“Improving the Security of Cryptographic Keys Using Modified Chaotic Models”

Author: Baikalov Daniil Evgenievich
Russia, Moscow

felix.trof@gmail.com

Abstract.
This paper explores methods for enhancing the robustness of cryptographic keys through the use of modified chaotic models. The logistic map and the tent map, which serve as key objects of study, are widely used in cryptography for generating pseudorandom sequences. The author proposes several modifications to these models, including the introduction of additional parameters, unique multipliers, and adaptive computation algorithms. The conducted research shows that such modifications increase the uniqueness index of sequences, making them more resistant to cryptanalysis and less predictable. The advantage of the proposed approach lies in its computational efficiency and suitability for use in modern information security systems. Experimental results confirm the effectiveness of the proposed methods, demonstrating their adaptability to various cryptographic tasks. Additionally, the paper provides implementation examples of the modified models in the Go programming language, including the generation of stream ciphers and improved pseudorandom sequences. This work opens up prospects for the development of new cryptographic primitives based on chaos theory and suggests specific directions for further improvement of data protection methods. Thus, the article makes a significant contribution to the advancement of cryptographic science and information security technologies, showcasing the effectiveness of chaotic models in strengthening key resistance.

Keywords: cryptographic keys, chaos theory, logistic map, tent map, random number generation, sequence uniqueness, chaotic models, cryptanalysis, stream ciphers, modified algorithms, data security, cryptographic primitives, information security, key robustness, pseudorandom sequences.

1. Introduction

Chaos theory is a branch of mathematics that studies dynamical systems characterized by sensitivity to initial conditions, nonlinearity, and unpredictability. A fundamental property of chaotic systems is that even minor changes in initial conditions can lead to drastically different system trajectories, making their long-term behavior highly unpredictable. Chaotic systems can be described using various mathematical models, such as differential equations and iterative maps.

Cryptography, as the science of information security, requires methods that ensure a high degree of randomness and unpredictability [1, p. 80]. In this context, chaos theory offers powerful tools for developing cryptographic algorithms, as chaotic systems possess properties ideally suited for generating cryptographic keys and stream encryption.

The use of chaos in cryptography began with the realization that chaotic systems can produce sequences with strong randomness characteristics. This potentially makes them more resistant to cryptanalysis compared to traditional methods. In particular, chaotic systems can be utilized to create cryptographic primitives such as random number generators and stream ciphers, which provide a high level of security for data protection.

2. Overview of Chaotic Models

2.1 Logistic Map

The logistic map is one of the simple deterministic chaotic models, described by the following recursive relation:

$$x_{n+1} = r * x_n * (1 - x_n)$$

where x_n is the current value of the state variable (usually within the interval [0, 1]), and r is the map parameter (often referred to as the growth parameter).

Logistic mapping is used in cryptography to generate pseudo-random numbers and stream encryption. To do this, an initial value is initializedx_0 (seed) and parameter r. Then the values are calculated sequentially.x_n, which can then be converted into bit sequences or other forms of data used in cryptographic primitives [2, p. 12].

The advantage of this model is that it does not require large computing resources for its implementation, and also has a high computing speed, due to the simplicity and iterative nature of its formula.

However, it is worth noting that depending on the parameter$r$and the initial valuex_0Sequences generated by the logistic map may be vulnerable to statistical attacks and cryptanalysis due to their predictability and repeatability.

It was decided to check the uniqueness of the generated values using logistic mapping. The values of the parameter were considered$r$from 0 to 4 in 0.1 increments with a static start parameter$x_0 = 0.6$. 2000 values were generated for each data set. The uniqueness index was derived using the following formula

$$UI = \frac{unique\ values}{all\ meanings}$$

A uniqueness graph was compiled.

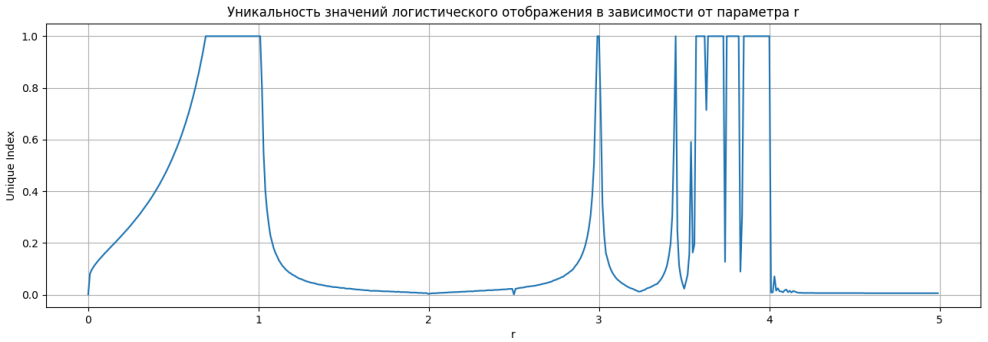

Fig. 1, Graph of the uniqueness of the logistic mapping values depending on the parameter r

Here the uniqueness index is equal to 1 when absolutely all obtained values out of 2000 are unique.

The observed dynamics also apply to shorter sequences.

Moreover, such uniqueness of the sequence values is inherent for any starting value x_0 for each value under study r You can verify this by studying the graph below.

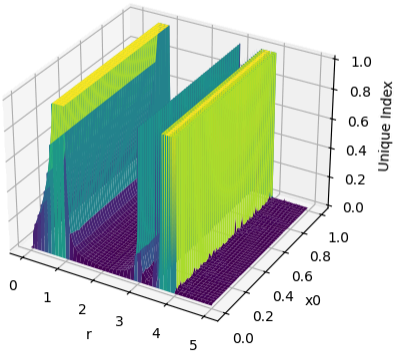

Fig. 2, Graph of the uniqueness of the logistic mapping values depending on the value of the starting value x and r

The average uniqueness index of logistic mapping values in such ranges is ~0.242. Recall that we are examining the first 2000 elements of the sequence.

To increase the uniqueness index of values and be able to use a larger range of parameter values r, it is proposed to modify the logistic mapping formula as follows. First, it is necessary to calculate the intermediate value q sequence element.

$$q = r * x_n$$

And then get the final value of the element

$$x_{n+1} = q * (1 - x_n),\ with\ q > 0.5$$

$$x_{n+1} = q * (1 + x_n),\ at\ q \leq 0.5$$

This approach will increase the average uniqueness index of the logistic mapping sequence values to ~0.389. A graph of the uniqueness of the elements of the sequence generated with the new approach was compiled.

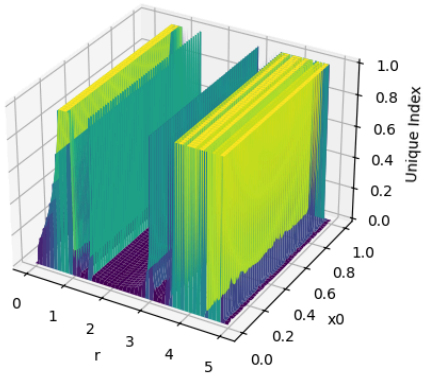

Fig. 3, Graph of uniqueness of logistic mapping values with modified multiplier

Below is a final version of a modification of the standard logistic mapping formula, aimed at ensuring absolute uniqueness of the sequence values in order to avoid predictability and repetition.

$$x_{n+1} = r * x_n * (1 - x_n) * d$$

$$d = \begin{cases} q + \left(\frac{n+1}{t}\right) \cdot (p - q) & (n+1) \mod 2 = 0 \\ p - \left(q + \left(\frac{n+1}{t}\right) \cdot (p - q)\right) + q & (n+1) \mod 2 = 1 \end{cases} \quad (1)$$

Where q- lower limit of the multiplier,p- the upper limit of the multiplier,t- the number of elements in the sequence.

This modification radically changes the uniqueness index. In the same ranges of values$r$ And x, for example,$q = 0.4$, A $p = 0.6$, the uniqueness graph looks like this.

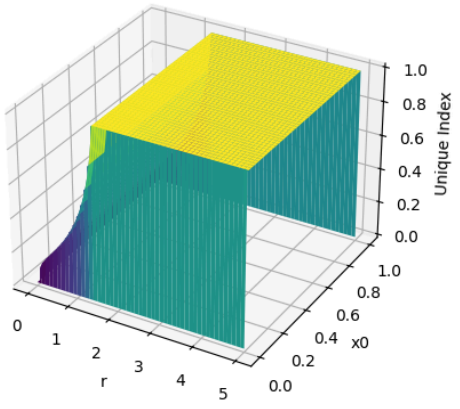

Fig. 4, Plot of uniqueness of logistic mapping values using an additional factor d, where r ranges from 0 to 5

The average uniqueness index is ~0.805. If you shift the range of the value$r$ from [0, 5] to [2, 7], then the uniqueness index will be equal to ~0.976.

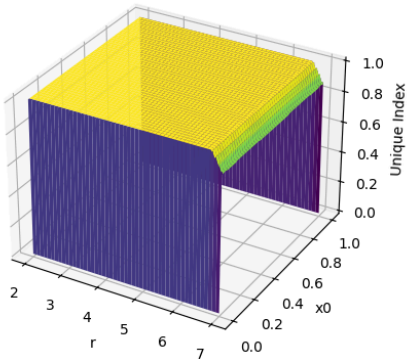

Fig. 5, Plot of the uniqueness of the logistic mapping values using an additional factor d, where r ranges from 2 to 7

In addition, this approach will provide even greater protection when generating a key using a logistic mapping, since to generate the correct byte sequence, it will be necessary to know not only the value of the parameters$r$ And x_0, but also the meaning of the parameters$q$ And p.

2.2 Tent Map

The Tent map is a simple one-dimensional deterministic chaotic system described by the following recurrence relation:

$$x_{n+1} = \begin{cases} \mu x_n, & x_n < 0.5, \\ \mu(1 - x_n), & x_n \geq 0.5 \end{cases}$$

Where x_n — the current value of the state variable (usually on the interval [0, 1]), andμ— a parameter that regulates the chaos of the system (atμ = 2The map of the Shatra is completely chaotic) [3, p. 24].

The Shatra map is used in cryptography to generate pseudo-random numbers and stream encryption. The initial value is initializedx_0, then the values are calculated sequentiallyx_n, which can be converted into bit sequences or other forms of data used in cryptographic primitives.

The tent map is simple to implement, just like the logistic mapping, does not require much computing power, but has the same disadvantages: predictability and repeatability.

It was decided to examine the tent map for the uniqueness of the sequence elements using the same principles as the logistic mapping:$\mu \in [0;\ 5]$, $x_0 \in [0,\ 1]$ and the number of elements equal to 2000. The following results of the uniqueness of the elements were obtained.

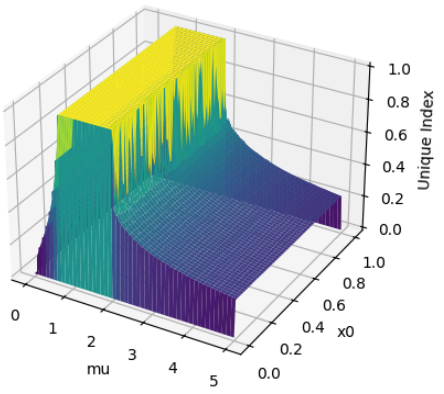

Fig. 6, Graph of the uniqueness of the tent map values with a sequence length of 2,000 elements

The average uniqueness index here is ~0.493. It can be seen that exceptionally unique elements appear in the sequence only when$\mu \in [0.5;\ 2]$.

However, if we reduce the number of elements in the sequence to 500, then the uniqueness index of the elements is already equal to ~0.954.

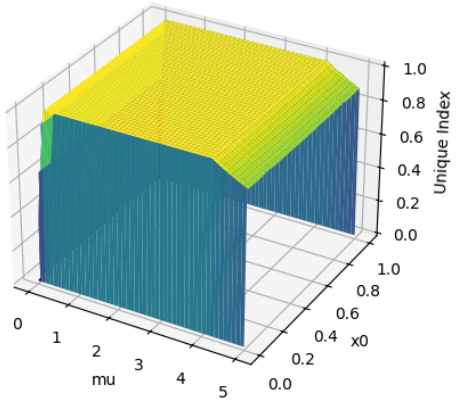

Fig. 7, Graph of the uniqueness of the tent map values with a sequence length of 500 elements

If we apply in this case the same approach that was proposed when generating logistic mapping values, i.e. introduce a unique multiplier for each element of the sequence based on 2 additional initial values that can be used as another measure of key security, then even on large ranges of values (2000 and more), the uniqueness index of elements rises to 0.75.

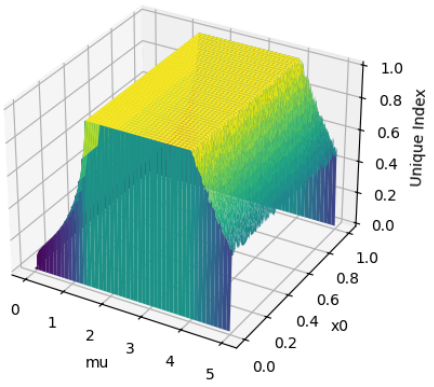

Fig. 8, Uniqueness of tent map values for a sequence length of 2,000 elements using an additional multiplier d

How the multiplier is calculated:

$$d = \begin{cases} q + \left(\frac{n+1}{t}\right) \cdot (p - q) & (n+1) \mod 2 = 0 \\ p - \left(q + \left(\frac{n+1}{t}\right) \cdot (p - q)\right) + q & (n+1) \mod 2 = 1 \end{cases} \quad (1)$$

Where q- lower limit of the multiplier,p- the upper limit of the multiplier,t- the number of elements in the sequence.

3. Implementation of the use of modified models

To demonstrate how the modified models work, it was decided to create a simple stream cipher that generates a byte stream of keys and then applies the XOR operation to each character in turn. Below is the Go code.

Function to generate logistic mapping:

```
func BLogisticMap(n int, r, x0, p, q float64) []float64 {
    m := []float64{x0}

    for i := 1; i < n; i++ {
        d := q + (float64(i)+1)/float64(n)*(p-q)

        if (i+1)%2 == 1 {
```

```
                d = p - d + q
            }

            m = append(m, r*m[i-1]*(1-m[i-1])*d)
        }

        return m
}
```

Function to convert a logistic map into a byte stream of keys.

```
func MapToByteStream(m []float64) []byte {
        byteStream := []byte{}

        for _, val := range m {
            byteStream = append(byteStream, byte(val*256))
        }

        return byteStream
}
```

Encryption and decryption function:

```
func StreamCipher(data []byte, byteStream []byte) []byte {
        encrypted := make([]byte, only(data))
        for in, select := range data {
            encrypted[ind] = val ^ byteStream[ind]
        }

        return encrypted
}
```

Input function:

```
func main() {
	toEncrypt := []byte("This is a new way to create logistic map.")
	bm := BLogisticMap(only(toEncrypt), 3.5, 0.56, 0.2, 0.8)

	res := StreamCipher(toEncrypt, MapToByteStream(bm))

	hexRes := hex.EncodeToString(res)

	fmt.Println(hexRes)
}
```

The result of the program will be a HEX representation of the encrypted byte stream:

dbc259151a19315528583e1d20572a141a521d014f0a060619291e77113f19230d37155e585b125f42

4. Analysis of existing models and benefits of proposed changes

Standard chaotic models such as logistic mapping and tent maps exhibit a high degree of randomness under certain parameters. However, with long-term use, repeating patterns may be observed, which reduces their resistance to cryptanalysis. This is because the predictability and periodicity of values make such systems vulnerable to attack.

To improve cryptographic strength, several modifications have been proposed aimed at increasing the uniqueness and unpredictability of the generated sequences.

1. Introducing intermediate values and conditions: Some modifications add intermediate values and conditions that change the system's behavior

depending on the current state. This allows for a significant increase in the complexity of the sequence, reducing the likelihood of repeating patterns.

2. Using additional parameters**:**Additional parameters such as multipliers and coefficients are introduced, which change dynamically during the sequence generation process. This complicates the process of restoring the correct sequence without knowing all the parameters and initial conditions.
3. Combination of different conditions**:**Some proposed modifications use different conditions and branches that depend on the current state of the system. This makes the sequences even more unpredictable and unique.

The benefits of the proposed changes are:

1. Increased Uniqueness: The uniqueness index of sequence values increases significantly, making them more resistant to analysis and prediction.
2. Increased cryptographic strength: Increasing the complexity and unpredictability of sequences reduces the likelihood of successful attacks on the system. This is achieved by adding new parameters and conditions that complicate the process of key recovery.
3. Flexibility and adaptability: The proposed modifications allow flexible configuration of system parameters depending on specific security requirements. This makes them universal for various cryptographic applications.

Uniqueness index, using different data samples as an example.

Length of sequence	**Without modification**		**With modification**	
	Logistic mapping	**Tent map**	**Logistic mapping**	**Tent map**
100	0.47	1.0	1.0	1.0
1 000	0.047	0.56	1.0	1.0

10 000	0.0047	0.056	1.0	0.172
100 000	0.00047	0.0056	1.0	0.02
1 000 000	0.000047	0.00056	1.0	0.0021

Table 1, Comparison of uniqueness of sequence values

Conclusion and further work

In conclusion, the presented study demonstrates that the use of chaotic models in cryptography is a promising direction for developing more secure cryptographic systems. The logistic mapping and the Shatr map, being simple to implement, have the potential to create cryptographic primitives resistant to attacks. The implementation of the proposed modifications allowed to significantly increase the uniqueness of the generated sequences, which is confirmed by experimental data. These results open up new opportunities for further improvement of cryptographic algorithms based on chaotic systems.

The study of the possibilities of chaotic models in cryptography opens up a number of promising areas for further research and improvement. Based on the results obtained, the following key areas for further work can be identified:

1. Optimization of model parameters
2. Analysis of resistance to different types of attacks
3. Application in various cryptographic primitives

References

1. Andreev Vsevolod Vladimirovich, Sapozhnikova Yulia Valentinovna, Fomichev Alexander Igorevich Deterministic chaos and information coding // Applied informatics. 2009. No. 6. URL: https://cyberleninka.ru/article/n/determinirovannyy-haos-i-kodirovanie-informatsii (date of access: 03.12.2024).

2. Kuznetsov Dmitry Yuryevich Continuous generalization of the logistic mapping // Bulletin of Moscow University. Series 3. Physics. Astronomy. 2010. No. 2. URL: https://cyberleninka.ru/article/n/kontinualnoe-obobschenie-logisticheskogo-otobrazheniya (date of access: 03.12.2024).
3. Shashikhin Vladimir Nikolaevich, Bogach Natalia Vladimirovna, Chuprov Viktor Aleksandrovich Analysis of cryptographic strength of encryption algorithms based on chaotic mapping of the tent type // Computer Science, Telecommunications and Management. 2012. No. 3 (150). URL: https://cyberleninka.ru/article/n/analiz-kriptostoykosti-algoritmov-shifrovaniya-na-osnove-haoticheskogo-otobrazheniya-tipa-tent (date of access: 03.12.2024).