Journal of Applied Mathematics & Bioinformatics, vol.5, no.3, 2015, 49-72
ISSN: 1792-6602 (print), 1792-6939 (online)
Scienpress Ltd, 2015

A Novel Audio Steganography Telecom System

Antonios S. Andreatos[1] **and Apostolos P. Leros**[2]

Abstract

This paper presents a novel telecommunication system based on audio steganography. Cryptographic as well as steganographic techniques have been employed to increase security. Initially the plaintext is encrypted using a symmetric cryptographic function based on a pseudo-random number generator (PRNG1), to produce the ciphertext; next, the ciphertext is randomly and evenly distributed (i.e., hidden) throughout a cover audio signal by means of a second PRNG, to produce the stego-signal. The resulting stego-sound is transmitted over the (insecure) channel. At the Receiver, the inverse operations take place, but in reverse order. Matlab simulation results as well as steganalysis tests which demonstrate increased security are presented. Using this system, two users may exchange confidential text messages hidden in attached audio files - among other applications.

[1] Division of Computer Engineering and Information Science, Hellenic Air Force Academy, Dekeleia Air Force Base, Dekeleia, Attica, TGA-1010, Greece.
E-mails: aandreatos.hafa@haf.gr, aandreatos@gmail.com.

[2] Department of Automation School of Technological Applications, Technological Educational Institute of Sterea Hellas, 34400 Psachna, Evia, Greece.
E-mail: lerosapostolos@gmail.com.

Article Info: *Received* : November 7, 2014. *Revised* : February 5, 2015.
Published online : September 15, 2015.

Mathematics Subject Classification: 94A60; 68P25; 11K45; 65C10
Keywords: Steganography; cryptography; steganalysis; cover audio signal; stego-system; random number generator; simulation; Matlab

1 Introduction

1.1 Cryptography alone is not enough

The recent boom in telecommunications worldwide and the transmission of personal and sensitive data via the Internet have raised the need for confidentiality. This is usually achieved via cryptography. Cryptographic techniques are widely used to encrypt the plaintext, transfer the ciphertext over the Internet and decrypt the ciphertext to extract the plaintext at the receiver side. However, a hacker or an intruder can easily perceive that the information being sent on the channel has been encrypted, since the ciphertext does not make sense. This can trigger the curiosity of a malicious hacker or intruder to conduct cryptanalysis attacks on the ciphertext (i.e., analyze the ciphertext vis-à-vis in relation to the encryption algorithms and decrypt the ciphertext completely or partially) [1]. For this reason, it would be more prudent to hide the secret information (either in plaintext or in ciphertext) by cleverly embedding it as part of a cover media (for example, an image, audio or video carrier file). In this way the existence of the hidden information cannot be easily perceived by the unintended recipients of the cover media. This idea forms the basis of steganography, which is the practice of hiding information by embedding the hidden (secret) message within other file types such as images, audio, video, text/pdf files, etc. [2]. This paper presents a confidential text communication system based on a combination of cryptography and steganography. To simplify the simulation as well as the presentation, encryption is performed by simply performing logical XOR between the cleartext and a Pseudo-Random Number Sequence (P-RNS). For this purpose Matlab's Pseudo-Random Number Generator (PRNG) which provides satisfactory results [3] has been used. However, classic symmetric encryption algorithms such as the Data Encryption Scheme (DES), the Advanced Encryption Scheme (AES) and the International Data Encryption Algorithm (IDEA) could have been used as well.

1.2 Introduction to audio steganography

Some widely accepted definitions of steganography are the following: Steganography is the effort of secret communication by hiding information in different kinds of data, usually redundant, without raising suspicions. Steganography involves hiding information in a cover (carrier) media to obtain the stego media, in such a way that the cover media is perceived not to have any embedded message for its unintended recipients [1]. According to Liu et al. [4], steganography is the art and science of hiding a secret message within an innocuous and open carrier medium, such as digital audio, image and video. To achieve covert communications without raising suspicion, media containing some hidden information (stego-objects) should be indistinguishable from media without any hidden information (cover-objects).

Typically, redundant files such as images are used as cover objects. Audio signals have an inherent redundancy and unpredictable nature, features which make them ideal for use as cover objects. This is further aided by the fact that the human ear is insensitive to small distortions of an audio signal. All these make audio a good candidate for use as a 'cover' medium to hide secret messages [4]. Wide use of the Voice over Internet Protocol (VoIP) and various Peer-to-Peer (P2P) protocols and audio services offer numerous opportunities for covert communication [4].

Free as well as commercial audio steganography software is widely available. Formats suited for injection include WAV, PCM, AVI, MIDI, MPEG, MP3, RIFF and VOC [2]. In this work we deal with WAV cover files.

1.3 Audio Steganography Algorithms

Digital audio provides a suitable cover for high- throughput steganography as a result of its transient and unpredictable characteristics [4].

In general, the most commonly used methods of audio steganography are [5],[6],[7]:

1. Least Significant Bit (LSB) coding;
2. Parity coding;
3. Phase coding;

4. Spread spectrum;
5. Echo data hiding;
6. Hiding the information in the frequency domain using Direct Cosine Transform (DCT) or Fast Fourier Transform (FFT) [8].

1.4 Steganalysis basics

Steganalysis is the science of detecting the presence of hidden data in cover media files. Meghanathan and Nayak define steganalysis as the mechanism of detecting the presence of hidden information in the stego media [1]. Steganalysis is emerging in parallel with steganography. Steganography and steganalysis have received a lot of attention in the past few years [1]. Most steganalysis techniques employ some form of statistical analysis or frequency spectrum analysis to determine the probability that the digital information distributions would naturally occur. Common steganalysis methods include pair analysis, discrete cosine transform (DCT) statistics, joint probability distributions and matching [9]. Steganalysis tools that detect and reveal steganography also exist [2], [12].

Compared to image steganalysis, audio steganalysis is relatively unexplored [4]. Johnson et al. [10] proposed a universal steganalysis algorithm that exploits the statistical regularities of recorded speech. Liu et al. [4] adopted audio quality metrics to capture the distortion introduced by the hidden information. Avcibas [11] proposed an audio steganalysis algorithm using content-independent distortion measures. All these audio distortion measurements seek ways to detect the presence of hidden messages using existing quality metrics. Audio quality metrics are used to capture the distortion introduced by the hidden information.

2 Related work

The idea of combining cryptographic and steganographic techniques to improve security is not new. In the past, similar systems have been presented.

Cryptography complements steganography rather than replacing it [8]. If the hidden message is encrypted, it must also be decrypted if discovered, which provides another layer of protection [13].

In 2012 Andreatos and Leros presented an image steganography telecom system for encrypted data based on a Chua's circuit chaotic noise generator [14]. In 2013, Andreatos and Leros presented a sound steganography telecom system for data cryptography, based on a Chua's circuit chaotic noise generator [15]. Also in 2013, Pitropakis et al. presented a steganographic technique for embedding hidden messages to Matroska multimedia containers [16]. The ciphertext is hidden in the audio file (WAV) of the Matroska container. In 2014, Leros and Andreatos presented a video steganography telecom system for encrypted data, based on a Chua's circuit chaotic noise generator over a reliable, noise-free communication channel [17]. Gupta et al. [6] used the RSA algorithm and the Diffie-Hellman algorithm to encrypt the data. The encrypted data substitute plain textual data in the image using the LSB substitution method.

3 System description

The objective of the proposed system is to transmit confidential text messages over a reliable Internet connection (TCP) using cryptographic and steganographic techniques.

3.1 The encryption process

Figure 3 presents a simplified block diagram of the proposed crypto-stego-system. The components of the system are: the secret message (cleartext), the encryption algorithm, the embedding algorithm, the secret (encryption and steganography) key, the cover (audio) signal, the Stego-Encoder (Transmitter), the (insecure) channel which is supposed to be error-free (i.e., reliable transmission over TCP), and the Stego-Decoder (Receiver).

The system uses symmetric cryptography based on a secret key. Since the initial message is English text, the first step is to encrypt the cleartext. This is done in three distinct phases. First the text is converted into ASCII

decimal numbers. Then, pseudo-random numbers are added to the numerical cleartext in order to alter the letter frequency and make the result robust against statistical cryptanalysis based on the letter frequency of the English language (Fig. 1). Figure 2 shows a letter frequency histogram of a real test file used; the similarity to the histogram of Figure 1 is evident.

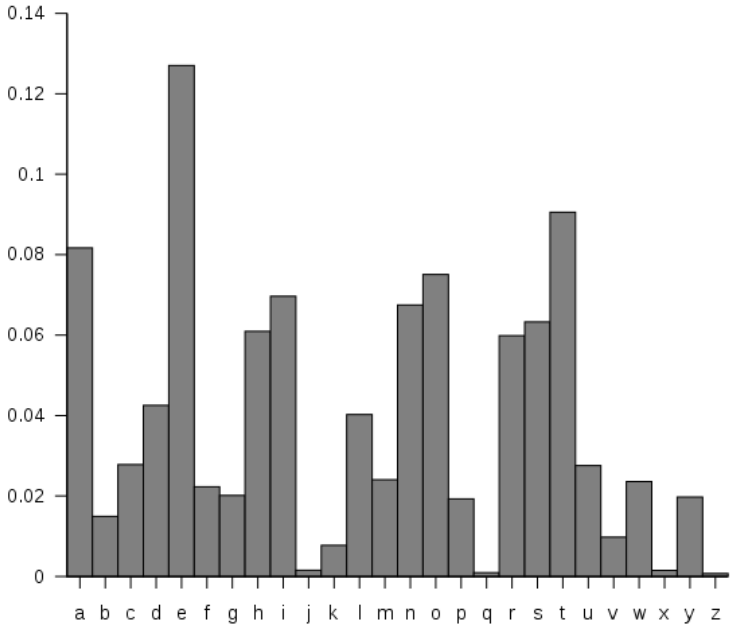

Figure 1: English letter frequency

Next, the result is converted into binary numbers and encrypted using the bitwise XOR function between that and an RNS. In our case a pseudo-RNS has been used. The result is the ciphertext. Next, the steganographic process follows.

3.2 The steganographic process

The cover media used in steganography are WAV audio signals. The system works with both mono and stereo audio signals. For best results, the length of the cover audio signals used should exceed by far the length of the ciphertext. The system computes the length of the cover audio signal to be used. Next, the ciphertext is padded with zeros up to the length of the cover audio signal. Then, this string is randomly and evenly shuffled using a second PRNG, to

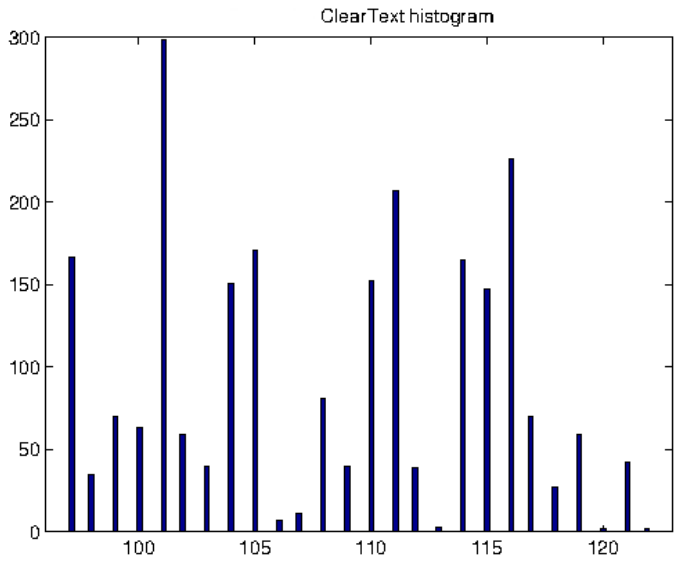

Figure 2: An example letter frequency histogram of a test file used

produce the scrambled ciphertext. The amplitude of this signal is then properly degraded, and the result is added to the cover audio signal to produce the stego-audio signal.

At the receiver, the scrambled ciphertext is first extracted from the incoming stego-signal; next, the ciphertext is reassembled using another instance of PRNG2; then, an instance of PRNG1 is used to produce an identical pseudo-random sequence which is subtracted from the ciphertext, in order to produce the initial plaintext. The whole system has been simulated in Matlab and its operation is described in the next section.

4 System operation

Using the proposed system we can encrypt clear text messages (in ASCII). The following example presents the encryption of a paragraph from Pericles' Funeral Oration as recorded by Thucydides (II, 41):

Example cleartext: "Rather, the admiration of the present and succeeding ages will be ours, since we have not left our power without witness, but have shown it by mighty proofs; and far from needing a Homer for our panegyrist,

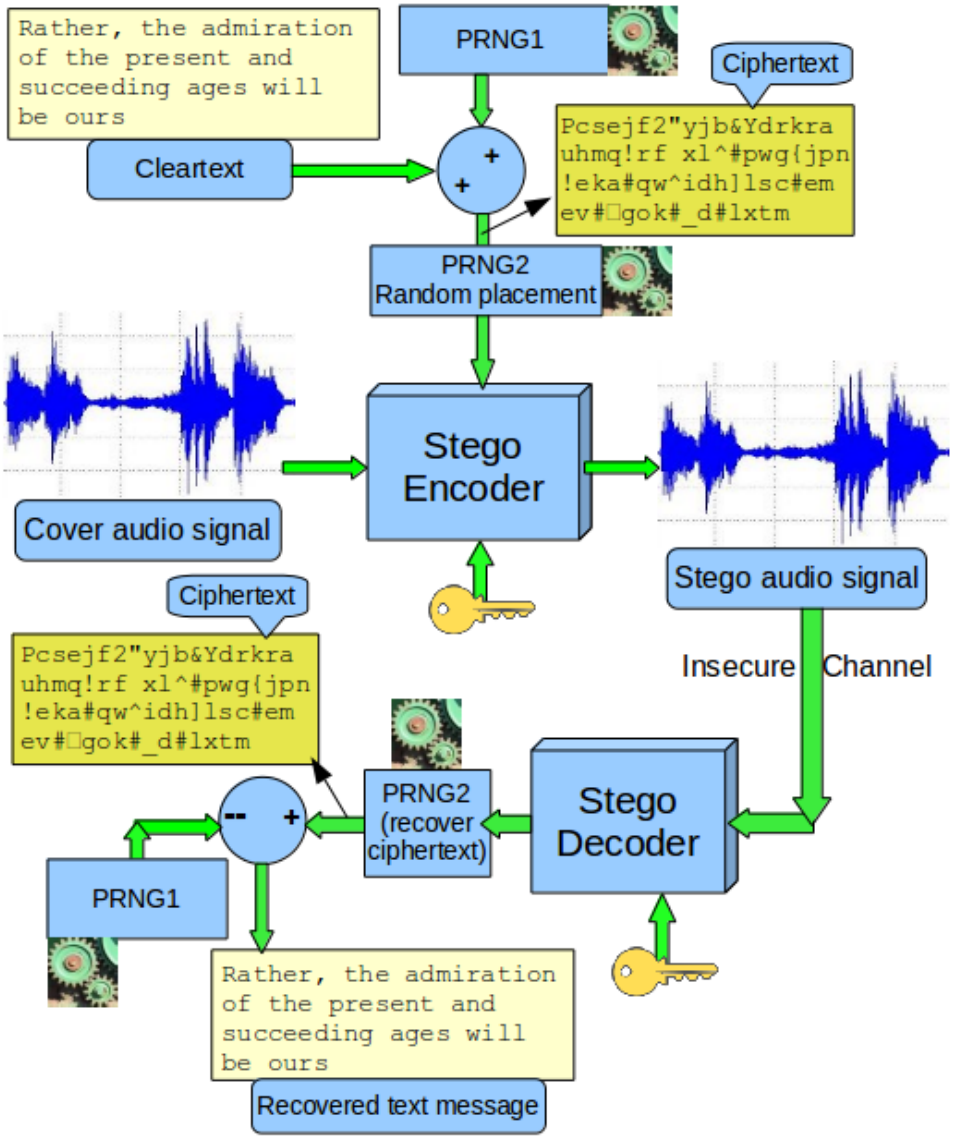

Figure 3: Simplified block diagram of the proposed stegosystem

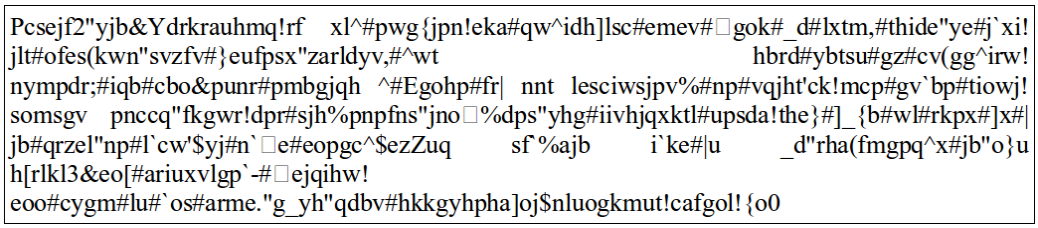

Figure 4: The corresponding ciphertext

or other of his craft whose verses might charm for the moment only for the impression which they gave to melt at the touch of fact, we have forced every sea and land to be the highway of our daring, and everywhere, whether for evil or for good, have left imperishable monuments behind us".

The corresponding ciphertext is shown in Figure 4, where '#' and boxes represent non-printable characters.

In this work we have used three different cleartext test files (in English): Two different quotes from Pericles' Funeral Oration and a text from Wikipedia (lemma 'Sega32'). The results below are from the latter test file, which contains 3840 characters (including 654 spaces).

The next step is to convert the cleartext into integer numbers for processing. The ASCII table has been used. For instance, space corresponds to 32 (20 hex), '1' corresponds to 49 (31 hex), 'A' corresponds to 65 (41 hex) and 'a' corresponds to 97 (61 hex). Printable characters span the range from 32 to 126 [19].

Figure 5 presents the histogram of test input file 'sega32'. As we can notice, space is the most popular character. The leftmost character is CR-LF (carriage return and line feed) which corresponds to decimal 13. Capital letters span the range from 65 to 90. Range from 97 to 122 corresponds to lowercase letters and is comparable to the histograms presented in Figures 1 and 2 above.

Figure 6 shows the cleartext represented as integer numbers.

4.1 Encryption function

The next step is to encrypt the cleartext. Therefore, we need a sequence of random numbers of the same length. This sequence must be regenerated at the Receiver. Algorithmic Pseudo-Random Number Generators (PRNGs)

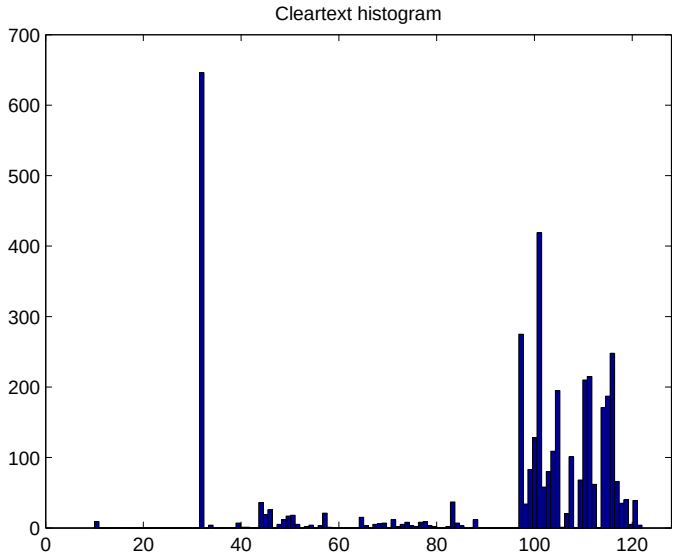

Figure 5: Histogram of test input file 'sega32'

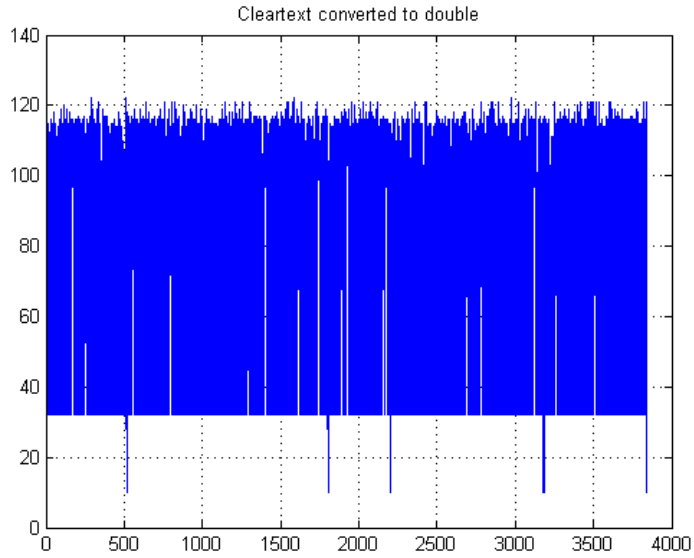

Figure 6: The cleartext represented as a series of integers

as well as Chaotic Random Number Generators (CRNGs) constitute the best candidates [3]. For the purposes of this work, Matlab's PRNG suffices [3]. Alternatively, a CRNG as that described in [15] could have been used. Figure 7 presents the histogram of Matlab's P-RNS which is uniformly distributed in the range [0-255]. It is worth noting that all good quality RNGs (including Matlab's PRNG) produce a histogram which approaches a flat curve as the length of the RNS increases [3], fact which denotes that all numbers in the range [0-255] have an equal probability of appearance.

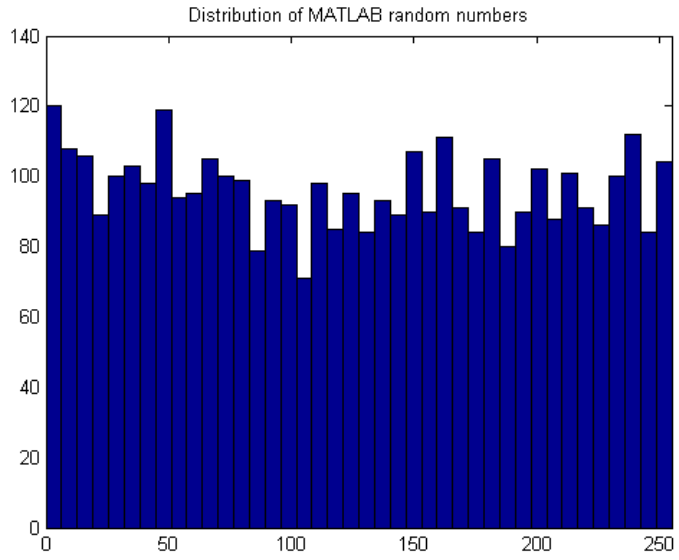

Figure 7: Histogram of Matlab's pseudo-random number sequence

Encryption is done simply by performing the bitwise Exclusive OR (XOR) function between the cleartext and the pseudo-random number sequence. The resulting sequence constitutes the ciphertext and is shown in Figure 8. The quasi-flat histogram of the ciphertext as opposed to the typical histogram of English text (Figures 1, 2) indicate good quality encryption [3], [20], [21].

The histogram of the resulting ciphertext is almost uniform in the range [0-255] and thus independent of the ciphertext (which follows the letter distribution of the language used, in our example the aforementioned distribution of Figures 1 and 2).

As we can see, the characteristic letter frequency pattern is no longer present. More complex encryption schemes such as the one described in [20]

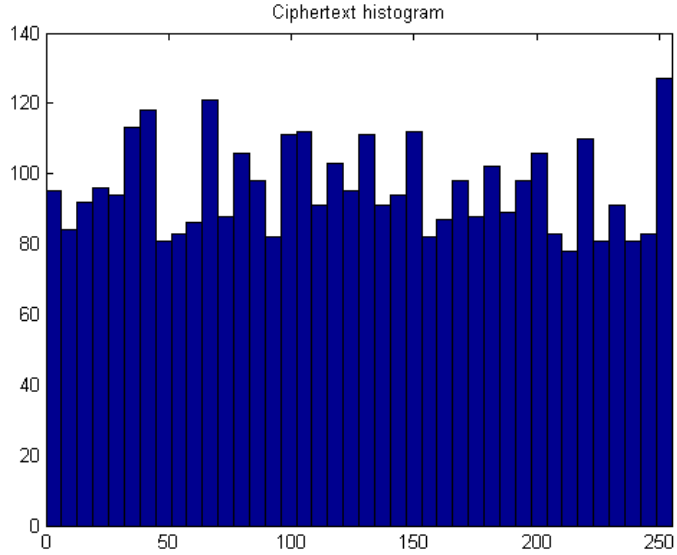

Figure 8: Histogram of the ciphertext

could also have been used. Next the ciphertext is converted into a sequence of real numbers (Fig. 9).

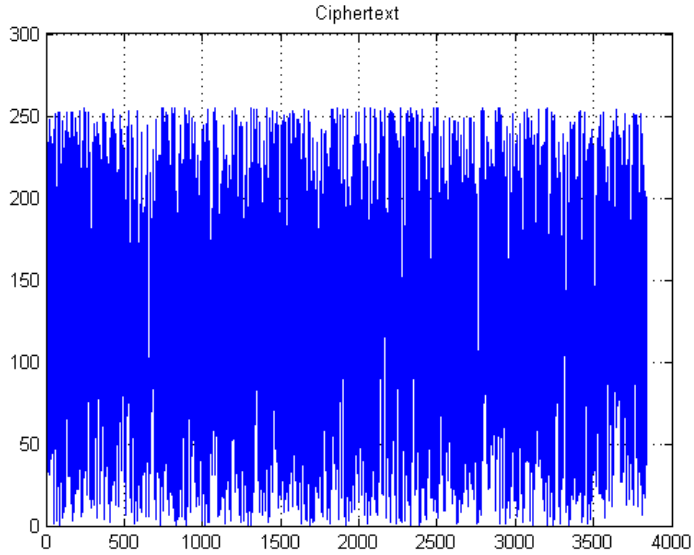

Figure 9: The ciphertext as a waveform of real numbers

4.2 Steganographic function

The steganographic method used here is different than those presented in paragraph 1.3. The method was initially presented by Andreatos and Leros in 2013 [15] and consists of three steps: First, the ciphertext string is padded with zeros to produce a much longer string equal to the length of the cover audio signal. The ciphertext waveform amplitude is compared to the amplitude of the cover audio stego signal and scaled down appropriately, in order to become unnoticeable. The gain (which is less than 1) is needed at the Receiver; subsequently, it makes part of the key. Figure 10 presents the cover audio stego signal.

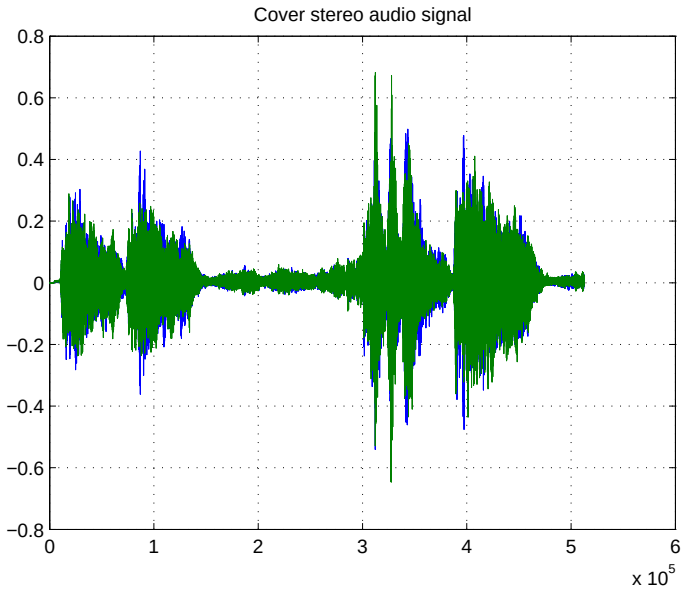

Figure 10: Cover audio stego signal

The second step is to shuffle the ciphertext and spread the samples throughout the whole length of the cover (digital) audio signal (in our example, 512453 samples). Here, a second PRNG has been used (PRNG2 in Figure 3).

Third, the two waveforms are finally added to make the steganographic audio signal to be transmitted to the Receiver through the insecure channel, for example, as an email attachment [17]. Figure 11 presents the stego signal.

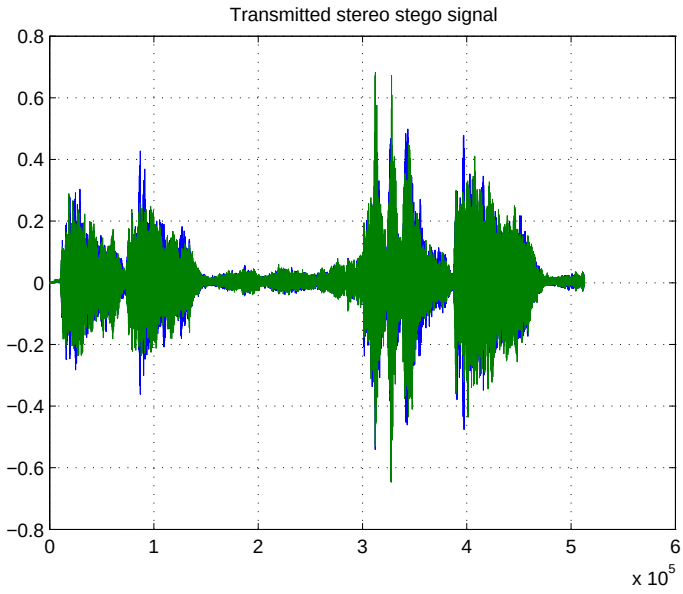

Figure 11: Stego signal transmitted over the insecure channel

4.3 Features and advantages

Three important features of stego-systems are capacity, security and robustness [22]:

- Capacity refers to the amount of information that can be embedded in a cover object.
- Security relates to an eavesdroppers inability to detect the hidden information, hence robustness against steganalysis attacks.
- Robustness refers to the amount of modification the stego-object can withstand before an adversary can destroy the information [22].

The capacity of the proposed system equals the maximum length of the ciphertext, which in turn equals the length of the cover signal. For instance, in our simulation,

- the length of the cover signal is 3841 characters or 3841 integers;
- the length of the ciphertext used is 512,453 x2 (double) = 1,024,906;
- the capacity of the presented example is 0.375%.

Security is examined in the next section. Finally, robustness is not an issue here since we have assumed reliable transmission over TCP.

The advantages of the proposed system are:

1. The cleartext uses all 7-bit ASCII characters, that is, not only letters, numbers and punctuation marks but also non-printable characters. Both lowercase and uppercase letters are accepted.

2. The ciphertext expands (almost) uniformly from 0 to 255, that is, contains both printable and non-printable characters, and seems independent from the cleartext.

5 Security issues

5.1 Key space

The key of the system contains the values of the following parameters:

1. The identity (id) code of the cover audio signal. The transmitting and the receiving parties can have a pool of cover audio signals to choose from, according to the length of the message and other factors. About 1000 cover audio signals is a reasonable number. Although 10 bits are enough, for protection reasons against brute-force attacks, not all of the ids should be meaningful; thus, we shall allocate 32 bits.

2. The amplification factor for mixing the ciphertext with the cover audio signal. In the experiments, values between 0.005 and 0.07 have been used. A reasonable value to use is 12 bits.

3. In our simulations, two PRNGs have been used. The algorithms of these PRNGs may not be constant but they better be selected from a predefined set. A reasonable PRNG id may use 16 bits (2x16=32 bits).

4. The seeds of the PRNG should be the same in the transmitting and the receiving parties. A reasonable assumption is to use 16 bits. In case of a Chaotic RNG, which has many more parameters, many more bits are needed [20] (2x16=32 bits).

5. A timestamp field, related with the date of transmission. Let us assume 8 characters or 64 bits.

The above factors sum up with (at least) 32+12+32+32+64 = 172 bits. In order to make the key longer and stronger, the following technique has been used: we take the message digest (hash sum) of each parameter according to the SHA-2 algorithm (consisting of 128 characters). From these we randomly select 15 characters (120 bits) for each field, thus we have an additional set of 7 x 120 bits = 840 bits. 172+ 840 = 1012 bits.

If we had taken all 128 characters then we would have 128*8 *7 = 7168 bits. Finally, we take the SHA-2 hash of the above and we get 128 characters more, corresponding to 1024 bits. Thus we can reach a key length of 2048 bits which is considered very strong against brute-force as well as other types of attacks with current computing power.

5.2 Cryptanalysis issues

5.2.1 The message is well hidden

The histogram of the cleartext has the typical distribution of the letters of the English language and spans the range of 7-bit binary numbers (0-127); on the other hand, the histogram of the ciphertext is quasi-flat, hence independent of the cleartext, and spans the range of 8-bit binary numbers (0-255). This is a quality indication of the encryption function. The following measures have been taken to enforce the encryption function: The ciphertext contains characters in the range [0-255], many of which are non-printable [19]; hence they don't look like text. The ciphertext is not consecutive but stuffed with zeros and scattered randomly in the cover audio, hence difficult to reconstruct.

Another way to measure the correlation between the cleartext and the ciphertext is to compute the correlation coefficient. By taking the relating function in Matlab we get correlation coefficient CC1 = 0.0112, which is satisfactory. We note that correlation coefficient close to 0 indicates no correlation at all while correlation coefficient close to 1 or -1 indicates strong correlation.

The correlation coefficient between the ciphertext (stuffed with zeros) and the permuted ciphertext is CC2 = -0.0012, even better than CC1. Thus, the

correlation between the cleartext and the permuted ciphertext will be even smaller.

5.2.2 Key strength

Another common test is to decrypt a ciphertext with a different key and check the result (keeping constant the steganography part). Indeed, our system passes this test. For instance, the result of decoding the ciphertext with a different key is as shown in Figure 12, where '#' represents non-printable characters.

Figure 12: Result of decoding the ciphertext with a wrong key

5.3 Steganalysis issues

5.3.1 Correlation of audio signals

In this section the correlation between cover and stego audio signals is presented. We begin by presenting the scatter diagram between the two signals. As we can see in the following Figure 13, the relationship is represented by a 45 degrees straight line, fact which indicates a very strong correlation.

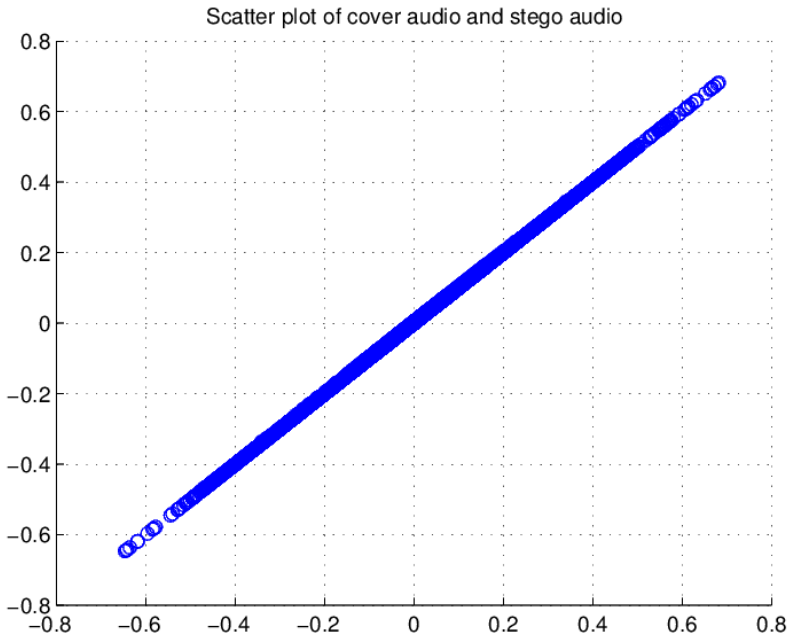

Figure 13: Scatter diagram between cover and stego audio signals

This result is also verified by the correlation coefficient between the cover and stego audio signals which is 0.9998 for degradation factor = 0.07 and 1.0000 for degradation factor = 0.01. Such a high correlation leaves no suspicion that the stego audio signal contains a secret message.

The first kind of tests are the subjective tests. Hearing tests check if the stego signal sounds like a normal audio signal or does it contain noisy artifacts. This is achieved by the degradation of the ciphertext before being added to the cover audio signal. The use of noisy cover audios is another option. Hearing tests check if the visual representation of the stego signal looks like a normal audio signal. The following figure presents a case with poor selection of the degradation factor as well as the cover audio. As a result, noisy artifacts can be observed.

5.3.2 Histogram analysis

A common steganalysis method is histogram analysis [2], [23]. Figures 15 and 16 show the histograms of the original and steganographic audio signals.

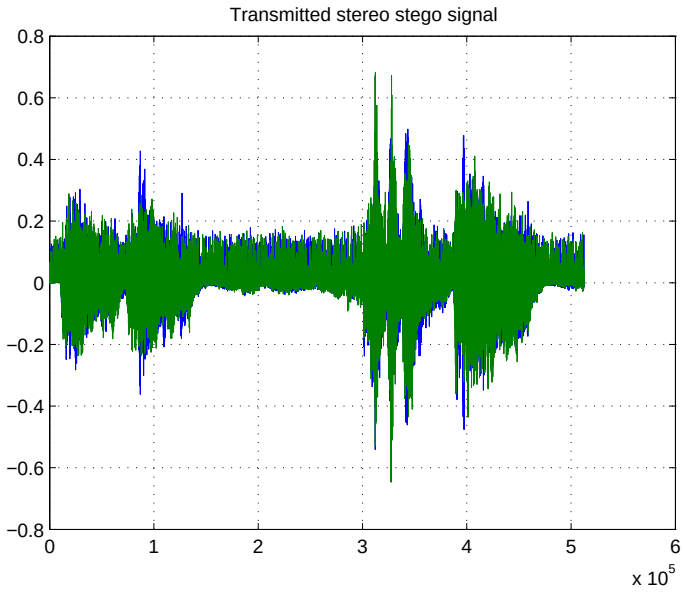

Figure 14: Poor selection of steganographic parameters

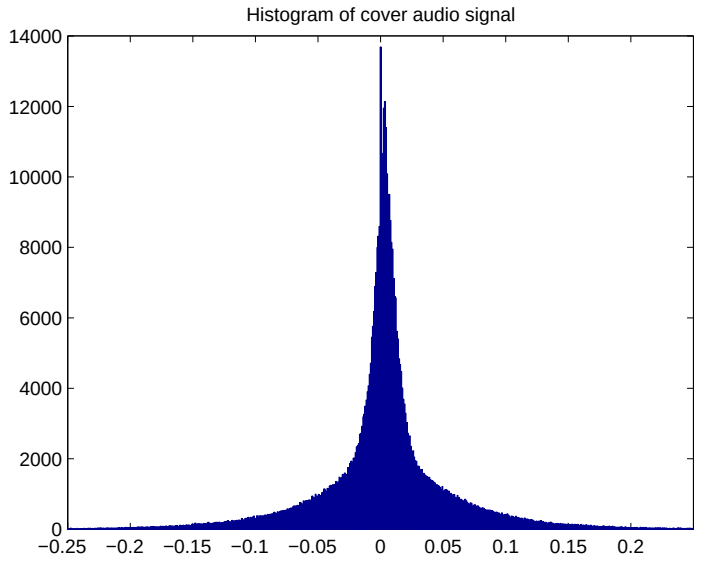

Figure 15: Histogram of the cover audio signal

Comparison of the two histograms verifies that the ciphertext is well hidden against this type of analysis. This happens for two reasons: first because the disturbance is small and second because the ciphertext is placed in the audible (low) frequencies, so that it is completely covered by the signal. Finally, Figure

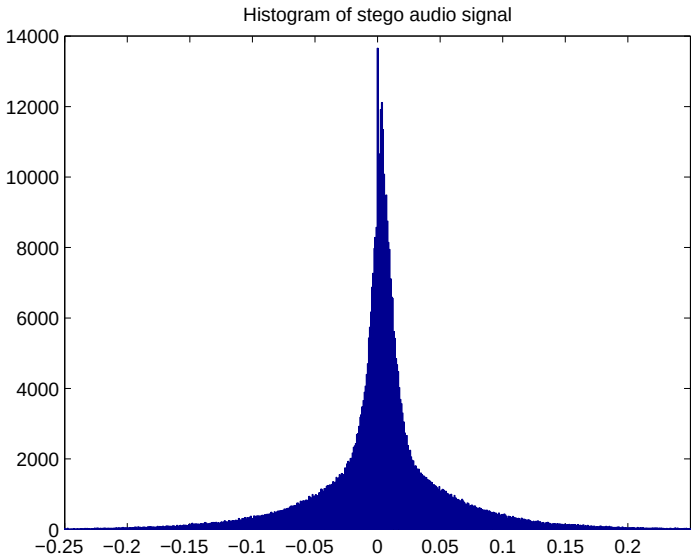

Figure 16: Histogram of the stego audio signal

17 presents the power spectral density of cover and stego audio signals which seem identical.

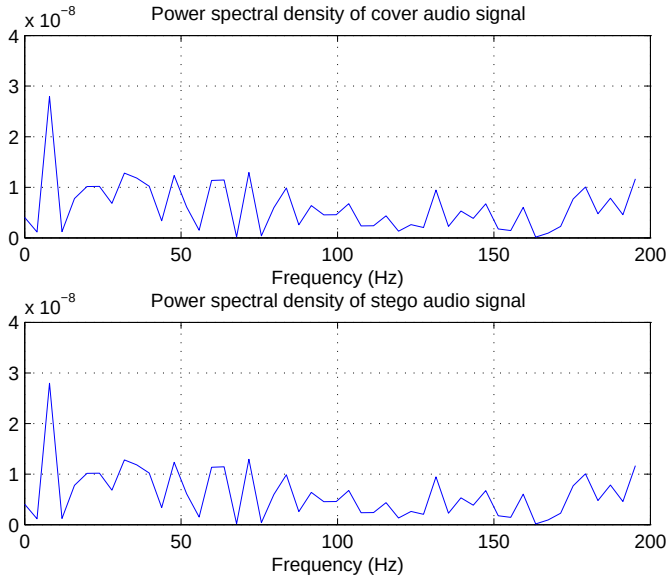

Figure 17: Power spectral density of cover and stego audio signals

6 Discussion and conclusion

In this paper we have presented a novel confidential text communication system for use by Internet applications running over reliable TCP connections. Confidentiality is assured by a double security approach: the cleartext is first encrypted and then hidden in a cover audio signal. The proposed system uses a secret 2048-bit symmetric key.

The advantages of the proposed system are:

1. The cleartext uses all 7-bit ASCII characters, that is, not only letters and numbers but also non-printable characters. Both lowercase and uppercase letters are accepted.

2. The ciphertext expands (almost) uniformly from 0 to 255, that is, contains both printable and non-printable characters, and seems independent from the cleartext.

3. The system design is relatively simple yet robust, since the encryption function is a simple XOR function and the steganographic function is a simple signal addition.

The cover audio signal used in our tests was a clip from Beethoven's 5th Symphony. In general, symphonic music is not the best case for cover audio signals because: a) they usually have high quality, so the ciphertext will produce audible artifacts of low quality which sound like noise; b) they often content short silence periods, which may visually reveal the ciphertext. It is preferable to use noisy signals such as old, analog and noisy recordings from vinyl discs. In this work we used Matlab's PRNG which provides satisfactory results [3]. In a real implementation, a CRNG such as that presented in [21] or [24] would render the system even more robust. Cryptanalysis and steganalysis tests performed and presented here demonstrate satisfactory results. The authors have successfully used the proposed system to exchange confidential messages hidden in attached cover audio files via e-mail.

References

[1] N. Meghanathan and L. Nayak, Steganalysis Algorithms for Detecting the Hidden Information in Image, Audio and Video Cover Media, *International Journal of Network Security & Its Application* (IJNSA), **2**(1), (January, 2010), 43-55.

[2] S. D. Dickman, An Overview of Steganography, James Madison University Infosec Techreport, Department of Computer Science, JMU-INFOSEC-TR-2007-002, (July, 2007).

[3] A. S. Andreatos and A. P. Leros, A comparison of random number sequences for image encryption, *Proceedings of MMCTSE, Mathematical Methods & Computational Techniques in Science & Engineering*, Athens, Greece, (November, 28-30, 2014).

[4] Y. Liu, et al., A Novel Audio Steganalysis Based on High-Order Statistics of a Distortion Measure with Hausdor Distance, *Information Security, Lecture Notes in Computer Science*, **5222**, (2008), 487-501.

[5] M. Nosrati, R. Karimi and M. Hariri, Audio Steganography: A Survey on Recent Approaches, World Applied Programming, ISSN: 2222-2510, **2**(3), (March, 2012), 202-205, www.waprogramming.com.

[6] S. Gupta, A. Goyal and B. Bhushan, Information Hiding Using Least Significant Bit Steganography and Cryptography, *I.J. Modern Education and Computer Science*, **6**, (2012), 27-34.

[7] S. Bhattacharyya and G. Sanyal, Feature Based Audio Steganalysis (FAS), *International Journal of Computer Network and Information Security*, **11**, (2012), 62-73. Published Online October 2012 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijcnis.2012.11.08.

[8] A. Rocha and S. Goldenstein, Steganography and Steganalysis in Digital Multimedia: Hype or Hallelujah?, *Journal of Theoretical and Applied Computing* (RITA), **15**(1), (2008), 83-110.

[9] J. Cazalas, T. Andel and J. McDonald, Analysis and Categorical Application of LSB Steganalysis Techniques, *Journal of Information Warfare*, **13**(3), (2014), http://www.jinfowar.com.

[10] M. K. Johnson, S. Lyu and H. Farid, Steganalysis of recorded speech, In Delp III, E.J., Wong, P.W. (eds.): Security, Steganography, and Watermarking of Multimedia Contents VII, **5681**, SPIE, (May, 2005), 664-672.

[11] I. Avcibas, Audio steganalysis with content-independent distortion measures, *Signal Processing Letters*, IEEE, **13**(2), (2006), 92-95.

[12] G. N. Sarage, Various Aspects of Steganography, *International Journal of Advanced Research in Computer Science and Software Engineering*, **4**(8), (August, 2014). Available online at: www.ijarcsse.com.

[13] N. F. Johnson and S. Jajodia, Exploring steganography: Seeing the unseen, IEEE Computer, **31**, (February, 1998), 26-34.

[14] A. S. Andreatos and A. P. Leros, A Stegosystem with advanced security features - Simulated in Matlab, *Proceedings of PCI 2012, 16th Panhellenic Conference on Informatics*, (October, 5-7, 2012), University of Piraeus, Greece, 111-116. DOI: 10.1109/Pci.2012.55.

[15] A. S. Andreatos and A. P. Leros, An audio steganography system using Chua Chaotic Noise Generator, *Proceedings of Athcon 2013*, Athens, Greece, (6-7 June, 2013).

[16] N. Pitropakis, C. Lambrinoudakis, D. Geneiatakis and D. Gritzalis, A Practical Steganographic Approach for Matroska based High Quality Video Files, *Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops*, Barcelona, Spain, (2013).

[17] A. P. Leros and A. S. Andreatos, *Video Steganography System for Secure Data Communication*, chapter 4.3 in *New Research Trends in Nonlinear Circuits: Design, Chaotic Phenomena and Applications*, Nova Science Publishers, 2014.

[18] Wikipedia, lemma: Letter frequency, http://en.wikipedia.org/wiki/ Letter_frequency, Accessed 5 Nov. 2014.

[19] Wikipedia, lemma: ASCII printable characters, http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters, Accessed 17 Nov. 2014.

[20] A. Andreatos and A. Leros, Secure image encryption based on a Chua chaotic noise generator, *Journal of Engineering Science and Technology Review*(JESTR), *Special Issue on Nonlinear Circuits: Theory and Applications*, **6**(4), (2013), 90-103.

[21] C. K. Volos, Image Encryption scheme based on coupled chaotic systems, *Journal of Applied Mathematics and Bioiformatics* (JAMB), **3**(1), (2013), 123-149.

[22] N. Provos and P. Honeyman, Hide and Seek: An Introduction to Steganography, IEEE *Security & Privacy*, **1**(3), (May-June, 2003), 32-44.

[23] K. Solanki, K. Sullivan, U. Madhow, B. S. Manjunath and S. Chandrasekaran, Provably Secure Steganography: Achieving Zero K-L Divergence using Statistical Restoration, *Proceedings IEEE International Conference on Image Processing 2006* (ICIP06), Atlanta, GA USA, (October, 2006).

[24] A. S. Andreatos and C. K. Volos, Secure Text Encryption Based on Hardware Chaotic Noise Generator, to appear in *Journal of Applied Mathematics and Bioiformatics* (JAMB), (2015).