Cryptography using Multiple – Chaos

R. Raja Kumar1*, R. Pandian2, T.Prem Jacob3, A.Pravin4 and P.Indumathi5

1Professor, Mathematics Department, Sathyabama Institute of Science and Technology, Sholinganallur, Chennai, India.
2Associate Professor, Department of Electronics and Instrumentation Engineering, Sathyabama Institute of Science and Technology, Sholinganallur, Chennai, India.
3, 4Associate Professor, Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Sholinganallur, Chennai, India.
5Professor, Department of Electronics Engineering, MIT Campus of Anna University, Chennai, India.

Contact email: rrajakumar.maths@sathyabama.ac.in

Abstract. Seemingly random, chaotic dynamic systems have state variables that move about in a non periodic, bounded fashion. The sensitivity to initial conditions of chaotic signals also holds an interesting pattern. A seemingly tiny change in the values of initial condition, can greatly affect the values of output. This means that the cross-correlation of two chaotic functions of same source is very low. This pattern of chaotic functions makes their use in the field of cryptography more interesting. A more complex version of chaos is the multiple-chaos, where two or more chaotic functions are used, enhances the security of the procedure. Additionally the dynamically varying look-up table, which is being transmitted along with the encrypted message, ensures high reliability in transmitting the data. Overall, this method ensures high security and high reliability in transmitting the data.

1. Introduction
Chaos theory has been used in several scientific applications. Among all, the most promising applications of chaotic system is their use in the field of chaotic encryption where the utilization of nonlinearities and the forcing of the dynamical system to a chaotic state will fulfill in our opinion the basic cryptographic requirements. Due to the nonlinear mechanisms that lead to a chaotic behavior, this one is too difficult to predict by analytical methods without the secret key (initial conditions and/or parameters) being known. This would reduce a potential attack to one category that of a brute force attack, in which any attempt to crack the key depends directly upon how long the key is[1].

Chaos is a particular state of a nonlinear dynamical system and appears only in certain conditions, e.g. for certain values of the system parameters and only in dynamical systems characterized by continuous values. The chaotic state can be observed in a first approach by the existence in the phase space of a chaotic attractor or fractal in which all the system trajectories evolve following a certain pattern (basin of attraction) but are never the same. In a more analytical approach the chaotic state can be very well studied by the Lyapunov exponents which globally characterize the behavior of a dynamical systems. Chaos will be observed only when there is at least one positive Lyapunov exponent and the total sum of all exponents is negative, that is the dynamical system has a stable but random like state called chaotic state [2].

M.S.Baptista proposed a chaotic crypto system using the ergodicity property of the simple low dimensional and chaotic logistic equation [3]. It encrypts the message text into the number of iterations needed for a chaotic map to reach a region on a phase space that corresponds to the text. He demonstrated
his approach using a simple one dimensional logistic map. In this work, to provide high security we use multiple-chaotic technique, that is more than one chaotic map is used for encryption. Here stream cipher technique is implemented meaning one character at a time is encrypted. Also this algorithm is proposed for the text messages. In this algorithm logistic map and tent map are the two chosen chaotic signals. For the characters in the odd position logistic map is used for encryption whereas for the characters in the even position tent map is used[4].

The rest of this paper is organised as follows: Section 2 summarises related researches. Section 3 summarises brief introduction of the proposed works. Section 4 summarises results and discussions. Section 5 summarises conclusion and future work.

2. Related Works
For a dynamical system to be defined as chaotic, it must have the following properties:

- It must be sensitive to initial conditions
- It must be topologically mixing
- Its periodic orbits must be dense

In mathematics, chaotic map is a map that exhibits some sort of chaotic behavior. Chaotic maps often occur in the study of dynamical systems. There are more chaotic functions which exhibits this function. For using multiple-chaos, chaotic maps have to be selected from the list of chaotic maps [5] mentioned in Table 1.

| Map                | Time domain | No. of space domains |
|--------------------|-------------|----------------------|
| Arnold’s cat map   | Discrete    | 2                    |
| Circle map         | Discrete    | 1                    |
| Duffing map        | Discrete    | 2                    |
| Exponential map    | Discrete    | 2                    |
| Gauss map          | Discrete    | 1                    |
| Henon map          | Discrete    | 2                    |
| Logistic map       | Discrete    | 1                    |
| Lorentz attractor  | Continuous  | 3                    |
| Rosslers map       | Continuous  | 3                    |
| Tent map           | Discrete    | 1                    |

For our algorithm Logistic map and tent map are chosen since both the maps are discrete and have one space domain. Both the maps are explained in detail in the following sections.

2.1. Logistic Map
The logistic map is a polynomial mapping (equivalently, recurrence relation) of degree 2, chaotic behaviour can arise from very simple non-linear dynamical equation represented in 1.1

\[ x_{n+1} = r * x_n (1 - x_n) \tag{1.1} \]

where, \( x_0 \) is a number between zero and one and \( x_0 \) represents initial value and \( r \) is a positive number.

The logistic map is one of the simplest forms of a chaotic process. Basically, this map, like any one-dimensional map, is a rule for getting a number from a number. The parameter ‘\( r \)’ is fixed, but if one studies the map for different values of ‘\( r \)’ (up to 4, else the unit is no longer invariant) it is found that ‘\( r \)’ is the catalyst for chaos. For values of ‘\( r \)’ chosen between 0 and 4, \( f(x) \) is a map from the interval \([0, 1]\) to itself.
At \( r=4 \), the resultant value will be either 0 or 1. To avoid that computation problem, we need to choose \( r \) value between 3.57 and 4. For any value of \( r \) there is at most one stable cycle. A stable cycle attracts almost all points. For an \( r \) with a stable cycle of some period, there can be infinitely many unstable cycles of various periods [6].

2.2. Tent Map

The tent map, very much like the logistic map, displays some very specific chaotic effects. The tent map with parameter \( \mu \) is the real-valued function \( f_\mu \), defined by 1.2,

\[
f_\mu := \mu \min\{x, 1-x\},
\]

the name being due to the tent-like shape of the graph of \( f_\mu \) as shown in Fig 1.

For the values of the parameter \( \mu \) within 0 and 2, \( f_\mu \) maps the unit interval \([0,1]\) into itself, thus defining a discrete time dynamical systems on it (equivalently a recurrence relation). In particular, iterating a point \( x_0 \) in \([0,1]\) gives rise to a sequence \( x_n \) as mentioned in 1.3,

\[
x_{n+1} = \begin{cases} 
\mu x_n & \text{for } x_n < \frac{1}{2} \\
\mu (1-x_n) & \text{for } \frac{1}{2} \leq x_n
\end{cases}
\]

where \( \mu \) is a positive real constant. Choosing for instance the parameter \( \mu=2 \), the effect of the function \( f_\mu \) may be viewed as the result of the operation of folding the unit interval into two, then stretching the resulting interval \([0,1/2]\) to get again the interval \([0,1]\). Iterating the procedure, any point \( x_0 \) of the interval assumes new subsequent positions as described above, generating a sequence \( x_n \) in \([0,1]\).

When more than one chaotic signal is used for encryption then it is called as multiple-chaos. It will provide high degree of security compared to existing Baptista type chaotic cryptography. In our proposed algorithm logistic map and tent map are used. The method by which both the maps are combined in our algorithm is explained in the following section.

3. Proposed Work

The disadvantage of Baptista approach is that, firstly, the resultant ciphertext usually concentrates at small number of iterations and so the distribution of ciphertext is not flat enough for high security. Baptista method does not ensure the reliability of the message.

In the proposed algorithm, to provide high security we use multiple-chaotic technique that is more than one chaotic map is used for encryption. Here stream cipher technique is implemented meaning one character at a time is encrypted. Also this algorithm is proposed for the text messages. In this algorithm logistic map and tent map are the two chosen chaotic signals. For the characters in the odd position logistic map is used for encryption whereas for the characters in the even position tent map is used.
The Advantages of Proposed Algorithm are listed below: For iterative values to become flat i.e., to make the iterative values distributive a threshold value is fixed. Look-up table is included to ensure the reliability of the received message.

3.1. Proposed Algorithm Encryption
1. Choose key $x_0$ & interval of interest $[x_{\text{max}}, x_{\text{min}}]$.
2. Sub divide the interval $[x_{\text{max}}, x_{\text{min}}]$ into ‘$s$’ sub-intervals
   - Subdivide the interval in the range $[x_{\text{min}} + (S-1)\epsilon, x_{\text{min}} + S\epsilon]$ Where $\epsilon = (x_{\text{max}}-x_{\text{min}})/S$
   - For alphabetical representation S represents number of ASCII characters
   - for image/audio/video $S = 256$
3. For characters in odd position value do $x_0:=r \times x_0 \times (1-x_0)$ (logistic map).
4. For characters in even position value do
   \[
   x_{n+1}= \begin{cases} 
   \mu x_n & \text{for } x_n < \frac{1}{2} \\
   \mu (1 - x_n) & \text{for } \frac{1}{2} \leq x_n 
   \end{cases}
   \] (tent map)
5. Use the final value of previous character obtained from the iterative function as initial value for the next character.
   The iterative value for each character is the encrypted message of that character.
   The encrypted value which is obtained from the proposed algorithm is transmitted to the receiver along with the initial value.

3.2. Proposed Algorithm Decryption
1. Get key $x_0$ from the transmitter.
2. Sub divide the interval $[x_{\text{max}}, x_{\text{min}}]$ into ‘$s$’ sub-intervals as obtained in the encryption process.
3. Get the encrypted values.
4. Keep decrementing the encrypted values for each character by 1 and do
   - $x_0:=r \times x_0 \times (1-x_0)$ (logistic map) for characters at odd position and
   - $x_{n+1}= \begin{cases} 
   \mu x_n & \text{for } x_n < \frac{1}{2} \\
   \mu (1 - x_n) & \text{for } \frac{1}{2} \leq x_n 
   \end{cases}$ (tent map) for characters at even position
5. Keep decrementing until the count reaches zero.
6. The sub-interval in which $x_0$ lies is found out. The character/value of the corresponding sub-interval is the decrypted value.

3.3. Iterative values
In order to make the distribution of ciphertext as flat enough a threshold is fixed for the number of iterations. In proposed algorithm it is chosen as 200. If the resultant iterative value is less than 200, it continues to iterate till the value become greater than 200 so that the distribution of ciphertext becomes flat enough. The same threshold is used in receiver side and the message is decrypted.

3.4. Look-up Table
As in Baptista, a multiple-chaotic encryption scheme for encoding the information that is to be transmitted to the receiver is proposed. An initial look-up table containing the mapping of each possible input combination to equal-width regions in the interval $[x_{\text{min}}, x_{\text{max}}]$ of the phase space of the logistic map should be set in advance. This initial mapping can be in order or at an agreed setting. For the encryption of the $i^{th}$ message block, let the logistic map iterate until the trajectory first falls into the region corresponding to the ASCII code of this message block. Similar to Baptista’s method, the iteration will continue if the current number of iterations is smaller than a pre-defined minimum number of iterations. This prevents cryptanalysis upon the loophole of zero or just a few iterations.
On the other hand, if the current number of iterations is large enough, it is sent immediately as the ciphertext and no random numbers need to be generated. This is an advantage to resource constraint computing environment such as smartcard because no additional hardware or software random number generators are required. Before encrypting the next message block, the look-up table is updated dynamically by exchanging the \( p \)th entry \( e_i \) with another entry \( e_j \). The interval \( v \) between these two entries is determined by the current value of \( x \) using the following formula as mentioned in 1.4.

\[
v = \left( \frac{x - x_{\min}}{x_{\max} - x_{\min}} \right) N
\]

(1.4)

where \( N \) is the total number of entries in the table, \( x_{\max} \) is the maximum value of \( x \) in the chosen interval and \( x_{\min} \) is the minimum value. Notice that the sum \( (i + v) \) may be larger than \( N \). In this case, modulus operation is performed so that the index increment is in a cyclic manner and the value of \( j \) is still within \( N \). As a result an equation 1.5 is obtained as mentioned below,

\[
j = (i + v) \mod N
\]

(1.5)

As the swapping of entries in the look-up table is determined by the \( x \) value which in turn depends on the plaintext, the final look-up table should be different for different messages. However, for two messages with only a single bit difference in the last block, their hash values can be very similar to each other, although not exactly the same. This is because the last message block can only affect two entries in the look-up table. A straightforward approach for solving this problem is to swap more pairs of entries during the encryption of a block of plaintext. However, as the location of the entries to be swapped is determined by a value modulo \( N \), some entries may never be affected. This approach still cannot guarantee that the hash values for two “very similar” messages are totally different. Therefore additional operations are to be incorporated.

A simple way is to perform the encryption and dynamic table updating processes for the whole message more than once so that the look-up table can be changed continuously during the second, third and later runs. As a result, previously proposed chaotic cryptographic scheme is generalised by allowing the swapping of multiple pairs of entries in the look-up table during the encryption of each input block.

In order to swap more pairs of entries in the look-up table during the encryption of a single plaintext block, the interval \( v \) is calculated, between two swapping entries. Starting from the current entry \( i \), we swap the entries at locations \( i \) and \( (i + v) \mod N \). Then we continue to swap \( (p - 1) \) pairs of entries starting from the entry next to the one last visited, i.e., \( ((i + v + 1) \mod N) \leftrightarrow ((i + 2v + 1) \mod N), \ (i + 2v + 2) \mod N) \leftrightarrow ((i + 3v + 2) \mod N), \ldots, \ 
((i + (p - 1)v + p - 1) \mod N) \leftrightarrow ((i + pv + p - 1) \mod N), \) where \( p \) is the total number of pairs to be swapped during the encryption of each plaintext block. After that, the look-up table updating process is completed and the iterations on the logistic map for the next plaintext block started. Then swapping of \( p \)-pairs in the look-up table is performed again by incrementing \( i \) and calculating a new value of interval. Notice that if the index \( i \) reach the bottom of the table, it will start from the top again.

Under the generalized chaotic cryptographic and hashing scheme, the operations do not necessarily stop after finished encrypting the last message block. If the number of runs \( r \) is greater than one, the whole process will repeat by encrypting from the first message block again using the current value of \( i \) and the look-up table. As a result, a tiny change in the last message block that affects \( p \) pairs of entries in the look-up table will be “amplified” by the subsequent runs of the encryption process on the whole message.

For the correct decryption of the ciphertext, the system parameter \( b \), the initial \( x \) value \( x_0 \) and the initial look-up table must be transmitted to the receiving end first. For the parameter \( p \) it can have different value for different messages and hence its security gets enhanced. This is possible as they are not critical parameters for initializing the decryption process and therefore can be encrypted and included in the ciphertext. As \( p \) is required for the subsequent updating of the look-up table, it should
be encrypted and transmitted first. After that the message is encrypted by using the proposed algorithm. At last the encrypted $p$, encrypted message and final look-up table is transmitted to the receiver.

At the receiver side, the parameter $p$ is obtained first by decrypting the first block of ciphertext with the agreed $b$, $x_0$ and the initial look-up table. Then the swapping of $p$ pairs of entries in the look-up table is performed. And the resultant look-up table is compared with received look-up table. If they are same, the message can be authenticated. Otherwise, it is alternated or the ciphertext sequence is corrupted during the transmission. Thus it checks the reliability of the message.

Unlike the ciphertext size, the encryption and decryption time depends on $p$, in addition to the message length. The higher the value of the parameter, the higher the security but the longer the encryption and decryption processes. A general guideline in selecting the value of the parameter is that all the entries in the final look-up table should be affected even only the last bit of the message is changed. However, for multi-media files with length much longer than $N$, the encryption time can be painfully long if $p$ is large. Therefore, a small value of $p$ is suggested. Notice that the minimum number of iterations is always 200.

In summary, we have generalized our recently proposed fast chaotic cryptographic scheme that is based on updating the look-up table dynamically. The number of pairs of entries in the look-up table to be swapped for each message block can be set differently for different messages. As a result, the security is further enhanced. Moreover, the final look-up table can be considered as a hash value or message authentication code for checking the authentication and the completeness of the message.

4. Simulation Results
An initial look-up table containing the mapping of each possible input combination to equal-width regions in the interval $[x_{\text{min}}, x_{\text{max}}]$ of the phase space of the logistic map should be set in advance. For the Encryption of $i^{th}$ message block,

we use proposed algorithm to iterate until the trajectory falls into the region corresponding to the ASCII code of this message.

At the transmitter side, an initial look-up table containing the mapping of each possible input combination to equal-width regions in the interval $[x_{\text{min}}, x_{\text{max}}]$ of the phase space of the logistic map should be set in advance. For the encryption of $i^{th}$ message block, we use proposed algorithm to iterate until the trajectory falls into the region corresponding to the ASCII code of this message. In order to be more secure we use different $p$ for different message block. Thus while transmitting we need to send, encrypted ‘p’ encrypted message i.e. iterative values and final look-up table.

At the receiver side initial look-up table is built and ‘p’ is decrypted. Based on the retrieved ‘p’ the initial look-up table is swapped following the same procedure as that of transmitter. Retrieve the final look-up table from the transmitter. Now the final look-up table is compared with the received look-up table. The similarity of the look-up table ensures the correctness of the retrieved message. This ensures the reliability of the transmitted message. The results are simulated using MATLAB.

5. Conclusion and Future Work
The proposed algorithm which uses multiple- chaos, that is two chaotic functions for encryption and decryption of text, ensures a high degree of security. The used chaotic functions are logistic map and tent map. And further, a dynamically varying look-up table, which varies according to the input messages, is being transmitted along with the encoded message. And a similar procedure is being carried out at the receiver side, and finally both the look-up table are compared. This ensures a high reliability in transmitting the message.

The work may be extended to encrypt and decrypt the image files. Also, the time complexity may be reduced by taking some appropriate measures.

References
[1] M.S.Baptista, “Cryptography with chaos”, Physics Letters A 240(1-2) pp. 50-54 (1998).
[2] Kwok-Wo Wong, “A combined chaotic cryptographic and hashing scheme”, Physics Letters A
307 pp. 292–298 (2003).

[3] W.K. Wong, L.P. Lee, K.W. Wong, “A modified Chaotic cryptographic Method”, Computer Physics Communications 138(3) pp. 234-236 (2001).

[4] K.W. Wong, S.W. Ho, C.K. Yung, “A Chaotic Cryptography Scheme for Generating Short Cipher Text”, Physics Letters A 310(1) pp. 67-73 (2003).

[5] G. Alvarez, F. Montoya, M. Romera And G. Pastor, “Cryptanalysis Of an Ergodic Chaotic Cipher”, Physics Letters. A,311 pp. 172-179 (2003).

[6] K.W. Wong, “A Fast Chaotic Cryptographic Scheme with Dynamic-Look Up Table”, Physics Letters A 298 (4) pp. 238-242 (2002).