Chaos for Stream Cipher

Ninan Sajeeth Philip
Department of Physics
Cochin University of
Science and Technology
Kochi - 682 022, India
nsp@stthom.ernet.in

K. Babu Joseph
Department of Physics
Cochin University of
Science and Technology
Kochi - 682 022, India
vc@cusat.ac.in

Abstract

This paper discusses mixing of chaotic systems as a dependable method for secure communication. Distribution of the entropy function for steady state as well as plaintext input sequences are analyzed. It is shown that the mixing of chaotic sequences results in a sequence that does not have any state dependence on the information encrypted by them. The generated output states of such a cipher approach the theoretical maximum for both complexity measures and cycle length. These features are then compared with some popular ciphers.

Keywords: chaos, stream ciphers, logistic equation, mixing of chaos

1 Introduction

With the advent of the world wide web and e-commerce, research in digital cryptography [1] has acquired a renewed momentum. In general, the algorithms used for cryptographic applications may be classified into two, namely, public key cryptography and secret key cryptography. Public key cryptography uses separate keys to encrypt and decrypt an information. In this system a person uses his private key and the public key of the legitimate recipient to encrypt a message. At the receiving side, the message is decoded using the public key of the sender and the private key of the recipient. The advantage of the scheme is that anyone can send an information securely to the person who has a public key and make sure that only the addressee will be able to read it. Thus they are mainly used for secure exchange of information between two persons who need not trust each other. Public key cryptosystems accomplish this by relying on the inability of present day tools to factor large numbers (eg. RSA like implementations) or on the difficulty in solving the discrete logarithm problem (Diffie-Hellman like implementations) etc. [2]. Since such computations take reasonable time on most implementations, they become vulnerable to a host of attacks. The nature of the key may be understood by monitoring the time taken for encryption and decryption. This is the basis of *timing attack*. A slightly different attack known as the *differential power analysis* monitors the power consumed by the encoding devices and deduce information on the secret key. A more generalized attack known as *differential cryptanalysis* maps the correspondence between the plain text and cipher text symbols (state vectors) to decode the secret key. On the other hand, secret key cryptography, popularly known as stream ciphers use the same key for encryption and decryption. Since this demands the secure exchange of secret keys between the users, their role is slightly different from that of public key cryptosystems. They are mainly used for the secure storage and retrieval of information. For example, in the banking sector or in the credit card application, there is no need for a public key. Since the user need not have to declare any public key, these ciphers are generally more secure compared to public key cryptosystems. Most of the stream ciphers use simple logical operations that are very fast compared to public key cryptosystems and are thus immune to differential power analysis or timing attacks. However, they could be vulnerable to *differential cryptanalysis*, *brute force attacks* and *known plaintext attacks* unless necessary precautions are taken.

Most stream ciphers rely on random number generators for encryption. At the transmitter side the secret information is added to a random carrier signal. At the receiving end it is subtracted out from the received signal to retrieve the information. Since a random process cannot be reproduced with certainty, it is not possible to generate the same random sequence at both the transmitter and receiver simultaneously. The general practice is to use pseudo-random numbers that can be generated at both sides without difficulty.

Studies in nonlinear dynamics shows that many of the seemingly complex systems in nature are described by very simple mathematical equations

[3, 4, 5]. Fractal images, for example, are capable of traversing the embedding space in minimum time, a feature that is referred to as the space filling ability of fractal images. Viewed from the cryptographic point of view, this is equivalent to the maximum cycle length [6], a feature expected from a good random number generator (RNG). Although chaotic systems appear to be highly irregular, they are also deterministic in the sense that it is possible to reproduce them with certainty. These promising features of chaotic systems attracted many researchers to try chaos as a possible medium for secure communication. Chaos was successfully implemented in analog secure systems following the works of Pecora and Carroll in 1990 [7, 8, 9]. However, implementations in digital systems have not yet been successful. The major reason why digital cryptography using chaos did not turn out to be effective is that digital circuits deal with finite number spaces. Chaos is a phenomenon in which even very small variations in the initial state of a system results in exponentially diverging evolution of its future states. When dealing with finite number space, chaos looses this diversity due to the restricted resolution of possible states. However, many researchers have pointed out that mixing of pseudo random sequences of finite precision are able to produce rather complex sequences [10, 11, 12, 13]. Their studies were however largely limited to linear systems. The present paper is an extension of these encouraging observations to nonlinear chaotic systems that have many additional features suitable for cryptography.

The organization of the paper is as follows: In section 2 we introduce the new cipher that generates a space filling sequence by mixing two chaotic sequences. Section 3 discusses the software implementation of the cipher. Section 4 presents some popular cryptanalysis techniques on the proposed system. Some methods to investigate the security aspects of a cipher in general are also discussed. Finally, Section 5 compares the new cipher with three popular ciphers.

2 Mixing of chaotic sequences

One major difference between digital and analog systems is the introduction of round-off errors in digital systems. This is observed to give some peculiar effects that are explained in detail in [14]. On the other hand, the assurance of exact reproducibility is an added advantage of digital systems. Studies on digital chaos generators have been reported to have more diversity compared to their analog counterparts [15]. This is encouraging, since cryptographic applications are mostly on digital platforms.

Matthews [16] in 1989 proposed an encryption algorithm using the principles of chaos. He generalized the logistic equation (see equation 1) with cryptographic considerations and used the last two decimal digits of the generated sequence to get the random sequence. This system was found to have various drawbacks as pointed out by Wheeler [17]. Matthews' algorithm gave way to one-way functions [18] in 1990.

A common route to chaos in most systems is through period doubling [3]. In the fully developed chaotic system, there exist simultaneously a large number of unstable periodic trajectories for the evolution of the system. At this stage even minor fluctuations in their initial conditions would cause the system to switch between different trajectories. Although the digital systems are restricted by the finite discrete space, it may be shown that a proper shuffling of these states can generate pseudo random sequences with many interesting properties. This is in agreement with the observations made by other researchers as it would be described below.

For the discussion of the proposed method of shuffling or mixing of chaotic states of a system, we take the logistic equation as an example. The logistic equation is an iterative equation in which every upcoming state x_{n+1} is defined as a nonlinear function of the present state x_n as:

$$x_{n+1} = 4\lambda x_n(1 - x_n) \quad with \quad 0 \leq \lambda \leq 1 \qquad (1)$$

If a graph is plotted with different values for the control parameter, λ, many interesting properties of non-linear systems may be observed. When the value of λ is close to zero, the sequence converges to some steady value after a few iterations. On increasing the value of λ it enters a periodic cycle. Increasing the value of λ further results in a feature known as period doubling and finally, at values of λ close to one, the sequence appears to be completely random. At this stage, even slight variations in the value of x_n at any instant would result in very large (exponential) variations in the subsequent evolution of the system. This is referred to as chaos [3]. It should be noted that in the logistic equation, the x_{n+1}th state is completely defined by x$_n$. This is what is known as dependence on initial conditions. Although the sequence of values generated by the system at this stage appears to be random, they are in fact highly deterministic as can be seen from their attractor structure shown in figure 1. This makes the system insecure for cryptographic applications.

To study the effect of mixing, let us consider two separate logistic equations. Let x_n and x'_n be two points on the nearby trajectories of such two systems. For simplicity we assume that the control parameter λ is the same for both the systems. We now define the evolution of a new system by the following sequence of equations:

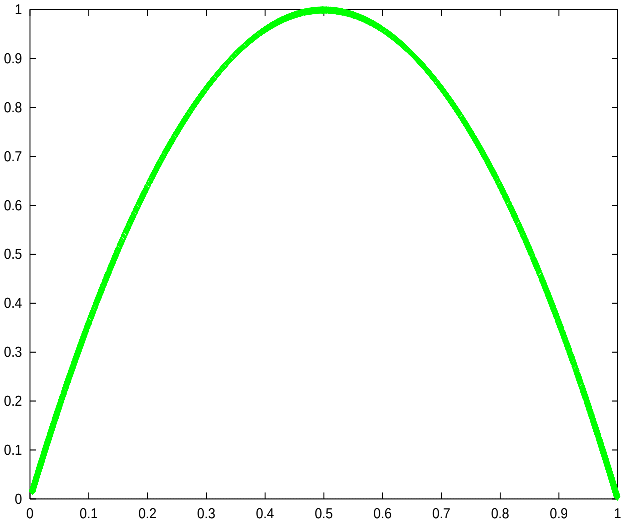

Figure 1: The attractor of the logistic equation is an inverted parabola. The peak of the parabola corresponds to the value of the control parameter λ

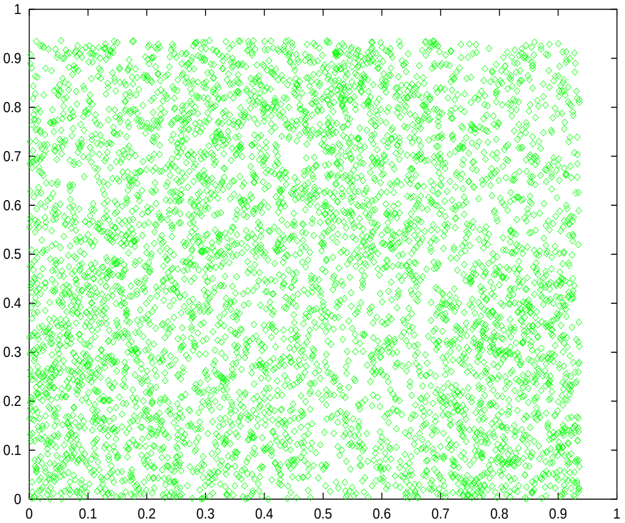

Figure 2: A fully developed chaotic system has many unstable states of excitation. The mixing of two nearby trajectories of the logistic equation destroy the attractor structure and the correlation of adjacent states. This is identical to a continuous perturbation on an unstable system.

$$\begin{array}{c} x_{n+1} = 4\lambda x_n(1-x_n) \\ x'_{n+1} = 4\lambda x'_n(1-x'_n) \\ P_n = x_n \oplus x'_n \\ x_n = x_{n+1} \oplus C_n \\ x'_n = x'_{n+1} \end{array} \qquad (2)$$

The $\oplus$ operator represent the XOR operation. We refer to equation (2) as the cipher equation in the rest of the paper.

As mentioned earlier, combining adjacent states of a system using XOR function is not new in cryptography. Ritter for example present the concept of pseudo-random shuffling [10] as a very efficient method for random sequence generation. MacLaren *et al.* [11] showed that combining two congruential generators would give a more complex random sequence than individual congruential generators. In cellular automata, Wolfram [12] suggested a linear array of elements a[0]...a[n] and an iterative equation a[i] = (a[i-1]) XOR (a[i] OR a[i+1]) to construct a RNG. The output was taken as a portion of the state vector. An analysis of such systems was made by Rietman [19]. Many combiners using linear feedback shift registers (LFSR) have also been attempted as alternate sources of random number generators [13]. However, these were on linear systems and due to their inherent linearity, they become vulnerable to a known plain text attack or correlation attack [20]. In the proposed system, the mixing is done on nonlinear systems at the onset of chaos. Thus the mixing is equivalent to a continuous perturbation on a chaotic system causing the system to jump from one unstable trajectory to another in accordance with the perturbation. The system is still deterministic as long as the nonlinear equations are defined. However, the sequence of iterative equations produces a high order Markov process due to the shuffling process in the finite space [21] making the generated sequence appear as random to the intruder. This is illustrated in figure 2 where instead of the attractor, we have a space filling *noise like* pattern in the phase space. The points in the plot will be identical in two instances if and only if the encrypted text and the initial conditions of the logistic equations are all the same. It is possible to prevent the occurrence of such a situation by initializing the logistic equation by some nonrepetitive parameter, e.g. a psedo random number generated using the time of the day as the seed. Although such a parameter appears to be insecure, this is not a serious issue since the actual security of the system is not based on the value of this parameter. Even a slight deviation in the value of this parameter causes the evolution of the system to diverge away exponentially from each other, an inherent property of chaotic systems, making an entirely new sequence even when the plaintext contents are the same.

Also note that in the iterative formula suggested in the cipher equation, C_n is the pseudo random cipher text and is defined outside equation 2. For the purpose of this paper, we will define $C_n = P_n \oplus y_n$ with y_n as the plaintext data stream. We will describe a multiple encryption scheme later in the paper which can improve the security of the system further by an exponential factor.

3 Implementation

In the software implementation, the starting value x_1 was selected as a pseudo random variable and $\mathrm{x}_1^{'}$ was computed from x_1. Since the value of x_1 is required to reconstruct the sequence, it was transmitted along with the cipher text. A dummy sequence of random numbers, the length of which is agreed by the legitimate users, were used to hide the location of x_1. It is possible to adopt any steganographic methods to hide the starting value, for example, as the nth bit of a set of bits in a dummy sequence. The data stream was then grouped into blocks of cipher texts with a set of synchronizing pulses to ensure proper alignment. All these signals are encrypted that they appear exactly like the encrypted information signal to the intruder. Grouping into blocks and synchronization pulses are not required by the algorithm, but are included to show that such locations may be used to place tracking codes, error correction codes or CRC codes.

4 Cryptanalysis

From cryptographic considerations, three measures on the ciphertext contribute significantly towards the security of the information ciphered by them. They are the Combiner Correlation [22], the Cycle-Length [6] and the Complexity Measure [23]. Combiner Correlation estimate the probability with which the output of the cipher becomes identical to that of the input [22]. In our example, it is the correlation between y_n and C_n . If for example the output follows the input by an unequal number of times, this information may be used to break the cipher. The simplest way to compute this correlation is by writing a table of the possible inputs and the corresponding outputs. For simplicity, it is assumed that the variables are of single bit boolean type. In table 1, for single variables like x_n, their actual value is indicated, while for comparison of two variables, we put a 1 to indicate that the variables are equal. For example $y_n = C_n$ has a value 1 to indicate that they are equal. As it can be seen (*last column*), the cipher text has equal probability of occurrence when the values of x_n, x_n' and y_n flips their states. Since the cipher equation uses the XOR operation, which is a bit level mod 2 addition operation, the same pattern repeats even when the variables are multiple bit words. Thus the proposed system will not leak out any information regarding the possible input state vectors in the cipher text.

x_n	x_n'	P_n	y_n	C_n	$y_n = C_n$
0	0	0	0	0	1
0	0	0	1	1	1
0	1	1	0	1	0
0	1	1	1	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	0	1
1	1	0	1	1	1
50%	50%	50%	50%	50%	50%

Table 1: The table shows the possible distribution of the cipher text and the intermediate states for the possible values of x_n and x_n'.

4.1 Differential Cryptanalysis

Differential cryptanalysis [24] makes use of some chosen plain texts to explore any possible correlation between the plain text and the cipher text state vectors. It is difficult to prove that a cipher would be equally strong in hiding vital information about itself irrespective of the plain text patterns enciphered by it. However, there are a few standard tests to verify such possibilities. One test is to look for the existence of indicative variations in the probability distribution density of the output states of the cipher text as a function of the plain text information masked by it. A reliable measure of this is the cycle length [6] of the ciphertext when the plain text consists of only one kind of symbol. Popular measures on cycle length such as the run-up tests are difficult on self-synchronous ciphers of the type discussed here. We thus do an indirect measurement of this feature. Information theoretical computations such as entropy measures are based on the probability distribution of state vectors in a sequence. A uniform distribution of state vectors at all possible values in a finite space is equivalent to a maximum cycle length since such a distribution demands the system to traverse all possible states before repetition. This can be directly measured as the probability of occurrence of each state vector in the finite space when the length of the sequence is in the order of the total number of possible states.

Two different sets of ASCII character sequences were used for the analysis. In the first round of tests, the sequences consisted of each with a single ASCII character repeated 9464 times. We say that the information is masked by the cipher, if it is not possible to identify any specific trend in the cipher corresponding to the distribution in the plain text. For ease of comparison, we computed the entropy distribution of the sequences. If the probability distribution density of the plain text character x in the cipher text is distributed over a range of M possible states and if the population density in each state is P_j, then the Shannon entropy of the distribution in unit bits is given by

$$H_x = -\sum_{j=1}^{j=M} P_j * log_2 P_j$$

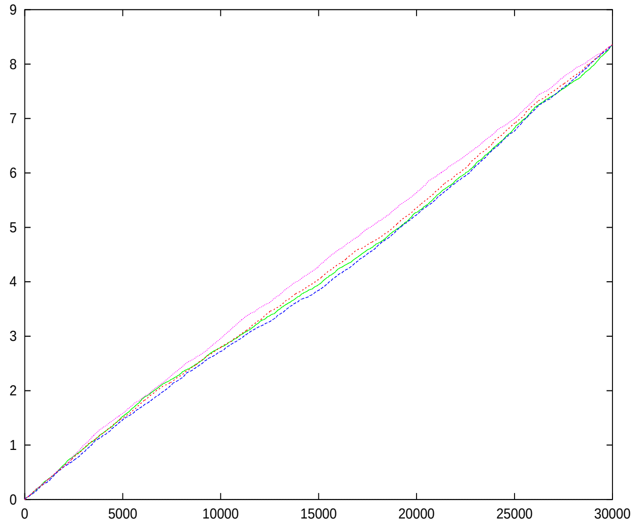

Figure 3: The entropy distribution of the proposed system as a function of bit count. The entire 16 bit number space was divided into 30,000 equal slots. The distribution is for eight bit ASCII characters represented by decimal digits 01, 30, 31 and 255. Note the linear increase in the entropy distribution as a function of the bit count. This indicates a uniform distribution at all bit levels and hence a maximum cycle length for the random sequence.

The graphical representation of this measure reveals any orientation of the distribution density towards the corresponding plain text information. The distribution of the entropy of the cipher texts corresponding to the ASCII characters represented by the decimal digits 01, 30, 31 and 256 are shown in figure 3. An enlarged portion of figure 3 is shown in figure 4. Figure 3 indicates that the entropy converges to the same value while its deviates slightly in the middle. Also it is observed that the entropy distribution increases linearly as a function of the bit-count. This means there is a uniform distribution of the random values over the entire state vector space. In other words, the cipher has effectively masked the information by generating a space filling sequence with uniform probability of occurrence at all levels, which is ideal from a theoretical point of view.

In the second round of tests, we used random sequences of English prose of equal length as plain text. It was observed that the entropy curves were more linear and indistinguishable compared to that of the single variable sequences.

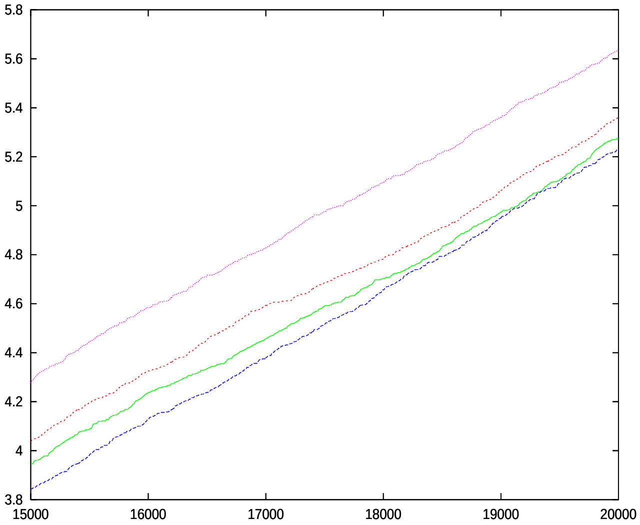

Figure 4: A magnified view of the central region of figure 3 shows a marginal deviation in the entropy distribution for each ASCII value. Although it appears to reveal some information regarding the plain text information, it remains to be known how far this could be used as a possible tool for breaking the cipher equation.

In both cases, the linearity of the curve ensures maximum cycle length and uniform distribution over all the possible state vectors [23], while the non-linearity of the chaotic system introduces necessary randomness. Thus we expect the cipher to be reasonably secure against differential cryptanalysis.

4.2 Complexity measure

The widely accepted test for the randomness of a sequence is the Chi-square test [21]. The cipher described in the present paper uses the cipher text itself as part of its input vectors. As a consequence, just like other self-synchronous stream ciphers, it is difficult to judge this cipher using Chi-square tests.

However, the randomness of a sequence may be estimated from its complexity. By complexity we mean the minimum number of bits required to represent the information contained in it. A direct measure of this is its compressibility [6]. Although the sequence is pseudo random, meaning that they may be compressed using the exact equation that generated them, since the details of the parameters of the equation (secret keys) are not available to the hacker, we discard this possibility. In this situation, a rough estimate of the complexity of a data set may be obtained using the Huffman algorithm [25] for data compression. The Huffman algorithm computes the redundancy of data in a sequence and then replaces them using some symbols that require lesser number of bits and thus reduces the size of the data sequence. We do not go into the details of the algorithm here, but would cite [26] as a practical reference. The ratio of the size of the original cipher text to that of the compressed text is a measure of the complexity or randomness of the cipher text. The higher the ratio, the lesser is the complexity.

For the analysis, a large set of English prose was chosen at random. Each plain text was about a few kilobytes in size. The choice was made taking into consideration the ease of comparison with earlier studies. We used the ASCII code to represent the English alphabets (8 bits per letter). The Linux version of *gzip software* with maximum compression option (gzip -9) was used to compress the files. This is the GNU lossless data compression package popular on most UNIX type machines. In all the test cases, the compression ratio was less than one since the coding resulted in files larger in size than the original files. This additional size was required to keep the coding information and the original file could not be compressed at all. This indicates that the complexity measure of the cipher text sequence is the same as the complexity of the embedding finite space and is the theoretical maximum.

4.3 Brute force attack

The straight forward attack on any cipher is the brute force attack. Except for the computational intensity, this is as simple as trying all the possible combinations of the variables in the system. In the cipher proposed by the cipher equation, the independent variables are λ, x_n and x'_n. Assume that the value of λ is significant up to k bits and that of x_n and x'_n each up to m bits. We say that a bit is significant if the system has a different evolution when the bit flips its value. Thus for brute force attack, the hacker requires a maximum of 2^{2m+k} attempts to break the value of λ. If we assume that $k = m = 16$ bits, the brute force attack requires a maximum of 2^{48} computations.

The general procedure of the attack is like this: take some samples from the middle of the ciphertext that appears to have the highest probability of containing some meaningful data. Reverse engineer using the cipher equation all the possible values of x_n, x'_n and λ until some meaningful text is generated. If the hacker has a known plain text database, the complete process may be automated. Then he only need to compare the two. If an exact match is found, he has succeeded in breaking the codes. He may now apply this information on the entire ciphertext and retrieve the location and method used to hide the starting value. This completely decodes the cipher.

One solution to this problem is to increase the number of bits used for the parameters in the cipher. But this is not foolproof, since the probability for getting the right guess in a reasonable number of iterations is not zero. We suggest multiple encryption as a better approach. Since a single flip of any of the bits of the parameters would result in a different sequence that diverge away from the original sequence exponentially at every point, in this case, the hacker has to either assume the ciphertext generated in the previous encryption or has to do $2^{n(2m+k)}$ computations to obtain the plaintext information. Here n is the number of times the encryption has been done. Certainly, there is an upper limit to the possible maximum value for n. This is mainly due to the finiteness of the embedding phase space and the block size. As computational power increases, there would be a proportional increase in the word size handled by the machine also. Thus it may be argued that there would always exist a safe margin between the attainable complexity and the time required for brute force attacks.

4.4 Plain text attack

We will now attempt a known plain text attack on the system. Let us assume that P_n represent a k-bit number with each bit represented by $\mathrm{P}_n[\mathrm{k}]$. Then, $\mathrm{P}_n[\mathrm{k}] = 0$ iff $\mathrm{x}_n[\mathrm{k}] = \mathrm{x}'_n[\mathrm{k}]$. The initial setting is such that x_0 and x'_0 are different. However, due to the mixing of x_{n+1} with C_n, there is a non-zero probability for $P_n = 0$. The risk with P_n going to zero is that this would cause the cipher to be same as the plain text information. However, it can be shown that P_n and P_{n+1} will never be zero simultaneously if y_n, the plain text is not a null vector. This is a simple condition since a null vector does not have any information content and is never transmitted.

In this situation, with a known plain text, the attacker is able to identify the locations in the cipher text that match the state vectors in the plain text and thus conclude that they correspond to locations where $P_n = 0$. Considering this possibility, the question is to what extent the attacker may extend this information to crack the secret codes, namely λ and some other critical parameters used to generate the Cipher text like the coding of the starting vector x_0.

Let at some level m, $P_m = 0$.

This means, $x_m = x'_m$.

$\Rightarrow x^*_m \bigoplus x_{m-1} \bigoplus x'_{m-1} \bigoplus y_n = x'_m$ with x^*_m representing x_{n+1} in the cipher equation.

$\Rightarrow x^*_m \bigoplus x_{m-1} \bigoplus x'_{m-1} \bigoplus x'_m = y_n$

$\Rightarrow (4\lambda x_{m-1}(1 - x_{m-1})) \bigoplus x_{m-1} \bigoplus x'_{m-1} \bigoplus (4\lambda x'_{m-1}(1 - x'_{m-1})) = y_n$

Due to the inability to solve the above equation for λ and x_0, we conclude that the cipher is resistant to known plain text attacks.

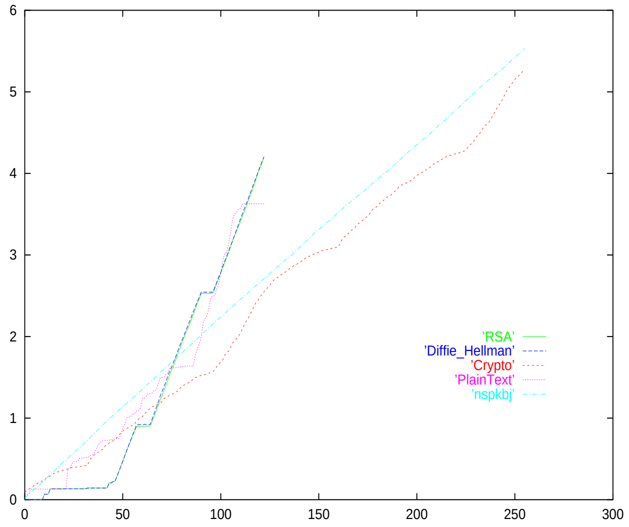

Figure 5: A comparative study of the proposed system with three popular ciphers is shown here. A flat region in the graph represents regions that does not have representative vectors in the cipher text. The ideal case is to span the entire phase space uniformly (a straight line with positive slope), as that would correspond to maximum cycle length for the generated random sequence.

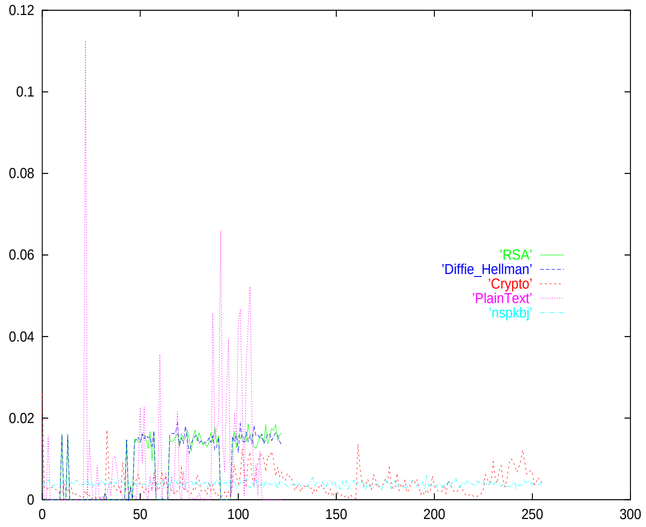

Figure 6: For an ideal cipher, the probability distribution density of the cipher text should be independent of the frequency distribution of the plain text data. The diagram shows the distribution of the plain text information along with the output of the three ciphers mentioned in the text and also the one proposed. The proposed cipher has almost ideal distribution (equal level) at all bit levels.

5 Comparison with other Ciphers

For a comparison of the new method with some of the popular ciphers, three popular ciphers were chosen. They are:

1. Crypto (v.2.9) 95—NT: Copyright 1998 by Gregory Braun. This software is a free-ware.

2. PGP 5.0 for Personal Privacy, Evaluation version, Copyright 1997 by Pretty Good Privacy and using RSA key of size 2048.

3. PGP 5.0 for Personal Privacy, Evaluation version, Copyright 1997 by Pretty Good Privacy and using DSS/Diffie-Hellman key of size 1024/3072.

The entropy distribution as a function of the state vector for all the above three ciphers along with that from the new method is shown in figure 5. It is seen that the proposed method produces the ideal distribution, a linear increase in entropy count at all bit levels in the finite space.

Resistance of a cipher to differential and linear cryptanalysis may be quantitatively measured to some extent as its ability to hide any tangible information regarding the population of the states in the plain text. This is to say that the probability distribution density of the cipher text should be independent of the frequency distribution of the plain text data. In an ideal case, the probability of occurrence at all possible levels of the iteration should be the same. This would result in a straight line parallel to the axis of possible states of the system. Figure 6 shows such a graph of the probability distribution density of the cipher texts produced by the popular ciphers mentioned above along with that by the new method. It may be noted that the new method produces a graph which is almost a straight line as required by the ideal case.

6 Conclusion

A new algorithm for secure communication using mixing of two or more chaotic sequences is discussed. The mixing of two near by trajectories of the logistic equation in the fully developed chaotic scenario is shown to generate pseudo random numbers of high complexity. The system appears to be resistant to brute force, known plain text and differential cryptanalysis. Since the computation is limited to finite space and mostly XOR operations, a successful attack based on Differential Power Analysis and Timing Attacks are not realistic on the system. Thus even for small values of the secret key size, the cipher appears to be secure to the popular methods of cryptanalysis. The paper also discuss some measures to identify significant features like cycle length and probability distribution density and compares the new cipher with some popular ciphers.

References

[1] G. Brassard, *Modern Cryptology - A Tutorial*, LNCS 325, Springer-Verlag, Berlin, 1988.

[2] W. Diffie, M. E. Hellman, "Privacy and Authentication: An introduction to Cryptography," in *Proc. IEEE*, vol. 67, no.3, pp. 397 - 427, 1979.

[3] P. Berge, Y. Pomeav and C. Vidal, *Order within Chaos*, Hermann, Paris, France, 1986.

[4] J. Gleick,*Chaos: Making a New Science*, Viking Penguin, New York, 1987.

[5] G. Nicolis and I. Prigogine, *Exploring Complexity*, W. H. Freeman and Company, New York, 1989.

[6] T. Ritter, "The Efficient Generation of Cryptographic Confusion Sequences," *Cryptologia*, vol. 15, no.2, pp. 81 - 139, 1991.

[7] M. Louis Pecora and L. Thomas Carroll, "Synchronization in Chaotic systems," *Physical Review Letters*, vol. 64, no. 8, pp. 821 - 824, 1990.

[8] H. Li and J. Chern, "Coding the chaos in a semiconductor diode for information transmission," *Physics Letters A*, vol. 206, pp. 217 - 221, 1995.

[9] Y. H. Yu, K. Kwak and T. K. Lim, "Secure communication using small time continuous feedback," *Physics Letters A*, vol. 197, pp. 311 - 315, 1995.

[10] T. Ritter, "Substitution Cipher with Pseudo-Random Shuffling: The Dynamic Substitution Combiner," *Cryptologia*, vol. 14, no. 4, pp. 289 - 303, 1990.

[11] M. MacLaren and G. Marsaglia, "Uniform Random Number Generator," *Journal of Association for Computing Machinery*, vol. 12 no. 1, pp. 83 - 89, 1965.

[12] S. Wolfram, "Cryptography with Cellular Automata (extended abstract)," in: *Advances in Cryptography, CRYPTO'85 Proceedings*, Springer-Verlag, pp. 429 - 432, 1986.

[13] V. Pless , "Encryption Schemes for Computer Confidentiality," *IEEE Transactions on Computers*, vol. C-26, no. 11, pp. 1133 - 1136, 1977.

[14] C. Beck, *Signal Processing VI: Theories and applications*, Elsevier, Amsterdam, 1992.

[15] J. Cernak, "Digital generators of chaos," *Physics letters A*, vol. 214, pp. 151 - 160, 1996.

[16] R. Matthews, "On the Derivation of a Chaotic Encryption Algorithm," *Cryptologia*, vol. 13, pp. 29 - 42, 1989.

[17] D. Wheeler, "Problems with Chaotic Cryptosystems," *Cryptologia*, vol. 13, pp. 243 - 250, 1989.

[18] D. Mitchell, "Nonlinear Key Generators," *Cryptologia*, vol. 14, pp. 350-354, 1990.

[19] E. Rietman, *Exploring the Geometry of Nature*, Windcrest Books, Blue Ridge Summit, P.A, 1989.

[20] T. Siegenthaler, "Decrypting a Class of Stream Ciphers Using Ciphertext Only," *IEEE Transactions on Computers*, vol. C-34, pp. 81 - 85, (1985).

[21] D. Knuth, *The art of Computer Programming, Vol.2, Seminumerical Algorithms*, 2nd ed., Addison-Wesley, Reading, Massachusetts, 1981.

[22] T. Ritter, "The story of Combination Correlation: A Literature Survey," Research comments from Ciphers by Ritter, (1996), URL: http://www.io.com/∼ ritter/ under Literature Surveys and Reviews.

[23] E. Fischer, "A theoretical Measure of Cryptographic Performance," *Cryptologia*, vol. 5, no.1, 1981.

[24] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," in: *Advances in Cryptology: Crypto'90*, Springer-Verlag, 1990.

[25] D. Huffman, "A method for the construction of Minimum-Redundency Codes," in: *Proceedings of the IRE*, vol. 40, pp. 1098 - 1101, 1952.

[26] S. Heller, *Efficient C/C++ Programming: Smaller, Faster, Better*, 2nd Edition, chapters 1-4, Academic Press Professional, 1991.