Simple Secrecy: Analog Stream Cipher for Secure Voice Communication

John Campbell

A Senior Thesis submitted in partial fulfillment
of the requirements for graduation
in the Honors Program
Liberty University
Spring 2015

Acceptance of Senior Honors Thesis

This Senior Honors Thesis is accepted in partial fulfillment of the requirements for graduation from the Honors Program of Liberty University.

_____

Carl Pettiford, Ph.D.
Thesis Chair

_____

Kyung Bae, Ph.D.
Committee Member

_____

Mark Shaneck, Ph.D.
Committee Member

_____

James H. Nutter, D.A.
Honors Director

_____

Date

Abstract

Voice signals are inherently analog, and some voice communication systems still utilize analog signals. Existing analog cryptographic methods do not satisfactorily provide cryptosecurity for communication systems due to several limitations. This paper proposes a novel means of provided cryptosecurity for analog signals without digitization; thereby avoiding the latency which results from ADC/DAC conversions. This method utilizes the principles of the digital stream cipher, generating instead a continuous pseudorandom analog key stream signal which is transformed with the original analog signal to create an encrypted ciphertext signal which is statistically independent of the original signal and the key stream signal. The transform is then inverted with the ciphertext signal and the same key stream signal to recover the original signal. The performance and characteristics of such a system has been measured and demonstrated through modeling and simulation.

Simple Secrecy: Analog Stream Cipher for Secure Voice Communication

Introduction

Technical Overview

Electronic communication systems transmit information from one point to another by means of electrical signals. Such a system has five components; the input transducer, the transmitter, the channel, the receiver, and the output transducer. As seen in Figure 1 below, the input transducer generates an electrical signal, known as the baseband signal, representing the input message. The transmitter modifies the baseband signal and transmits the new signal through the channel. The receiver receives this signal from the channel and processes it in order to recover a distorted version of the baseband signal. This is then output to the output transducer which generates the output message [1].

Figure 1: Communication System [1]

In general, a communication system is classified according to the characteristics of its input message, baseband signal, and transmitted signal. The information in the input message can be either discrete (i.e., alphabetic characters) or continuous (i.e. audible speech). The baseband signal representing this information can either be an analog or digital representation. In general discrete input messages are represented with digital

baseband signals, but continuous input messages can be represented with a digital signal through the use of an analog to digital converter (ADC) or more directly with an analog signal. The transmitted signal which is generated from baseband signal can also be analog or digital in nature, but practically, digital signals and low frequency analog signals are suitable only for communication through conductive media; therefore, high frequency analog signals are used for transmission in wireless communications systems in order to minimize component size and increase propagation distance [1]. To achieve this, the baseband signal, whether digital or analog, is represented in the analog transmitted signal by variation in amplitude, frequency, or phase.

Communication security (COMSEC) describes the technical and procedural means implemented to deny unauthorized access to a communication system's information and maintain communication integrity. COMSEC has four components; transmission security, emission security, physical security, and cryptosecurity [2]. Emission security and physical security both relate to the physical equipment in the system architecture and the secondary radiation emanating from this equipment. Transmission security relates to protecting the transmitted signals from interception or exploitation through means other than cryptanalysis. Cryptosecurity constitutes the implementation of cryptographic systems to preserve the confidentiality of the information in a communication system.

A cryptographic system allows for secure communication over insecure channels by preventing outside parties from corrupting or eavesdropping on the messages. Such a system has four main components; the original message, known as the plaintext, the shared secret, known as the key, the cryptographic algorithm, and the encrypted version

of the message, known as the ciphertext [3]. The plaintext and the key are input into the cryptographic algorithm, generating the ciphertext which is transmitted through the channel. Once the ciphertext is received, the plaintext is recovered from the ciphertext using an appropriate cryptographic algorithm and the secret key. The cryptographic algorithm can take one of two forms; a code or a cipher; additionally, the plaintext, the key, and the ciphertext can either be digital or analog.

Cryptanalysis is the process of breaking a cryptographic system [3]. The most basic method of cryptanalysis is to directly observe the ciphertext (e.g. audio signal or image). This is known as a ciphertext-only attack which relies on the intuition of the observer to make sense of any information which leaks through the cryptographic system. More advanced methods rely on a primarily mathematical analysis of ciphertext and/or the plaintext. Generally, cryptanalysis is used to refer to the more advanced methods and the manual observation is known as eavesdropping or listening. Two main factors contribute to the security of a cryptographic system. The first is the strength of the cryptographic algorithm. With a strong algorithm, the ciphertext is statistically independent of the plaintext [4]. The second is the number of possible keys known as the keyspace. Having a larger number of possible keys all of which will generate a different ciphertext given the same plaintext decreases the chance of success for brute force attacks.

Objective

The primary objective of this paper is to investigate the use of a novel analog cryptographic system (analog scrambler) to provide cryptosecurity for analog voice communication systems.

Scope and Constraints

For the purposes of this thesis, an analog cryptographic system (analog scrambler) is considered to be any cryptographic system which performs cryptographic operations on analog plaintext and ciphertext signals utilizing a digital key and a cipher based cryptographic algorithm. Analog voice communication systems will be considered to have continuous audio messages as input, analog baseband signals, and analog transmitted signals. Both systems perform all of their operations on continuous analog signals without performing any digitization operations. Figure 2 below illustrates such a system.

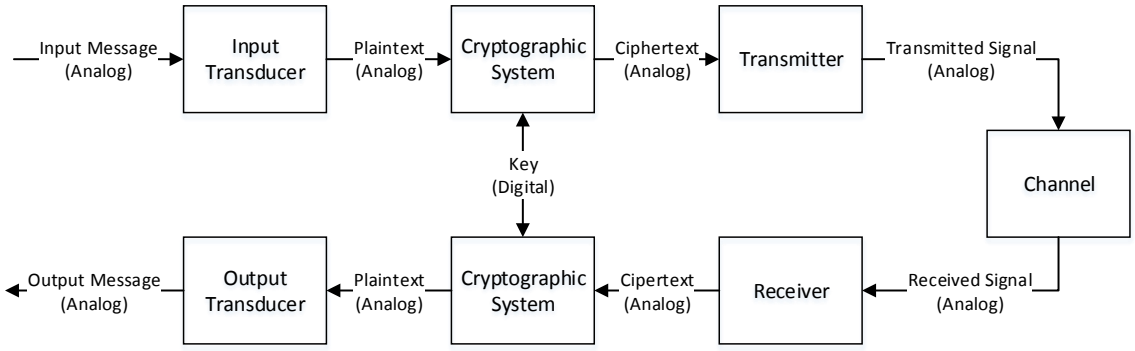

Figure 2: System Design Objective

Motivations

The motivation for implementing an analog cryptographic system within an analog voice communication system is twofold. The plaintext and ciphertext signals of a cryptographic system must be in the same form as the baseband or transmitted signal when implemented in a communication system. So in an analog voice communication system, the plaintext and ciphertext messages must be continuous analog signals; therefore, analog to digital (ADC) and digital to analog conversion (DAC) procedures

would be necessary for digital signal processing based encryption in such a system. Bypassing these conversion procedures by implementing an analog cryptographic system allows for cryptosecurity to be implemented without any reduction in throughput or signal quality which would occur as a result of ADC/DAC procedures [5]. This also avoids the incremental cost increase associated with the ADC/DAC functionality and the digital signal processing architecture.

Literature Review

Analog cryptographic systems have been around since the late 19th Century but only became prevalent in World War II [6]. They remained as the dominant technical means of secure communication until the start of the digital age. As they were common for a long period of time, there is a great deal of literature discussing various systems and methods which have been implemented, but because they reached obsolescence over thirty years ago, little has been written on the subject in recent years. In their place digital cryptographic systems have become the prevalent means of secure communication. As a result, a great deal of research and academic writing has been done on these systems. The following is a summary of pertinent writings on existing analog cryptographic systems and applicable digital cryptographic systems.

Existing Analog Methods

Early analog speech cryptographic systems manipulated either the amplitude or frequency of the input signal in order to scramble the speech [6]. Frequency inversion was commonly applied as a means of speech scrambling and can be implemented within modern DSP ICs [7]. This method inverts the frequency of each component signal within a given range to create the ciphertext as seen in Figure 3.This method provided

reasonable security against basic eavesdropping as it inverted the typical frequency patterns of normal human speech, making it mostly unintelligible to the human listener [6]. But it does not provide adequate security against even basic cryptanalysis as the effective keyspace of such a system is equal to one for a given frequency range. This is because the entire range of selected frequencies is inverted in all cases, and the behavior of the system is static.

Figure 3: Frequency Inversion [7]

Other methods operating on the frequency domain of the signals were also implemented. Different patterns of frequency shifting could be carried out offering increased cryptosecurity due to the increased keyspace. Eventually, this methodology led to spread spectrum systems. By first separating the signals into several frequency domain components with band-pass filters and then shifting each of these indecently in the frequency domain according to the specified pattern, spread spectrum systems spread the original signal across a wider bandwidth making interception, jamming, and cryptanalysis more difficult. These systems were common early in the 20th century [8] and some

remain in use in the modern context. While they offer strong security against basic eavesdropping, spread spectrum systems alone cannot offer sufficient cryptosecurity [9]. This is in part due the limited number of components which can be generated and the resulting limited keyspace. Just like frequency inversion, spread spectrum can be implemented using digital signal processing methods, but inherently both can be implemented using analog signal processing techniques only; therefore, the methods themselves can be labeled as analog.

Other historical cryptosecurity methods were labeled as “analog” but do not meet the definition of analog systems being used in this paper [10] [11] [12]. Three examples of these methods are sample permutation, block permutation, and frequency inversion [12]. With these methods single samples or blocks of samples are rearranged or inverted to create the ciphertext. Figure 4 demonstrates an example of sample permutation. These offer varying degrees of effectiveness against eavesdropping, but all offer limited security against strong cryptanalysis [12]. These methods were likely labeled as analog methods because their affect could be directly represented and observed in the time domain as shown in Figure 4, but as they operate by sampling the signal and manipulating the stored samples, they are not truly analog systems.

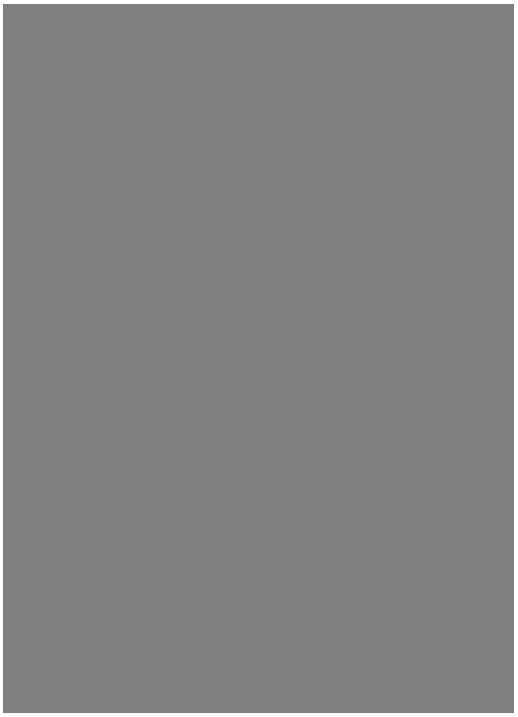

Figure 4: Sample Permutation [12]

Each of these systems, whether truly analog or sample based, share one common factor. They are generally secure against eavesdropping attacks, but are not secure against strong cryptanalysis. One primary reason for this is that the scrambling pattern remains fixed rather than changing in a pseudorandom manner [5]. This creates a scenario where the ciphertext has a statistical correlation with the plaintext, allowing successful cryptanalysis attacks. Another reason common in many of the methods is a limited keyspace. These are the same two factors which directly correlate to a strong cryptographic system, further echoing the cryptographic weakness of these systems.

Applicable Digital Methods

Digital cryptographic systems have become the standard means of providing communication security. These cryptographic algorithms used in these systems fall into two general categories; block ciphers and stream ciphers [3]. In a block cipher, the

message is broken into blocks on which the encryption algorithm operates along with the key. Similarly, the ciphertext is broken into blocks of data which are fed into the decryption cipher along with the key. The security of the system can be increased by utilizing a different key for each block in one of several modes of operation; the most basic of these is the Electronic Code Book mode of operation. In this mode both the encryption and decryption algorithms take a block of data with a given length and the users key as their input and output a given length data block. This process is repeated for each block of the message using the same key for each block. The complete ciphertext is found by combining all of the output ciphertext blocks. To recover the message, the complete ciphertext is broken into blocks, similarly to the message, and passed along with the key into to decryption cipher. Block ciphers represent the most common form of digital cryptographic algorithms, but their operation inherently requires that the plaintext and ciphertext undergo sampling and storage procedures, similar to the sample permutation algorithm discussed earlier [12]; therefore, they cannot be considered for implementation within the scope of this thesis.

A stream cipher can be thought of as a block cipher wherein the block is one single unit in length. In digital systems, stream ciphers operate on each individual bit of the message and ciphertext, processing one bit at a time [3]. To do this, a pseudorandom stream is generated based on the input key. This stream is combined with the message to create the ciphertext which is then combined with a similar or identical stream to reveal the message once again. In order to provide security, the key stream must be longer than the message and possess pseudorandom characteristics. This can become a severe limitation as the length of the message increases, but if these constraints are met, the

system will approximate the operation of a one-time pad which provides perfect security. Due to the fact that they operate on the smallest possible segment of data, stream ciphers generally have no discernable latency, and their throughput is not subject to any additional limitations. Also, the continuous nature of stream ciphers allows them to operate without any memory.

Proposal

Because existing analog cryptographic systems do not offer sufficient security against cryptanalysis, a novel method of providing cryptosecurity through analog signal processing is proposed.

Design

The proposed design will utilize the fundamentals of stream cipher cryptography using analog signal processing methods on continuous analog signals. To achieve this, a pseudorandom analog signal is generated based upon the input private key. This signal is combined with the analog plaintext signal through some transform to create an analog ciphertext signal; completing the encryption cycle. At the receiver, an identical pseudorandom signal is generated and combined with the ciphertext signal with some transform to recover the plaintext signal; completing the decryption cycle. Two components are necessary for the operation of this system; the stream generation protocol and the analog transform which interact in a manner demonstrated in Figure 5. The remainder of this thesis will be dedicated to establishing requirements for these two components, analyzing available means of performing these operations, and demonstrating the performance of the chosen methods.

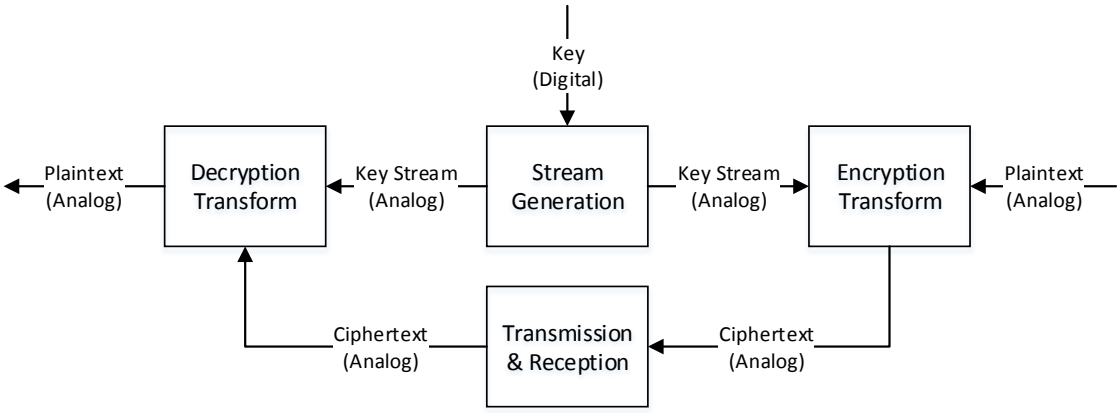

Figure 5: Proposed System Architecture

Requirements

To provide adequate cryptosecurity, the methods used to implement the stream cipher with analog signal processing techniques must adhere to the principles of secure cryptographic systems. As stated earlier, the primary characteristics of strong cryptographic systems are a sufficiently large key space and a strong cryptographic algorithm which relates to the statistical independence between ciphertext and the plaintext. To satisfy these criteria and operate as an effective cryptographic system, the chosen component methods should meet the following requirements.

Key stream generation. In a stream cipher, the proper operation of the encryption/decryption cycle depends in large part on the ability to generate the same key stream on both ends of process when given the same key. Without the ability to generate a duplicate of the encrypting key stream, the receiver will not be able to decrypt the ciphertext; therefore, a primary requirement for the effective operation of the proposed system is the repeatability of the stream generation for any given input key.

The cryptographic strength of a stream cipher algorithm is derived from the length and perceived randomness of the key stream. For the ciphertext to be statistically independent of the plaintext, the key stream must be pseudorandom and have a length longer than that of the plaintext. Therefore, to implement a stream cipher with analog methods, the system must generate an analog signal which has pseudorandom characteristics and a duration greater than the plaintext signal duration. For this system a pseudorandom stream with a duration of at least one hour will be deemed sufficient.

Cryptographic security also depends on the number of possible keys which can produce a unique encryption. Within a stream cipher, this number is equal to the number of keys which can generate a unique key stream. While having a sufficient large key space is generally and arbitrary requirement, a keyspace with 2^{128} possible keys is the minimum requirement for the stream generation component of the proposed system. Also, as per the initial design constraints, the key must be a digital value.

Transform. The ability to invert the cryptographic algorithm to recover the plaintext from the ciphertext is a critical component in any cryptographic system. The same is true for the analog stream cipher system being proposed; therefore, the analog transform which operates on the key stream signal and the plaintext signal must have an inverse transform which can recover the plaintext signal from the ciphertext signal combined with the key stream signal. This relationship is demonstrated in Equation 1 where F represents the encryption transform, X_p represents the plaintext signal, X_c represents the ciphertext signal, and F^{-1} represents the inverse of the encryption transform which is the decryption transform. When the decryption transform is applied to the ciphertext the resultant signal is the plaintext signal due to the inverse relationship of the

transforms. The encryption transform component of the analog stream cipher system design must, therefore, have an inverse transform from which to recover the plaintext signal.

$$F(X_p) = X_c \therefore F^{-1}(X_c) = F^{-1}[F(X_p)] = X_p$$

Equation 1: Inverse Transform

To satisfy the statistical independence criteria for cryptographic systems, the ciphertext signal must be statistically independent of the plaintext signal. This is easily achievable due to the pseudorandom behavior of the key stream, but stream ciphers offer an additional challenge. Because the key stream must be kept secret to preserve secrecy, the ciphertext must also be statistically independent of the key stream; therefore the analog transform must combine the key stream and plaintext signals in such a way that the resulting ciphertext signal does not reveal any information about the key stream of the plaintext signals.

Analysis

Given these requirements, several existing key stream generation methods and analog transforms have been analyzed to ascertain if they represent a viable option for implementation within an analog stream cipher cryptographic system.

Key Stream Generation

The performance and characteristics of the key stream generation methods will be evaluated according to their ability to generate a pseudorandom signal of a given length based on a particular digital key and recreate the same signal given the same key. This behavior is generally known as pseudorandom as the signal itself is not truly random;

rather, it is statistically random over a given observation window but will eventually repeat itself outside that window.

Noise generation. There are many different methods of noise generation utilizing electrical components to generate low power random signals. For example, random noise can be generated by forcing semi-conductor devices to operate in their breakdown region [13]. These signals are statistically random over any observation window, but their behavior is not conducive for use in a stream cipher system because the generated signals are truly random and cannot be repeated for the decryption cycle. Random noise generation is, therefore, not a suitable solution for analog key stream generation.

Feedback shift register. A feedback shift register (FSR) is commonly used with stream ciphers when speed takes a higher priority than security [3]. An FSR can generate a pseudorandom binary sequence using a shift register and XOR gates in a feedback configuration. The sequence of this bit stream is determined by the initial conditions of the shift register and the placement of the feedback gates. The length of the pseudorandom region of the output signal and the size of the keyspace are dependent on the number of bits in the shift register. With the feedback gates properly positioned within an n-bit shift register, the system is capable of generating a pseudorandom bit stream 2^n bits in length [14]. The keyspace for such a system is also 2^n as the initialization of the shift register constitutes the key input into the system. Under a normal configuration, the output from such a system is a pseudorandom bit stream which is generated by reading the last bit in the shift register. A feedback shift register is not suitable for this application as it is vulnerable to a known plaintext attack [15].

Cryptographically secure pseudorandom number generator. A pseudorandom number generator (PRNG) generates a stream of pseudorandom integer numbers based on initialization conditions and, in some cases, environmental entropy. Many of the commonly available PRNGs are not suitable for implementation in cryptographic systems as there output becomes predictable after the observation of previous output values [3]. In order to implement a PRNG in a cryptographic system then, it is necessary to utilize a cryptographically secure pseudorandom number generator (CSPRNG). A CSPRNG generates a stream of integer numbers which is both pseudorandom and cryptographically secure if for a given sample size the output values are statistically independent and the sequence cannot be predicted using known plaintext cryptanalysis techniques. Meeting these criteria satisfies the cryptographic requirements for implementation in this design. The requirement which governs the period of the stream must also be addressed. To this end, the length of the pseudorandom period in the time domain is represented in Equation 2. Assuming a 3 kHz clock frequency, for the period to be greater than one hour, the period of the PRNG must be greater than $10{,}800{,}000 \approx 2^{24}$.

$$Period = \frac{P_{cycles}}{f_{clock}}$$

Equation 2: CSPRNG period

Block cipher CSPRNG (Fortuna). One method of implementing a CSPRNG is to utilize a block cipher in counter (CTR) mode. With this method, successive incrementing values are encrypted using a block cipher constant key value as shown in Figure 6. The resulting output stream will be a sequence of a cryptographically secure pseudorandom numbers. The period of this sequence depends on the nature of the block

cipher and the input key. In general this period is at least 2^n where n is the bit size of the state space within the block cipher [16]. Therefore, any secure block cipher with a state space larger than 24 integer bits is suitable for implementation in the proposed system.

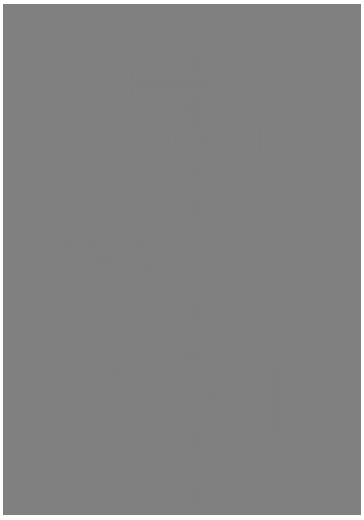

Figure 6: CTR Mode PRNG [17]

The Fortuna CSPRNG utilizes a secure block cipher of the designers choosing and also includes a reseeding procedure which expands the pseudorandom period by restarting the counter or modifying the key based on collected entropy values [18]. As the entropy collected at any given time or location is not repeatable, this reseeding methodology is not suitable for the intended cryptographic system; therefore, a CSPRNG based on a CTR mode block cipher alone will be considered to be sufficient provided it meets the established requirements for the pseudorandom period.

Analog signal from CSPNRG. In order utilize a CSPRNG to generate a pseudorandom analog signal; the n-bit pseudorandom number can be input into an n-rung R-2R ladder. R-2R ladders are used to convert an n-bit digital input to an analog dc voltage [19]. The input values to the R-2R network will change according to the

pseudorandom sequence with a frequency determined by the internal clock of the pseudorandom generator, and the output will be a jagged, discontinuous signal. A low-pass filter with a cutoff frequency just above the clock frequency could be used to smooth out the output signal without losing any of the random information in the system. As the output of the R-2R network is always positive, the generated signal will have a positive DC offset. In order to remove this and reduce the signal bandwidth, a high-pass filter can be used. In order to optimize performance, the bandwidth of the key stream signal should be limited to that of the plaintext signal (usually 30 Hz to 3 kHz); therefore a band-pass filter cutoff at these frequencies could implemented, and the register clock frequency can be set less than 3 kHz. The output then is a continuous analog signal with pseudorandom amplitude meeting all design requirements.

Transform

The performance and characteristics of the signal transforms will be evaluated according to their behavior when operating on two non-identical input signals which will model the plaintext and key stream signals. The invertibility of the transform will be evaluated alongside the degree to which the key stream and plaintext signals are obscured.

Signal summation. The simplest means of combining the two signals is to sum their values using a summing amplifier or similar circuitry for the encryption cycle. This transform would be invertible as decryption could be accomplished by inverting the key stream signal and summing the resultant signal with the ciphertext signal by simple arithmetic as seen in Equation 3.

$$X_p + X_s = X_c \therefore -X_s + X_c = X_p$$

Equation 3: Signal Summation Cryptographic Cycle

While the inversion of this transform may be straightforward, any one of several basic analyses demonstrates that it does not sufficiently conceal the component signals. A basic eavesdropping attack could easily defeat this method as all it effectively accomplishes is adding noise to the original signal which is something that the human ear is especially good at ignoring in order to perceive the other, more important audio signals. Also, although the resultant signal as seen in Figure 7 appears to bear little resemblance to the original signals in the time domain, but in the frequency domain, the two signals are entirely separate and can be separated using filtering techniques. This is the principle behind frequency domain multiplexing.

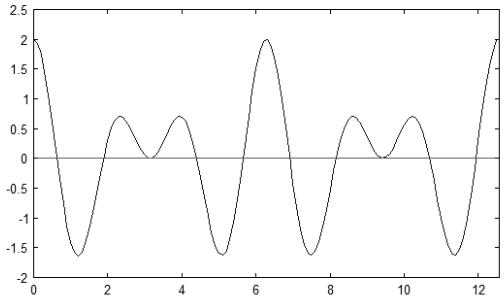

Figure 7: Signal Summation [Cos(2x)+Cos(3x)]

Amplitude modulation. Another means of combining the two signals is amplitude modulation which modulates the amplitude of the carrier signal to correspond with the amplitude of the baseband signal as seen in Figure 8 where m(t) is the baseband signal and $\cos\omega_c t$ is the carrier signal. The inversion of this transform is easily achieved through a demodulation transform, but it does not sufficiently conceal the component

signals for the following reason. The resultant signal of the amplitude modulation transform retains the frequency of the carrier signal and the amplitude of the baseband signal. This phenomenon can be directly observed in Figure 8. Also, using a frequency domain analysis the resultant signal is shown to simply be the baseband signal shifted to a higher frequency range centered on the carrier signal frequency [1]. These characteristics mean that the amplitude modulation does not satisfactorily combine and conceal the plaintext and key stream signals to be considered for use in an analog stream cipher cryptosecurity system.

Figure 8: Amplitude Modulation of Sinusoids in the Time Domain [1]

Angle modulation. Like amplitude modulation, angle modulation operates on a carrier signal and a baseband signal. Angle modulation has two classes which are closely related; frequency modulation and phase modulation. In frequency modulation, the frequency of the carrier signal is modulated in proportion to the amplitude of the baseband signal [1]. When the amplitude of the baseband signal is high, the frequency of the carrier signal is shifted up, the when the amplitude is low, the frequency of the carrier signal is shifted down. This behavior can be seen in Figure 9.

Figure 9: Frequency Modulated Signals [1]

Similarly with phase modulation, the instantaneous phase of the carrier signal is modulated in proportion to the baseband signal. Both frequency and phase modulation have two primary categories. Narrowband signals have a bandwidth approximately equal to twice the baseband signal bandwidth and can be readily derived from an arbitrary carrier signal. Wideband signals have a much larger bandwidth (theoretically infinite) but can only be generated with a sinusoidal carrier signal or indirectly from a narrowband signal [1]. To implement angle modulation in the proposed system, either the pseudorandom stream signal or the baseband plaintext signal must serve as the carrier signal; therefore, the chosen angle modulation method must be able to utilize an arbitrary carrier signal. Narrowband angle modulation then is the only available method. To implement narrowband angle modulation, the baseband signal, or its integral, is modulated with the carrier signal which has been shifted by negative ninety degrees. The resultant signal is subtracted from the original carrier signal generating a signal which approximates the behavior of an angle modulated signal [1]. This process is outlined in Figure 10 where the first diagram describes phase modulation and the second describes frequency modulation.

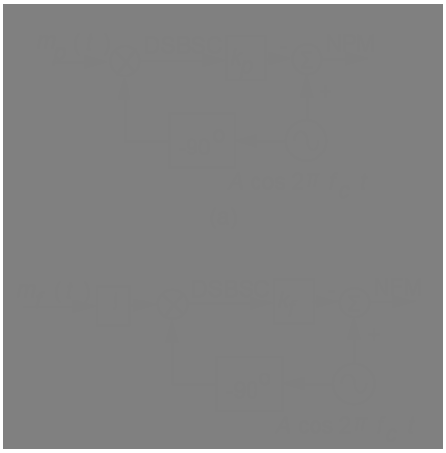

Figure 10: Narrow Band Angle Modulation Implementation [20]

While it is a linear transform, narrowband angle modulation modulates the angle of the carrier signal which, when combined with the pseudorandom behavior of the carrier signal, makes the baseband signal difficult to distinguish with an auditory analysis or other ciphertext only attack; additionally, the key stream carrier signal is sufficiently modified as well. Narrowband angle modulation is therefore a suitable transform for use in analog stream cipher cryptographic system.

Conclusion

The following design description and performance demonstration summarize the detailed design which is meant to fulfill the primary proposal of this paper by implementing stream cipher cryptography using analog signal processing techniques.

Design

For ease of use, the system accepts an n-bit string as the private key for the cryptographic system. This key is input into the SHA-256 cryptographic hash function generating a 256-bit string representation of the private key. This string is input into an AES block cipher set in counter-mode. The pseudorandom output from this system is sent

into a 128 rung R-2R ladder network. The pseudorandom analog signal output of this network is put through a band-pass filter with passband of 30 Hz to 3 kHz. The pseudorandom analog signal output by the filter serves as the carrier signal for the narrowband angle modulation of the plaintext signal and the narrowband angle demodulation of the ciphertext signal. Figure 11 shows a block diagram representing this operation.

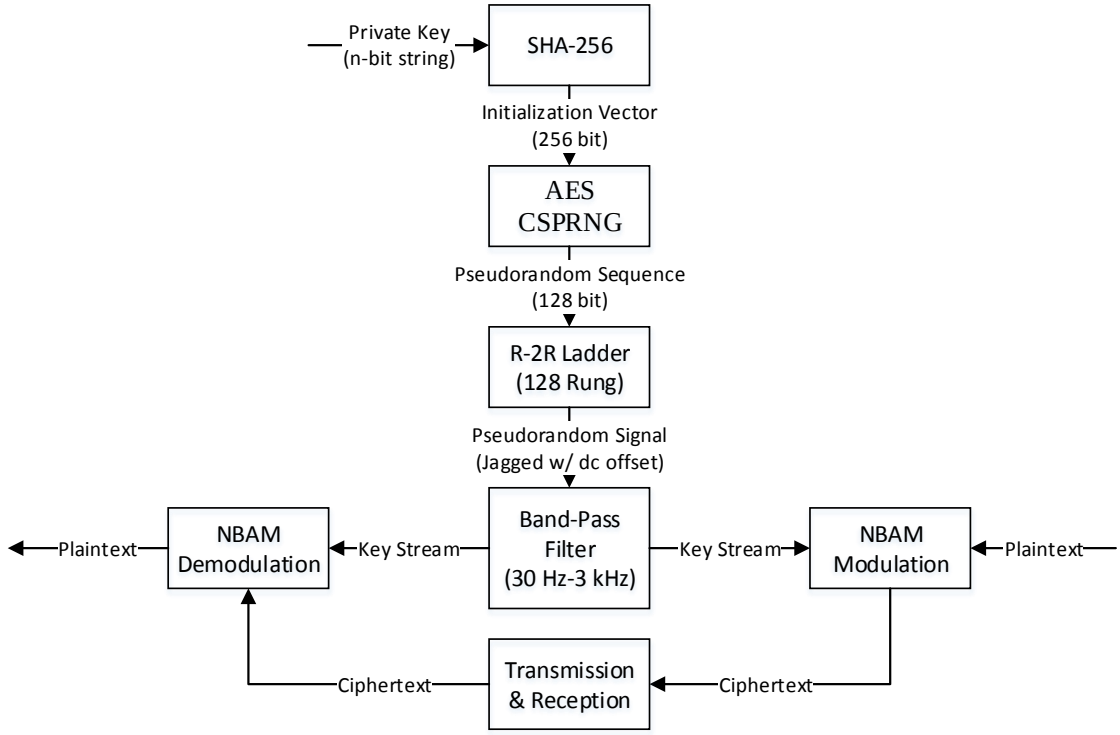

Figure 11: Detailed System Design

Implementation

The implementation would be quite simple. The n-bit private key string could be input into the system using an ASCII encoded character string collected from a standard input device (i.e. a keyboard or keypad). The SHA-256 hash function and the AES counter-mode CSPRNG could be implemented using the appropriate software on any

processor unit with the proper capabilities. The 128 rung R-2R ladder network could be with network of 256 resistors. The angle modulation and demodulation could be implemented with appropriate analog signal processing modules.

Demonstration

The viability of this design has been demonstrated utilizing digital modeling and simulation. The primary tool used for this process was MATLAB. The source code for this simulation can be found in Appendix A. The first step was to demonstrate and verify the use of random number generation to generate a pseudorandom analog signal. Implementing a CSPRNG was not feasible within MATLAB, so the built in MATLAB random number generation functionality was used to model a PRNG which generated a stream of pseudorandom analog values with a clock frequency of 3 kHz and a sampling frequency of 44.1 kHz. The resultant signal contained high frequency components and a significant DC offset as expected. This signal was then filtered using a fourth order Butterworth band-pass filter with 30Hz and 3 kHz cutoff frequencies. The resultant frequency had a reduced DC offset and high frequency components as expected and was output to an uncompressed audio file using the built-in “audiowrite” function. Both signals can be seen in Figure 12 below.

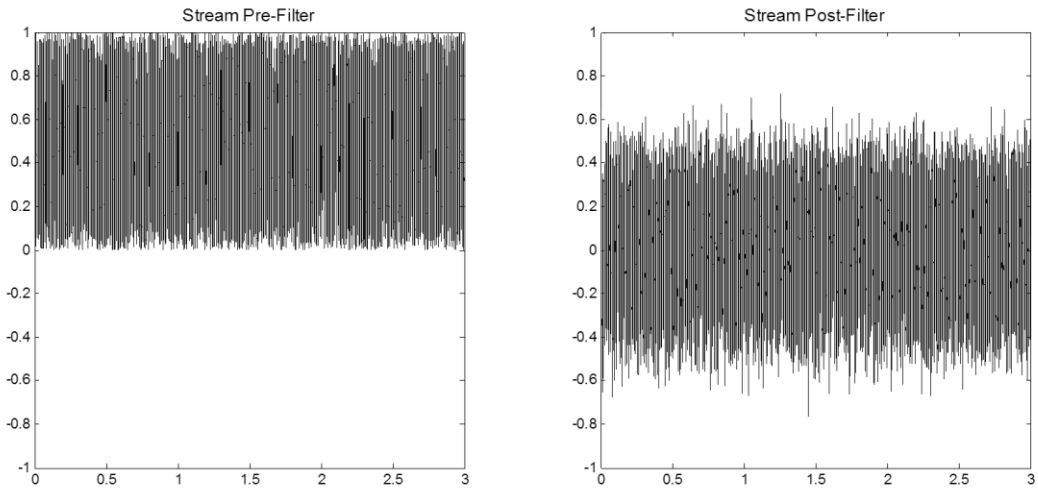

Figure 12: Stream Generation Simulation

The second step in the process was to simulate and verify the encryption and decryption transforms using a voice signal and the filtered pseudorandom stream. The voice signal was input from an uncompressed audio file using the built-in "audioread" function. The ciphertext signal was generated by performing narrowband phase modulation with the plaintext signal as the baseband signal and the filtered pseudorandom stream as the carrier signal. Modulation was accomplished using the algorithm represented in Figure 10. The built-in Hilbert transform function was used to perform the negative ninety degree phase shift on the stream signal. The resultant signal was modulated with the baseband signal through multiplication, and the resultant signal was subtracted from the original pseudorandom stream. The final ciphertext signal, seen in Figure 13, presented a phase modulated version of the original pseudorandom stream as expected. A basic demodulation procedure was adopted for simplicity. First the pseudorandom stream was subtracted from the ciphertext signal. Then the resultant signal was demodulated using a simulated multiplicative demodulator. The output signal which is shown in Figure 13 resembled the original plaintext to a certain degree as expected.

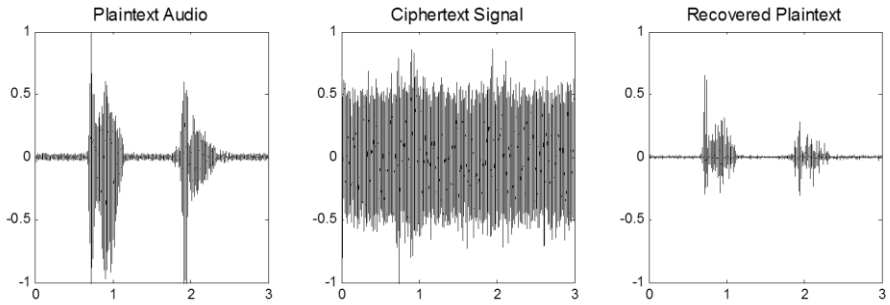

Figure 13: Simulated Audio Signals

The third step was to perform an auditory analysis of resultant ciphertext and plaintext signals. This was completed by listening to the generated ciphertext and recovered plaintext signals in order to ascertain the effectiveness of the system. Through this method it was found that the narrowband phase modulation encryption transform was effective against an auditory ciphertext only attack as the plaintext speech was indistinguishable in the resultant signal. The narrowband phase demodulation decryption transform did not perform satisfactorily as the resultant recovered signal was noisy and distorted. The original message was distinguishable but with great difficulty.

Recommendation & Follow Up

Due to the computational complexity of generating the pseudorandom number stream and the ineffectiveness of the demodulation procedure, implementation of the proposed system is not recommended. But as the narrowband phase modulation encryption transform was found to be effective and the theoretical benefits of the proposed system could possibly be significant, the concept of an analog stream cipher encryption may be pursued through further research outside of the scope of this project. Additional work should focus on developing more efficient means of generating the pseudorandom stream, addressing the problem of synchronization, and improving the

functionality of the decryption algorithm or investigating alternative transforms. It is also recommended that a full cryptographic analysis of the chosen algorithms and transforms be conducted to further verify or contradict the conclusion of this paper. Finally a prototype implementation of this system should be constructed to confirm total system feasibility.

Bibliography

[1] B. Lathi, Modern Digital and Analog Communication Systems, Oxford: Oxford University Press, 1998.

[2] National Security Agency, *National Information Systems Security Glossary,* Ft. Meade, Maryland: National Security Agency, 2000.

[3] W. Trappe and L. C. Washington, Introduction to Cryptography with Coding Theory, Upper Saddle River, NJ: Pearson, 2005.

[4] A. Gersho, "Perfect Secrecy Encryption of Analog Signals," *IEEE Journal on Selected Areas in Communications,* pp. 460-466, 1984.

[5] P. S. Parwinder, S. Bhupinder and P. A. Satinder, "Need of Secure Voice Encryption and its Methods a Review Paper," *International Journal of Advanced Research in Computer Science and Software Engineering,* 2012.

[6] D. Kahn, The Codebreakers, New York: The Macmillan Company, 1973.

[7] J. Ahmed and N. Ikram, "Frequency-Domain Speech Scrambling/Descrambling Techniques Implementation and Evaluation on DSP," in *IEEE Multi Topic Conference,* Islamabad, 2003.

[8] W. R. Bennet, "Secret Telephony as a Historical Example of Spread-Spectrum Communications," *IEEE Transactions of Communications,* pp. 98-104, 1983.

[9] W. Shaw, Cybersecurity for SCADA Systems, Lake Arrowhead, CA: PennWell

Corp, 2006.

[10] N. S. Jayant, R. V. Cox, B. J. McDermott and A. M. Quinn, "Analog Scramblers for Speech Based on Sequential Permutations in Time and Frequency," *The Bell Systems Technical Journal,* vol. 62, no. 1, pp. 25-46, 1983.

[11] B. Goldburg, S. Sridharan and E. Dawson, "Design and Cryptanalysis of Transform-Based Analog Speech Scramblers," *IEEE Journal on Selected Areas in Communications,* pp. 735-744, 1993.

[12] N. Jayant, "A Comparison of Four Methods for Analog Speech Privacy," *IEEE Transactions of Communications,* pp. 18-24, January 1981.

[13] F. C. Fitchen and C. D. Motchenbacher, Low-Noise Electronic Design, Hoboken, New Jersey: Wiley, 1973.

[14] L. Xu, J. Han, T. Zhang and H. Wang, "Dual-Channel Pseudo-Random Signal Generator with Precise Control of Time Delay between Channels," in *Intrumentation and Measurement Technology Conference,* Ottowa, Canada, 2005.

[15] N. Li, "Introduction to Cryptography Lecture 10," 2005. [Online]. Available: https://www.cs.purdue.edu/homes/ninghui/courses/Fall05/lectures/355_Fall05_lect10.pdf. [Accessed 5 March 2015].

[16] J. Salmon, M. Moraes, R. Dror and D. Shaw, "Parallel random numbers: As easy as 1, 2, 3," in *International Conference for High Performance Computing,*

Networking, Storage and Analysis, Seatle, WA, 2011.

[17] H. C. Hudde, "Building Stream Ciphers from Block Ciphers and Their Security," 18 February 2009. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.392.9064&rep=rep1&type=pdf. [Accessed 10 March 2015].

[18] R. McEvoy, J. Curran, P. Cotter and C. Murphy, "Fortuna: Cryptographically Secure Pseudo-Random Number Generation In Software And Hardware," in *Irish Signals and Systems Conference,* Dublin, 2006.

[19] L. Nashelsky and R. L. Boylestad, Electronic Devices and Circuit Theory, Upper Saddle River, New Jersey: Pearson, 2012.

[20] C. Lee, "Angle Modulation," 9 September 2002. [Online]. Available: http://engineering.mq.edu.au/~cl/files_pdf/elec321/lect_fm.pdf. [Accessed 30 March 2015].

[21] B. P. Lathi, Linear Systems and Signals, Oxford: Oxford University Press, 2004.

[22] S. C. Kan and N. S. Nayant, "On Speech Encryption Using Waveform Scrambling," *The Bell System Technical Journal,* pp. 781-808, 1977.

[23] A. Matsunaga, K. Koga and M. Ohkawa, "An Analog Speech Scrambling System Using the FFT Technique with High-Level Security," *IEEE Journal on Selected Areas of Communications,* pp. 540-547, 1989.

Appendix A: Simulation Source Code

```
clear %Clear MATLAB memory

%Constants
duration = 3; %seconds audio duration
nyquist = 44100; %Hz sampling rate
maxF = 3000; %Hz upper cutoff frequency
minF = 30; %Hz lower cutoff frequency
length = duration*nyquist; %total samples
t = 0+duration/length:duration/length:duration; %time vector

%Build Filter Entity
order = 4; %filter order
lowCut = minF/nyquist; %lower cutoff
highCut = maxF/nyquist; %upper cutoff
[b,a] = butter(order,[lowCut, highCut],'bandpass'); %filter poles

%Input Files
inputFile = 'D:\Documents\Project\Test.wav';
streamFile = 'D:\Documents\Project\Stream.wav';
encryptFile = 'D:\Documents\Project\Encrypt.wav';
decryptFile = 'D:\Documents\Project\Decrypt.wav';

%Input Plaintext
plaintext = audioread(inputFile, [1 length]); %audio signal from file
plaintext = filter(b,a,4.*plaintext); %filter signal
%Plot audio signal
figure(1);
subplot(1,3,1);
plot(t,plaintext);
axis([0 duration -1 1]);
title('Plaintext Audio');

%Generate PRS
streamRaw = zeros(length,1); %initialize vector
tmp = rand(); %initialize first value
for n = 1:length
    %Get new value at 3kHz
    if(mod(n, floor(nyquist/maxF)) == 0)
        tmp = rand(); %pseudorandom float
    end
    streamRaw(n) = tmp; %assign value to vector
end
%Plot pseudorandom stream
figure(2);
subplot(1,2,1);
plot(t, streamRaw);
axis([0 duration -1 1]);
title('Stream Pre-Filter');

%Filtering the PRS
stream = filter(b,a,streamRaw); %perform DSP filter
%Plot filtered stream
subplot(1,2,2);
plot(t, stream);
```

```
axis([0 duration -1 1]);
title('Stream Post-Filter');
audiowrite(streamFile,stream,nyquist); %output stream audio file

%Transform
ciphertext = stream - 2.*imag(hilbert(stream)).*plaintext;
%Plot ciphertext
figure(1);
subplot(1,3,2)
plot(t, ciphertext);
axis([0 duration -1 1]);
title('Ciphertext Signal');
audiowrite(encryptFile,ciphertext,nyquist); %output ciphertext audio
file

%Decrypt
plaintextRecover = (ciphertext-stream); %Remove stream noise
plaintextRecover = plaintextRecover.*imag(hilbert(stream)); %Demodulate
plaintextRecover = filter(b,a,plaintextRecover); %Continue demodulation
%Plot recovered plaintext
figure(1);
subplot(1,3,3);
plot(t, plaintextRecover);
axis([0 duration -1 1]);
title('Recovered Plaintext');
audiowrite(decryptFile,plaintextRecover,nyquist); %output plaintext
audio
```