GiLBCSteg: VoIP Steganography Utilizing the Internet Low Bit-Rate Codec

Garrett Calpouzos¹, Cihan Varol^{2,*}

¹WhiteHat Security, 5120 Woodway Dr., Houston, TX, 77056, United States

²Department of Computer Science, Sam Houston State University, Huntsville, TX, 77341, United States

*Corresponding Author: cvarol@gmail.com

Copyright © 2013 Horizon Research Publishing All rights reserved.

Abstract GiLBCSteg is a steganography algorithm that utilizes an open source low bit-rate audio codec (iLBC) to be used in Voice over Internet Protocol (VoIP) applications. The GiLBCSteg achieves this by hiding data within the compression process of audio signals, a required step in successful live VoIP applications. Specifically, GiLBCSteg is the first study in the literature that alters the linear spectral frequencies indices used by iLBC to encode hidden data within the audio signal. This encoding algorithm not only produces mild distortion, less than 5 dB in difference which is not readily noticeable to the human ear, but also do not transmit the hidden message as a separate file.

Keywords Audio codec, iLBC, Linear Spectral Frequency, VoIP Steganography

1. Introduction

Steganography is the act of hiding a secret message within the contents of a more public one, with the purpose of keeping unintended recipients from knowing the hidden information is there [1]. While the technical aspect of steganography is definitely a science, because steganography deals primarily with the perception of those who are attempting to break it, it can require a more artistic touch compared to a cryptographic approach.

There are many practical applications for steganography. Military, intelligence agencies, and other government entities, often require the use of covert communication. While they definitely use cryptography to hide the contents, sometimes even the sending of a message can be detrimental to some operations, or lead to an attack of the data transmission source, such as through jamming or physical force [2]. Counter intelligence agencies also have an interest in data hiding, as greater understanding and research in the area can lead to a greater effectiveness at detecting concealed data. Entities wishing to avoid detection by federal organizations are also highly interested in concealing data, such as criminals, freedom of speech activists wishing to

circumvent censorship, and those very concerned with data privacy, for obvious reasons [2].

With the internet as wide spread as it is today, it only makes sense that we try to utilize these connections as a more readily available audio communication method than traditional phone lines. This is achieved through the use of Voice over Internet Protocol technologies, or VoIP [3]. More than 270,000 people use VoIP technologies to communicate in western European countries, and over 12% of businesses in the United States have implemented this technology into their day to day business operations [3]. VoIP has become, and is becoming more prevalent. This means that there is a significant amount of audio data traffic on the internet and this makes VoIP and other audio based transmissions useful for steganography.

Audio data tends to be very copious. Because of this, there is a lot of room to hide bits for steganography. This is generally achieved by implementing steganography within an audio codec. An audio codec performs two functions. First, it encodes raw audio signals with esoteric mathematical models into a computer useable format, and then decodes the audio data into sound waves that can be transmitted through speakers and listened to. Second, because of audio data's voluminous nature and the fact that bandwidth can be limited for network transmission; a codec also compresses audio data for transmission and decompresses it again for usability. Thus a codec is required for nearly all usage of digitized audio data, including use in VoIP [4]. Therefore, one is able to hide data within the encoding process of the audio, which can be resilient to steganalysis.

iLBC codec is getting attention from many open source VoIP applications because of its open source platform and being a low bit rate audio codec [5]. Keeping this in mind, in this work we introduce a new framework which alters the linear spectral frequencies used by iLBC to encode the hidden data within the audio signal, namely GiLBCSteg. This will be the first study that embeds secret messages into iLBC codec to test the distortion level and success of the steganography transmission.

In Section 2, other related works in VoIP Steganography is

discussed. The details of the iLBC codec are provided in Section 3. The GiLBCSteg algorithm is explained in Section 4, and the paper is finalized with the results and discussion section at the end.

2. Literature Review

Research in the area of VoIP steganography has been steadily progressing over the years. Using the G.729a codec, Tian, et al. added the traditional least significant bit (LSB) steganography algorithm to VoIP steganography [6]. More research has been done in steganalysis of this sort of algorithm. For example, Liu and associates have developed a LSB audio steganalysis technique that involves examining the Hausdorff distance to measure the amount of distortion in the audio signal [7]. With the results they achieved, they were able to detect the presence of LSB steganography in audio signals [7]. While this method is relatively straight forward, since it is also a well-known and analyzed approach, it is substantially easy to identify this type of steganographic practices.

Another study into VoIP steganography was conducted by Naofumi Aoki [8]. The author used the closed source high bit rate G.711 audio codec to embed semi-lossless steganography in it. High bit rate audio codecs are conducive to steganography for two primary reasons. First, this codec is lossless, meaning that none of the data is lost during the encoding or compression process. Second, this codec contains a significant amount of redundant data. This redundant data makes the algorithm resilient to data loss or corruption, as sometimes happens when transmitting information over a network. The author utilized this fact and replaced the redundant data with the hidden data [8]. This had the advantage of creating absolutely no distortion in the audio signal, assuming the original signal was not corrupted in any manner. As there is no distortion typically induced, it would be impossible for one to be able to detect the presence of steganography utilizing the end audio signal [8]. However, although the high bit rate G.711 codec does make steganography easier and provides some advantages, it also requires a large amount of data to be transmitted. This can be problematic in limited bandwidth situations. It also seems that one who is familiar with this algorithm may be able to defeat it, by comparing the redundant sections of the transmitted data. In other words, the lack of matching redundant audio data in this algorithm would indicate the presence of hidden data.

Another proposed steganography algorithm is the LACK, or Lost Audio Packets method [9]. This method attempts to leverage the RTP protocol that VoIP uses to transverse the network. The RTP protocol is similar to the UDP protocol, in that packets may be lost or received out of order. In a typical VoIP session, if a packet is delayed in reaching its destination, the receiving computer normally just drops it, as it is typically too late to integrate it into the bit-stream and maintain VoIP's real-time quality. The LACK method

utilizes this by purposely delaying packets, and then simply rewriting them with the hidden data. The receiving computer will then, instead of simply dropping the packet, will save and extract the hidden data from it [9]. The problem with this algorithm is that it relies entirely on the assumption that an eaves dropper will simply drop any delayed packet like a normal VoIP receiver would. It seems not unlikely that a malicious middle party would not simply capture all of the packets, and then notice the unusual structure of the delayed packets, thus causing the interloper to examine it more closely.

One novel algorithm was proposed by Huang, et al. [10]. They utilized the proprietary G.723.1 algorithm, which is often used for VoIP communication because of its low bit-rate nature, to conduct VoIP steganography. It achieves this by using a method called pitch period prediction. The codec calculates 4 pitch period predictions per frame. Huang and associates altered the pitch period predictions slightly allowing for up to one bit of hidden data to be encoded per pitch period prediction, thus allowing 4 bits per frame. Modifying the pitch period predictions just slightly keeps the amount of distortion from becoming noticeable, while hiding the data within the algorithm itself. Therefore, just applying it to the binary of either the input or output signal of the algorithm makes it harder to detect [10].

As indicated above, most of the research into VoIP steganography has been done on closed source or proprietary high-bit rate audio codecs. High bit-rate codecs have redundant data in them, allowing arbitrary data to be hidden in these spots without distorting the signal. While this can be effective, high bit-rate audio codecs can also be too demanding on bandwidth to be conducive to VoIP transmissions. Also being a closed source audio codec, the exact inner workings of the codec cannot be examined. Because of the stated reasons, there is a dire need for creation of a VoIP steganography algorithm that could be applied to an open-source low bit rate audio codec. Therefore, in this study iLBC algorithm codec was used to hide data inside the VoIP telecommunications.

3. The iLBC Codec

The iLBC algorithm is an open source audio codec, used primarily for compressing audio for use in VoIP applications [5]. Since it was designed to run in less than ideal circumstances, iLBC limits the amount of sound degradation that occurs in situations that cause packets to be lost or delayed. In addition to this, the computational complexity is similar to that of other proprietary audio codec, such as the G.729A codec [5]. It is designed to take uncompressed 16-bit PCM (Pulse-code Modulation) audio with a sample rate of 8000 Hz as input. iLBC supports payload bit rates of either 13.33 kilobits per second with an encoding length of 30 ms, or a bit rate of 12.50 kilobits per second with an encoding length of 20 ms [5]. In this particular work we have based our experiments for audio samples with a length of 30 ms. The

30 ms frames are divided into 240 audio samples which are organized into 6 sub-blocks/frames of 40 samples. iLBC processes each block without looking at the next block, so that if some packets are dropped on a network, the signal isn't interrupted significantly. The iLBC algorithm send the signal through a filter to cut off any frequencies above 90 Hz, as frequencies above this don't add much to the quality, but would increase the packet size. After filtering out this noise, the algorithm then uses the autocorrelation method and applies the Levinson-Durbin recursion to the resulting Toeplitz matrix. This results in two sets of 10 linear predictive coding, a form of pitch period prediction, or LPC coefficients [5].

The first set of LPC coefficients is centered over the second sub frame, while the second set is centered over the fifth sub-frame. Working with LPC coefficients is computationally complex. Therefore, these two sets are converted into two sets of linear spectral frequencies, or LSFs. These two sets are then quantized using a memory-less split vector quantization method, leaving 3 quantization indices for each set, with the first representing the first 3 LSF coefficients, the second representing the next 3, and the third representing the last 4 LSF coefficients. The quantized index numbers then only need six bits to represent the first index,

and seven bits to represent the second and third index. This is a total of twenty bits per set, for a total of 40 bits per block. Because the decoder requires these coefficients, they are sent along as part of the packet.. The LSF coefficients are then interpolated and used to calculate the residual for all of the sub-blocks. The iLBC algorithm then examines the residuals of all of the sub blocks and finds the two consecutive sub-blocks with the highest energy. It then selects the two halves of these two sub-blocks with the highest energy. These two halves are referred to as the start state and are encoded with scalar quantization. The start state is then used in the construction of a dynamic codebook that is then used to encode the rest of the block. Once this is done, iLBC packs all of these into the bit stream, and sends it down the network to the receiving side. Once the encoded block is received by the decoder, it extracts and interpolates the LSF coefficients, and extracts and decodes the start state. These two pieces of data are then used to reconstruct the codebook that was used by the encoder. This codebook is then used to decode the rest of the block. Once this is done, iLBC sends the signal through an enhancement algorithm to improve quality. If the packet is lost, the algorithm uses a packet loss concealment algorithm to attempt to lower the amount of distortion.

Table 1. GiLBCSteg Encoding Algorithm

Algorithm:

```

1: ENCODING
2: COLLECT random samples of 10 seconds, 60 seconds, 300 seconds, 600 seconds, and 1200 seconds from data set
3:   READ input from user
4:   IF: input = 1
5:     START Encoding Process
6:     RUN initEncode and initcount functions to initialize the data
7:     LOOP:
8:       READ bytes from the steganography file
9:       RUN binstring to generate 8 bit representation of each byte
10:      STORE representation
11:    END LOOP:
12:    START encoding the audio one block at a time
13:    IF: (LSFindex >= 7) //make sure that the shorter one is left unaltered before manipulating
14:      LOOP:
15:        COLLECT next bit (dBit) of stega. data to be inserted into the encoded audio
16:        IF: LSFindex %2 == 0  && dBit == 0
17:          LSFindex = LSFindex
18:        IF: LSFindex %2 == 1  && dBit == 0
19:          LSFindex++
20:        IF: LSFindex %2 == 0  && dBit == 1
21:          LSFindex++
22:        IF: LSFindex %2 == 1  && dBit == 1
23:          LSFindex = LSFindex
24:      END LOOP:
25:    END IF:
26:    Rest of the block is encoded and inserted into the bit stream as described in RFC 3951.
27:  END IF:

```

4. The GiLBCSteg Steganography

Class 1 bits are given the greatest amount of protection by the uneven level protection (ULP), of the RTP protocol [11], thus giving the greatest possible protection for the integrity of the hidden data, while at the same time causing the least amount of distortion. Because of these, the steganography algorithm proposed in this paper attempts to provide data hiding by altering the quantized LSF coefficients, since they are the largest value that falls within the class 1 bit. The hidden message can then be extracted on the receiving end

by examining the last bit of the extracted LSFs. In order to reduce distortion, only the larger of the LSF indices in each set are used as shown in the GiLBCSteg Encoding Algorithm pseudo code (Table 1). This allows for the encoding of 1 nibble, or 4 bits, of data into each block. To provide a frame of reference, a 60 second audio sample contains approximately 24,000 blocks of data are used. This would be enough space for about 1 kilobyte of hidden data to be sent every 5 seconds.

Figure 1. Spectral Frequency Analysis of the GiLBCSteg Algorithm

Figure 2. Comparison of Differences in dB Level

In order to test the algorithm, 20 minute voice recordings have been created. As a first task, random samples are collected from the data sets. The length of these excerpts lasted ten seconds, sixty seconds, five minutes, ten minutes, and twenty minute of the original recording. The unmodified iLBC codec was run on the test cases to create a baseline for comparison. Also, same data set applied to the steganography algorithm (modified iLBC). In order to test the robustness of this method for steganography, rather than sending a test message through the algorithm, instead it was set it up to change every linear spectral frequency so that the last bit was always modified. This would add the greatest amount of distortion that any hidden data could theoretically produce. At the end, a spectral frequency analysis was applied against each sample to test the success of the algorithm.

The spectral frequency analysis results of our algorithm are shown in Figure 1. The graphs are presented with the largest and most broad sample at the top, and descending with the smallest sample at the bottom. The unmodified decoded audio signals are on the left, whereas the ones on the right are the decoded audio signals that had the steganography algorithm applied to. Differences are small, although somewhat more noticeable in the analyses of the smaller speech samples.

In order to achieve a closer examination, the differences between dB levels of various frequencies are analyzed (Figure 2). The average young adult human ear can typically hear sound levels in a frequency range of between approximately 20 Hz and 1600 Hz [12]. Taking this into consideration, frequencies near the lowest and highest frequencies in the average human perceptible range were chosen, as well as three additional between those frequencies.

After comparing the audio samples, it was found that the unaltered audio samples consistently had a higher dB level

than the samples that had been altered, although most of the time the difference between the two stayed within 5 dB, as shown in Figure 2. The difference in dB level did exceed 5 dB at some points, with the difference in dB level for set A reaching 6.087307 dB at 2585.9375 Hz, but this was above the range a typical human can hear. Therefore, it can be neglected. The algorithm provided similar results in a low-bit rate audio codec framework compared to other non-VoIP audio steganography algorithms. Also, based on these results, it is nearly impossible that an individual simply listening to the recording would be able to discern the presence of the steganography data. Moreover, the steganography algorithm is not able to be detected by any of the well known steganalysis techniques currently available, since the algorithm provides a way to embed hidden messages in the LSF Coefficients, causing least amount of distortions which makes steganalysis harder to accomplish. However, it is possible that it may be discerned by comparing the cover medium to the original unaltered audio, which is true for all of the steganography algorithms. We should keep in mind that in VoIP situations, which this is intended for, there should be no unaltered media for comparison.

5. Discussions and Conclusion

VoIP-ready smartphones and VoIP ISP's have an extensive increase in number of users. These new gadgets/technology are started to be used as part of steganography experiments for good or harmful usage purposes. Most of the current research in VoIP steganography has focused on closed source or proprietary high-bit rate audio codecs. However, the algorithm proposed in this paper leverages the linear spectral frequencies of the open source iLBC codec. The presented algorithm provided

close results in iLBC codec framework compared to other non-VoIP audio steganography algorithms.

Using an open source baseline, the research described here is intended to serve as a springboard for more research in the area of VoIP steganography. Currently, although some steganalysis methods are present, they are not universal and efficient enough to be practically deployed in the network to perform real-time detection. Therefore, as a future work in this area we would include a steganalysis solution for the algorithm we created. Additionally, other variables in the iLBC codec could be examined potentiality for data hiding.

REFERENCES

- [1] Tactics and Tools for Security Practitioners, Waltham: Syngress, 2011.
- [2] F. A. P. Petitcolas, R. J. Anderson, M. G. Kuhn. Information Hiding : A Survey. IEEE Special Issue on Protection of Multimedia Content, 1999.
- [3] M. Larsen. VOIP statistics and Predictions. [Online]. Available: <http://www.thevoipstore.net/articles/voip-statistics-and-predictions.html>. [Accessed 27 April 2012].
- [4] excITingIP.com. Licenced & Open Source Voice Codecs in IP Telephony (VOIP) – A brief Introduction. 18 January 2011. [Online]. Available: <http://www.excitingip.com/1088/licenced-open-source-voice-codecs-used-in-ip-telephony-voip/>. [Accessed 27 August 2012].
- [5] S. Andersen, A. Duric, H. Astrom, R. Hagen, W. Kleijn, J. Linden. Internet Low Bit Rate Codec (iLBC). IETF RFC3951. 2004.[Online]. Available: <http://www.ietf.org/rfc/rfc3951.txt>. [Accessed 7 February 2012].
- [6] H. Tian, K. Zhou, Y. Huang, D. Feng J. Liu. A Covert Communication Model Based on Least Significant Bit Steganography in Voice over IP. The 9th International Conference for Young Computer Scientists, 18-21 November 2008, Hunan, China.
- [7] Y. Liu, C. Corbett, R. Archibald, B. Mukherjee, D. Ghosal. A Novel Audio Steganalysis Based on High-Order Statistics of a Distortion Measure with Hausdorff Distance. 11th International Conference on Information Security, 15-18 September 2008, pp. 487-501, Taipei, Taiwan.
- [8] N. Aoki. A Semi-lossless Steganography Technique for G.711 Telephony Speech. Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 15-17 October 2010, Darmstadt, Germany.
- [9] M. Hamdaqa, L. Tahvildari. ReLACK: A Reliable VoIP Steganography Approach. Fifth International Conference on Secure Software Integration and Reliability Improvement, 27-29 June 2011, Jeju Island, Korea.
- [10] Y. Huang, C. Liu, S. Tang, S. Bai. Steganography Integrating into Low-bit Rate Speech Codec. IEEE Transactions on Information Forensics and Security, Volume 7, Issue 6, pp. 1865-1875, December 2012.
- [11] A. Duric, S. Andersen, S. Real-time Transport Protocol (RTP) Payload Format for internet Low Bit Rate Codec (iLBC) Speech. IETF RFC3952. 2004. [Online]. Available: <http://www.ietf.org/rfc/rfc3952.txt>. [Accessed 12 July 2012].
- [12] K. Daqrouq, I. N. Abu-Isbeih, O. Daoud, E. Khalaf. An investigation of speech enhancement using wavelet filtering method. International Journal of Speech Technology, June 2010, Volume 12, Issue 2, pp 101-115.