
Journal of Informatics and Web Engineering

Vol. 3 No. 3 (October 2024)

eISSN: 2821-370X

Towards Analysable Chaos-based Cryptosystems: Constructing Difference Distribution Tables for Chaotic Maps

Je Sen Teh^{1,2*}, Abubakar Abba³

¹School of IT, Deakin University, 3216 Waurin Ponds, Victoria, Australia

²Deakin Cyber Research and Innovation Centre, Deakin University, 3220 Geelong, Victoria, Australia

³School of Computer Sciences, Universiti Sains Malaysia, 11800 Gelugor, Pulau Pinang, Malaysia

*corresponding author: (j.teh@deakin.edu.au; ORCID: 0000-0001-5571-4148)

Abstract – Chaos-based cryptography has yet to achieve practical, real-world applications despite extensive research. A major challenge is the difficulty in analysing the security of these cryptosystems, which often appear ad hoc in design. Unlike conventional cryptography, evaluating the security margins of chaos-based encryption against attacks such as differential cryptanalysis is complex. This paper introduces a straightforward approach of using chaotic maps in cryptographic algorithms in a way that facilitates cryptanalysis. We demonstrate how a chaos-based substitution function can be constructed using fixed-point representation, enabling the application of conventional cryptanalysis tools such as the difference distribution table. As a proof-of-concept, we apply our method to the logistic map, showing that differential properties vary based on the initial state and number of iterations. Our findings demonstrate the feasibility of designing analysable chaos-based cryptographic components with well-understood security margins.

Keywords—Chaos Theory, Cryptography, Cryptanalysis, Differential Cryptanalysis, Cryptosystem

Received: 12 July 2024; Accepted: 03 September 2024; Published: 16 October 2024

This is an open access article under the [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/) license.



1. INTRODUCTION

Chaos theory is a mathematical field that studies nonlinear dynamical systems that have the *butterfly effect*. In other words, even the slightest variation in their inputs can lead to a big difference in their outputs over time. We can observe chaos in real-world phenomena such as climate, population and traffic. Chaotic maps are iterative functions that have the aforementioned chaotic behaviour. The logistic map is an example of how complex behaviours can

arise from simple dynamical equations and has been used as part of a discrete-time demographic model [1]. The logistic map is defined as in Equation (1).

$$x_{n+1} = rx_n(1 - x_n), \quad (1)$$

where $x_n \in [0,1]$ and $r \in [0,4]$ is a control parameter. As the value of r increases from 0 to 4, the behaviour of the logistic map evolves from a fixed value, to periodic and finally chaotic.

Chaos-based cryptography is a popular research field that investigates the use of chaotic maps in cryptography. Researchers have sought to apply chaotic maps in cryptography due to their properties such as pseudo-randomness, complexity and sensitivity to changes in their inputs. Although these properties are indeed analogous to some of the requirements of cryptographic algorithms, most chaos-based algorithms are designed in an ad hoc manner and stray from widely studied design paradigms [2]. Since this complicates the analysis of their security margins, the designers of chaos-based cryptosystems rely mainly on statistical testing to *prove* their algorithms are secure.

When it comes to symmetric-key cryptosystems, algorithms such as the Advanced Encryption Standard (AES) [3] or ISO-standard PRESENT [4] have been designed based on well-studied design paradigms such as the substitution-permutation network (SPN). These designs facilitate third-party cryptanalysis efforts which builds confidence in the security of these ciphers over time. These designs facilitate third-party cryptanalysis efforts which builds confidence in the security of these ciphers over time, which can then be used to encrypt data-at-rest (e.g. in cloud storage [5]) and data-in-transit. We can determine the exact security margins of these ciphers against state-of-the-art cryptanalysis techniques such as differential cryptanalysis [6] or boomerang attacks [7]. The simplicity of their designs allows attacks to be modelled as a system of constraints that are then solved by constraint solvers to provide strict security bounds. Boolean satisfiability (SAT) and satisfiability modulo theory (SMT) solvers are commonly used in cryptanalysis efforts [8], [9], [10].

Since chaos-based algorithms generally do not conform to any cryptographic design paradigms, applying conventional cryptanalysis tools to evaluate their security is a difficult task. Most third-party cryptanalysis results are dedicated methods that exploit specific design flaws [11], [12], [13]. Instead, designers of chaos-based cryptosystems resort to statistical tests such as the number of pixels change rate (NPCR) and unified average change intensity (UACI) to provide some form of evidence that their designs are secure against differential cryptanalysis [14], [15], [16], [17], [18], [19]. However, it is well-established that passing statistical tests is not an indicator of security [20].

In this paper, we aim to narrow the gap between conventional and chaos-based cryptography. We introduce a simple approach to adopting chaotic maps as a cryptographic building block that can be analysed using classical cryptanalysis techniques such as differential cryptanalysis. Specifically, we can build difference distribution tables (DDTs) for them, much like substitution boxes (S-boxes). The use of fixed-point representation [21] allows the computation of chaotic maps using interpretable binary strings, eliminating the complexity of floating-point representation. We provide examples of DDTs for 4-bit and 8-bit variants of the chaos-based substitution functions based on the logistic map. Note that these substitution functions are not meant to be used as is in the design of actual cryptosystems and are mainly as a proof-of-concept.

The rest of this paper is structured as follows – Section 2 first provides background information including previous methods to analyse the security of chaos-based cryptosystems against differential cryptanalysis. Section 3 then introduces the proposed chaos-based substitution function based on fixed-point notation and how we can construct DDTs for its different variants. In Section 4, we analyse the resulting DDTs and also show that we can also trivially derive boomerang connectivity tables. The paper is concluded in Section 5.

2. PRELIMINARIES

2.1 Differential Cryptanalysis

Differential cryptanalysis was originally introduced by Biham and Shamir to cryptanalyse the Data Encryption Standard (DES) [6]. Since then, it has been applied not only to block ciphers but other cryptographic primitives such as hash functions [22] and stream ciphers [23], spawning variants such as differential-linear [24], boomerang [7] and inspiring the development of neural cryptanalysis [25].

When analysing a block cipher, the goal of a cryptanalyst is to find a pair of plaintexts (P_1, P_2) with a difference $\alpha = P_1 \oplus P_2$ that leads to a pair of ciphertexts (C_1, C_2) with a difference $\beta = C_1 \oplus C_2$. Here, we define a difference as an XOR difference but there are also other variants such as rotational differences [26]. The probabilistic propagation of an input difference to an output difference, $\alpha \rightarrow \beta$ through the cipher is known as a differential trail or differential characteristic.

If a differential trail holds a sufficiently high probability, it can be used as a statistical distinguisher in key recovery attacks. Estimating a trail's differential probability, P requires analysing the nonlinear components of a block cipher such as S-boxes. If an S-box receives a nonzero difference as its input, it is considered *active* and imposes a penalty on the overall differential probability because there are many possible output differences that occur with varying probabilities for a given input difference. Differential propagation through an S-box can be described using a DDT that captures the distribution of all possible input-to-output differences. Each row corresponds to an input difference while each column corresponds to an output difference. The table entries count the number of possible pairs of inputs with a given input difference that would lead to a particular output difference. The DDT of the block cipher PRESENT [4] is shown in Table 1.

Table 1. Differential Distribution Table Of PRESENT

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	4	0	0	0	4	0	4	0	0	0	4	0	0
2	0	0	0	2	0	4	2	0	0	0	2	0	2	2	2	0
3	0	2	0	2	2	0	4	2	0	0	2	2	0	0	0	0
4	0	0	0	0	0	4	2	2	0	2	2	0	2	0	2	0
5	0	2	0	0	2	0	0	0	0	2	2	2	4	2	0	0
6	0	0	2	0	0	0	2	0	2	0	0	4	2	0	0	4
7	0	4	2	0	0	0	2	0	2	0	0	0	2	0	0	4
8	0	0	0	2	0	0	0	2	0	2	0	4	0	2	0	4
9	0	0	2	0	4	0	2	0	2	0	0	0	2	0	4	0
A	0	0	2	2	0	4	0	0	2	0	2	0	0	2	2	0
B	0	2	0	0	2	0	0	0	4	2	2	2	0	2	0	0
C	0	0	2	0	0	4	0	2	2	2	2	0	0	0	2	0
D	0	2	4	2	2	0	0	2	0	0	2	2	0	0	0	0
E	0	0	2	2	0	0	2	2	2	2	0	0	2	2	0	0
F	0	4	0	0	4	0	0	0	0	0	0	0	0	0	4	4

2.2 Evaluating The Security of Chaos-based Ciphers Against Differential Cryptanalysis

As of July 2024, out of the 300 or so chaos-based cryptography papers indexed in Scopus, over 250 papers are related to image or video encryption. The go-to justification for designing image or video-specific encryption algorithms is that conventional encryption algorithms are not suitable for multimedia data [27] and are mainly for textual data [28]. These claims are unfounded since conventional algorithms such as AES are the ones being used daily to encrypt all types of data. One example would be the instant messaging service, WhatsApp which relies on AES-256 for its end-to-end encryption [29]. In his paper, Zhang also refuted these claims by showing that AES can efficiently encrypt images [30].

Since the majority of chaos-based encryption algorithms were designed for images or videos, analyses of their security against differential cryptanalysis were performed based on pixel statistics rather than using conventional cryptanalysis approaches. NPCR and UACI are the two main metrics used. Multiple cipherimage pairs (C_1, C_2) are first obtained by encrypting plainimage pairs with a random 1-bit difference. Let $D(i, j)$ be a function that returns 1 if the pixels at position (i, j) in both cipherimages are different, or 0 otherwise. NPCR and UACI values are then given by Equation (2) and Equation (3) respectively.

$$NPCR(C_1, C_2) = \frac{\sum_{i,j} D(i, j)}{MN} \times 100\%, \quad (2)$$

$$UACI(C_1, C_2) = \frac{1}{MN} \sum_{i,j}^{M,N} \frac{|C_1(i,j) - C_2(i,j)|}{L} \times 100\%, \quad (3)$$

respectively, where M and N represent the number of rows and columns in an image, $L = 255$ is the maximum gray level value for an 8-bit pixel and $C(i, j)$ is a pixel at position (i, j) . The ideal values for NPCR and UACI are 99.6094% and 33.4635% respectively [31]. For example, the encryption scheme proposed by Jun and Fun achieved values of 99.6166% and 32.74455% [18]. Attaining close to these ideal values supposedly implies security against differential attacks. There also exist ciphers that rely on S-boxes generated using chaotic maps or chaos-based S-boxes. In some cases, some designers do analyse the DDTs of the chaos-based S-boxes [32], [33], [34] while others do not [18], [27], [35].

3. METHODOLOGY

3.1 Chaos-based Substitution based on Fixed-Point Representation

The proposed substitution function takes an a -bit input and produces a b -bit output, where a and b need not be the same size. The substitution function is not bijective and must be used in a construction that does not require a round function to be invertible, such as the generalized Feistel network [36]. As a proof of concept, we will show how the logistic map (Eq. 1) can be used to construct the proposed substitution function. The substitution function based on the logistic map may not possess optimal cryptographic properties but is presented for its simplicity and ease of understanding. The same methodology can be applied to any other chaotic map to develop other substitution functions with improved security.

We represent real numbers as 32-bit unsigned fixed-point numbers, with 2 of the most significant bits (MSBs) representing integers and the remaining 30 least significant bits (LSBs) representing fractions. We fix $r = 3.9999999990686774$ ($r = 0x\text{FFFFFFF}$ in fixed-point representation) to ensure that the logistic map has chaotic behaviour while at the same time minimising the number of bits required for integers (only 2 bits are required to represent "3"). We implement all the necessary fixed-point arithmetic operations required to iterate the logistic map shown in Equation 1. For details about how fixed-point operations work, please refer to [21]. All codes related to this paper, including the fixed-point implementation, are found at https://github.com/diffsearch/chaos_ddt.

Let the initial condition (or initial state) of the logistic map x_0 be a user-selected value between 0 and 1. Then, given an a -bit binary input y where $a \leq 30$, we compute $x_0 \oplus y$ before iterating the logistic map i times. We extract b LSBs from x_i as the output of the substitution function. The proposed substitution function is flexible since different values of the initial state x_0 , number of iterations i , input size a and output size b can be selected depending on the user's requirements. If used in a cryptographic algorithm, these parameters can also be initialised based on a security parameter like the secret key. The entire process is summarised in Figure 1.

Algorithm 1 Chaos-based Substitution

Inputs: Initial state x_0 , control parameter r , a -bit input y , number of iterations i , output size b

```

 $x \leftarrow x_0 \oplus y$  ▷ Perturb the initial state
while  $i \neq 0$  do ▷ Iterate the chaotic map  $i$  times
     $x \leftarrow rx(1 - x)$  ▷ Logistic map equation
     $i \leftarrow i - 1$ 
end while
Return:  $b$  least significant bits of  $x$ 

```

Figure 1. Algorithm 1 On Chaos-based Substitution

3.2 Constructing Difference Distribution Tables

Generally, block ciphers process their inputs as fixed-length words in 4-bit denominations. The word length depends on their underlying building blocks, e.g. if 4-bit S-boxes or 16-bit modular addition operations are involved, then the word sizes are usually 4 or 16 bits respectively. With that in mind, we will analyse the difference distribution for the proposed substitution functions with specific word lengths.

Let the proposed substitution function be denoted as $Sub_{a,b}(y, i)$, where y denotes an a -bit input, b denotes the output length in bits and i the number of chaotic map iterations. Generating the DDT for $Sub_{a,b}$ follows the same procedure as S-boxes – for all possible input a -bit input differences α , we enumerate all combinations of (y, y') where $y \oplus y' = \alpha$. We then compute the output difference β after substituting both y and y' individually and increment the corresponding DDT entry indexed by (α, β) . Figure 2 illustrates the entire process.

Algorithm 2 Generating Difference Distribution Tables

```

1: Inputs: Input size  $a$ , output size  $b$  number of iterations  $i$ 
2: _____
3: for  $\alpha \leftarrow 0$  to  $2^a$  do                                ▷ Loop through all input differences
4:   for  $y \leftarrow 0$  to  $2^a$  do                                ▷ Loop through all possible input values
5:      $y' \leftarrow \alpha \oplus y$                                 ▷ Calculate second value in the pair
6:      $\beta \leftarrow Sub_{a,b}(y, i) \oplus Sub_{a,b}(y', i)$         ▷ Calculate output difference
7:      $DDT[\alpha][\beta] \leftarrow DDT[\alpha][\beta] + 1$           ▷ Increment DDT entry
8:   end for
9: end for
10: Return: Difference distribution table  $DDT$ 

```

Figure 2. Algorithm 2 On Generating Difference Distribution Tables

4. ANALYSIS OF THE DIFFERENCE DISTRIBUTION TABLES

We examine DDTs for $a = b = 4$ and $a = b = 8$ which are arguably the most commonly used S-box sizes. The following experiments are conducted:

- Fix x_0 and determine the impact of the number of iterations i on the upper-bound (worst-case) differential probability P_w
- Fix i and determine the impact of the initial state x_0 on P_w

Differential probability is calculated as $P = \frac{c}{2^a}$, where c is the value of an entry (count) in the DDT and 2^a is the total number of possible input combinations for a -bit pairs. For example, if a table entry for a 4-bit substitution is 2, the differential probability is calculated as $P = \frac{2}{2^4} = 2^{-3}$. As a designer, the goal is to minimize c , which also minimizes P_w . S-boxes used in standardised block ciphers such as PRESENT (4-bit S-boxes) and AES (8-bit S-boxes) have maximum counts of $c = 2$ and $c = 4$ respectively.

4.1 DDT Analysis Based on the Number of Iterations

We first begin with substitution functions where $a = b = 4$. We randomly select $x_0 = 0.05129266157746315$ or $x_0 = 0x03486104$ in fixed-point representation. These are *nothing-up-my-sleeve* numbers randomly selected from the numbers of π . We first generate a DDT for $i = 1$ as an example, depicted in Table 2. The highest count in the DDT is $c = 6$, which corresponds to $P_w = \frac{6}{2^4} = 2^{-1.415}$. We then repeat the experiments $\forall i \in [1, 500]$. The majority of DDTs have $P_w = 2^{-1.415}$ ($c = 6$) while there were 49 instances with $P_w = 2^{-2}$ ($c = 4$). $P_w = 2^{-0.415}$ ($c = 12$) only occurs once when $i = 3$ and $2^{-0.678}$ ($c = 10$) thrice when $i \in \{5, 173, 427\}$. This implies that these worst-case scenarios (DDTs with high counts) occur less frequently when i increases. The results of this experiment are illustrated in Figure 3.

Table 2. 4-bit DDT Where $x_0 = 0.005129266157746315, i = 1$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	6	2	0	2	0	0	0	0	4	0	0	2
2	0	0	0	0	0	0	0	2	2	6	0	4	0	0	0	2
3	0	0	0	2	2	0	0	2	0	0	0	2	2	4	2	0
4	0	4	6	2	0	0	2	0	0	0	0	0	0	0	2	0
5	0	0	2	0	0	2	4	2	0	0	2	0	0	2	2	0
6	0	0	0	0	0	0	2	0	2	4	2	4	0	2	0	0
7	0	0	2	0	0	4	0	0	0	2	0	0	2	0	2	4
8	0	0	0	4	0	4	0	2	0	0	0	0	2	2	0	2
9	4	2	0	0	0	0	0	4	0	2	2	2	0	0	0	0
A	0	0	0	0	4	0	0	2	0	0	4	0	4	0	2	0
B	2	0	2	2	0	0	0	0	2	0	0	4	0	0	4	0
C	0	2	2	0	0	2	0	4	0	0	0	0	2	2	2	0
D	0	2	2	2	0	2	2	0	2	2	0	2	0	0	0	0
E	0	0	0	0	2	4	0	0	2	2	0	0	2	0	4	0
F	2	2	2	0	0	0	0	0	2	2	2	0	2	2	0	0

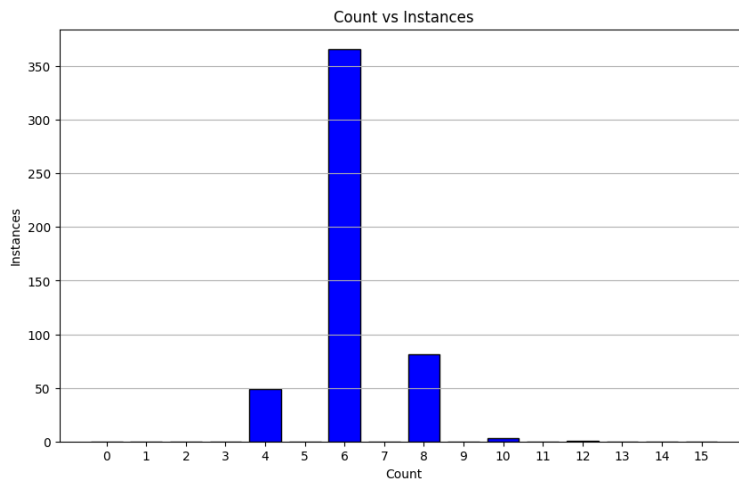


Figure 3. Comparing 4-bit DDTs $\forall i \in [1,500], x_0 = 0.05129266157746315$

We repeat experiments for $a = b = 8, \forall i \in [1,100]$, the results of which are illustrated in Figure 4. We limit the experiment to 100 iterations because constructing an 8-bit DDT takes significantly longer than its 4-bit counterpart. The majority of DDTs have $P_w = \frac{10}{2^8} = 2^{-4.678}$ and $P_w = \frac{12}{2^8} = 2^{-4.415}$. The distribution of DDTs with these upper-bound probabilities is illustrated in Figure 5. We note that at least 10 iterations are required before DDTs with lower counts are generated, implying that having a larger number of iterations positively impacts differential properties.

4.2 DDT Analysis Based on the Initial State x_0

Next, we fix the number of iterations to study the impact of different initial states x_0 on P_w . We set $i = 500$ (Figure 6) based on our observations in Section 4.1. We first examine DDTs for $a = b = 4$. We select 30 x_0 values with a hamming weight of 1 ($x_0 \in \{0x00000001, 0x00000002, 0x00000004, \dots, 0x4000000\}$) to determine how specific bits affect probability values. The results in Figure 6 show that if any of the four LSBs are active, the

resulting DDTs have the lowest $P_w = \frac{4}{2^4} = 2^{-2}$. This is unsurprising because only the $a = 4$ bits of the input are modified before iterating the chaotic map. Taking this into consideration, we enumerate all DDTs for $x_0 \in [0x00000001, 0x0000000F]$ and found that all of them also have $P_w = 2^{-2}$.

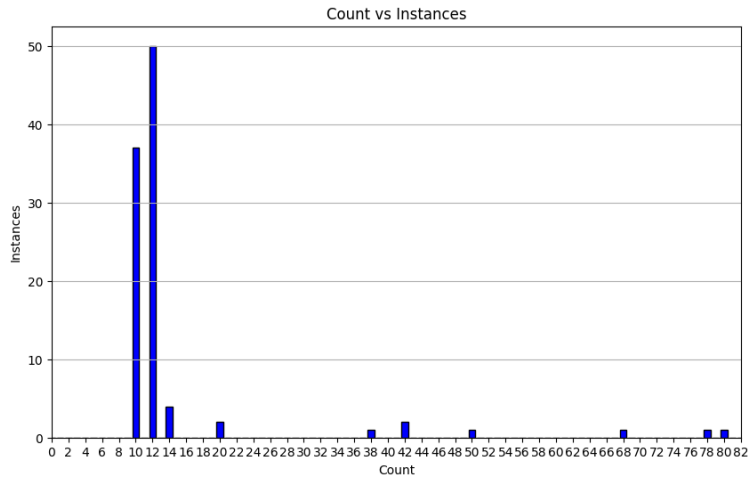


Figure 4. Comparing 8-bit DDTs $\forall i \in [1,100], x_0 = 0.05129266157746315$

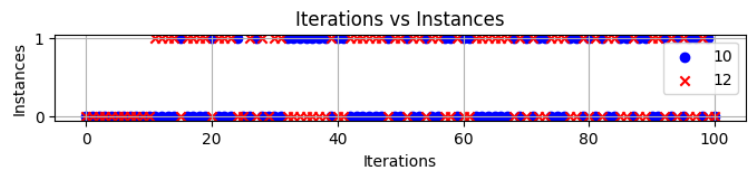


Figure 5. Comparing 8-bit DDTs $\forall i \in [1,100], x_0 = 0.05129266157746315$ Where $c \in \{10,12\}$

Next, we repeated the experiment for $a = b = 8$ and $i = 50$ (Figure 7). Results in Figure 7 show that if any of the eight LSBs are active, we get DDTs with the lowest counts of $c = 10$ or $P_w = 2^{-4.678}$. Upon enumerating all DDTs for $x_0 \in [0x00000001, 0x000000FF]$, we also found that they all have $P = 2^{-4.679}$. This supports our previous conjecture that the bits that have the biggest impact on differential probability are those that the substitution function’s inputs have directly modified.

4.3 Boomerang Connectivity

Boomerang attacks are an extension of differential cryptanalysis [7], whereby a cipher is divided into two halves that is connected by a single *switching* round. An attacker first needs to find differential trails for the upper and lower halves of the cipher, and check if they are compatible by using a boomerang connectivity table (BCT). The BCT defines all possible switching scenarios and is very similar to the DDT. More information about BCTs can be found in [37]. Since the proposed chaos-based substitution function is not bijective, we need to instead construct what is known as a Feistel BCT (FBCT) [38]. An example of a 4-bit FBCT where $x_0 = 0.05129266157746315$ ($x_0 = 0x03486104$) when $i = 100$ is shown in Table 3. The various known properties of the FBCT can be clearly seen such as diagonal symmetry, the ladder switch (fixed values for the first column and row) and the Feistel switch (fixed values diagonally).

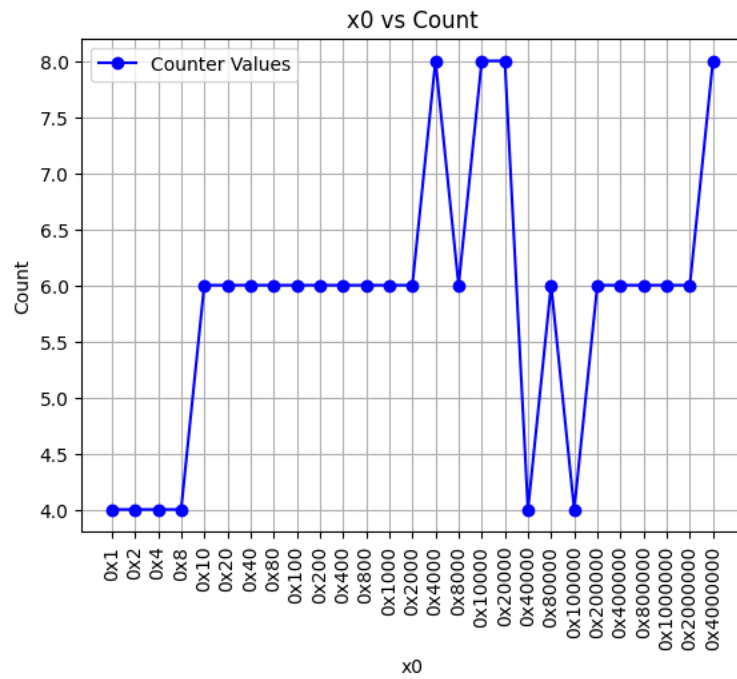


Figure 6. Comparing 4-bit DDTs Where 1 bit Of x_0 Is Active And $i = 500$

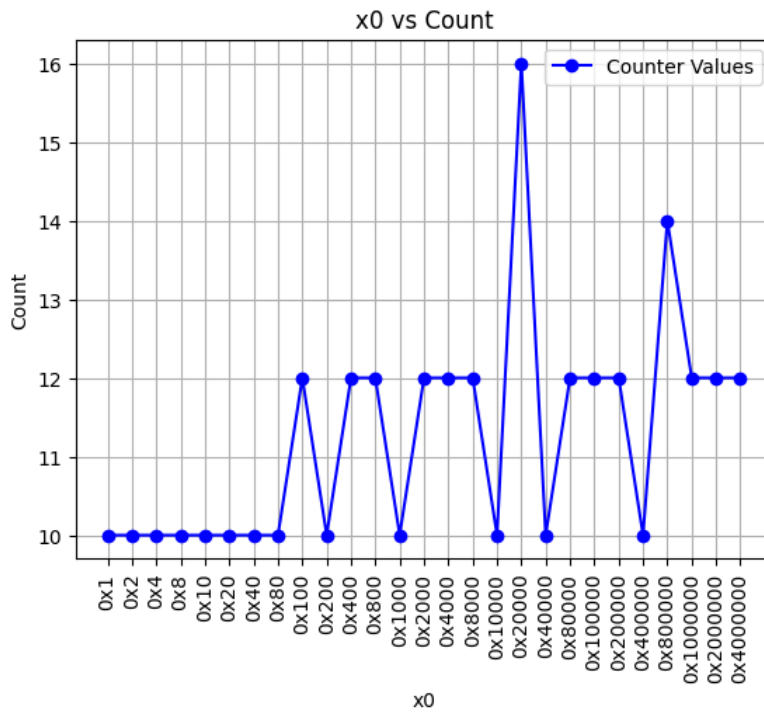


Figure 7. Comparing 8-bit DDTs Where 1 Bit Of x_0 Is Active And $i = 50$

Table 3. 4-bit FBCT Where $x_0 = 0.005129266157746315, i = 100$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
1	16	16	4	4	4	4	4	4	0	0	0	0	0	0	0	0
2	16	4	16	4	0	0	0	0	0	4	0	4	0	4	0	4
3	16	4	4	16	0	0	0	0	0	0	0	0	0	0	0	0
4	16	4	0	0	16	4	0	0	4	0	0	0	4	0	0	0
5	16	4	0	0	4	16	0	0	0	0	0	0	0	0	0	0
6	16	4	0	0	0	0	16	4	0	0	4	0	4	0	0	0
7	16	4	0	0	0	0	4	16	0	0	0	0	0	0	0	0
8	16	0	0	0	4	0	0	0	16	0	0	0	4	0	0	0
9	16	0	4	0	0	0	0	0	0	16	0	4	0	0	0	0
A	16	0	0	0	0	0	4	0	0	0	16	0	4	0	0	0
B	16	0	4	0	0	0	0	0	0	4	0	16	0	0	0	0
C	16	0	0	0	4	0	4	0	4	0	4	0	16	0	0	0
D	16	0	4	0	0	0	0	0	0	0	0	0	0	16	0	4
E	16	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0
F	16	0	4	0	0	0	0	0	0	0	0	0	0	4	0	16

5. CONCLUSION

One of the main problems in chaos-based cryptography is that their ad hoc designs are often highly complex and hinders third-party cryptanalysis efforts. In this paper, we address this issue by introducing a straightforward approach to using chaotic maps that facilitates cryptanalysis. This goal was achieved by adopting fixed-point representation in place of floating-point, which allows us to compute chaotic maps using straightforward binary operations. We first introduce a simple chaos-based substitution function that can accommodate various input and output sizes. It is a one-way substitution function that can be used in stream ciphers or block ciphers. We show how we can easily construct a difference distribution table to analyse the substitution function's security against differential cryptanalysis. The impact of varying different chaotic map parameters on differential properties was examined in several experiments involving the logistic map as a proof-of-concept. As future work, we will explore the use of other more complex chaotic maps and how well they fare against other conventional cryptanalysis methods such as linear cryptanalysis.

ACKNOWLEDGEMENT

The authors received no funding from any party for the research and publication of this article.

AUTHOR CONTRIBUTIONS

Je Sen Teh: Conceptualization, Data Curation, Methodology, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing;

Abubakar Abba: Writing – Review & Editing;

CONFLICT OF INTERESTS

The authors declare that they have no conflict of interest.



REFERENCES

- [1] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976, doi: 10.1038/261459a0.
- [2] J. S. Teh, M. Alawida, and Y. C. Sii, "Implementation and practical problems of chaos-based cryptography revisited," *Journal of Information Security and Applications*, vol. 50, p. 102421, 2019, doi: 10.1016/j.jisa.2019.102421.
- [3] J. Daemen and V. Rijmen, *The Design of Rijndael*. 2002. doi: 10.1007/978-3-662-04722-4.
- [4] A. Bogdanov et al., "PRESENT: An Ultra-Lightweight Block Cipher," in *Lecture notes in computer science*, 2007, pp. 450–466. doi: 10.1007/978-3-540-74735-2_31.
- [5] J.-F. Lai and S.-H. Heng, "Secure File Storage On Cloud Using Hybrid Cryptography," *Journal of Informatics and Web Engineering*, vol. 1, no. 2, pp. 1–18, 2022, doi: 10.33093/jiwe.2022.1.2.1.
- [6] E. Biham and A. Shamir, "Differential Cryptanalysis of the Full 16-round DES," in *Springer eBooks*, 2007, pp. 487–496. doi: 10.1007/3-540-48071-4_34.
- [7] D. Wagner, "The Boomerang Attack," in *Lecture notes in computer science*, 1999, pp. 156–170. doi: 10.1007/3-540-48519-8_12.
- [8] F. Lafitte, "CryptoSAT: a tool for SAT-based cryptanalysis," *IET Information Security*, vol. 12, no. 6, pp. 463–474, Apr. 2018, doi: 10.1049/iet-ifs.2017.0176.
- [9] R. Ankele and S. Kölbl, "Mind the Gap - A Closer Look at the Security of Block Ciphers against Differential Cryptanalysis," in *Lecture notes in computer science*, 2019, pp. 163–190. doi: 10.1007/978-3-030-10970-7_8.
- [10] J. S. Teh and A. Biryukov, "Differential cryptanalysis of WARP," *Journal of Information Security and Applications*, vol. 70, p. 103316, 2022, doi: 10.1016/j.jisa.2022.103316.
- [11] H. Fan, C. Zhang, H. Lu, M. Li, and Y. Liu, "Cryptanalysis of a New Chaotic Image Encryption Technique Based on Multiple Discrete Dynamical Maps," *Entropy*, vol. 23, no. 12, p. 1581, 2021, doi: 10.3390/e23121581.
- [12] J. M. K. Mastan and R. Pandian, "Cryptanalytic attacks on a chaos-based image encrypting cryptosystem," *International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2021, doi: 10.1109/icacite51222.2021.9404679.
- [13] H. Wen, Y. Lin, L. Yang, and R. Chen, "Cryptanalysis of an image encryption scheme using variant Hill cipher and chaos," *Expert Systems With Applications*, vol. 250, p. 123748, 2024, doi: 10.1016/j.eswa.2024.123748.
- [14] P. Cao and L. Teng, "A chaotic image encryption algorithm based on sliding window and pseudo-random stack shuffling," *Nonlinear Dynamics*, vol. 112, no. 15, pp. 13539–13569, 2024, doi: 10.1007/s11071-024-09727-0.
- [15] M. Alawida, "A novel DNA tree-based chaotic image encryption algorithm," *Journal of Information Security and Applications*, vol. 83, p. 103791, 2024, doi: 10.1016/j.jisa.2024.103791.

- [16] S. Zhu, X. Deng, W. Zhang, and C. Zhu, "Secure image encryption scheme based on a new robust chaotic map and strong S-box," *Mathematics and Computers in Simulation*, vol. 207, pp. 322–346, 2023, doi: 10.1016/j.matcom.2022.12.025.
- [17] H. R. Shakir, S. A. Mehdi, and A. A. Hattab, "A new four-dimensional hyper-chaotic system for image encryption," *International Journal of Power Electronics and Drive Systems/International Journal of Electrical and Computer Engineering*, vol. 13, no. 2, p. 1744, 2022, doi: 10.11591/ijece.v13i2.pp1744-1756.
- [18] W. J. Jun and T. S. Fun, "A New Image Encryption Algorithm Based on Single S-Box and Dynamic Encryption Step," *IEEE Access*, vol. 9, pp. 120596–120612, 2021, doi: 10.1109/access.2021.3108789.
- [19] A. Alghafis, N. Munir, M. Khan, and I. Hussain, "An Encryption Scheme Based on Discrete Quantum Map and Continuous Chaotic System," *International Journal of Theoretical Physics*, vol. 59, no. 4, pp. 1227–1240, 2020, doi: 10.1007/s10773-020-04402-7.
- [20] M. Preishuber, T. Hutter, S. Katzenbeisser, and A. Uhl, "Depreciating Motivation and Empirical Security Analysis of Chaos-Based Image and Video Encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2137–2150, 2018, doi: 10.1109/tifs.2018.2812080.
- [21] R. Yates, *Fixed-Point Arithmetic: An Introduction*. Digital Signal Labs, 2024. [Online]. Available: <http://www.digitalsignallabs.com/downloads/fp.pdf>.
- [22] D. Khovratovich, "Cryptanalysis of Hash Functions with Structures," in *Lecture notes in computer science*, 2009, pp. 108–125. doi: 10.1007/978-3-642-05445-7_7.
- [23] H. Wu and B. Preneel, "Differential Cryptanalysis of the Stream Ciphers Py, Py6 and Pypy," in *Lecture notes in computer science*, 2007, pp. 276–290. doi: 10.1007/978-3-540-72540-4_16.
- [24] H. Wu and B. Preneel, "Differential-Linear Attacks Against the Stream Cipher Phelix," in *Lecture notes in computer science*, 2007, pp. 87–100. doi: 10.1007/978-3-540-74619-5_6.
- [25] S. Parikibandla and S. Alluri, "Low area field-programmable gate array implementation of PRESENT image encryption with key rotation and substitution," *ETRI Journal*, vol. 43, no. 6, pp. 1113–1129, 2021, doi: 10.4218/etrij.2020-0203.
- [26] T. Jakobsen and L. R. Knudsen, "The interpolation attack on block ciphers," in *Lecture notes in computer science*, 1997, pp. 28–40. doi: 10.1007/bfb0052332.
- [27] W. Zhang, H. Yu, Y.-L. Zhao, and Z.-L. Zhu, "Image encryption based on three-dimensional bit matrix permutation," *Signal Processing*, vol. 118, pp. 36–50, 2015, doi: 10.1016/j.sigpro.2015.06.008.
- [28] Y. Liu, P. Liu, and Z. Qin, "An image encryption scheme based on Arnold map and DNA sequence," *Multimedia Tools and Applications*, vol. 77, no. 24, pp. 31531–31548, 2018, doi: 10.1007/s11042-017-5388-2.
- [29] T. Gao and Z. Chen, "A new image encryption algorithm based on hyper-chaos," *Physics Letters A*, vol. 372, no. 4, pp. 394–400, 2007, doi: 10.1016/j.physleta.2007.07.040.
- [30] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1996.

- [31] H. Wu, "The Stream Cipher HC-128," in *Lecture notes in computer science*, 2008, pp. 39–47. doi: 10.1007/978-3-540-68351-3_4.
- [32] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, Jan. 1991, doi: 10.1007/bf00630563.
- [33] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed. New York, NY, USA: Wiley, 1996.
- [34] M. Matsui, "Linear Cryptanalysis Method for DES Cipher," in *Lecture notes in computer science*, 1994, pp. 386–397. doi: 10.1007/3-540-48285-7_33.
- [35] D. Wagner, "Cryptanalysis of the Alleged RC4 Keystream Generator," in *Fast Software Encryption*, vol. 1267, S. Vaudenay, Ed., *Lecture Notes in Computer Science*, vol. 1267, Springer Berlin Heidelberg, 1997, pp. 18–28, doi: 10.1007/BFb0052345.
- [36] K. Jain, "Side Channel Attacks: Ten Years After Its Publication and the Impact on Cryptographic Algorithm and Devices," in *IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2011, pp. 257–261, doi: 10.1109/TrustCom.2011.154.
- [37] N. Ferguson et al., "Improved cryptanalysis of Rijndael," in *Lecture notes in computer science*, 2001, pp. 213–230. doi: 10.1007/3-540-44706-7_15.
- [38] H. Wang, J. Hu, X. Hu, and C. Zhu, "Design and Implementation of Lightweight Encryption for Image Based on Chaotic Systems," *Mathematics*, vol. 9, no. 17, p. 2022, 2021, doi: 10.3390/math9172022.

BIOGRAPHIES OF AUTHORS

	<p>Je Sen Teh is a lecturer at the School of IT, Deakin University. He is also a researcher at the Deakin Cyber Research and Innovation Centre. His research interests include cybersecurity and cryptography. He can be contacted at: j.teh@deakin.edu.au</p>
	<p>Abubakar Abba received his B.Sc (Hons.) in Computer Science from Bayero University Kano, 2010, M.Sc in Computer System and Engineering from University of East London, 2017 and PhD at the School of Computer Sciences, Universiti Sains Malaysia. He is a lecturer and researcher at the Federal College of Education, Zaria Kaduna State, Nigeria. His research interests include Information Security, Cryptography and Cyber security. He can be contacted at: abbatahiru@gmail.com</p>

