Indian Journal of Science and Technology Vol. 4 No. 5 (May 2011) ISSN: 0974- 6846

Improving security of communication systems using CHAOS

R. Raja Kumar[1], A. Sampath[1] and P. Indumathi[2]

[1]*Mathematics Department, Sathyabama University, Chennai, India*
[2]*Electronics Engineering Department, M.I.T Campus, Anna University, Chennai, India*
rrkmird@yahoo.com, indu@mitindia.edu, dr_asampath@yahoo.co.in

Abstract

The principal design objective of this paper is to facilitate extremely secure digital communication using chaotic shift keying in existing frequency-hop spread spectrum systems. Generally in chaotic shift keying, the incoming digital bit stream is mapped onto one of the available chaotic signals. Our scheme proposes the use of antipodal chaotic signals, which means they are signals which are out of phase with each other by 180°. Data bits 0 and 1 can be mapped onto either of the antipodal signals using required convention. Also, in FH-spread spectrum (FH-SS) techniques, there exists a p-n code generator based on the output of which the frequency synthesizer selects a particular carrier frequency. Usual FH-SS systems use MFSK modulation in the stage following the frequency hopper. As our case involves the use of chaotic reference signals and their basic shapes need to be preserved throughout, we propose the use of binary frequency shift keying. The digital data, which is now in the form of chaotic signals, is turned into a BFSK signal. This resultant signal is mixed with a carrier frequency at the frequency synthesizer. This carrier is chosen by the frequency hopper, whose output, in turn depends on the output of a p-n sequence. We assume the transmission channel to contribute additive white noise. At the receiver, we propose to perform FSK demodulation as a first stage recovery to get back the chaotic signal. Now this recovered signal is either an exact copy of the chaotic reference signal or its inverse. To decide which of the two was sent by the transmitter, we can use a correlator receiver, which uses a locally generated chaotic reference signal.

Keywords: Chaotic shift keying, orthogonal chaotic shift keying, pulse code modulation.

Introduction

Over recent years a great deal of research has focused on chaotic communication schemes (Saha *et al*., 2004; Raja Kumar *et al*., 2009). The primary driver for this interest is that chaos based schemes are inherently highly secure and have spectral efficiencies which give good noise rejection. There are two basic types of transmission schemes utilizing these chaotic signals. The first method relies on ideas which choose a particular state of a chaotic system to transmit and then the same state is used in the receiver to synchronize a similar chaotic circuit and allow regeneration of the complete set of chaotic states needed for decoding the incoming message sequences. These methods, although attractive, have not proved to be sufficiently robust with noisy transmission channels. The second type of chaos communication is characterized by the transmission of a reference signal. The most successful of these types of transmission has been differential chaos shift keying (DCSK) which introduced chaotic processes into existing correlation based schemes. This method transmits a chaotic function for half of the symbol interval and then a duplicate or inverted version of the same signal in the second half representing a '1' or '0'. This is exactly analogous to a BPSK scheme and decoding is achieved by correlation of both halves of the signal. Furthermore, there exists another transmission scheme termed quadrature chaos shift keying (QCSK) which characterizes a chaotic signal within orthonormal subspaces and utilizes this to choose the transmitted signal.

In principle, the foundation for chaos in communication stems from the basic principles of chaos theory and its dynamics. Chaos theory describes the behaviour of certain nonlinear dynamical systems that may exhibit dynamics that are highly sensitive to initial conditions. As a result of this sensitivity, which manifests itself as an exponential growth of perturbations in the initial conditions, the behaviour of chaotic systems appears to be random. This happens even though these systems are deterministic, meaning that their future dynamics are fully defined by their initial conditions, with no random elements involved. This behaviour is termed as 'deterministic 'chaos' or simply 'chaos'.

Chaotic behaviour has been observed in the laboratory in a variety of systems including electrical circuits, lasers, oscillating chemical reactions, fluid dynamics, and mechanical and magneto-mechanical devices. Observations of chaotic behaviour in nature include the dynamics of satellites in the solar system, the time evolution of the magnetic field of celestial bodies, population growth in ecology, the dynamics of the action potentials in neurons and molecular vibrations. Everyday examples of chaotic systems include weather and climate. All these inherent properties of a chaotic system enable us to successfully utilize chaotic and seemingly random behaviour for telecommunication purposes. As mentioned previously, amongst the greatest advantages it can provide lies the fact the message sequences to be transmitted are extremely secure and this feature of chaos communication enables us to utilize this scheme in

Indian Journal of Science and Technology Vol. 4 No. 5 (May 2011) ISSN: 0974- 6846

military applications where information security is of utmost priority.

The objective of this work was to enable security of information through chaos, assuming that the other resources such as bandwidth were available aplenty.

Methodology

Security in CHAOS

As we have been emphasizing all along, the chief utility of chaos based systems is that we are now provided with better means to combat security threats that are inherent to any present-day communication system. The key to how this is done forms the basis of our proposed scheme. Chaos, once decidedly used in a communication system, can take multiple forms in any block of the system we desire (Kolumban, 2000). Hence the understanding of general stand-alone chaotic systems is extremely critical to our understanding of how chaos works out when infused into an otherwise normal communication system. An understanding of the underlying mathematical concepts is also extremely imperative for the proper utilization of all the inherent properties of chaotic systems.

Systems that exhibit mathematical chaos are deterministic and thus orderly in some sense; this technical use of the word chaos is at odds with common parlance, which suggests complete disorder. A related field of physics called quantum chaos theory studies systems that follow the laws of quantum mechanics. Recently another field, called relativistic chaos, has emerged to describe systems that follow the laws of general relativity (Kurian *et al.,* 2005). But throughout the design of our scheme we have restricted ourselves to a very simple degree of use of chaotic systems and its properties, the primary reason for this being the fact that our focus was on how to use chaos to improve our communication system.

Spread spectrum with CHAOS

The principle of spread spectrum (SS) system is to spread the original information over a broad bandwidth of frequencies. These systems require the spreading sequence to provide a white noise like auto and cross correlation properties. Conventionally, a pseudorandom or a pseudo-noise (PN) sequence generator is used for spread spectrum systems, but it suffers from the periodicity problem (which is a fixed number of states where the state-machine runs through each state in a deterministic manner). This periodicity behaviour of pseudorandom sequence compromises the overall system security; moreover, it reduces the system capacity as well. In contrast, these noise-like sequences can as well be generated by a chaotic generator. A chaotic generator is an unlimited states state-machine; therefore it can produce non-repeating sequences. This non-periodic behaviour of the chaotic generator offers potential advantage over conventional pseudo-noise based system in terms of security, synchronization and system capacity of a spread spectrum (SS) communication (Luca *et al*., 2005).

Therefore, there have been an increasing number of schemes that utilize chaos theory in SS communication for both analog and digital systems. Such schemes include but are not limited to chaotic masking, chaos modulation, chaos shift keying (CSK), and chaotic CDMA sequence. Although the early chaotic communication schemes have performance characteristics much worse when compared to the traditional binary phase shift keying (BPSK) and binary frequency shift keying (BFSK) modulations, the bit-error-rate (BER) characteristics of many recently suggested schemes are very close to the ideal ones. Hence the combined properties of spread spectrum with chaos give the user an edge in security in communication. This is the key design objective of our scheme.

System structure

Data bits 0 and 1 can be mapped onto either of the antipodal signals using required convention. Also, in FH-spread spectrum (FH-SS) techniques, there exists a p-n code generator based on the output of which the frequency synthesizer selects a particular carrier frequency. Usual FH-SS systems use MFSK modulation in the stage following the frequency hopper. As our case involves the use of chaotic reference signals and preservation of their basic shapes throughout, we propose the use of binary frequency shift keying. The digital data, which is now in the form of chaotic signals, is turned into a BFSK signal. This resultant signal is mixed with a carrier frequency at the frequency synthesizer. This carrier is chosen by the frequency hopper, whose output in turn depends on the output of a p-n sequence. We assume the transmission channel to contribute additive white noise. At the receiver, we propose to perform FSK demodulator as a first stage recovery to get back the chaotic signal (Fig. 1).

Now this recovered signal is either an exact copy of the chaotic reference signal or its inverse. To decide which of the 2 was sent by the transmitter, we can use a correlator receiver which uses a locally generated chaotic generator to produce the same reference signal exactly. The problem of synchronization is not addressed here; hence the transmitter and receiver are expected to be in perfect synchronism with respect to the p-n code generators and the chaotic waveform generators.

Security against intruders is provided by the hopping of the frequency bands of the carrier frequency. Even if the jammer is monitoring all of the possible frequency bands, the chaotic signal would not mean anything to him. Further, the initial values to the polynomial keeps changing for every specified number of bits conforming to a possible code book. In order to enhance security, the generator polynomial itself can be changed for every specified number of bits. This means that even for a

Indian Journal of Science and Technology Vol. 4 No. 5 (May 2011) ISSN: 0974- 6846

Fig. 1. System structure.

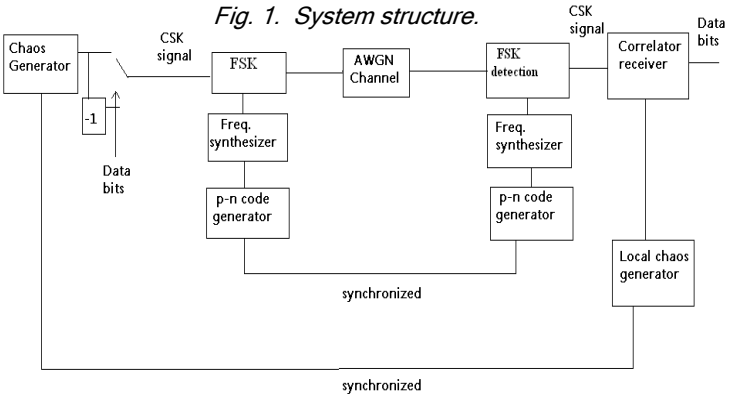


Fig. 2. Samples & quantized values.

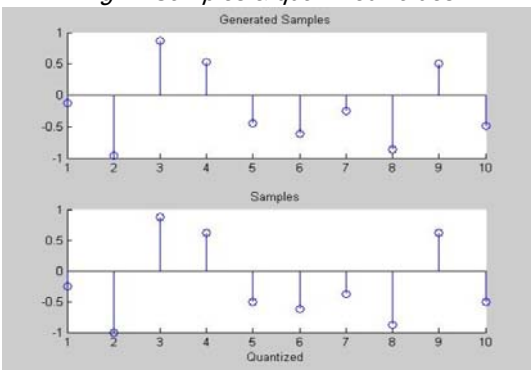


Fig. 3. Encoded stream.

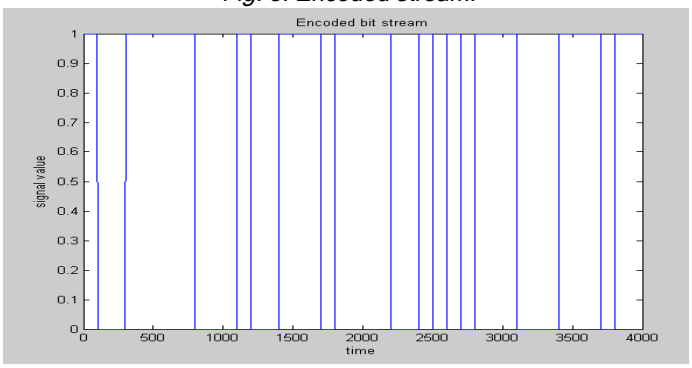


Fig. 4. FSK output.

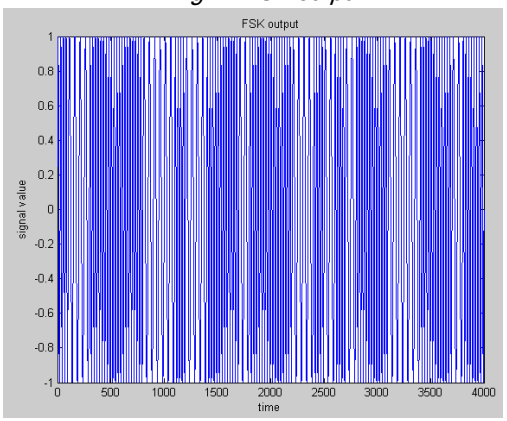


Fig. 5. Components in FSK output.

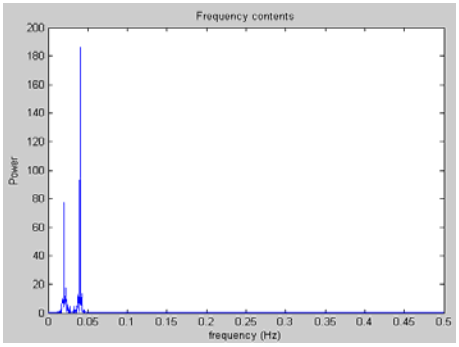


Fig. 6. Mixer output for p-n code 00.

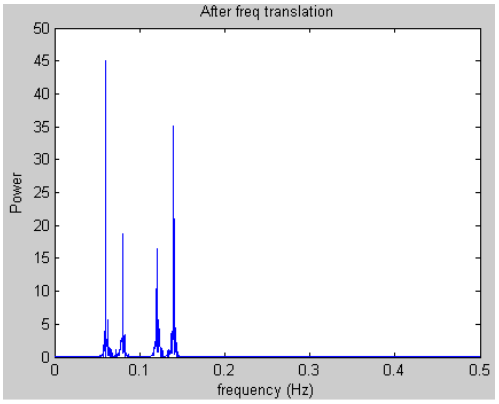


Fig. 7. Mixer output for p-n code 01.

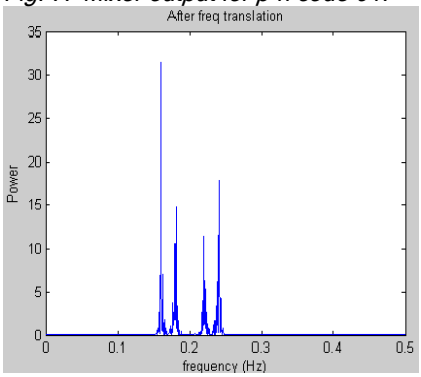


Fig. 8. Mixer output for p-n code 10.

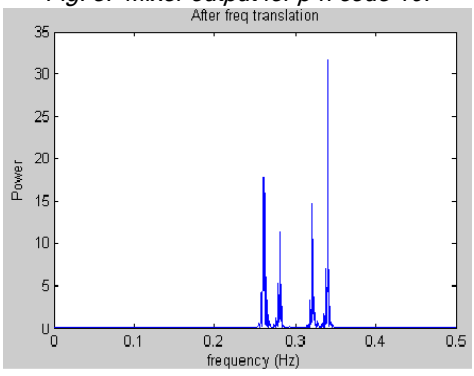


Fig. 9. Mixer output for p-n code 11.

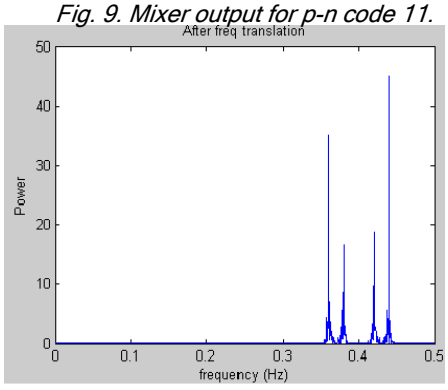
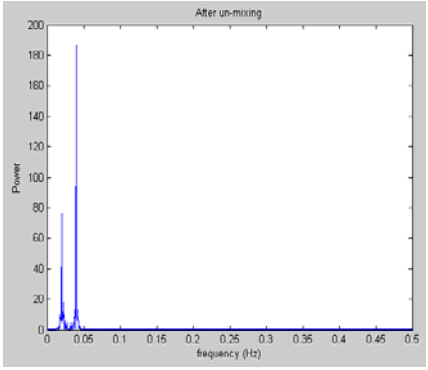


Fig. 10. Un-mixed signal.



Indian Journal of Science and Technology Vol. 4 No. 5 (May 2011) ISSN: 0974- 6846

same data bit (say 1) being transmitted at 2 different occasions, we will have 2 different transmitted signals. This provides added security.

Correlator receiver

The final stage of our receiver is the correlator receiver. Also an integral part of this correlator receiver is the local chaos generator. This is an exact match of the chaos generator that is used at the transmitter to generate samples. Since the issue of synchronization is not addressed here at all, we require both the chaos generators to be in the same state at the same instant of time. So when the local chaos generator gives the receiver a set of samples generated locally, the correlator receiver performs the task of comparing the incoming message signal with the locally generated signal. If both the signals are one and the same, we can conclude that the bit is a 1. However if the two signals are out of phase by 180^{o}, the bit transmitted by the sender was a 0. In our scheme, we have made use of the combined advantages of spread spectrum and chaos. Together they form a very secure communications technology which will enable military entities to exchange data, given all of the required resources (such as abundant bandwidth) and mathematical formulae.

Transmitter design

The transmitter part of our scheme starts with the chaos generator. The chaos generator is one of the key components of the design which contributes mainly to the security aspect of our scheme; this block can be substituted by a hardware model governed by differential equations in the real world. Using this chaos generator, we can generate either continuous time signals to be used for transmission of our message sequences or as an alternative; we can also use discrete time samples. In our scheme we make use of discrete samples that are in effect, outputs from the chaos generator. In order to simulate chaotic signal generators (including both continuous & discrete signals) in MATLAB or in any other programming tool, we make use of a governing equation recursively. This governing equation is usually an ordinary differential equation (Raja Kumar *et al.,* 2006), though exceptions exist in certain cases, where simple quadratic or cubic equations can satisfy necessary conditions (Saiki & Yamada, 2008). Usually, the equation governing the chaos generator is given a set of initial conditions based on which the system generates future chaotic values. Even a small change in the initial value brings about drastic changes in the outputs, hence proving the property of sensitivity to initial conditions. Now, once the samples have been generated, we can utilize these signals or samples at the desired blocks in our system. These samples in our scheme undergo pulse code modulation after quantization. And once this is accomplished, a host of digital communication techniques are applied.

This transmitter generates samples based on a set of governing equations. Literature survey of chaotic systems yielded a few examples given by the following:

$$X_{n+1} = 1- a (X_n)^2 \quad (1)$$

This system becomes non-linear, dynamic and exhibits chaotic behaviour for the value of a =2 and for the initial value of $X_0 = \{-1,1\}$ (Saiki & Yamada, 2008).

$$X_{n+1} = 4 X_n^3 - 3 X_n \quad (2)$$

Generators like these can be used to produce chaotic waveforms and their antipodal signals. Equations like these are used to generate sample sets until a certain limit. This sample set and its antipodal set are both used to map data bits 0 and 1. In our simulation exercise, we have used eqn. (1) shown above to generate samples. Also each data bit is represented by 10 samples. The initial value for X_0 is taken as 0.126 and the value of 'a' is taken as 2. Using these initial conditions, we can generate samples recursively. The samples are represented diagrammatically as shown in Fig. 2.

The program code gets the data bit that is to be transmitted as the input of the program and accordingly generates these samples. The samples generated above are for an input data bit of '1', with the specified initial conditions. For a '0', the samples would appear as though they were multiplied by -1.

Encoding & FSK

Once pulse code modulation of the samples is done as shown above, we end up with a stream of bits. Now we have used 10 samples to represent each bit and the quantization is done using 16 levels. Hence each sample can be represented by 4 bits. Therefore, we now have a stream of 40 bits to represent our original data bit. After the bit stream has been encoded, frequency shift keying has to be implemented on it. This is shown in Fig. 3.
Also the FSK signal is obtained as follows (Fig. 4): This output is generated using 2 different normalized frequency values for 0 and 1 respectively. The frequency contents of this FSK output are also shown using Fourier analysis. This is done so as to have a visual interpretation of the frequency hop that will occur in the very next stage of our transmitter. The following components represent the direct and unmixed output of the FSK block. For frequency hopping, we mix this output with appropriate carriers. The various components are represented using Fourier analysis as shown in Fig. 5.

The next stage involves the mixing of the above shown FSK output with a carrier that is chosen by the output of the p-n code generator. For the ease of programming, the frequency hopper is hopped over only 4 different carrier frequencies. And one of these 4 frequencies is chosen based on a two-bit input which is given by the user.

Indian Journal of Science and Technology Vol. 4 No. 5 (May 2011) ISSN: 0974- 6846

Frequency hopping (Fig. 6, 7 & 8):

Similarly, the mixer outputs for the other two p-n codes are shown as follows (Fig. 9): To sum up the working of transmitter, it converts a single data bit into a sequence of 40 bits in our case. Since the principal design objective of our scheme is security, we assume that we have enough resources (especially bandwidth) that allow such a translation of one bit into forty bits. These bits are handled as in a normal communication system, with pulse code modulation applied over the samples, followed by frequency shift keying and the hopping of carrier frequency based on the output of a p-n generator.

Receiver design

In the construction of the receiver, we have performed the exact inverse operation carried out in the transmitter. The "unmixer" component is the first part of the receiver which retrieves the FSK signal from the frequency hopped signal. The FSK signal is now processed to get back the bit stream that was originally sent at the transmitter. Now these bits are taken four at a time and the quantized values are recovered. Now from these quantized values, we retrieve the actual samples. In order to make a decision about the transmitted bit, we use exactly the same system to locally generate samples in the receiver just as in the transmitter. We compare this sample set to the received sample set- if it is the same set of 10 samples, we conclude that the transmitted bit was a 1, if it is an antipodal set, a 0 was sent. The unmixer component of the receiver requires the p-n code that was used at the transmitter during transmission of the signal. Once the input is given, the unmixer is responsible for giving back the FSK signal. From this FSK signal, the subsequent stages recover the message sequence originally sent at the transmitter. Fig. 10 shows the recovered FSK signal from the mixed signal.

Receiver stages

Once the FSK signal has been recovered by the unmixer circuit, the next step is to recover the bit stream from this FSK signal. Originally at the transmitter, this bit stream was encoded before being converted into an FSK signal. The recovery of this bit stream is the main concern once the unmixer has performed its operation. Once this bit stream is recovered, the next phase is to convert this stream of 40 bits into a single data bit. This is done by grouping 4 bits into each of 10 sets. Now these ten sets each represent a sample. Using all these 4 bit combinations, we can reconstruct only the quantized values that were sent at the transmitter. But the correlator receiver needs the actual samples to work with. Hence we have to nullify the quantization error that occurred at the transmitter so as to get the samples back. The programming approach that we have used to counter quantization error is a simplistic one. Since the number of quantization levels used was 16, each sample can differ from the other by a maximum value of 0.1249, because MATLAB takes into account only 4 bits post the decimal point. Since this difference is known, we can determine the range between which quantized values fall and hence the samples can be reconstructed.

Determining the data bit

The samples have been recovered from the received signal. Our next job is to generate locally a set of samples. For this synchronization is required between the transmitter and receiver with respect to the initial conditions. Once similar initial conditions are used at both ends, our job is to compare the locally generated samples with the received samples. If these samples are an exact match, we can successfully conclude that the data bit sent at the transmitter was a 1. Otherwise, if the two sample sets are found to be antipodal to each other, a 0 was the data bit sent at the transmitter.

The impact of different governing equations of the chaotic signal generator on the communications system can be considered for future work.

Conclusion

The inherent advantages of both frequency hopping and chaos with respect to information security resulted in a very good communications technique. By virtue of the simulation exercise, we were able to prove that the same data bit can be represented as two different data signals at two instances of time, thus proving the robustness of this technique.

References

1. Kolumban G (2000) Theoretical noise performance of correlator-based chaotic communications schemes. *IEEE Trans. on Crcuits & Systems I: Fundamental Theory & Applications.* 47(12), 1692-1701.
2. Kurian AP, Puthusserypady S and Su Myat Htut (2005) Performance enhancement of DS/CDMA system using chaotic complex spreading sequence. *IEEE Trans. on wireless communications.* 4(3), 984-989.
3. Luca MB, Azou S and Burel G (2005) A complete receiver solution for a chaotic direct sequence spread spectrum communication system. *IEEE Int. Symp. on Circuits & Systems, Kobe, Japan.*
4. Raja Kumar R, Indumathi P and Sampath A (2009) The Mathematics of chaos. *Int. J. Intelligent Elec. Sys.* 3, 73-76.
5. Saha P, Banerjee S, Roy A and Chowdhury S (2004) Chaos, signal communication and parameter estimation. *Physics Letts. A.* 326,133-139.
6. Saiki Y and Yamada M (2008) Time averaged properties along unstable periodic orbits and chaotic orbits in 2 map systems. *Nonlin. Processes Geophys.* 15, 675–680.