Analysis of a CSPE-based Audio Watermarking technique from a cryptographic perspective

James Molloy, Petar Stefanov, Darren Healy, Ron Healy, Jian Wang

Correspondence email: **rhealy@cs.nuim.ie**

Abstract

In this paper we intend to analyse, from a cryptographic perspective, a particular audio watermarking technique based on the Complex Spectral Phase Evolution (CSPE) algorithm. We will investigate the scheme mathematically (through the domains of practical cryptography and information security), theoretically (determining the characteristics of the scheme to evaluate its strengths and weaknesses) and experimentally (evaluating the results of the algorithm with different inputs, keys and variables to see if the technique has practical potential). We will investigate whether the technique replicates characteristics of Perfect Secrecy, a One-Time Pad, Random Oracle and the Avalanche effect. We will show how the CSPE audio watermarking scheme under investigation compares to current similar encryption techniques. Finally, we will outline possible areas to improve the technique and provide recommendations that will increase the security of the system so that it may be used in the fields of covert communication, steganography and cryptography.

Keywords

Covert communication, audio watermarking, cryptanalysis, steganography, information security

1.0 Introduction

Steganography, defined as "the art of concealing a message, image or file within another message, image or file", [1] is derived from the Greek words steganos, meaning "protected", and graphei, meaning "writing". This concept, a form of information hiding, has been widely used throughout history. Herodotus, a historian of ancient Greece, tells the story of a message being tattooed on a slave's head, hidden by the growth of his hair and exposed by the message carrier advising the intended recipient to shave his head again. The message allegedly carried a warning to Greece about Persian invasion plans. While there are drawbacks to this method, it highlights the early use of Steganography to hide information.

Audio watermarking, a technique whereby data is hidden in an audio signal, is considered a form of Steganography through its ability to 'hide' the presence of other data. This is done by embedding (adding) the information to be watermarked ω, within a host audio signal *s*, to produce a watermarked signal *s'*.

$$s + \omega = s' \quad (1)$$

Steganography can be combined with Cryptography, the art of protecting information by transforming it into an incomprehensible format, [2] to add an extra layer of security. This increases the strength and security during transmission; even if the presence of the message is discovered, it will be harder to understand.

In this paper, cryptographic concepts will be investigated and discussed to evaluate the relative strength and security of this CSPE based audio watermarking cryptographic system. They include:

- *Perfect Secrecy*: satisfied if the ciphertext (encoded message) gives no extra information about the original plaintext than already known about the plaintext.
- *One Time Pad* (OTP): a type of encryption that, if used correctly, has been proven to be impossible to crack.
- *Random Oracle*: a function mapping each unique input to a relative specific output.
- *Avalanche effect*: a small change in the input will produce a large change in the output.

The system analysed in this paper will be compared against RC4 to identify the similarities between both systems and highlight potential dangers to our system. RC4 is the most widely used software stream cipher at present and is used in protocols such as Secure Socket Layer (SSL) to protect internet traffic and Wired Equivalent Privacy (WEP) to secure wireless networks. While remarkable for its simplicity and speed in software, RC4 has weaknesses that argue against its use in new systems. We will consider these weaknesses while evaluating our system.

The CSPE-based audio watermarking algorithm will be discussed in more detail in Section 2. The system will be investigated mathematically and theoretically in Section 3. Experimental results will be presented in Section 4. We will also discuss the potential weaknesses and strengths of this system and provide recommendations in Section 5.

2.0 CSPE Background

The Complex Spectral Phase Evolution (CSPE) algorithm, introduced by Short and Garcia in [3], can be used to analyse and detect the presence of frequency components that exist in an audio signal. It has been shown in [4] that, due to the algorithms computationally efficient design, it provides a more accurate set of results than other signal decomposition techniques such as Discrete Fourier transform (DFT).

2.1 Audio-based Covert Encryption System

The CSPE algorithm can be used as the basis for an Audio-based Covert Encryption System (ACES) for security purposes, by introducing cryptographic aspects such as encryption (the process of transforming a message into a coded message) into the process. While [4] addressed the use of audio watermarking as a technique for content identification and tracking, the basic system in [4] can be adapted to provide a platform for covert communications and secret data transmission. The main principle of the encryption process is displayed in Figure 1.

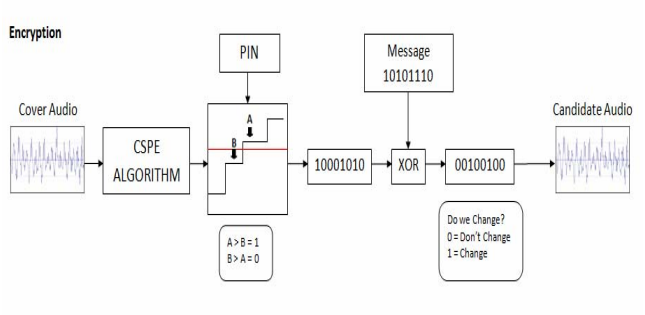

Figure 1: Audio-based Encryption Process

The signal intended as the cover or host audio is segmented into frames of uniform length depending on the length of the plaintext. The frame length is derived by dividing the length of the cover signal in samples (S_t) by the length of the plaintext in bits (P_l). This provides the maximum frame size (F_{max}) required to embed the plaintext in the whole audio signal exactly once.

$$S_t / P_l = F_{max} \quad (2)$$

Note, as explained in [4], that it is clearly possible to repeat a plaintext message multiple times within a cover audio signal to increase robustness and accuracy of recovery of the message. However, for the purpose of this paper, repeat embedding is not considered in any great detail primarily as the intended application domain is different and in view of the *One Time Pad* concept

For each frame of the cover audio, an FFT (Fast Fourier transform) analysis is performed twice; firstly on the signal of interest (S_0) and secondly on the same signal, but shifted one sample in time (S_1). The FFT is an efficient algorithm to compute the DFT (Discrete Fourier transform) [5]. The DFT analyses frequency components present in a sample created from a function, such as an audio signal.

Figure 2 illustrates the CSPE algorithm proposed by Short and Garcia in [3].

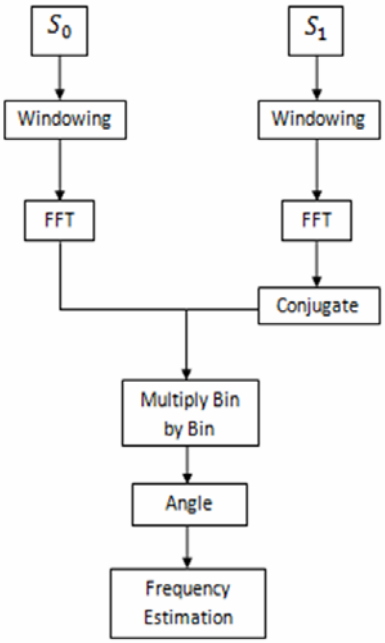

Figure 2: The Flow Diagram of CSPE

To improve the accuracy of the FFT, both samples are first passed through a window. This helps to minimise the problem of spectral leakages (energy from one frequency leaks into another frequency) in the sample. Then, by multiplying the initial FFT spectrum by the complex conjugate of the sample-shifted FFT spectrum, a frequency dependent function is formed from which the exact values of the frequency components in the signal can be detected.

The algorithm produces a graph with a staircase-like appearance, illustrated in Figure 3, where the horizontal parts indicate the exact frequency components in the signal.

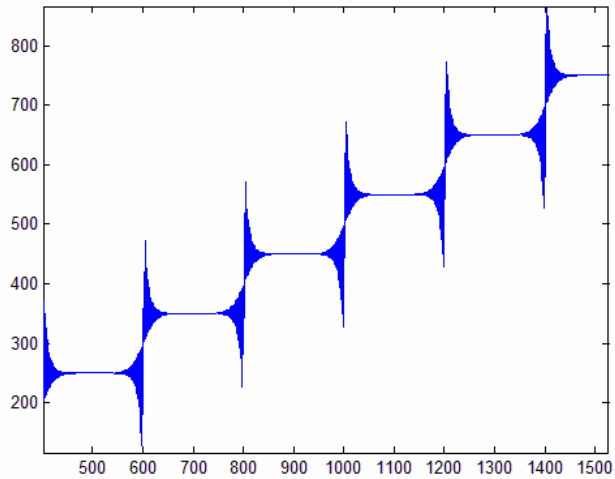

Figure 3: Output of the CSPE process for a multi-frequency signal and a NutallC3 window

It can be seen clearly from Figure 3, that the signal analysed contains frequency components at 250 Hz, 350 Hz, 450 Hz, 550 Hz, 650 Hz, 750 Hz and 850 Hz.

In order to 'embed' data, the user first chooses a value, for example, an arbitrary value within the human hearing range (≈ 100 Hz – 20,000 Hz), known as the user-defined Personal Identification Number (PIN). The PIN is used as a reference value during the ACES process for each frame so that the closest frequencies above and below the reference value (user-defined PIN), that actually exist in the cover signal, can be identified. Note that there is no need for a frequency component to exist which matches the user-defined value.

The magnitudes (the size of the sound wave, a value between 0 and 1) of the obtained frequencies are then compared using the following rules.

$$A > B = 1; \quad B > A = 0 \tag{3}$$

According to this rule (3), if the magnitude value of the component above the user's reference value (*A*) is greater than the magnitude value of the component below the reference value (*B*), then a '1' would be produced. Conversely, if the magnitude value of the component below the reference value (*B*) is greater than the magnitude value of the component above the reference value (*A*), a '0' is produced.

The reverse of rule (3) can be chosen by the user, however, such that;

$$A > B = 0; B > A = 1 \tag{4}$$

Comparing the components present in the output of the CSPE process on a frame by frame basis, and using rule (3) or (4) as required, a binary bit sequence is created, composed of '1's and '0's that represent the relationship between two components that exist in the frame, above and below the reference value. It is critical to note, however, that the components above and below the reference value are likely to be different in every frame when using real audio signals.

The sequence output from this analysis of the relationship between components in the cover signal is then applied to the plaintext binary bit sequence using bitwise 'Exclusive OR' (XOR). A bitwise XOR takes two bit patterns of equal length and performs the logical XOR operation on each *pair* of corresponding bits. The output in each

position is 1 if only *one* of the pair of bits is a 1. It will be 0 if both are 1 or both are 0. This is equivalent to being 1 if the two bits are different, and 0 if they are the same. For the ACES system, this means that if the cover audio presents components that already represent a 1 and we want to embed a 1, nothing changes. Similarly, if the cover audio presents components that already represent a 0 and we want to embed a 0, nothing changes. However, if cover audio presents components that do not match the plaintext bit to be embedded (in other words, the pair of bits in the XOR are different), then we need to modify the cover signal to embed the plaintext bit.

C	P	C XOR P
0	0	0
1	0	1
0	1	1
1	1	0

Figure 4: XOR Truth Table

In Figure 4, column 'C' represents the cover audio bit and column 'P' represents the plaintext bit. The final column is the result of the XOR operation. With a new binary bit sequence (essentially, the XOR result for each bitwise pair), we identify the components in the original audio signal need to be changed to embed the plaintext.

If a '1' is produced as a result of XOR, this means that we need to change the magnitude of component *A* in that frame so that it is less than the magnitude of component *B*. This process is repeated for every frame where the bitwise XOR produces a 1. We expect that half of the frames will require a change in magnitude, a point that will be addressed further on in the paper. When this task is complete, the component modification process will have ended and the plaintext will have been embedded in the signal, producing a cipher audio (audio equivalent of a ciphertext).

2.2 Decryption

The decryption process is much simpler than the encryption process. The recipient of the cipher audio needs to know various parameters to be able to decrypt the encoded message.

These include the frame length, the user-defined PIN, the window length and the Rule. If the recipient is in possession of all of these parameters, they are then able to input them into the ACES system to retrieve the plaintext embedded in the audio signal by simply analysing the relationship between the components directly above and below the user defined reference value. The process is illustrated in Figure 5.

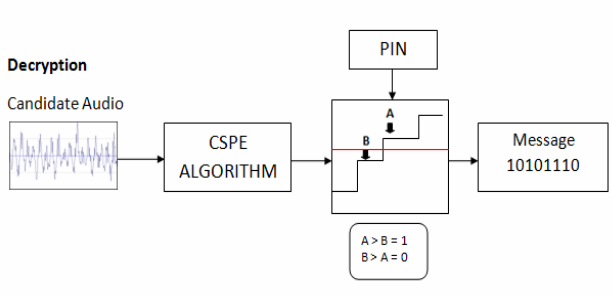

Figure 5: Decryption Process

3.0 Cryptology

Cryptology, described as the study of cryptography and cryptanalysis (the art of code breaking), makes use of several different aspects from other fields (information theory, computer science) to evaluate the strength of a cryptosystem. In Section 3.1, we discuss some of these traits and characteristics.

3.1.1 Perfect Secrecy

Perfect Secrecy, a property brought about by Claude Shannon during WWII [6], implies that the ciphertext gives absolutely no information about the plaintext, except for the maximum possible length of the message. The aim of Perfect Secrecy is to make guessing the plaintext as hard to do given the ciphertext as it is without it. A cryptosystem has Perfect Secrecy if it satisfies the following three conditions.

1. The cryptosystem is Shannon secure (knowing the ciphertext does not make it any easier to decipher).

$$Prob(x|y) = Prob(x) \quad \forall\ x \in P, y \in C$$

2. The cryptosystem is perfectly message indistinguishable. This means for every pair of plaintext messages, the probabilities that either message (m_1, m_2) maps to a given ciphertext must be equal.

$$\text{Prob}[Enc(m_1, k) = c] = \text{Prob}[Enc(m_2, k) = c]$$

3. The cryptosystem, with equal size spaces $|K| = |C| = |P|$, is perfectly key ambiguous. This condition holds if and only if each key chosen is truly random and for all $x \in P, y \in C$, there exists a unique key k such that $y = Enc_k(x)$.

3.1.2 One-Time Pad

The One-Time Pad (OTP) [7], derived from the Vernam cipher, was proven by Claude Shannon to have Perfect Secrecy. With One-Time Pads, each bit or character from the plaintext is encrypted, using modular addition, with a bit from a secret random key of the same length as the plaintext, resulting in a ciphertext. If the generated key satisfies certain conditions, it has been proven to be perfectly secure (shown below). These conditions are that the key is truly random, it is as large as or greater than the plaintext, it is never reused and it is kept secret. If all these conditions are met, then the ciphertext will be impossible to decrypt or break without knowing the key.

Take any $m \in M$ and $c \in C$, where M = Message Space, C = Ciphertext Space

Let $k^* = m \otimes c$.

Note that:

$$\underset{k \leftarrow K}{Prob}[Enc(k, m) = c]$$

$$= \underset{k \leftarrow K}{Prob}\left[\left(k \otimes [m) = c\right]\right]$$

$$= \underset{k \leftarrow K}{Prob}\left[k = c \otimes m\right]$$

$$= \underset{k \leftarrow K}{Prob}[k = k^*]$$

$$= \frac{1}{2^l}$$

where l = length of message

Since the equation is true for every $m \in M$, it follows that for every $m_1, m_2 \in M$, we have:

$$\underset{k \leftarrow K}{Prob}[Enc(k, m_1) = c] = \frac{1}{2^l}$$

$$\underset{k \leftarrow K}{Prob}[Enc(k, m_2) = c] = \frac{1}{2^l}$$

Therefore, this implies:

$$\underset{k \leftarrow K}{Prob}[Enc(k, m_1) = c] = \underset{k \leftarrow K}{Prob}[Enc(k, m_2) = c]$$

This establishes Perfect Secrecy of the One-Time Pad.

3.1.3 Random Oracle

A Random Oracle is a theoretical black box (system which can be viewed solely as having an input and an output but having no knowledge of its internal workings) in which there is one unique output per input. Theoretically, in our case, the input is the frame length, the PIN and the Rule; assuming the audio signal is a constant in this scenario. The black box would refer to the CSPE process and the output is the generated key. If the key is unique for each different input, and if repeating an input provides the same output, then our system will satisfy the properties of a Random Oracle.

3.1.4 Avalanche Effect

The Avalanche effect is a desirable quality in cryptographic algorithms. It is evident if, when the input is slightly changed (such as changing one bit) the output changes significantly (half the output bits change) [8]. If the cipher does exhibit the avalanche effect to a significant degree, it is also said to have good randomisation. The strict avalanche criterion (SAC) is a generalization of the avalanche effect. It is satisfied if, whenever a single input bit is complemented, 50% of the bits in the output sequence will change.

3.1.5 Randomness

Random tests are used to analyze the distribution pattern of numerical sequences. With stream ciphers, these sequences consist of just 1's and 0's, also known as bit sequences. Random bit sequences possess a high measure of unpredictability.

If you toss a coin up in the air ten times in a row, there is no way to predict how many times it will land on heads or tails. In information theory, entropy is a measure of disorder or, more precisely, unpredictability. A series of coin tosses is known to have zero entropy. If you have a bit

sequence of length 16, it is said you have 16 bit entropy. In other words, you cannot consistently predict successfully 16 times whether the next bit is going to be a 1 or a 0. This is very important in the area of randomness. If an attacker can predict the next bit in your sequence, your system is considered insecure.

In our case, we will be testing the generated keys to see if they are statistically random (contain no recognizable patterns or regularities). Statistical randomness, however, does not imply true randomness. True randomness is a property that cannot be created using a computer algorithm. If a bit sequence passes a statistical random test, it is said to be locally random (pseudo-random; appears to be random). Due to the fact that true randomness is difficult to assess, pseudo-randomness is sufficient for most stream ciphers.

The following three statistical random tests (also known as Hypothesis tests) were published by Kendall and Smith in the journal of the Royal Statistical Society in 1938. [9]

1. Frequency Test: Number of 1's and 0's should be roughly equal.
2. Serial Test: Number of two digit combinations (00, 01, 10, 11) should be roughly equal.
3. Poker Test: Number of times subsequences of length 'm' occur in the full sequence should be roughly equal.

Each test is designed to use Pearson's chi-square table [10]; a statistical tool used to compare each tests chi-squared value with the corresponding chi-squared number from the table. According to the assigned degree of freedom (number of values in the final calculation of a statistic that are free to vary) for the test, the test passes if the tests chi-squared value is less than the chi-squared from the table.

3.2 Attacking the system

An Attack is a successful or unsuccessful attempt at breaking part or all of a cryptosystem. The methods for attacking a stream cipher can be classified according to the information available to the cryptanalyst, the aim of the attack, or the way the attack is done. It is often assumed that the attacker has knowledge of the cryptographic algorithm, but not the key. As a result, due to the general algorithm often being publicly known, the confidentiality of the message depends on the secrecy of the key (Kerckoff's law).

Ciphertext-only Attack: An attack model for where the attacker is assumed to have access only to a set of ciphertexts. For this type of attack to be successful, the plaintext would need to have several bit repetitions. The various statistical techniques used for attacking the ciphertext are:

(i) *Frequency Analysis* – Studying the rate of occurrences.
(ii) *Traffic Analysis* - intercepting and examining messages in order to deduce information from patterns.
(iii) *Brute force Attack (Exhaustive Key Search)* - systematically checking all possible keys until the correct key is found. This type of attack requires a large amount of computing power and a large amount of time to run. For the majority of encryption algorithms, a brute force attack is impractical due to the large number of possibilities.

Known-plaintext Attack: The attacker has a set of ciphertexts to which he knows the corresponding plaintext and is at liberty to make use of them to try and uncover further secret information; typically, the key.

Chosen-plaintext Attack: An attack model which presumes that the attacker has the capability to choose arbitrary plaintexts to be encrypted and obtain the corresponding ciphertexts. Any cipher that can prevent chosen-plaintext attacks is then also guaranteed to be secure against known-plaintext and ciphertext-only attacks. The two distinguished forms of chosen plaintext attacks are:

(i) *Batch chosen-plaintext attack* - the cryptanalyst chooses all plaintexts before any of them are encrypted.
(ii) *Adaptive chosen-plaintext attack* - the cryptanalyst makes a series of interactive queries, choosing subsequent plaintexts based on the information from the previous encryptions.

Chosen-ciphertext Attack: An attack model where the cryptanalyst gathers information, at least in part, by choosing a ciphertext and obtaining its decryption under an unknown key. The aim is to deduce the key.

3.3 Similar Cryptographic Systems

The RC4 algorithm, developed by Ron Rivest of RSA [11], is a shared key stream cipher algorithm that requires a secure exchange of a secret key. The algorithm is used identically for encryption and decryption as the data stream is simply XOR'ed with the pseudo-randomly generated key sequence.

Several similarities exist between RC4 and our system. Both systems, if non-random keys are generated, will show vulnerability to attacks and highlight weaknesses in the system. RC4 is malleable which means that an adversary has the power to transform a ciphertext (e.g. by bit flipping) into another ciphertext, which would in practice, alter the plaintext decrypted. This attack is commonly known as a "Man in the Middle" attack. Fortunately, however, Man in the Middle attacks can be detected with the use of Message Authentication Code (MAC's) and digital signatures.

4.0 Experiments & Analysis

A number of experiments were designed and carried out to identify the characteristics displayed by this system and whether or not certain desirable cryptographic properties were present. The majority of experiments were carried out using Matlab, our main programming tool, while others were tested via Eclipse, a Java programming IDE (Integrated Development Environment). All experiments were carried out using 128 bit sequences, the equivalent to 16 ASCII characters [12]. One hundred experiments were conducted on three different audio signals, each from a different genre (rap, pop/rock and house), for each set of input parameters (900 experiments). Three input parameters were used throughout the course of these experiments:

(i) Random PIN, Constant Frame Length & Constant Plaintext
(ii) Constant PIN, Reducing Frame Length & Constant Plaintext
(iii) Random PIN, Reducing Frame Length & Constant Plaintext

The results of these experiments were used to test for randomness, amongst other properties. The overall results show that the key was locally random (pseudo-random) 82.44% of the time, according to the statistical tests which were discussed in the previous section.

As well as testing for randomness, the results of the aforementioned experiments were analysed to determine if the system displayed the characteristics of a Random Oracle. The output of the ACES algorithm, the generated key, was tested to identify if the key was unique for each different input. Based on the experiments carried out, it was ascertained that the output (key) was distinct for each input 100% of the time and that the key was never repeated for any different input tested. It was also noted that if the same input was used twice, then the same output would be achieved. This is an important trait of the Random Oracle model.

Along with a random key and the Random Oracle model, another desirable cryptographic property that was tested for was the Avalanche effect. The same set of input parameters as used previously were chosen with a minor change made to one of the inputs progressively throughout each set of experiments. Firstly, the PIN was changed by 120 Hz each time leading to changes in the key by between 41-60%, averaging 50.15%. Similarly, for the second set of experiments, the frame length was slightly altered each time by a change of 512. This resulted in a change in the key by between 36-57%, averaging 47.4%. While the recorded averages are not overly large percentages, it still lends credence to the idea that this system displays certain positive attributes of the Avalanche effect.

After the previously discussed experiments were completed, we wanted to verify the idea proposed earlier in the paper (Section 2) that 50% of the magnitudes of the original audio would require a change to be able to embed the plaintext into the signal. To do this, we modified the input into the ACES algorithm each time and recorded the number of 1's that were present in the XOR result. It was calculated that the number of 1's present in the XOR ranged between 35-67%, averaging 50.09%, thus proving the proposed concept.

Lastly, an experiment was written to test if the system was susceptible to a Chosen-Plaintext Attack. This was done by examining if any relationship existed between the plaintext and the XOR for a randomly generated key. The experiment was carried out on 3 audio signals with an average difference of 49%, ranging from 39-64%, in the bits between the plaintext and

XOR being recorded. Based on these results, we can conclude that there is no obvious relationship between the plaintext and the XOR.

Experiment	Sig 1	Sig 2	Sig 3	Average
Randomness	83.33%	83.33%	80.66%	82.44%
Random Oracle	100%	100%	100%	100%
Avalanche Effect (Change in PIN)	49.65%	50.45%	50.35%	50.15%
Avalanche Effect (Frame Size change)	47.73%	47.53%	46.94%	47.4%
50% Magnitude Change	49.83%	50.59%	49.86%	50.09%
Chosen Plaintext Attacks	48.05%	51.85%	47.1%	49%

Figure 5: Full set of Results

5.0 Evaluation & Recommendations

With numerous sets of experiments having been completed, it has allowed us to identify where we feel the strengths and weaknesses of this system lie. Due to the attacker requiring specific information to decrypt the message, such as frame length, the PIN and the Rule, the complexity of this system is increased. This strengthens the security of the system thus making it harder to break.

One of the major weaknesses of the system that became apparent to us was that the attacker would always know the length of the audio signal. However, as shown by (2), the attacker must know either the plaintext length to compute the frame size, or the frame size to compute the plaintext length. Due to neither of these being provided by the system, it makes the knowledge of the length of the audio signal redundant. This is also a strength of the system in the fact that, even if the attacker were to try and brute-force attack the system, he would not know what the length of the message was, thus reducing the likelihood of finding the correct plaintext amongst all the messages provided as a result of the brute-force attack.

Along with the frame length being unknown, two other elements are also kept secret; the PIN and the Rule. Taking this into consideration, we feel that this system promotes a secure method of data hiding. Despite this, however, we are of the opinion that improvements can still be made to improve this system.

(i) The plaintext could be encrypted before embedding it into the audio signal, for example, by using a hash function [13]. Due to the fact that each resulting message produced after a brute-force attack is equally likely to be the correct plaintext, the attacker can narrow down the list of messages by eliminating any messages that do not make any sense. If the message was encrypted beforehand, the attacker would have no way of confidently identifying the correct message, thus increasing the security of the message.

(ii) Another enhancement to the system would be to use a signal made up of white noise. White noise is a type of random signal observed to have a flat spectrum over a defined frequency band (such as the range of human hearing), where the frequencies and magnitudes are random themselves. As the signal is created randomly, the attacker has no original signal to compare the cipher-audio to.

Another way of implementing this concept would be to record a voice memo, or even something as natural as the noise of the wind, and use this audio signal as the basis for your encryption. Once the process is complete, you would then delete the voice memo, providing the attacker with no original to compare to. This mimics the characteristics of a OTP as the voice memo is used only once, and even if the exact same message was recorded a second time, the frequencies present would differ.

(iii) Another way to increase the strength of this system would be to use, what the attacker assumes to be the original, an easily accessible audio signal. An example of this would be using a song from the charts or a well-known speech. Then, before starting the encryption process, you would alter your own private copy of the accessible audio signal, use the edited version for the process and then delete it. As such, because the attacker is unaware of the alteration, he may try to compare the cipher-audio to the accessible unedited original audio signal. Since it has been edited before encryption, there would be no way the attacker could compare the signals and confidently find the plaintext, and it would also serve to confuse the attacker.

(iv) Instead of using a constant PIN for each frame, you could use either a constantly changing random PIN or a function to implement the PIN. The main drawback to a constantly changing random PIN per frame is that the intended recipient would need to know all of these PINs to decrypt the message, making it an unattractive proposal.

We could use a function instead, however, for each individual digit of the PIN per frame. A PIN, as explained previously, is chosen by the user, and in this case the user also defines a function of that PIN. Each individual digit of the PIN is placed into the function for each different frame and this acts as a reference value for that frame. The recipient, or attacker, would need to know the function as well as the PIN, making the system more secure. Due to the size of the PIN compared to the amount of frames, the digits would repeat themselves. However, due to the frequency components constantly changing on a frame by frame basis, we do not believe that this would cause any patterns to emerge.

Conclusion

We have presented a CSPE based audio watermarking cryptographic system that has been analysed mathematically, theoretically and experimentally. The results highlight that this system has similar characteristics to a One-Time Pad. It also contains certain desirable cryptographic properties such as the Random Oracle model and the Avalanche effect. Due to the fact that the key does not appear to be truly random according to our results, this system cannot be considered to satisfy perfect secrecy. We have also made recommendations that we feel would improve the system, allowing it to be used securely in the area of covert communications.

References

[1] Merriam-Webster Online Dictionary
http://www.merriam-webster.com/dictionary/steganography

[2] Webopedia
http://www.webopedia.com/TERM/C/cryptography.html

[3] K. M. Short and R. A. Garcia, "Signal Analysis using the Complex Spectral Phase Evolution (CSPE) Method", Audio Engineering Society 120th Convention, May 2006, Paris, France.

[4] J. Wang, R. Healy, and J. Timoney, "Perceptually Transparent Audio Watermarking of Real Audio Signals Based On The CSPE Algorithm", 21st Irish Signals and Systems Conference, 23rd - 24th June 2010, UCC, Cork, Ireland.

[5] M. T. Heideman, D. H. Johnson and C. S. Burrus, "Gauss and the History of the Fast Fourier Transform", Archive for History of Exact Sciences, Vol. 34, No. 3, 1985, pp. 265-277.

[6] C. E. Shannon, "Communication Theory of Secrecy Systems", pp. 679-683, Bell Systems Technical Journal, Vol. 28, pp. 656-715, 1948.

[7] D. Rijmenants, "The Complete Guide to Secure Communications with the One Time Pad Cipher", 2010.

[8] S. Ramanujam and M. Karuppiah, "Designing an Algorithm with high Avalanche Effect", IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.1, January 2011

[9] M.G. Kendall & B. Babington Smith: "Randomness and Random Sampling Numbers", Journal of the Royal Statistical Society, Vol. 101, No. 1 (1938), pp. 147-166.

[10] Chi Square Distribution Table
http://www.medcalc.org/manual/chi-square-table.php

[11] Ronald L. Rivest, "RSA security Response to weaknesses in key scheduling algorithm of RC4", Technical note, RSA Data Security, Inc., 2001.

[12] ASCII Table
http://www.asciitable.com/

[13] B. Prenell, "Analysis and Design of Cryptographic Hash Functions", February 2003