BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **20**, No 3
Sofia • 2020 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2020-0031

Coverless VoIP Steganography Using Hash and Hash

S. Deepikaa, R. Saravanan

Vellore Institute of Technology, Vellore, India
E-mails: deepikaa.s@vit.ac.in rsaravanan@vit.ac.in

***Abstract*:** *Performing secure and robust embedding and extracting in real time voice streams without deteriorating the voice quality is a great challenge. This paper aims on hiding the secret data bits in the voice packets without modifying any data in the cover thereby improving the embedding transparency and becomes robust against the steganalysis attacks using coverless approach. Initially a hash array is built with the frame size. The cover bit position is identified from the hashing function. The hash array is marked with a flag value to indicate that the particular sample consist of the secret message bit. The hash array is attached with the VoIP samples, at the receiver side the hash table is separated, and the secret bits are extracted based on the hash array. The experimental results conducted on a VoIP prototype proved to be simpler and effective in terms of the computational complexity, undetectability and voice quality at both sender and receiver end.*

***Keywords*:** *Hash, VoIP.*

1. Introduction

Hiding data into another file such as image, audio, video is the art of steganography. There are number of algorithms proposed and successfully implemented in real time. [3] Author compares many static digital audio steganography methods. The secret data is hidden in another file called cover file. The whole file cannot be used for data hiding. Choosing the cover for steganography is a tedious job. Only the portion of the file, whose modifications remains unnoticed or whose modifications don't affect the voice quality factors must be chosen. One of the trending researches is done on choosing dynamic covers rather than the static cover. Static cover has more security threats of getting detected and interpreted. The effectiveness of the embedding lies not only in hiding the secret message in another file but in also hiding its existence without leaving a clue. One such dynamic cover steganography is VoIP steganography, which aims at hiding the secret message in VoIP packets. The VoIP (Voice over Internet Protocol) is a technique of communicating with the remote person by streaming the audio and video in terms of VoIP packets through internet. The effectiveness of the steganography is measured by stegnographic bandwidth, cost, robustness and transparency.

The VoIP steganography can be done by modifying the VoIP payload or modifying time relations or combining both [4]. The first method is the most common and simplest method of hiding secret data. The methods involved in modifying the payload may be easily sniffed by the attacker on continuous monitoring or by comparing the payload packets at different networks. The modification in the payload packet may also lead to the degradation of voice quality.

The survey of the existing methods on VoIP steganography is done in [4], [20]. Some methods are highlighted in this section for basic understanding. These methods perform embedding and extraction during the audio encoding and decoding process. Since the hiding is done even before the VoIP packet is framed, Integrity is maintained. This feature improves the undetectability but it is not completely undetectable. The steganalysis based on the content comparison do not reveal the secret data or existence of it, but steganalysis based on some statistical features aids detecting the existence of secret data.

LSB Steganography. This method has been age old but simple technique followed from image steganography and extended to latest covers. This method replaces the Least Significant Bit (LSB) with the secret message bit. So that the changes in the least significant bit do not affect the audio and the human auditory system does not differentiate much of the minute changes undergone. This feature helped many researchers to propose various methods on LSB modification. LSB method also provided more bandwidth that allowed us to hide more information, but the static steganalysis method detected the existence of the message from its static properties. One of the effective LSB methods was Matrix embedding which was first proposed by C r a n d a l l [1] and later developed by other researchers. Hybrid Matrix embedding methods [8-11, 18] yielded fruitful results rather than the pure matrix embedding.

Quantization index Modulation. Quantization index modulation is the process of hiding and extracting the secret data during the modulation and demodulation step using the index values in the codebook. The codebook rules are framed in such a way that to hide the secret data inside the cover. Various methods have been proposed on codebook division, which plays important role in embedding and extracting. Some of the prominent methods are discussed in [13-16].

Spread Spectrum Methods. Spread Spectrum method is the process of hiding and extracting the secret messages along the Frequency domain. This signal spreading is done by adding some noise and along the noise the secret data is hidden and certain methods use a specific frequency factor.

Pitch Modification Methods. Pitch modification methods aims at hiding and extracting the secret data along the special pitch parameter F0 [6, 12]. The authors contributed to VoIP steganography based on the pitch parameter fundamental frequency F0 [2]. The author also proposed a method where audio packets do not modify but new packets are added by introducing intentional delays. The intentional delays may give a clue of presence of the secret data [22]. The author shared the areas that aids in detecting the presence of the secret data with a case study. This may be beneficial for those involved in proposing new methods to embed the secret data in VoIP [17]. Speech codec properties also provide us space for hiding the secret data

but the limitation is that it may not be compatible with other speech codecs sometimes.

Coverless Steganography. The Coverless steganography is the process of hiding the secret data into the cover file without modifying the cover data. The coverless steganography has been first proposed on images. Many authors have contributed towards hiding the secret data without tampering the cover. This paper aims at using the coverless steganography into live data streaming. The paper utilizes the hashing concept as per the title "Coverless Steganography using Hash and Hash" in two ways.

Hash serves two purposes. When hash is used as data structure, the hash table helps organizing the data with respect to the index. When hash is used for authentication, the hash value is generated and appended along with the plain text. This paper utilizes both the services of hash and hence the title describes that first hash is for identifying the cover bit position depending on the hash array index. The second hash is for appending the hash array along with the VoIP samples.

To start up with simple method, the hash array concept of data structures is used. For example: In linear hashing, the hash table supports the ⟨key, value⟩ concept. The value is stored in the hash table depending on the key. This hash array is to be sent to the other party to help them extract the secret data. This idea was inspired from the hash code utilized in network security for authentication purposes. The generated hash array is attached with the audio packets like a hash code and sent to the receiver. The receiver extracts the hash array from the packet and sends only the voice samples to the audio player, meanwhile the algorithm processes the samples received to extract the secret message with the help of hash array.

The paper also aims at reducing the degradation of voice by not modifying the cover bits. The cover is not subject to any changes and instead the position of cover bit is selected in such a way that secret data is embedded with proposed algorithm, which in turn improves undetectability and transparency. The hacker is unable to extract the secret message without the cover bit position that depends on the hash function. In this way the integrity of the data and cover is maintained. The remaining sections are organized as follows; the related works on coverless steganography in images are described in Section 2. The proposed algorithm is explained with an illustration in Section 3. The results and graphs are organized in Section 4. The conclusion and future enhancements are quoted in Section 5.

2. Related works

Coverless steganography impresses the researchers with the fact that, the cover audio need not be modified, but complexity lies in finding the cover features which helps to hide the secret data without changing the cover. It is noted from previous works, that the Coverless steganography on images is done in two ways. One is generating some functions that involves texture synthesis and hide secret data inside it. Another method is using indexing using some mapping functions. This paper focuses on second method. Some works that contributes to first method can be referred in

[24-26]. Some notable works on indexing is highlighted below for basic understanding.

Z h e n et al. [19] proposed a method of hiding the secret data using shared hash method. The author divides the secret image into segments of equal length as same as the length of the hash code. The hash code is constructed based on a map designed using quad tree structure. The map is built based on the SIFT features of the image. The secret data is extracted from the hash code at the receiver end.

Z o u et al. [21] proposed a method, which hides the secret information using the average of sub images. A hashing algorithm was used to generate the hashing sequence, which in turn is mapped with the hash array using a dictionary with some set of rules framed. The image retrieval is based on multi-level index structure.

C h e n et al. [23] proposed a method, in an effort to improve the hiding capacity in images. The image is split into image blocks. Each block can hide one bit. For retrieval of the message bits, the author uses double level index

The coverless steganography works on audio and VoIP are not reported. This paper is the first effort to implement the coverless steganography work on VoIP.

3. Proposed method

The paper proposes a coverless steganography, in which the cover is not modified to hide the secret data but only refers the samples that contain the secret data. The audio voice is streamed across the UDP protocol.

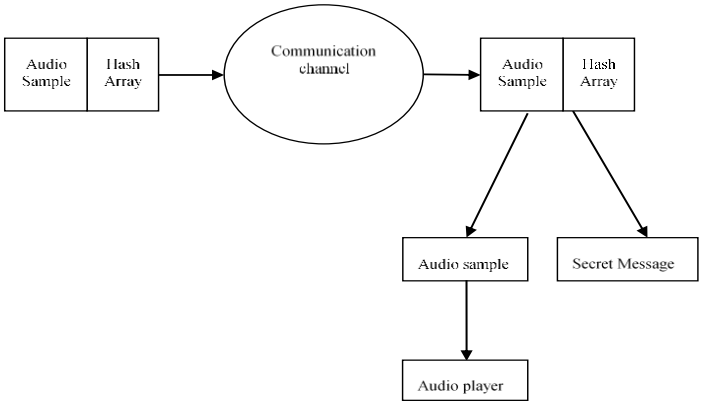


Fig. 1. Proposed model of coverless steganography

3.1. Embedding

The embedding algorithm shows the step by step instruction of how the secret message is hidden in the VoIP samples with the help of hashing. According to the hash table concept of data structures, the array index, where the data is to be stored, is decided on some hashing function. Similarly, the array index plays the major role in identifying the cover bit position in this algorithm.

The hash array plays an important role in informing the receiver about the presence of the secret bits. The size of the hash array is same as the size of the voice samples. To avoid attracting the hacker's attention on the hash array, the hash array

is designed in such a way as to create an illusion of the binaural audio [26], which is 3D audio that can be recorded from the smart phones to produce an effect of real time audio in both ears. This audio differs from the stereo, which has the same values for both left and right ear.

The mono audio recorded is visualized as binaural audio while sending hash array with VoIP samples. At the receiver side, the mono audio is again extracted back in a new array which is played in the audio player while the hash array guides the receiver to extract the secret bits from the cover.

The frame size is decided by the sender and receiver.

Step 1. Split the audio into set of samples $S = \{s_1, s_2, s_3, \ldots, s_f\}$ in a VoIP frame, where f is total number of samples per VoIP frame. Let each sample has n cover bits say, $s_p = \{c_1, c_2, c_3, \ldots, c_n\}$, where $p = \{1, 2, ..., f\}$, represents p-th sample out of f samples.

Step 2. Split the secret message M of size k, where $M = \{m_1, m_2, \ldots, m_k\}$.

Step 3. Initialize H to be the hash array of size f for every new frame, where f is the total number of samples per VoIP frame; $H = \{h_1, h_2, \ldots, h_f\}$.

Step 4. For each sample s_p in S, choose the cover bit position to hide the secret data using the equation

$$\text{Cover bit position } (i) = \log_2^p \bmod n, \tag{1}$$

where $p = \{1, 2, \ldots, f\}$ represent the index of the hash array and n is the length of a sample.

Step 5. If the c_i value obtained by the equation is 0 then increment the position by 1, since the range starts from 1 to n and the mod operation may give the result 0, else the i value remains the same.

For example, if index of the hash array is 2 then 2^1 is 2, so the c_1 bit of the sample is identified as the cover, if the index is 8 then 2^3 is 8. So the c_3 is chosen as cover bit position to hide.

Step 6. Perform XOR of secret message bit m_j and cover bit c_i as per the equation

$$h_p = \begin{cases} 1 & \text{if } c_i \oplus m_j = 0, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where $p = \{1, 2, \ldots, f\}$ is the number of samples per VoIP frame and also represents the array index of hash array, i represents the cover bit position identified from Equation (1), and $j = \{1, 2, 3, \ldots, k\}$, and k is the length of the secret message.

Step 7. If the Hamming distance between the message bit and the cover bit is 0 as per the equation, then hash array of index $p(h_p)$ is marked with the s_p sample value else the hash array value is set to 0.

Step 8. Repeat the steps 3-7 until all the message bits are hided.

Step 9. If message bits are completely hidden, then the hash array values are simply made 0 to indicate that there are no secret data hidden. All the frames have the hash array at the end, despite secret data embedded. This is to ensure the VoIP frames are of same length.

Step 10. Attach the hash array at the end of each frame and send it to the receiver.

VoIP frame = Voice samples + hash array.

3.2. Extracting

The VoIP frames generated from the sender is received at the receiver side. Each VoIP frame is processed individually to obtain the audio samples and the hash array. The message bits extracted from each frame are concatenated to obtain the whole message. The following steps elaborate the steps of extraction.

Step 1. Receive the VoIP frame with audio samples and hash array concatenated with it.

Step 2. Extract the first *f* samples from the frame sent and store in a new array. This array also helps us to extract the secret message.

Step 3. The voice samples stored in a new array are sent to the audio player.

Step 4. Extract the hash array after *f* frames from the frame and store in an array.

Step 5. Initialize p=0 and Read the p-th voice sample and p-th index in the hash array.

Step 6. if h_p !=0 then calculate the cover bit position using Equation (1). The algorithm is known to the sender and the receiver. According to the proposed method, the secret key is the cover bit position. So the cover bit position selection may slightly differ with respect to the size of the cover bits.

Step 7. The secret message bit from the cover is extracted from the cover bit position obtained and is stored in an array.

Step 8. Concatenate the message bits to form the whole message.

Step 9. If h_p=0, then that voice sample is omitted from extraction.

Step 10. Read the next frame and repeat the steps 2-9 until the whole message is extracted.

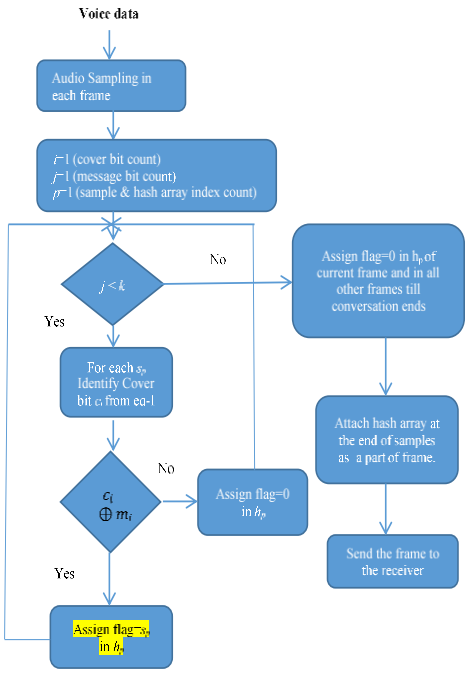


Fig. 2. Embedding algorithm

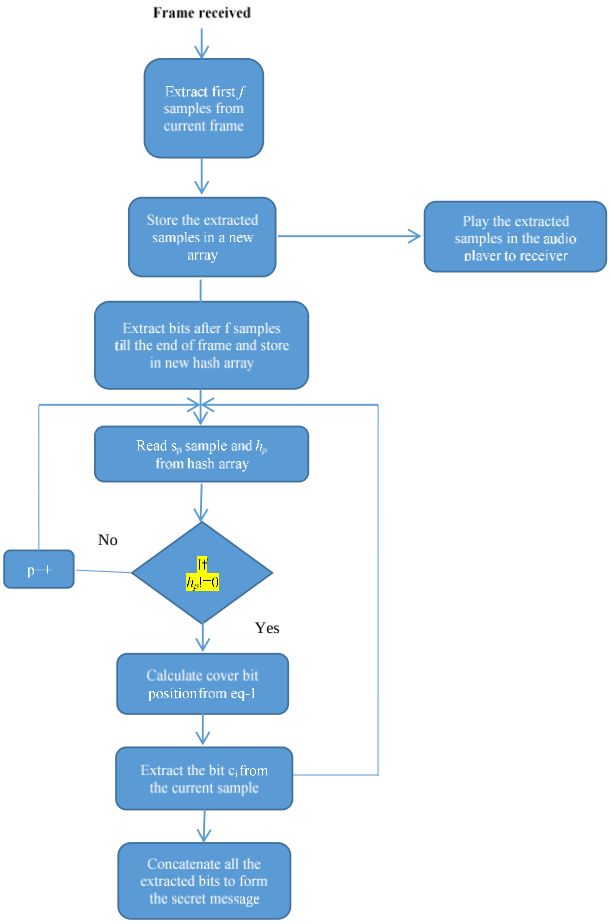


Fig. 3. Extraction algorithm

3.2. Illustrations

Let us assume the sample size n=8, where C= {c_1, c_2, ..., c_8}. The total number of samples per frame f=1024, where S={s_1, s_2, …, s_{1024}}. Assume k message bits to be hided k=8, where M= {0, 1, 0, 0, 1, 0, 0, 0}and the cover bits s_1= {10000000}, s_2={01001100} and so on. The message bit m_1=0 is to be hided in sample s_1. The cover bit position for hash array with index 1 is identified by the Equation (1). So the cover bit position as per the equation is 0. But the range starts from 1, so the c_1 bit is chosen as cover bit position from sample s_1. The XOR between c_1 of sample s_1 and m_1 is 1, so the hash array of index 1 is marked with 0, which means data is not hided. The message bit m_1=0 is to be hided in sample s_2. As per the equation, the cover bit position for the hash array with index 2 is 1. The XOR between c_1 of sample s_2 and m_1 is 0, so the hash array of index 2 is marked with s_2, which means the data is hidden in the sample s_2 at c_1 bit position. This process continues till the whole message is hidden. The surplus hash array indices are simply marked with 0 to specify, no hidden messages inside.

A sample calculation is picturized in the Table 1. The audio with 8 bits per 1 sample is taken. Depending on the index number of the hash array, the cover position is highlighted. The secret message bits M is XORed with the cover position highlighted. If the XOR result is 0 then hash array value is 1 else 0. The hash array is concatenated with the audio sample and sent to the receiver.

Table 1. Sample calculation

Index No	Secret message bit M	Audio sample (*S*)								Hash array
1	0 (m_1)	1	0	0	0	0	0	0	0	00000000
2	0 (m_1)	0	1	0	0	1	1	0	0	01001100
3	1 (m_2)	0	1	1	1	1	1	0	1	00000000
4	1(m_2)	0	1	1	1	1	0	1	1	01111011
5	0 (m_3)	1	0	0	0	0	0	0	0	10000000
6	0 (m_4)	0	1	1	0	1	1	0	0	00000000
7	0 (m_4)	0	1	0	0	0	0	0	1	01000001
8	1 (m_5)	0	0	0	0	0	0	0	0	00000000

4. Experimental results and analysis

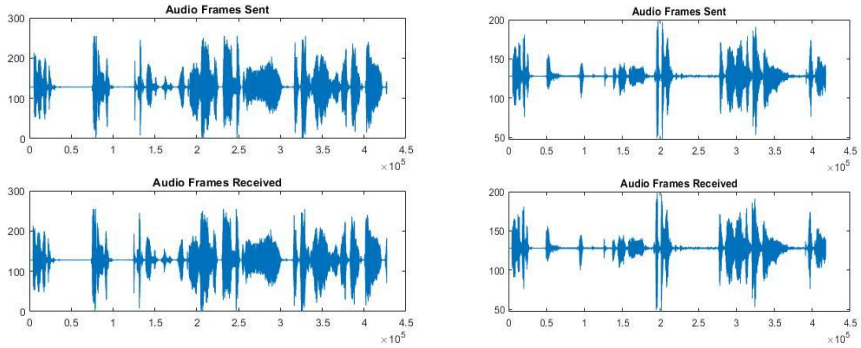
4.1. Experimental setup and results

The proposed algorithm is implemented by developing a VoIP prototype based on UDP protocol using Matlab and the secret message is hidden into the audio samples with the flag values set in the hash array. The audio samples along with the hash array is converted into a VoIP packet and sent across the internet to the sender. At the sender side, the VoIP packets are received and the hash array and audio samples are separated. The flag values in the hash array help us to extract the secret message from the respective sample and cover position. The audio samples are passed through the audio player at the sender side.

Initially 15 Byte message in the character format is hidden in the VoIP call of duration 10 s and this helps us to verify the capacity of the audio samples to hide the secret message of given length. The same experiment was carried out with increasing the call duration to 20 s and 30 s. By gradually increasing the message size, the experiment is repeated until 45 Byte message lengths.

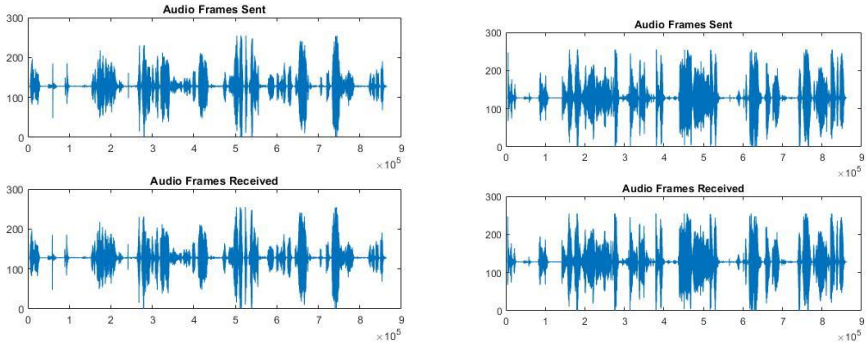
Thus the experimental results show that the secret message can be hidden in the cover without any modification and with no voice quality degradation within the same network. The graph is plotted for the samples sent after embedding the secret data at the sender side and at the receiver side. Fig. 4a shows the graph plotted for the samples with 15 B of secret message hidden in it. The call was tested for 10 s. The top graph represents the samples plot at sender side. The bottom graph represents the samples plot at the receiver side. Fig. 4b shows the graph of the audio samples with 45 B hidden at 10 s duration of VoIP call. Figs 5a and 5b display the audio samples with call duration of 20 s and message length of 15 B and 45B, respectively. Figs 6a and 6b report the audio samples tested for 30 s with message of length 15 B and 45 B, respectively.

In each image, the graphs are arranged in the following order, the top graph represents the sender side samples. The bottom graph represents the receiver side samples.



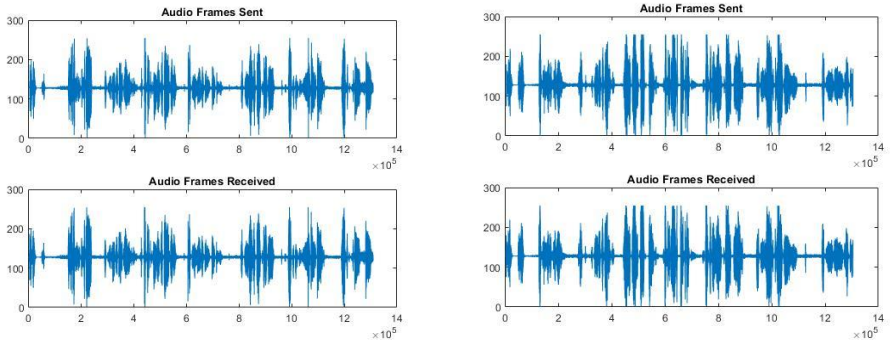
(a) (b)

Fig. 4. 15 B of Message hidden in 10 s (a); 45 B of Message hidden in 10 s (b)



(a) (b)

Fig. 5. 15 B of Message hidden in 20 s (a); 45 B of message hidden in 20 s (b)



(a) (b)

Fig. 6. 15 B of message hidden in 30 s (a); 45 B of message hidden in 30 s (b)

4.2. Performance analysis

The proposed algorithm aims to achieve undetectability by not modifying the original cover bits. Thus, the voice quality degradation is completely avoided during embedding and extraction. The performance of the algorithm proposed is analyzed in three aspects such as capacity, audio quality, undetectability.

4.2.1. Capacity

The proposed algorithm can hide of 1 bit per sample (8 bits). According to hashing functions, each VoIP frame consisting of 1024 samples can hold an average of 300 to 400 bits per VoIP frame. A VoIP conversation of 10 s consists of 400 VoIP frames approximately. So, the 10 s of VoIP conversation can hold 200 B of secret data. Thus, the proposed algorithm hides considerable amount of data into the VoIP packets.

4.2.2. Audio quality

The audio quality is measured in two ways; the first method uses the PESQ-LQO & PESQ-MOS prescribed by ITU-T and the implemented using [29]. The second method uses MSE and PSNR values calculated for the signals at the sender side and the receiver side.

PESQ: Perceptual Evaluation of Speech Quality, as defined in the ITU-T P.862 standard, is an objective method to evaluate the speech quality with the help of objective MOS (Mean Opinion Score) and the PESQ MOS is mapped to the MOS LQO (Listening Quality Objective) which is to evaluate the listening speech quality. These evaluations provide a score ranging from 1 to 5. The higher the score is better the quality achieved. Fig. 7 shows the PESQ MOS and its mapping MOS-LQO at different lengths (10-155 B) of secret messages embedded in a 10 s VoIP conversation. The PESQ results emphasize the fact that the increase in the secret message bits did not affect the speech quality. All the results yielded a score around 4-4.4 which is a proof for good speech quality.

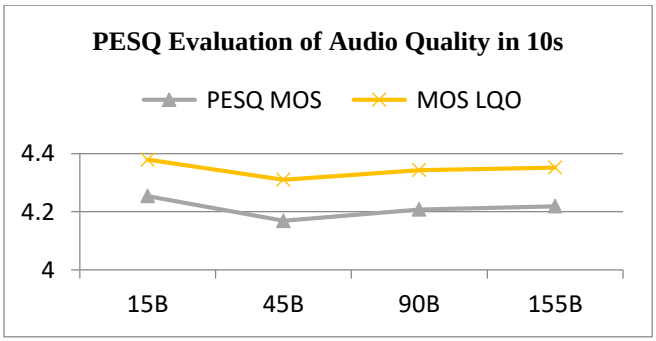


Fig. 7. PESQ evaluation of speech quality

PSNR: Peak Signal to Noise Ratio represents the ratio of the originally constructed signal and the noised, i.e., reconstructed signal. This is used as one of the metrics to judge the signal quality due to noise or modified sample values. This is normally represented in terms of decibels. PSNR is calculated using the equation

$$\text{PSNR} = 20.\log10\left(\frac{\text{Max}}{\sqrt{\text{MSE}}}\right). \quad (3)$$

This can also be represented as

$$\text{PSNR} = 20.\log10(\text{Max}) - 10.\log10(\text{MSE}). \quad (4)$$

Here Max represents the maximum value of signal. For example if audio signal is represented in 8 bits per 1 sample, then the Max value is $((2^8) - 1) = 255$.

MSE: Mean Square Error is the mean of all the differences between the original signal and noised signal in a VoIP frame. MSE is calculated using equation

(5) $$\text{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[s(i,j) - r(i,j)]^2,$$

where *m* and *n* represent the rows and columns used to represent the VoIP samples in a single VoIP frame; *s* represents the original signal (here sent frame), and *r* represents the noised signal (here received frame).

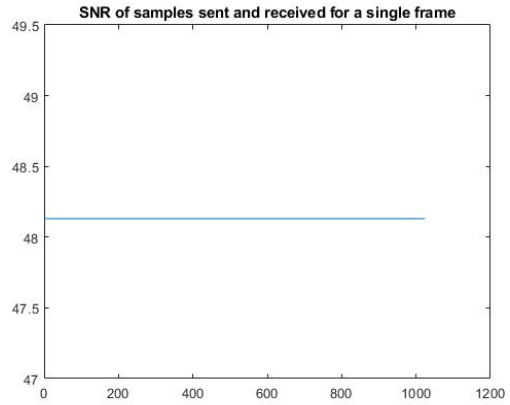


Fig. 8. PSNR of the VoIP frame

The typical PSNR value achieved from the audio compression techniques ranges from 30-50 dB. Higher the score is the better quality achieved. The PSNR value is calculated for all the samples to measure the quality of voice. The PSNR value for a single frame (1024 samples) is plotted in Fig. 8. The PSNR value is calculated with the help of MSE. The MSE for the VoIP samples is zero; this is because of nil modification done to the cover frames. PSNR is considered as 20log 10 (Max). The PSNR value obtained is 48dB approximately, which is good score to describe the audio quality. Since the MSE is 0, 10log10(MSE) will give infinity and PSNR also represents infinity [28]. So, this is neglected during the PSNR calculation. The obtained PSNR value shows that the cover frames sent and received are the same. Hence, the audio quality is not compromised for data hiding.

4.2.3. Statistical undetectability

The Steganalysis aims at detecting the presence of a secret message or the extracting the message itself. During the process of embedding, the statistical properties are modified. The human auditory system will not recognize the slight variations in the audio quality due to some external noise factors. But analyzing the cover properties may disclose the presence of modifications done during embedding. The statistical steganalysis is a process to detect the changes that are made to the statistical properties of the cover. [27] The author used Mann-Whitney U-test or Mann-Whitney-Wilcoxon **(**MWW), Wilcoxon rank-sum test to prove the statistical undetectability.This algorithm implements the MWW to prove the tolerance towards the statistical steganalysis.

The Mann-Whitney U-test or Mann-Whitney-Wilcoxon (MWW), Wilcoxon rank-sum test is a popular non-parametric test of null hypothesis. This test takes randomly selected two independent samples which are from the same distribution.

According to our proposed algorithm, the cover bits are same at both sides before embedding and after embedding. Thus, a sample VoIP frame is plotted in Fig. 8 for our visual understanding.

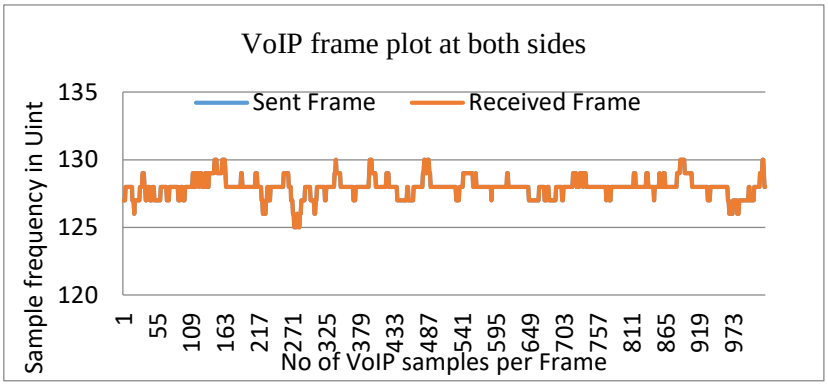


Fig. 8. Sample VoIP frame plot at sender side and receiver side

So, according to the algorithm Null hypothesis H_0 the cover bits are same before and after embedding at both sides.

Alternate hypothesis H_1 – the cover bits are modified due to embedding at both sides. The null hypothesis was tested at the significance level of 0.05 in Matlab using the ranksum() function. The results are displayed in the Table 2.

Table 2. Statistical and Null hypothesis test results

Call Duration	Secret message length	Frame count	H_0 test results			
			P	*H*	*Z* value	Rank sum
10 s	15 B	408	0.9857	0	–0.018	1.83×10^{11}
10 s	45 B	409	0.9819	0	–0.0227	1.75×10^{11}
20 s	15 B	845	0.9993	0	9.40×10^{-4}	7.48×10^{11}
20 s	45 B	843	0.9992	0	9.93×10^{-4}	7.45×10^{11}
30 s	15 B	1280	0.9926	0	–0.0092	1.72×10^{12}
30 s	45 B	1275	0.9998	0	1.96×10^{-4}	1.70×10^{12}

From the test results, the *H* value is 0, therefore the H_0 cannot be rejected. Therefore, the results show that the cover bits do not differ before embedding, after embedding, and at both the sender side and receiver side.

This test shows that the statistical properties of the cover do not get modified due to data hiding. Hence, it can withstand the statistical steganalysis techniques thereby contributing to undetectability. If the statistical properties of the cover are modified due to embedding, the samples cannot

5. Conclusion

This paper proposes coverless steganography in VoIP using hash that do not modify the cover data, instead select the position of cover bit matching the secret bit using a

hashing function. The hash array was built and the flag values inside the hash table were set to sample value, if the embedding has been done, else the flag value remained 0. The hash array was sent to the receiver along with the audio samples as a VoIP frame. The VoIP frames are received and the secret message was extracted based on the hash array flag value. Thus, the algorithm was implemented and tested on a VoIP prototype based on UDP protocol in Matlab. The results proved that the algorithm successfully sent and received the secret message without degrading the voice quality. The statistical undetectability is proved using Mann Whitney U Test. The results also show that there is no additional delay caused due to the computation. The voice quality was assured in terms of PSNR and PESQ values. The hash array may occupy some additional bandwidth during a VoIP conversation but this additional bandwidth does not attract the third party as it resembles a binaural audio. The future work concentrates on improving the coverless techniques by separating the hash array with some efficient key sharing techniques.

6. References

1. C r a n d a l l, R. Some Notes on Steganography. Posted on Steganography Mailing List, 1998, pp. 1-6.
2. M a z u r c z y k, W., J. L. L u b a c z. A VoIP Steganographic Method. – Telecommunication Systems, Vol. **45**, 2010, No 2-3, pp. 153-163.
3. D j e b b a r, F., B. A y a d, K. A. M e r a i m, H. H a m a m. Comparative Study of Digital Audio Steganography Techniques. – EURASIP Journal on Audio, Speech, and Music Processing, Vol. **1**, 2012, Article No 25.
4. M a z u r c z y k, W. VoIP Steganography and its Detection – A Survey. – ACM Computing Surveys (CSUR), Vol. **46**, 2013, No 2, Article No 20.
5. T i a n, H., J. L i u, S. L i. Improving Security of Quantization-Index-Modulation Steganography in Low Bit-Rate Speech Streams. – Multimedia Systems, Vol. **20**, 2014, No 2, pp. 143-154.
6. J a n i c k i, A. Novel Method of Hiding Information in IP Telephony Using Pitch Approximation. – In: Proc. of 10th International Conference on Availability, Reliability and Security (ARES'15), IEEE, 2015, pp. 429-435.
7. Z h o u, Z., H. S u n, R. H a r i t, X. C h e n, X. S u n. Coverless Image Steganography without Embedding. – In: Proc. of International Conference on Cloud Computing and Security. Springer, Cham., 2015, pp. 123-132.
8. T i a n, H., J. Q i n, S. G u o, Y. H u a n g, J. L i u, T. W a n g, Y. C a i. Improved Adaptive Partial-Matching Steganography for Voice over IP. – Computer Communications, Vol. **70**, 2015, pp. 95-108.
9. T i a n, H., J. Q i n, Y. H u a n g, Y. C h e n, T. W a n g, J. L i u, Y. C a i. Optimal matrix embedding for Voice-over-IP Steganography. – Signal Processing, Vol. **117**, 2015, pp. 33-43.
10. Q i n, J., H. T i a n, Y. H u a n g, J. L i u, Y. C h e n, T. W a n g, X. A. W a n g. An Efficient VoIP Steganography Based on Random Binary Matrix – In: Proc. of 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC'15), IEEE, 2015, pp. 462-465.
11. L i u, P., S. L i, H. W a n g. Steganography Integrated into Linear Predictive Coding for Low Bit-Rate Speech Codec. – Multimedia Tools and Applications, Vol. **76**, 2017, No 2, pp. 2837-2859.
12. J a n i c k i, A. Pitch Based Steganography for Speex Voice Codec. – Security and Communication Networks, Vol. **9**, 2016, No 15, 2923-2933.
13. L i u, J., H. T i a n, J. L u, Y. C h e n. Neighbor-Index-Division Steganography Based on QIM Method for G. 723.1 Speech Streams. – Journal of Ambient Intelligence and Humanized Computing, Vol. **7**, 2016, No 1, pp. 139-147.
14. J i a n g, Y., S. T a n g. An Efficient and Secure VoIP Communication System with Chaotic Mapping and Message Digest. – Multimedia Systems, 2017, pp. 1-9.

15. H u a n g, Y., H. T a o, B. X i a o, C. C h a n g. Steganography in Low Bit-Rate Speech Streams Based on Quantization Index Modulation Controlled by Keys. – Science China Technological Sciences, Vol. **60**, 2017, No 10, pp. 1585-1596.
16. L i u, P., S. L i, H. W a n g. Steganography in Vector Quantization Process of Linear Predictive Coding for Low-Bit-Rate Speech Codec. – Multimedia Systems, Vol. **23**, 2017, No 4, pp. 485-497.
17. J a n i c k i, A., W. M a z u r c z y k, K. S z c z y p i o r s k i. Influence of Speech Codecs Selection on Transcoding Steganography. Telecommunication Systems, Vol. **59**, 2015, No 3, pp. 305-315.
18. T i a n, H., J. S u n, C. C. C h a n g, J. Q i n, Y. C h e n. Hiding Information Into Voice-Over-IP Streams Using Adaptive Bitrate Modulation – IEEE Communications Letters, Vol. **21**, 2017, No 4, pp. 749-752.
19. Z h e n g, S., L. W a n g, B. L i n g, D. H u. Coverless Information Hiding Based on Robust Image Hashing. – In: Proc. of International Conference on Intelligent Computing, Springer, Cham, 2017, pp. 536-547.
20. D e e p i k a a, S., R. S a r a v a n a n. VoIP Steganography Methods, a Survey. – Cybernetics and Information Technologies, Vol. **19**, 2019, No 1, pp. 73-87.
21. Z o u, L., J. S u n, M. G a o, W. W a n, B. B. G u p t a. A Novel Coverless Information Hiding Method Based on the Average Pixel Value of the Sub-Images. – Multimedia Tools and Applications, Vol. **78**, 2019, No 7, pp. 7965-7980.
22. J a n i c k i, A., W. M a z u r c z y k, K. S z c z y p i o r s k i. Steganalysis of Transcoding Steganography. – Annals of telecommunications-annales des telecommunications, Vol. **69**, 2014, No 7, pp. 449-460.
23. C h e n, X., A. Q i u, X. S u n, S. W a n g, G. W e i. A High-Capacity Coverless Image Steganography Method Based on Double-Level Index and Block Matching. – Mathematical Biosciences and Engineering, Vol. **16**, 2019, No 5, pp. 4708-4722.
24. W u, K. C., C. M. W a n g. Steganography Using Reversible Texture Synthesis. – IEEE Trans. Image Proc., Vol. **24**, 2015, No 1, pp. 130-139.
25. X u, J., X. M a o, X. J i n. Hidden Message in a Deformation-Based Texture. – Vis. Comput., Vol. **31**, 2015, pp. 1653-1669.
26. B e g a u l t, D. R., E. M. W e n z e l. Techniques and Applications for Binaural Sound Manipulation. – The International Journal of Aviation Psychology, Vol. **2**, 1992, No 1, pp. 1-22.
27. J i a n g, Y., S. T a n g, L. Z h a n g, M. X i o n g, Y. J. Y i p. Covert Voice over Internet Protocol Communications with Packet Loss Based on Fractal Interpolation. – ACM Transactions on Multimedia Computing. – Communications, and Applications (TOMM), Vol. **12**, 2016, No 4, p. 54.
28. S a l o m o n, D. Data Compression: The Complete Reference. 4th Ed. Springer, 2007. 281 p. ISBN 978-1846286025. Retrieved 26 July 2012.
29. P r o d e u s, A. PESQ MATLAB Driver MATLAB Central File Exchange. Retrieved 21 June 2020. (**https://www.mathworks.com/matlabcentral/fileexchange/47333-pesq-matlab-driver**)

Received: 25.01.2020; Second Version: 07.06.2020; Third Version: 21.06.2020; Accepted: 09.07.2020