganalysis of VoIP is to determine whether secret information is embedded in the LSBs of G.723.1 encoded VoIP speeches. Having investigated the principle of the G723.1 codec and the CNT characteristics of their speech parameters, we suggest a novel steganalysis method in this study, which is based on second statistical detection and regression analysis. The proposed method does not only detect the hidden information embedded in the compressed VoIP speech, but also estimate the embedding capacity precisely.

3. Second Statistics Based Steganalysis Algorithm

In this section a new steganalysis method is described in detail. First, all the bits in each frame of

the G.723.1 codec are divided into six speech parameters according to the steganographic algorithm introduced above. Statistical analysis of these parameters is then conducted by using Poker test. Finally, the steganalysis method is used to determine the embedding capacity.

3.1 Poker Test for Speech Parameters

Poker test is one of the statistical tools used to study and predict random phenomena. The test starts with a set (sequence) called the sample space, which relates to the set of all possible outcomes, denoted by $S = \{x_1, x_2, \ldots\}$. Assuming that the sequence S consists of N random variables like integers, the whole sequence is described as S^N, and the number of times the variable (integer) i occurs in the sequence is denoted by $n_i(S^N)$, $0 \le i \le 2^L-1$, where L is an integer used to define the largest variable in the sequence.

Suppose the whole sequence S^N is divided into N/L segments. If all the values of $n_i(S^N)$ with $i = 0$, 1, 2, …, 2^L-1, i.e. the frequency of each variable occurring in the sequence, are calculated, thus, the normalized variance of a random variable in the sequence, f_{Tp}, which is the expected square deviation of that variable from its expected mean, can be computed by the following formula:

$$f_{Tp} = \left(\frac{L \times 2^L}{N}\right)^2 \sum_{i=0}^{2^L-1}\left(n_i(S^N) - \frac{N}{L \times 2^L}\right)^2 \qquad (1)$$

where $N/(L\times 2^L)$ is the expected number of times the variable i occurs in the segment if each variable has the same probability of appearing, and $L\times(2^L)^2/N$ is used to normalize the variance. The advantage of using the normalized variance is that it enables comparisons between different samples.

To perform a poker test on the G.723.1 compressed speech, the whole sequence (S^N) is required to construct first. In terms of the bit allocation of the G.723.1 codec (Table 1) and the Capacity of Noise Tolerance (CNT) listed in Table 2, all bits in each audio frame are divided into six speech

parameters, denoted by $s1$, $s2$, $s3$, $s4$, $s5$, $s6$, and $S = \{s1, s2, s3, s4, s5, s6\}$. Statistical analysis is then performed on these speech parameters, respectively.

The bit sequence of the speech parameter s_i in the frame k is defined as

$$s_i^k = b_i^0, b_i^1, \ldots, b_i^j \quad (2)$$

with $i = 1, 2, \ldots, 6$, and $j = 0, 1, \ldots, sum$

where sum is the total number of bits in the parameter s_i that can be used to embed messages. As different parameters have different numbers of Least Significant Bit (LSB), the values of sum vary for different parameters. Thus the whole bit sequence of the frame k is given by

$$Y_k = s_1^k, s_2^k, \ldots, s_6^k \quad (3)$$

Assuming the total number of frames in a compressed speech sample is M. The bit sequence of the parameter s_i for all frames, X_i^M, can be constructed as follows

$$X_i^M = s_i^0, s_i^1, \ldots, s_i^j \quad (4)$$

with $i = 1, 2, \ldots, 6$, and $j = 1, 2, \ldots, M$

where s_i^j denotes the bit j of the parameter s_i. The subscript i of s_i^j stands for the sequence number of the parameter, and the superscript j denotes the bit sequence number in the parameter. So the whole bit sequence of M frames can then be described as

$$X^M = X_1^M, X_2^M, X_3^M, X_4^M, X_5^M, X_6^M \quad (5)$$

In accordance with the above Poker test algorithm, a series of experiments were conducted to examine all the speech parameters of the G.723.1 codec, such as $s1$, $s2$, $s4$, $s5$, and $s6$, except for $s3$ because its CNT level is too high to be used for embedding messages. For example, in order to calculate f_{Tp} of the parameter $s6$, the bit sequence S_6^N is constructed using Equation (4), where N = 152790×8, 152790 is the least number of the frames the tested speech sample is divided into, which is

the threshold value determined by observing a number of experiments, and 8 is the number of bits in the parameter *s*6 that can be used to embed messages. Hence $L = 8$ and $N/L = 152790$ are obtained; $M = N/L$ represents the number of the tested frames.

In the experiments random messages were embedded in different speech parameters of the G723.1 codec, leading to a number of stego-speech streams (data sets) with embedding rates varying from 0% to 100% in 10 percent increments. A chaos random sequence generator was used to create the embedding positions so as to achieve random embedding. The f_{Tp} values for different speech parameters were calculated, and the results are listed in Table 3.

Table 3: f_{Tp} values for different speech parameters before and after embedding messages

Parameters	Embedding rates (%)										
	0	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
*s*1	130.346	115.801	102.851	91.342	81.465	73.072	66.213	60.844	57.039	54.742	53.989
*s*2	130.335	117.878	106.788	97.019	88.512	81.317	75.465	70.908	67.594	65.608	64.915
*s*4	130.348	115.796	102.878	91.425	81.437	73.083	66.178	60.812	57.007	54.732	53.983
*s*5	130.505	118.020	106.818	96.968	88.427	81.193	75.317	70.774	67.487	65.554	64.939
*s*6	50.950	47.941	45.287	42.910	40.884	39.156	37.739	36.623	35.851	35.372	35.209

Analysis of the data in Table 3 shows the value of f_{Tp} decreases as the embedding rate increases for each speech parameter. This is probably due to the reason that the randomness of stego-speech streams becomes more significant when the embedding rate increases. The f_{Tp} values for different speech parameters can then be used to compute the number of times the bit occurs in the frame, which

is required by the second statistical method (detailed in the next section) to estimate the embedding capacity.

3.2 Second Statistical Detection and Regression Analysis

For the parameter set of the G723.1 codec, $S = \{s1, s2, \ldots, s6\}$, the value of $n_i(S^N)$ for each speech parameter can be computed by using the Poker test algorithm described in the preceding section, respectively. Thus the following equation yields

$$y_i = \{n_0(s_i^N), n_1(s_i^N), \ldots n_J(s_i^N)\} \quad (6)$$

with $i \in \{1, 2, 4, 5, 6\}$

Let $Y = \{y1, y2, y4, y5, y6\}$, a new bit sequence Y is constructed, and a new frequency value $n_y(s^N)$ can then be obtained for the sequence Y. A detailed analysis by drawing curves using SPSS (statistics software) reveals great statistical regularity, as shown in Figure 1. Note that the calculations are based on the parameter $s6$, i.e. Pulse positions (Ppos).

The arrows in Figure 1 denote the frequency points at different levels of data embedding. Figure 1 shows the frequency point increases non-linearly as the embedding rate increases. A number of repeated experiments were conducted so as to allow the establishment of the statistical relationship between the frequency point and the embedding rate. Regression analysis was chosen to seek for the statistical law, i.e. a mathematical function between the frequency point and the embedding rate, as shown in Equation (7)

$$p = a_0 + a_1 m + a_2 m^2 + a_3 m^3 \quad (7)$$

where p is the embedding rate, m is the frequency point of the sequence Y, and a_0, a_1, a_2 and a_3 are the

coefficients.

Fitting Equation (7) with plenty of similar data sets, obtained from the G.723.1 compressed stego-speech samples having 152790 frames, leads to attaining the regression coefficients on Equation (7) such as a_0, a_1, a_2 and a_3. So Equation (7) is re-written as

$$p = 0.000668 + 1.940232m - 1.511366m^2 + 0.588058m^3 \qquad (8)$$

The estimation curve (Model) in Figure 2 is drawn according to Equation (8), and the experimental curve (Observed) is based on experimental results. Comparisons between the estimated and experimental results indicate that Equation (8) accurately simulated the relationship between the embedding rate and the frequency point. Hence, with Equation (8) ones can easily compute the corresponding embedding rates for different frequency points.

However, there is an unsolved issue, i.e. how to decide whether a sampled speech contains hidden messages. If a secret message is embedded in the sampled speech, the embedding rate cannot be assumed to be 0% when computing the first frequency point in Figure 1. In fact, the correct relationship between the frequency point and the embedding rate cannot be derived when blind detection is performed. If so, how to compute the embedding rate in case of blind detection? To solve this problem, a novel method based on the second steganography is suggested to estimate the embedding rate as follows.

The second steganography with random embedding positions in the same sampled speech as the first steganography does is suggested to decide whether the sampled speech contains hidden messages, and determine how much information has been embedded. Suppose the first embedding rate is p_1, and the second embedding rate is p_2. After completion of two steganography processes (i.e. embedding a

message in the sampled speech at p_1 followed by embedding another message at p_2), $p_1 \times p_2$ percent of Least Significant Bits (LSBs) in the speech parameters like *s*6 are changed twice, and half of the bits in the parameters are converted to the original values again. Hence, the embedding rate equals the percent of bits that have been changed only once after the second steganography, given by

$$\text{Embedding rate} = (p_1 + p_2 - p_1 \times p_2) / 2 \quad (9)$$

In the experiments, the first embedding rates were fixed at $p_1 = 30\%$, and the second embedding rates were varied, i.e. $p_2 = 10\% \times i$ with $i = 0, 1, \ldots, 10$. Initially the sampled VoIP speech embedded a message at the first embedding rate of 30%, and then embedded another message at the second embedding rates varying from 0% to 100% in 10 percent increments, respectively. Figure 3 shows the experimental results, depicting the relationship between the frequency of $n_y(s^N)$ and the second embedding rate (p_2) while the first embedding rate (p_1) remained constant. The black coattail arrows in the figure are related to the first embedding rates, and the common arrows are in relation to the second embedding rates.

Close analysis of the experimental results (Figure 3) shows that the first embedding rate (black coattail arrows) stands out well against the second embedding rate (common arrows) in the first figure (the second embedding rate is 0%) and in the sixth figure where the second embedding rate is 100%. For the same sampled speech, the results for the two data embedding process at $p_1 = 30\%$ and $p_2 = 0\%$ should be the same as those for the single data embedding process at $p_1 = 30\%$. So the frequency point is related to the first embedding rate only when the second embedding rate is 0%. Therefore, as long as the frequency points in Figure 3 are obtained, the embedding rate for the first steganography can then be determined by using Equation (7).

4. Performance Evaluation

A series of experiments were performed to evaluate the proposed steganalysis method. Seven groups of the compressed speech sampled from the G.723.1 codec with 6.3kp/s LOC (Lines of Communication) were employed as cover objects. Random messages with different lengths were embedded in each of the compressed speech cover objects, respectively, to achieve ten different embedding rates. A chaos sequence generator was used to create random embedding positions in the compressed speech streams so as to ensure the messages were randomly spread out over the embedding positions. To simplify the experiments, 128 bits keys were used for steganography.

Figure 4 illustrates the testing process in which stegnography and steganalysis in VoIP streams were carried out over an intranet called CERNET. Alice and Bob communicated secretly, so the VoIP streams between them contain secret messages, which were embedded in the parameters of ITU-T G.723.1 (6.3Kbps) compressed speech streams. Similarly covert communications also occurred between John and Smith. Using our proposed steganalysis method, Mary as a warden monitored the router connected with the network by examining all transmitted streams between Alice, Bob, John and Smith so as to decide whether a transmission contains a hidden message and to estimate the embedding rate.

The embedding rate is defined as the secret message length divided by the length of the stego VoIP stream. The real embedding rates varied from 0% to 100% in 10 percent increments, which are listed in the second row of Table 4. Seventy test data sets were used to perform statistical analysis.

The proposed steganalysis method was utilised to compute the estimated embedding rates, and the results are listed in Table 4.

Table 4: The real and estimated embedding rates for different compressed speech cover objects

Cover objects	Embedding rates									
	0.10	**0.20**	**0.30**	**0.40**	**0.50**	**0.60**	**0.70**	**0.80**	**0.90**	**1**
Cover 1	0.1412	0.1522	0.3070	0.4293	0.4626	0.6161	0.7129	0.7665	0.9237	0.9700
Cover 2	0.1650	0.1802	0.3330	0.3636	0.5013	0.5467	0.6948	0.7796	0.8665	1.0176
Cover 3	0.1650	0.1802	0.3636	0.4356	0.5230	0.6095	0.6948	0.8147	0.9083	1.0176
Cover 4	0.1802	0.1984	0.3636	0.4005	0.5467	0.5726	0.6664	0.7593	0.8538	0.9573
Cover 5	0.1802	0.1984	0.3636	0.4005	0.5013	0.6009	0.6990	0.7967	0.8985	1.0176
Cover 6	0.1802	0.1984	0.3636	0.4005	0.4773	0.5726	0.6664	0.7967	0.8985	1.0176
Cover 7	0.0007	0.1911	0.3528	0.3752	0.4717	0.5892	0.7047	0.8200	0.9442	1.0176

Statistical tools are normally used for quantifying the accuracy and precision of a measurement or approximation process. Accuracy is the degree of closeness of a measured or calculated quantity to its actual (true) value, indicating proximity to the true value. So the absolute error, which is the magnitude of the difference between the real value and the approximation, is an indication of accuracy. Precision is the degree to which further measurements or calculations show the same or similar results. In statistics, standard deviation is a measure of the variability or dispersion of a statistical population, a data set, or a probability distribution. A low standard deviation indicates that the data points tend to be very close to the mean, whereas a high standard deviation indicates that the data are spread out

over a large range of values. So the standard deviation of a group of repeated calculations should give the precision of those calculations.

Table 5: Accuracy and precision in estimating data embedding rates

Real embedding rate	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1
Estimated embedding rate (Mean)	0.1446	0.1856	0.3496	0.4007	0.4977	0.5868	0.6913	0.7905	0.8991	1.0022
STDEV	0.0650	0.0168	0.0219	0.0260	0.0300	0.0245	0.0181	0.0231	0.0312	0.0266
Absolute error	0.0446	0.0144	0.0496	0.0007	0.0023	0.0132	0.0087	0.0095	0.0009	0.0022

Table 5 lists the accuracy and precision of the proposed steganalysis method in estimating data embedding rates. A small standard deviation (STDEV) indicates that the estimated embedding rates for the seven compressed speech samples (their results are listed in Table 4) are clustered closely around the means at different levels of data embedding. Therefore, the experimental results show the proposed steganalysis method has great precision in determining the embedding rate in most circumstances, and acceptable errors occur at low embedding rates.

5. Conclusions

In this paper we have suggested a novel method capable of detecting the hidden message within a

compressed VoIP speech and estimating the embedding rate as well. The experimental results have shown the proposed steganalysis method is quite effective and accurate. To the best of our knowledge, this is the first practical implementation of the steganalysis of the compressed VoIP speech.

This work is an initial exploration of the detection of covert channels in the compressed VoIP speech and there is room for improvement. Other steganalysis methods for detecting hidden information in other compressed speeches such as iLBC and G.729.a are the subjects of future work.

Acknowledgments

This work was supported in part by grants from the National High Technology Research and Development Program of China (863 Program, No. 2006AA01Z444), the National Foundation Theory Research of China (973 Program, No. 2007CB310806), and the National Natural Science Foundation of China (No. 60703053, and No. 60773140).

References

[1] Avcibas, I.: 'Audio steganalysis with content-independent distortion measure', IEEE Signal Processing Letters, 2006, 12, (2), pp. 92-113

[2] Fridrich, J., and Goljan, M.: 'Practical steganalysis of digital images-state of the art'. Security and Watermarking of Multimedia Contents IV, San Jose, USA, January 2002, pp. 1-13

[3] Fridrich, J., Goljan, M., and Hogea, D.: 'Steganalysis of JPEG images: Breaking the F5 algorithm'. Proc. 5th Information Hiding Workshop, Toronto, CA, May 2002, pp. 23-35

[4] Fridrich, J., Goljan, M., and Du, R.: 'Detecting LSB steganography in colour and gray-scale

images', IEEE Multimedia, 2001, 8, (4), pp. 22-28

[5] Ru, X.M., Zhang, H.J., and Huang, X.: 'Steganalysis of audio: Attacking the steghide'. Proc. Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, China, August 2005, pp. 3937-3942

[6] Zhang, T.: 'Image steganalysis of bit randomicity'. PhD thesis, Tsinghua University, 2003

[7] Westfeld, A., and Pfitzmann, A.: 'Attacks on steganographic systems'. Lecture Notes in Computer Science, vol. 1768, Springer-Verlag, Berlin, 2000, pp. 61-76

[8] Westfeld, A.: 'Detecting low embedding rates'. Proc 5th Int. Workshop Information Hiding, Noordwijkerhout, The Netherlands, October 2002, pp. 324-339

[9] Mazurczyk, W., and Kotulski, Z.: 'New VoIP traffic security scheme with digital watermarking'. Proc SafeComp 2006, Lecture Notes in Computer Science 4166, Springer-Verlag, Heidelberg, 2006, pp. 170-181

[10] Kuhn, D.R., Walsh, T.J., and Fries, S.: 'Security considerations for Voice over IP systems'. National Institute of Standards and Technology, U.S. Department of Commerce, 2004

[11] Kraetzer, C., Dittmann, J., Vogel, T., and Hillert, R.: 'Design and evaluation of steganography for Voice-over-IP'. Proceedings of 2006 IEEE International Symposium on Circuits and Systems, Island of Kos, Greece, May 2006, pp. 2397-2340

[12] Tian, H., Zhou, K., Jiang, H., Huang, Y., Liu, J., and Feng, D.: 'An adaptive steganography scheme for Voice over IP'. Proc. IEEE International Symposium on Circuits and Systems, Taipei, Taiwan, May 2009, pp. 2921-2925

[13] Mazurczyk, W., and Szczypiorski, K.: 'Steganography of VoIP streams', Lecture Notes in Computer Science, 2008, 5332, pp. 1001-1018

[14] Dittmann, J., Hesse, D., and Hillert, R.: ‘Steganography and steganalysis in Voice-over IP scenarios: operational aspects and first experience with a new steganalysis tool set’. Security, Steganography, and Watermarking of Multimedia Contents VII, SPIE vol. 5681, Jan. 2005, pp. 607-618

[15] Wray, J.C.: ‘An analysis of covert timing channels’. Proc. IEEE Computer Society Symposium on Research in Security and Privacy, May 1991, pp. 2-7

[16] Dittmann, J., and Hesse, D.: ‘Network based intrusion detection to detect steganographic communications channels - on the example of audio data’. Proc. 6th IEEE Workshop on Multimedia Signal Processing, Siena, Italy, September 2004, ISBN 0-7803-8579-9.

[17] Gul, G., Dirik, A.E., and Avcibas, I.: ‘Steganalytic features for JPEG compression-based perturbed quantization’, IEEE Signal Processing Letters, 2007, 14, (3), pp. 205-208

[18] Kraetzer, C., and Dittmann, J.: ‘Mel-Cepstrum based steganalysis for VoIP-steganography’. IS&T/SPIE Symposium on Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII, San Jose, USA, January 2007.

[19] Liu, Q., Sung, A.H., and Qiao, M.: ‘Temporal derivative-based spectrum and mel-cepstrum audio steganalysis’, IEEE Transactions on Information Forensics and Security, 2009, 4, (3), pp. 359-368

[20] Su, Y., and Huang, Y.: ‘Steganography-oriented noisy resistance model of G.729a’, IMACS Multi-conference on Computational Engineering in Systems Applications, 2006, 1, pp. 11-15

[21] Xiao, B., Huang, Y., and Tang, S.: ‘An approach to information hiding in low bit-rate speech stream’, IEEE GLOBECOM 2008, 2008, pp. 371-375

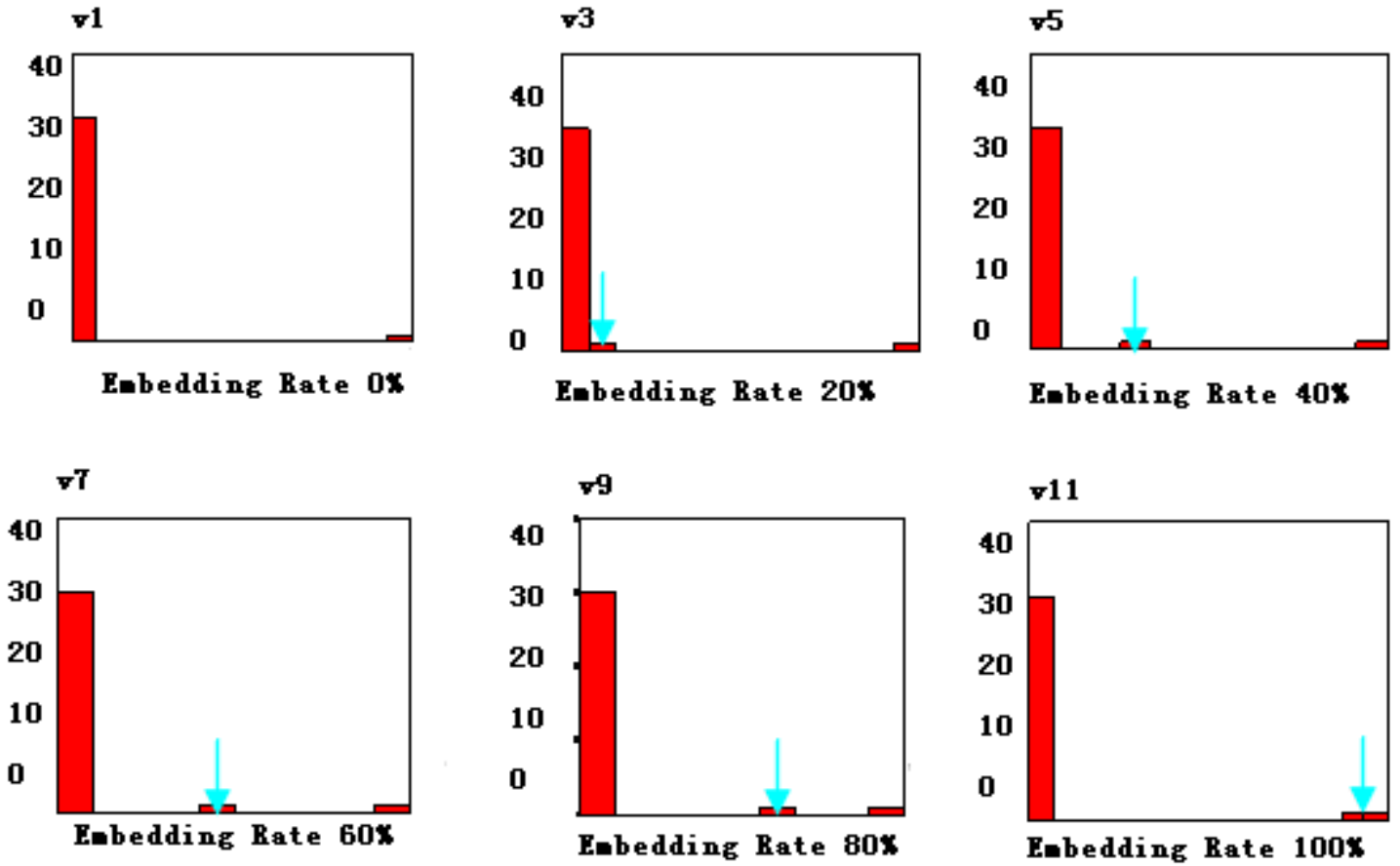

Figure 1: The frequency of $n_y(S^N)$ at different levels of data embedding (Y-axis is the frequency and X-axis is the frequency point)

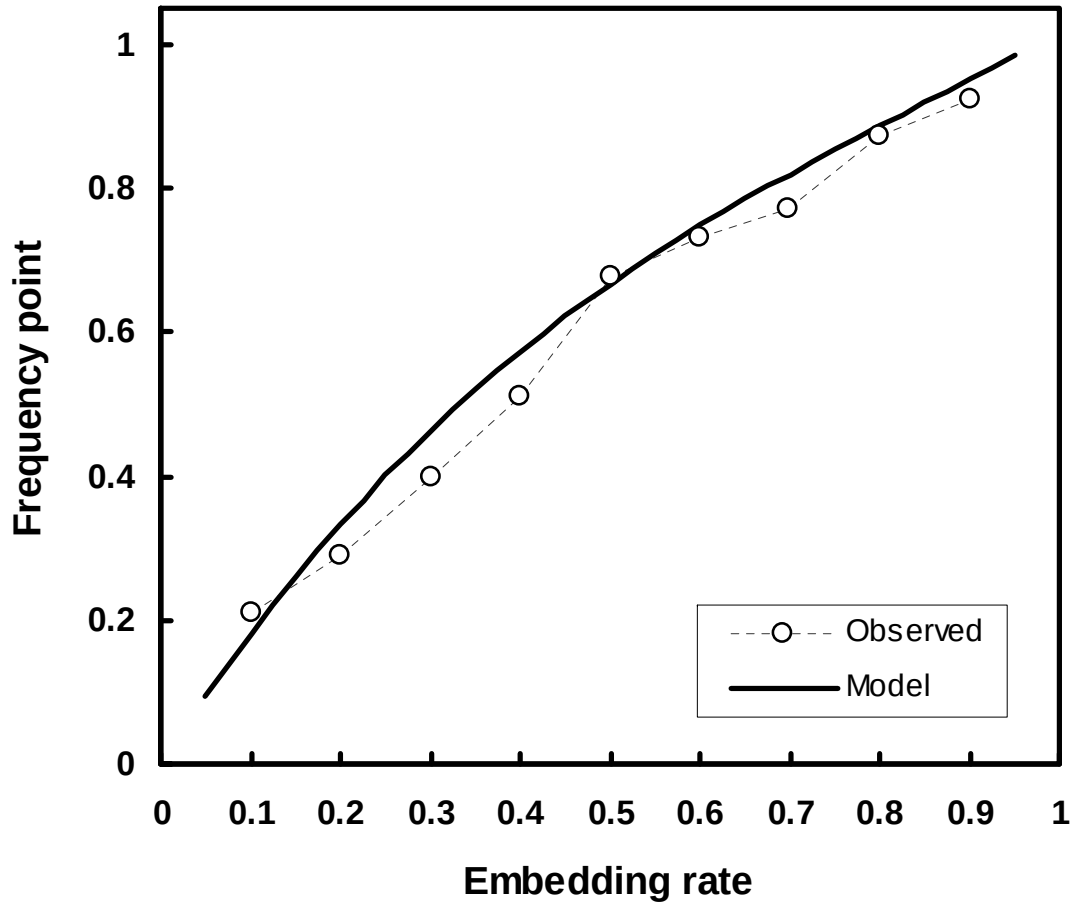

Figure 2: Relationship between the embedding rate and the frequency point

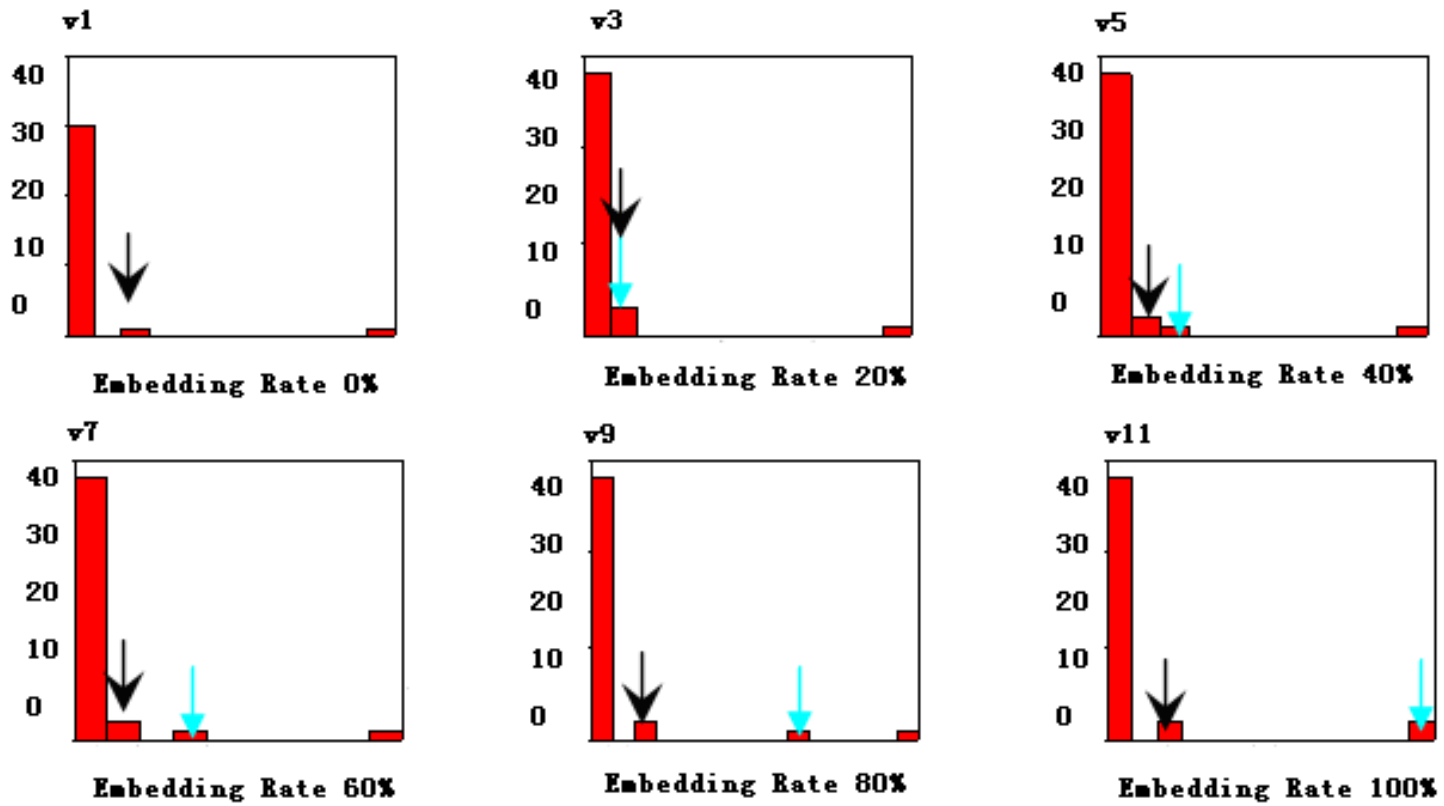

Figure 3: The frequency of $n_y(S^N)$ at varying second embedding rates with a constant first embedding rate (Y-axis is the frequency and X-axis is the frequency point).

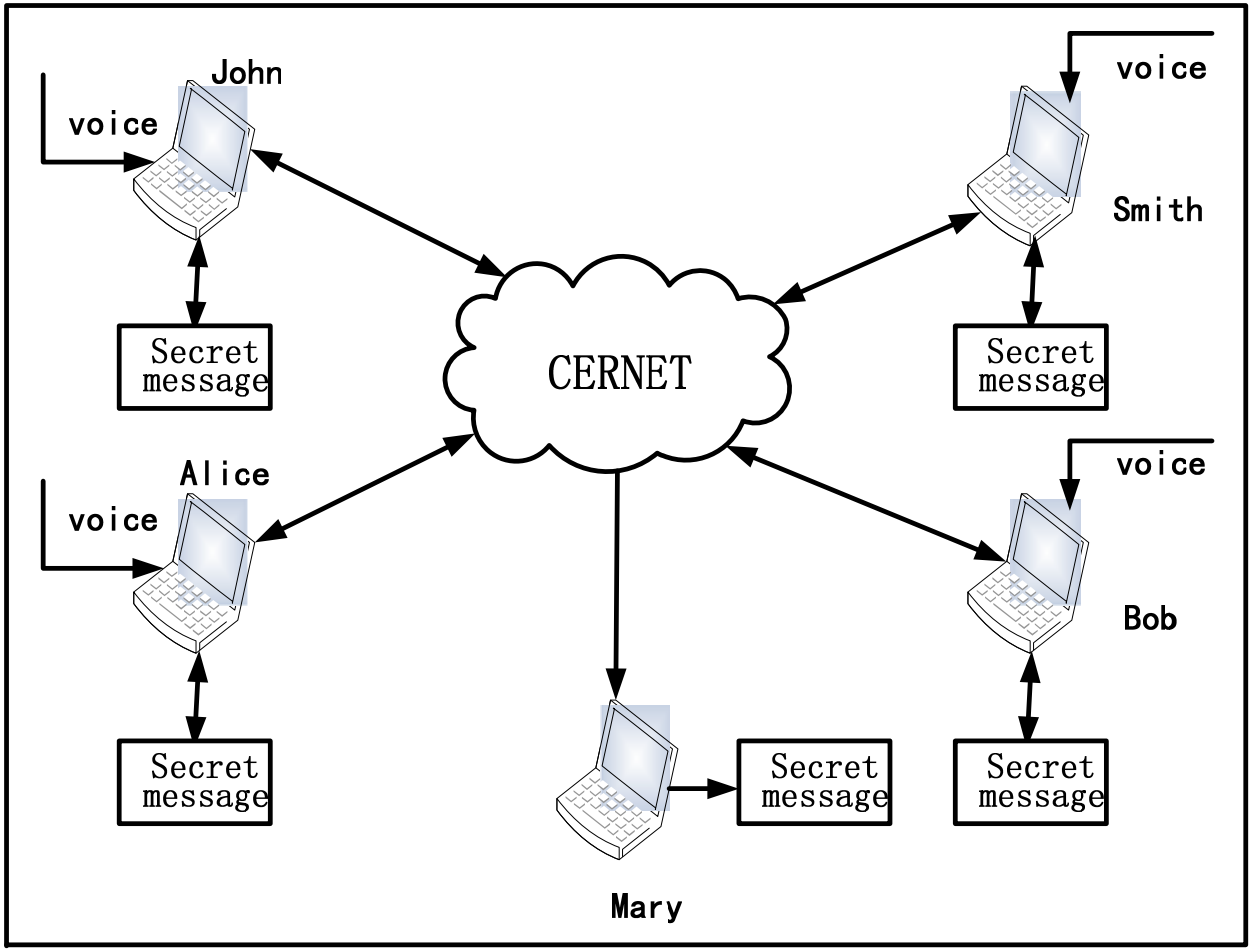

Figure 4: Sketch of the testing setup