Logistic Map based Cryptography

Ashay Wakode
Mechanical Engineering Department
IIT Bombay
Mumbai, India
ashaywakode@iitb.ac.in

Dyanesh Pawaskar
Mechanical Engineering Department
IIT Bombay
Mumbai, India
pawaskar@iitb.ac.in

***Abstract*—The report is a modest review of a work done in the intersection of fields of cryptography and chaos. A cryptographic technique based on logistic map is presented as main result. Various key concepts required for understanding the technique are introduced before hand. A modification introduced in the main result is compared with the original technique. A comparison with existing popular cryptographic methods is also performed. Lastly, required future work on the main result is discussed.**

***Index Terms*—Logistic Map, Symmetric Cryptography, Shadowing, Ergodicity**

I. Introduction

People have been using Cryptology as long as 2500 years. It is used to communicate between two host parties without a third-party knowing about the contents of communication unless otherwise permitted by the communicating hosts. Cryptology involves two processes, the first one is cryptography, in which the message is converted into a secret message (code), which then is transmitted to other host party, which performs second process of cryptology, cryptoanalysis. Cryptoanalysis is the conversion of secret message into a comprehendable material [1]. With the development in the fields of number theory and computing, mankind has produced more efficient (requiring lesser time and energy), more powerful (harder to break) and complex (requiring usage of complex mathematical concepts). These cryptologic techniques are at the heart of computer security. New techniques are being developed at a faster rate recently due to the advent of powerful computers, which can easily break a code and may get unwanted access to information. The cryptology is simply referred to as cryptography in much of recent scientific literature, So, here on in this paper "Cryptograhy" will refer to both making and breaking of code.

Our world is filled with processes, phenomenons and systems which are not governed by linear dynamical mathematical expressions but by more complex non-linear dynamical expressions. Chaos theory is the study of such non-dynamical systems. One of the many striking property of such systems is the heavy dependence of such systems on the initial conditions. Such systems evolve in very different way once the initial condition is changed very slightly [2]. One of such systems is the evolution of an animal population described in the most simple manner. Logistic map is a very simple way of modelling such a system. More information about the logistic map is presented in the following section.

Logistic map has a very useful property of producing random numbers which can be exploited in cryptography. Such random numbers produced aren't truly random and therefore called pseudo-random numbers, the reason for the same is presented in next section. There are many ways in which pseudo-random number producing property can be used for developing cryptographic technique. One such method was introduced by Baptisa in 1998 [3]. The introduced method falls under the category of a symmetric cryptography (described in next section). [4] further enhances the method by suggesting changes in the working of the method. But, even the enhanced method is not used for secure communication in day to day life since it has some lacunae. Methods like RSA, AES, Hash function and many more are used on a daily basis. A comparison of the method introduced by Baptisa and popular methods is presented to throw light on the lacking aspects method.

This paper is effort to showcase the work done by Baptisa. Section II introduces important concepts required for understanding the cryptographic method in [3] which is showcased in Section III. Section III talks about the method and tweak performed in [4]. Further, a comparison between the original method and the tweaked method is presented. This section ends with a discussion of popular cryptographic methods and Baptisa's method. Lastly, the paper concludes with the suggestions on further work to make the method more usable.

II. Preliminaries

A. Logistic Map

A simple mathematical model for predicting an animal population x_{n+1} at time t_{n+1} for a current animal population of x_n at time t_n is given as,

$$x_{n+1} = b \times x_n \times (1 - x_n) \tag{1}$$

Here, b (also denoted as r) is called tuning parameter [5]. This quadratic recurrence relation is known as **Logistic Map**. The range of b for studying the eq.1 is $b \in [0, 4]$. Since, for $b > 4$, $x_n \to -\infty$, as $t_n \to \infty$. eq. 1 has been studied for $b < 0$ [6], but for the purpose of this report we don't need to work with this range of b.

Here, Fig. 1 represents the relation between equilibrium value of x_n against b with some initial condition $x_0 \in (0, 1)$.

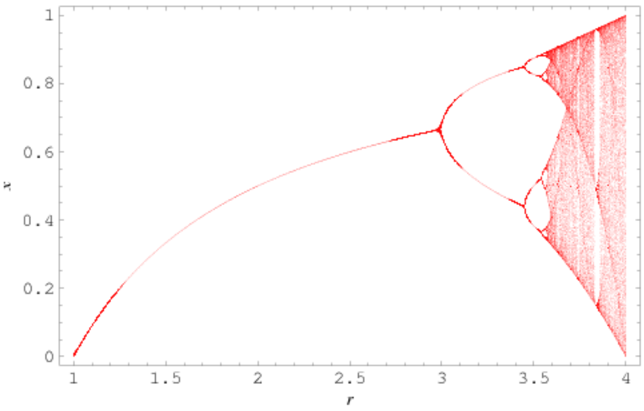

Fig. 1. x (x_n as n $\to \infty$) vs b

Fig. 1 is the bifurcation diagram of Logistic map [6]. Many interesting properties can be deduced from the bifurcation diagram.

B. Ergodicity

A system is said to be ergodic if, the time average of the system trajectory equals the ensemble average. Time average ($\bar{X}$) is the average of system state for all the points on a single system trajectory as $t \to \infty$, mathematically,

$$\bar{X} = \lim_{x\to\infty} 1/T \times \int_0^T x(t)\,dx \tag{2}$$

Ensemble average is the average of the system state at some time interval $(t, t+dt)$ over infinitely many system trajectories, as the time interval shrinks to t. Mathematically,

$$\langle X \rangle = \lim_{N\to\infty} 1/N \times \sum_{n=1}^{N} x(t) \tag{3}$$

Above concepts can be easily understood by observing Fig.2. Here, Green rectangle represents the time average for finite time, while Red rectangle represents the Ensemble average for 5 trajectories. Once we let time and number states go to infinity, we get the Time and the Ensemble averages of the system.

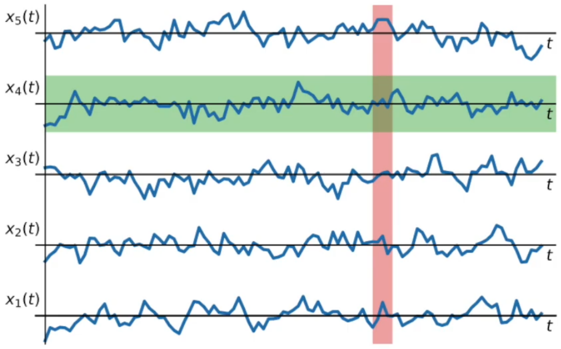

Fig. 2. State trajectories

Next question comes to mind is, whether the Logistic Map is ergodic. The Logistic Map is ergodic for $b \in (1, 4)$ [7]. This has also been numerically verified in [8].

Here, the bifurcation diagram shows that for some values of b the x equilibriates to every $x \in [xi, xo]$. This means that the state (x_n) of the Logistic map will never equilibriate to some $x^* \in [xi, xo]$ and hence will never revisit the value it had previously landed upon. Since a new value for the state is obtained for each time iteration, we get random numbers eq.1 is iterated. So, random numbers can be generated from the Logistic map. But, the numbers aren't truly random as the eq.1 can be iterated the previously iterated number of times to get the same random number. Therefore, the numbers obtained from Logistic map are called pseudo-random.

C. Cryptography and Classification

[9] throws light upon various cryptographic methods and their classification. Cryptography can be majorly divided into Classic and Modern Cryptography. The classification can be observed in the fig. 3. There are further sub-division to classic and modern cryptography.

- **Classic Cryptography** : The cryptographic methods developed before 1900 that use simple mathematical tricks and were easy to crack come under this category. One famous example is Caeser cipher. In this method, the message, mainly alphabetical strings were encrypted letter by letter. The trick was to shift the characters in the strings by a fixed number n (which was predecided by the communicating parties). For example, "WERTY" becomes "XFSUZ" for $n = 1$.
- **Modern Cryptography** : The cryptographic methods that are developed after 1900 and use advanced mathematical concepts like Number Theory. Modern Cryptography works on the concept of "Key". A key is a message that can be transmitted to receiver and is used alongside a code or a message. A key provides necessary information to convert a message into a code or to convert the code into a meaningful message. There are two types of keys, which are
  - Public Key : This key is used by transmitter to convert message into a code and is then sent to the receiver alongwith the code. The receiver uses the same key to decipher the code and get the message.
  - Private Key : This key is generated by each of the communicating parties and is never communicated with anyone. This key is bestowed with a single task of deciphering the received code and nothing else. This key is mostly used alongwith a Public key, in which case the Public key encrypts the message and Private key decrypts it.

  Modern Cryptography can be further classified into Symmetric and Asymmetric Cryptography.
  - Symmetric Cryptography uses only Public key for communication. Baptisa's Cryptographic method is one of the examples of Symmetric Cryptography, the method is explained in the coming section.

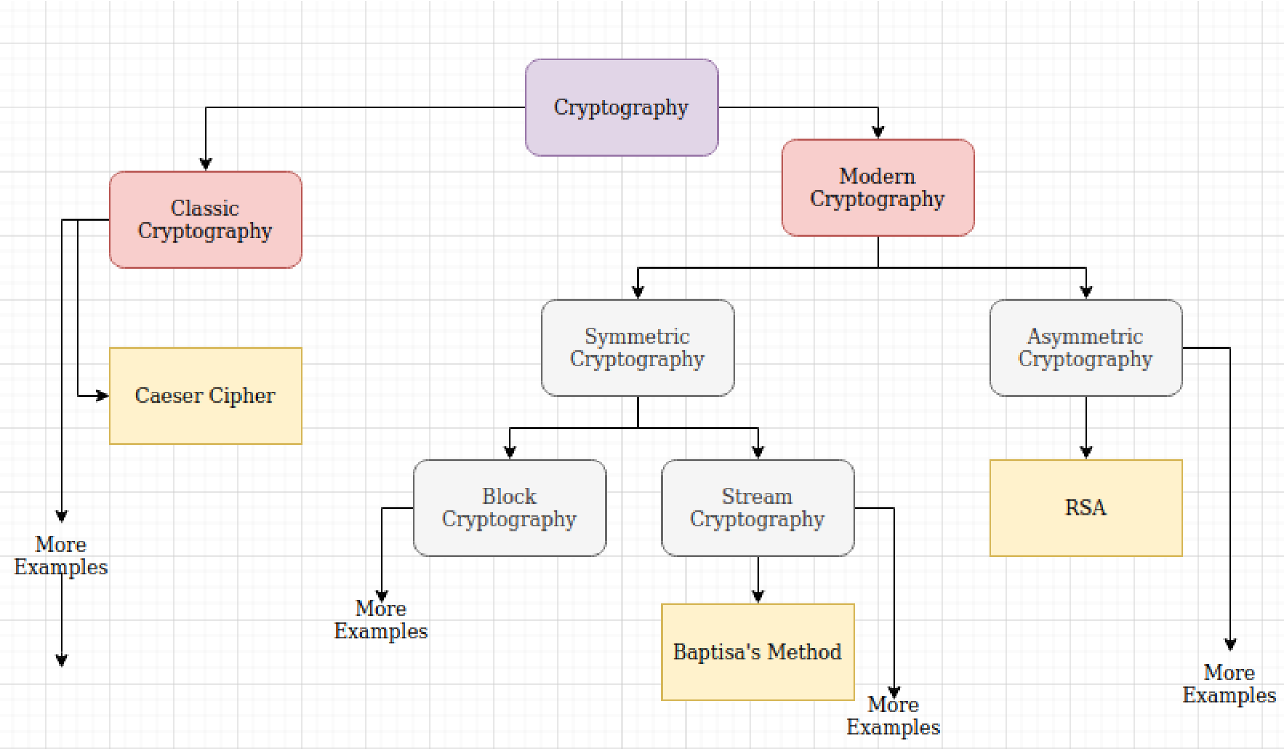

Fig. 3. Classification of Cryptographic methods

  - Asymmetric Cryptography uses both Public and Private keys secure communication. RSA is an example of Asymmetric Cryptography, it uses concepts from modular arithmetic like Euler-Totient function which are beyond the scope of this paper, but more details can be found in [9].

III. Main Result

A. Cryptographic Technique

1) Without Shadowing: Baptisa's method is a symmetric stream cryptographic method. In a stream cryptographic method, encryption and decryption are done character-by-character, in contrast to block cryptographic method wherein encryption and decryption is done block of characters of message and code respectively at a time.

The method was implemented in **Python 3** and has been made publicly available on **Github** [10]. The Baptisa's method is explained in following steps:

- **Step 1:** A value of b for which the Logistic map do not equilibriate is chosen. If the $b = 4$, then the Logistic map is ergodic for entire interval of $[0, 1]$. In which case $[0, 1]$ would be used for further. If $b = 3.78$ then the interval that needs to be used is $[0.2, 0.8]$, that is the interval which for which the Logistic map equilibriates to all the values of the interval.
- **Step 2:** Now, the string of characters to be used for creating the message is decided. The number of characters in the string is taken in powers of 2, 5 or both. This step will get clear as the method proceeds. Suppose the number of characters in the string is S, then the interval $[x_{min}, x_{max}]$ (interval chosen in **Step 1**) is divided into S equal smaller intervals. The size of each interval is given by,

$$\epsilon = (x_{max} - x_{min})/S \tag{4}$$

- **Step 3:** Each of the smaller interval is assigned a unique character from the string chosen in **Step 2** in the order they occur in the string. This can be seen in the fig. 4. Here, ϵ needs to have finite decimal places, or else the more number (more specifically that will be equal to the limit of machine's finite number representation) of decimal places will need to be stored and used in further computations to distinguish between the smaller intervals. And therefore the number of characters in the string is preferred to be powers of 2, 5 or both in **Step 2**.
- **Step 4:** A value of x_o, the initial condition for the Logistic Map is chosen. x_o can be anything between 0 and 1. Now, the Logistic Map can be formed,

$$x_{n+1} = b \times x_n \times (1 - x_n) \tag{5}$$

- **Step 5: Encryption:** Suppose, a message ="Hi" needs to encrypted. The first character "H" is encrypted to start with. The eq. 5 is iterated until the state (x_n) lands in the interval assigned to the character "H". The number of iterations (assume it be 174) required is noted and it represents "H" in the code. Next, eq. 5 is again iterated with initial condition as x_n until the state lands in the interval assigned to character "i". The number of

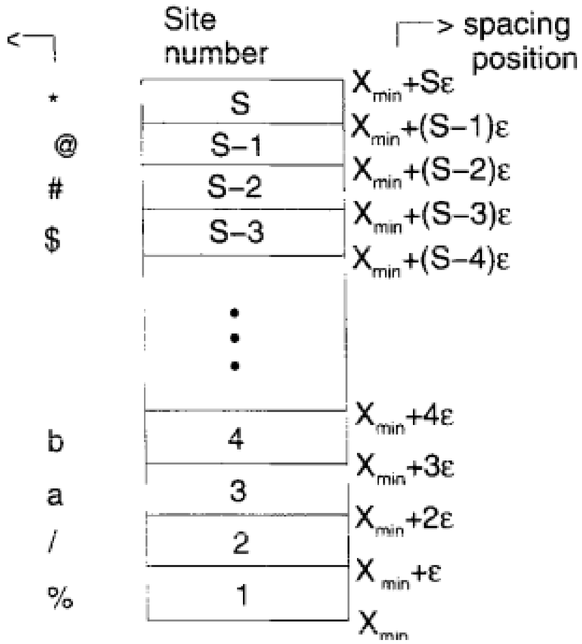

Fig. 4. Division of Interval and assignment of characters

iterations (assume it be 235) required after encrypting ”H” represents ”i” in the code. So, the code becomes,

$$174 \quad 235$$

- **Step 6: Communication:** The transmitter needs to send following with receiver to for successful communication,
  Code = 174 235
  Public key 1 = b
  Public key 2 = x_o
  Public key 3 = x_{min}
  Public key 4 = x_{max}
  In this method it is assumed that the communicating parties have already decided upon the usage of this method, order of Public Keys and the string of characters to be used for message formation.
- **Step 7: Decryption:** The receiver forms the Logistic map (eq. 5) and designates the smaller intervals obtained after the division of $[x_{min}, x_{max}]$ to the characters of the string, based upon the information from Public keys. Then, eq. 5 is iterated for 174 times, and the location of the state in the smaller intervals is identified to get ”H”, which means the state of the Logistic map after iterating 174 lies in the interval assigned to ”H”. The current state is used as initial condition and is iterated 235 time to get ”i” in the similar fashion.

This method has one major advantage, if x_o is changed even slightly, the code changes drastically, this is because of the fact that Logistic Map is chaotic.

2) With Shadowing: [4] suggests a tweak to make the Baptisa's cryptographic method stronger. The new method uses following equation for iteration instead of eq.5 for iterating.

$$x_{n+1} = \begin{cases} b_1 \times x_n \times (1 - x_n), & if\ n \leq j_1 \\ b_2 \times x_n \times (1 - x_n), & if\ j_1 \leq n \leq j_2 \\ b_3 \times x_n \times (1 - x_n), & if\ j_2 \leq n \leq j_3 \\ \ldots \\ b_m \times x_n \times (1 - x_n), & if\ j_{m-1} \leq n \leq N \end{cases} \tag{6}$$

For user-defined, N, b_i and j_i, i from 1 to $m-1$.
The algorithm with shadowing also has been implemented in [10].

B. Comparison between the method with shadowing and without shadowing

- The number of public keys in case of Baptisa's method with shadowing $(2m + 3)$ is larger than that without shadowing (4) and hence making it difficult to crack the code for any unwanted eavesdropper (referred to as hacker from now on).
- Both methods were analysed for the time required for the encryption. The time required for the encryption as a function of number of characters in the message is plotted in fig. 5. The required for the encryption increases as the number of characters increases, but times for methods remains in the same order and hence the lesser complexity of method without shadowing is not a clear advantage over method with shadowing.

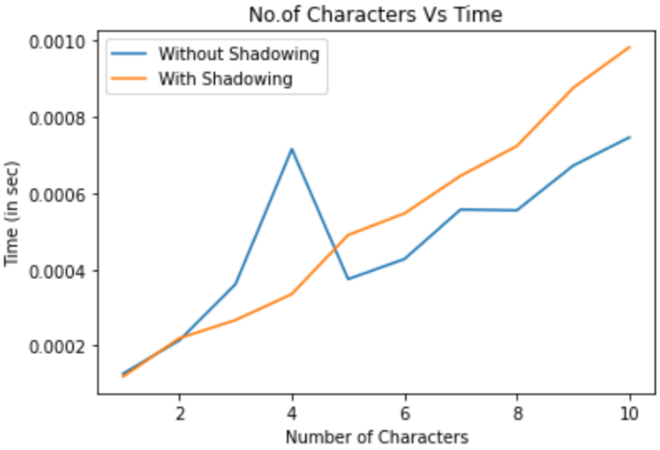

Fig. 5. Time Analysis

- The trajectory of the Logistic Map can be shadowed by other trajectory and hence the code can be cracked even if the hacker does not posses the exact values of keys. The concept of shadowing comes from the fact that, even if the hacker is not able to accurately determine b, and if,

$$\mid b - b_{guess} \mid \leq \delta \tag{7}$$

  For some small positive δ, which will depend upon ϵ. Given above information the hacker can mimic the true trajectory (given he knows other keys accurately) and thereby crack the code [4]. Which makes the method without shadowing vulnerable.

But, this is not the case with the method with shadowing, since the hacker will be required to know $2m$ parameters to generate eq. 7. Hence, the method augmented with shadowing is more powerful than method without shadowing.

C. Comparison Logistic based cryptographic technique and other popular methods

- Logistic based method or Baptisa's method is symmetric cryptographic method, while the most popularly used methods are asymmetric. The asymmetric methods are popular since they use Private key, which is needed to decrypt a message. And it is mathematically and computationally very difficult to guess and hence generate the Private key using Public keys (Only information which a hacker has other than code). This is due a mathematical concept of **one-way functions** which is used to generate Private key, In which we can generate $f(a)$ for given a, but a can't be easily generated easily from $f(a)$.
- Code generated from Baptisa's method (with or without shadowing) is difficult for computers to crack, but with technological advance in computing, one day these methods will be cracked in a matter of minutes. This is not a matter of concern to methods like **RSA**, wherein the computer will need to work out some mathematical algorithm (to be developed by a human) to crack the code.

IV. Conclusion and Future Work

The Baptisa's cryptographic method was studied for shadowing and without shadowing cases. The Baptisa's method with and without shadowing was implemented on a computer to know about it's practical applicability. Further, It was found that the method with shadowing is much better than that without shadowing. It was also found that the method has weaknesses that can be attributed to the way it works. But, there are various ways it can be made powerful in terms of vulnerability to attacks. It can be used with other methods like AES, AES is a symmetric cryptographic methods which needs to generate public keys, those public keys can be generated using Baptisa's method, thereby making the AES (in this modified) more powerful [11]. Since the aim is to establish secure communication, **RSA** (or any other more asymmetric cryptograhic method) can be used to communicate the Public keys of Baptisa's method and then communicate using Baptisa's method. So, combinations of Baptisa's method with other methods can be looked at. Also, efforts can be put in the direction of making Baptisa's method an asymmetric method, by developing the required math for generating Private key.

References

[1] Dooley, John F. History of cryptography and cryptanalysis: Codes, Ciphers, and their algorithms. Springer, 2018.

[2] Boeing, Geoff. (2016). Visual Analysis of Nonlinear Dynamical Systems: Chaos, Fractals, Self-Similarity and the Limits of Prediction. Systems. 4. 37. 10.3390/systems4040037.

[3] Baptista, M. S. ”Cryptography with chaos.” Physics letters A 240, no. 1-2 (1998): 50-54.

[4] Smaoui, Nejib, and Ali Kanso. ”Cryptography with chaos and shadowing.” Chaos, Solitons and Fractals 42, no. 4 (2009): 2312-2321.

[5] Tsuchiya, Takashi, and Daisuke Yamagishi. ”The complete bifurcation diagram for the logistic map.” Zeitschrift für Naturforschung A 52, no. 6-7 (1997): 513-516.

[6] Weisstein, Eric W. ”Logistic Map.” From MathWorld–A Wolfram Web Resource. https://mathworld.wolfram.com/LogisticMap.html

[7] https://www.math.uvic.ca/faculty/aquas/ds/ergodic.html

[8] timeandensembleaverages.py @ https://github.com/wakodeashay/Logistic-map-based-encryption-and-decryption.git

[9] Menezes, Alfred J., Paul C. Van Oorschot, and Scott A. Vanstone. Handbook of applied cryptography. CRC press, 2018.

[10] https://github.com/wakodeashay/Logistic-map-based-encryption-and-decryption.git

[11] Bose, Ranjan, and Amitabha Banerjee. ”Implementing symmetric cryptography using chaos functions.” In Proceedings of the 7th International Conference on Advanced Computing and Communications, pp. 318-321. 1999.