

## Sequence pairs related to produced graphs from a method for dividing a natural number by two

Mohammad Zeynali Azim<sup>\*1</sup>, Saeid Alikhani<sup>†2</sup>, Babak Anari<sup>‡3</sup> and Bagher Zarei<sup>§4</sup>

<sup>1,3,4</sup>Department of Computer Engineering, Shabestar Branch, Islamic Azad University, Shabestar, East Azerbaijan, Iran

<sup>2</sup>Department of Mathematical Sciences, Yazd University, 89195-741, Yazd, Iran

---

### ABSTRACT

This paper is about producing a new kind of pairs which we call MS-pairs. To produce these pairs, we use an algorithm for dividing a natural number  $x$  by two for two arbitrary numbers and consider their related graphs. We present some applications of these pairs that show their interesting properties such as unpredictability, irreversible, aperiodicity and chaotic behavior.

*Keyword:* Algorithm, Graph, Diamond, DGBT

AMS subject Classification: 11Y16, 05C63.

---

### ARTICLE INFO

*Article history:*

Research paper

Received 13, February 2023

Received in revised form 16, April 2023

Accepted 17 May 2023

Available online 01, June 2023

---

\*mo.zeynali@iaushab.ac.ir

†Corresponding author, S. Alikhani: alikhani@yazd.ac.ir

‡anari@iaushab.ac.ir

§zareibagher@iaushab.ac.ir

# 1 Introduction

In [4] a new method (which is called MZ-algorithm), has presented for dividing a natural number  $x$  by two and used graphs as models to show MZ-algorithm. As an example, Figure 1 illustrates the MZ-algorithm for division number 375 by two.

Every digit in a number denoted by a vertex and edges of graph draw based on MZ-algorithm. We have shown the division of the number 458 by two in Figure 2. This graph (Figure 2) which we call it division graph by two (DGBT) is a path of order 13, i.e.,  $P_{13}$  (see [4]).

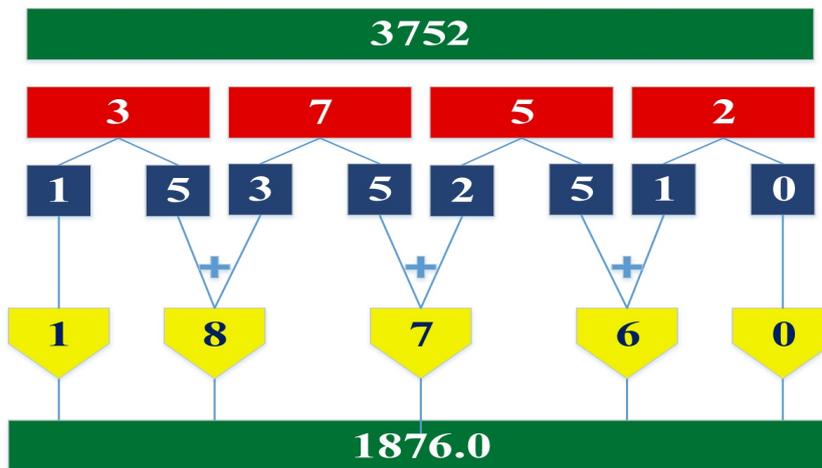


Figure 1: Algorithm for the division number 3752 by two.

Applying  $k$ -times of the MZ-method for the number  $x$ , creates a graph with unique structure that is denoted by  $G_k(x)$  and is called DGBT. It is easy to see that  $G_k(n)$  is not tree for  $k > 1$ , since the graph has cycle. See the graph  $G_2(375)$  in Figure 3 (see [4]).

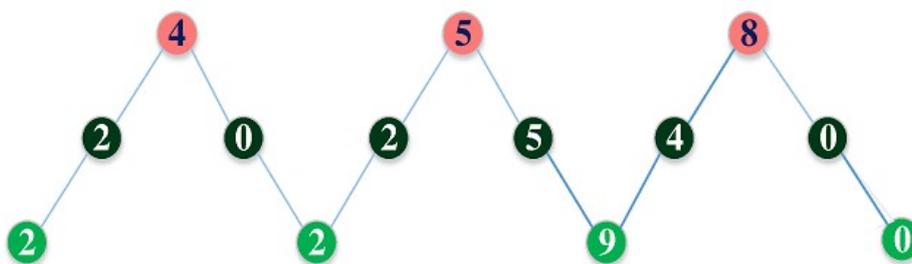


Figure 2: The graph  $G_1(458)$ .

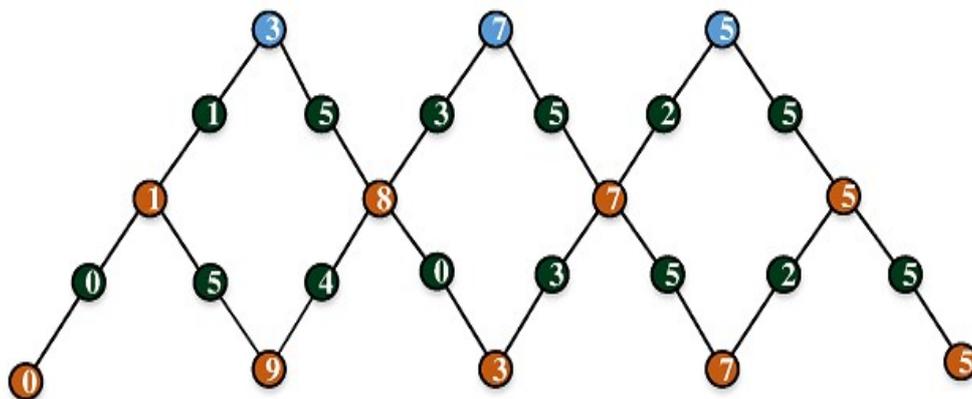


Figure 3: The graph  $G_2(375)$ .

1 INTRODUCTION

Since DGBT is an infinite graph, sometimes is better to show some of DGBT in bitmap model. In bitmap model each numbers in 0, 1, 2, ..., 9 represented by unique color. See Figure 4.

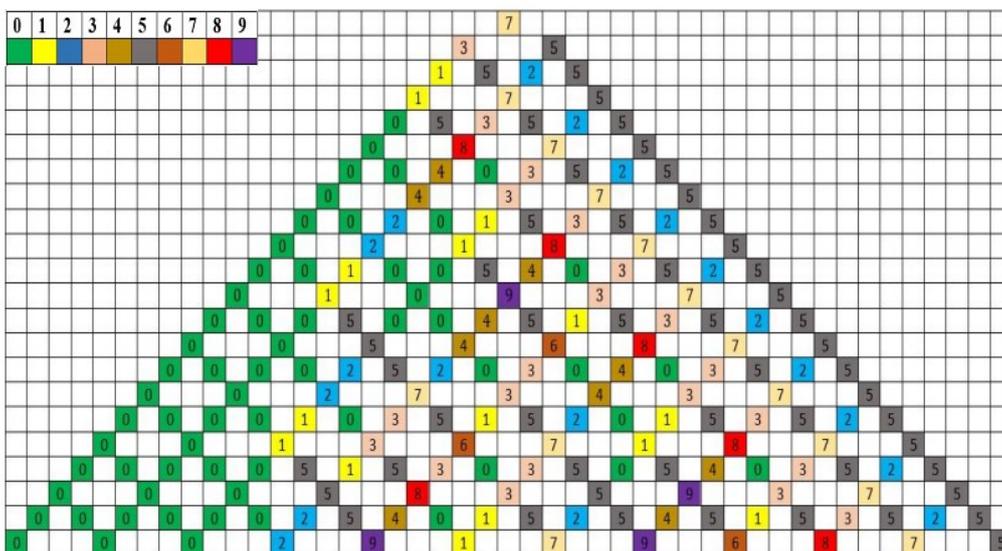


Figure 4: Bitmap model for the graph  $G_{11}(7)$

It is easy to see that the number of cycles  $C_8$  in the graph  $G_k(x)$  is  $\frac{k-1}{2}(2d + k - 2)$ . We show the figures of these cycles  $C_8$  in  $G_k(x)$  similar to diamond. After finding all diamonds in the

graph  $G_k(x)$ , we label them by number 0, 1, 2, ... and write these label inside diamonds (Figure 5).

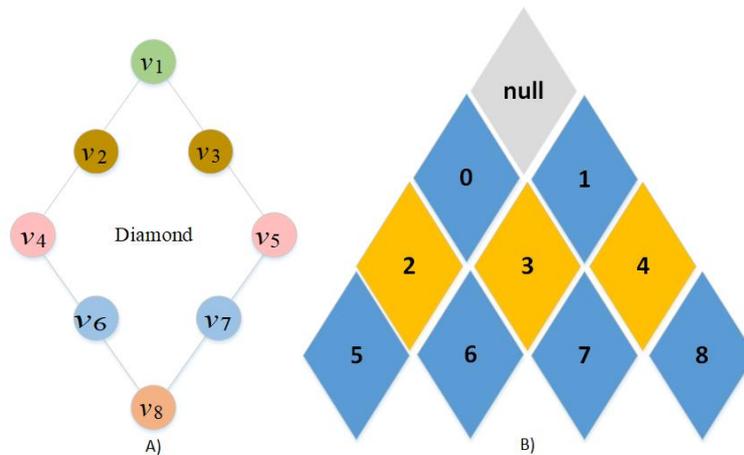


Figure 5: A) Diamond structure. B) Nine diamonds related to level 1 to 11 in DGBT.

We consider these diamond structures and use them to produce a sequence of pairs, that we call these pairs MS-pairs. In Section 2, we introduce MS-pairs and investigate some of their properties. In Section 3, we present some of applications and behavior of these pairs in cryptography and chaos theory.

## 2 MS-pairs Generation

We start this section by introducing MS-pairs. First we state the following definition:

**Definition 2.1.** Consider the diamond  $d_i$  in the graph  $G_k(x_1)$ , and the diamond  $d_j$  in the graph  $G_k(x_2)$ . We say that these two diamonds are equal, if all values of vertices in the diamond  $d_i$  are equal to all values of vertices in the diamond  $d_j$ . Note that we compare the value of a vertex in the diamond  $d_i$  with the value of the same vertex (topologically) in the diamond  $d_j$ .

The following statement gives the definition of MS-pair related to graph  $G_k(x_1)$  and  $G_k(x_2)$ :

**Definition 2.2.** The MS-pair related to graphs  $G_k(x_1)$  and  $G_k(x_2)$  is denoted by  $(i, j)$  and the first and the second components of this pair obtain as follows. If the diamond  $d_i$  from the graph  $G_k(x_1)$  is equal to the diamond  $d_j$  from the graph  $G_k(x_2)$ , then we consider the label of these two diamonds as the first and the second component of a pair, i.e.,  $(i, j)$ .

Observed that by considering the graphs  $G_k(x_1)$  and  $G_k(x_2)$  and applying MZ-algorithm  $k$ -times (for large enough number  $k$ ) we can produce sequences of MS-pairs. These pairs have interesting properties such as unpredictability, irreversible, aperiodicity and chaotic behavior and so they are useful and applicable in cryptography, chaos theory, random number generation, stegnaography, password hashing, and unique identifier generators. Note that if we

have all of the MS-pairs that generated from  $G_k(x)$  and  $G_k(y)$ , we cannot predict or determine the root numbers of graphs, i.e.,  $x$  and  $y$ .

The following are some of the properties of diamonds. We consider and use Figure 6 for convenience. According this figure, the layers denoted by 1 to  $n$ . The vertices of a diamond is labeled with  $v_1, v_2, \dots, v_8$ . As observe that the first vertex i.e., the vertex  $v_1$  of all diamonds are in the odd levels. Also we need at least five layers in DGBT to have diamond structure.

Let use the notation  $dg$  for the number of nodes in the root of DGBT (which is the number of digits in the number that used for constructing DGBT).

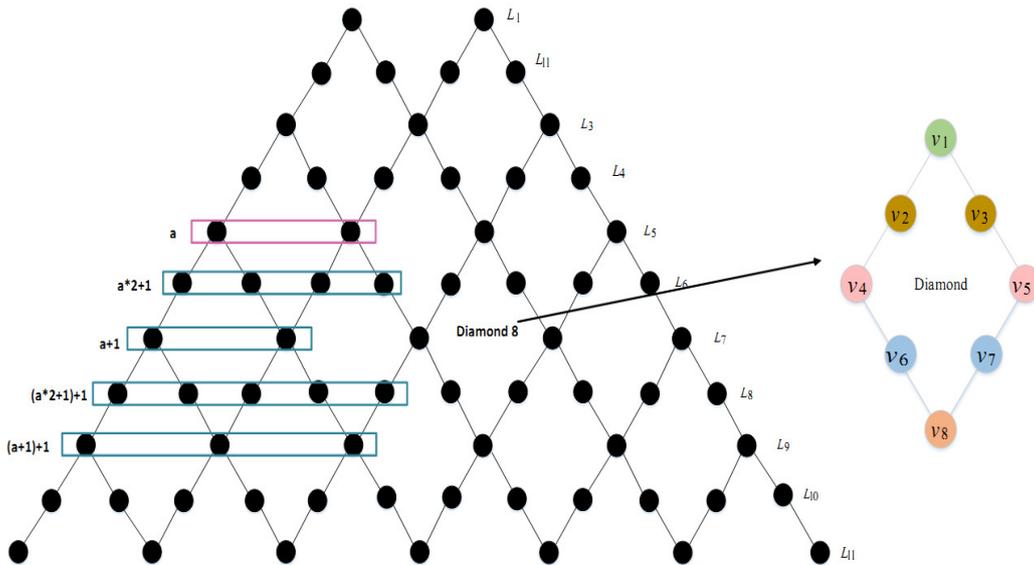


Figure 6: The graph  $G_5(n)$ .

With these notation, we state the following easy theorem:

**Theorem 2.3** Let  $L$  be the level of the graph DGBT.

- (i) If  $L$  is odd, then the number of nodes in that level is  $onc = \lfloor \frac{L-1}{2} \rfloor + dg$ .
- (ii) If  $L$  is event, then the number of nodes in that level is  $enc = 2 \lfloor \frac{L-1}{2} \rfloor + dg$ .
- (iii) The total number of nodes from the level  $L_1$  to the level  $L_n$  is

$$tn = \sum_{L=1}^n \left( \left\lfloor \frac{L-1}{2} \right\rfloor + dg \right) ((L+1)_2 + 1),$$

where the notation  $(x)_2$  is the reminder of the division of  $x$  by two.

We end this section by the following properties:

**Theorem 2.4**

(i) If  $d$  is the number of diamonds in the DGBT graph  $G_l(n)$ , then

$$d = \binom{l+1}{2} - \binom{n}{2}.$$

(ii) If  $G_k(x)$  is an infinite DGBT, then the number of its diamonds is infinite.

(iii) MS-pairs are unique. In other words, there is no two numbers  $x$  and  $y$  such that MS-pairs produced by  $G_k(x)$  and  $G_k(y)$  are exactly the same.

### 3 Some applications of the MS-pairs

In this section we state a behavior and some applications of the MS-pairs.

#### 3.1 A behavior of MS-pairs

In chaos theory, the butterfly effect is the sensitive dependence on initial conditions in which a small change in one state of a deterministic nonlinear system can result in large differences in a later state [1]. In this subsection, we observe that a small change in the root number of DGBT, causes very large changes in the correspond MS-pairs.

Consider two graphs  $G_k(x)$  and  $G_k(y)$ , where  $x$  is a number that we can consider it as a time (which is a number with at most six digit, for example the time 7:19:27 consider as number 71927) and  $y$  is an arbitrary constant number. We have shown the DGBT for the time 7:19:27 in Figure 7.

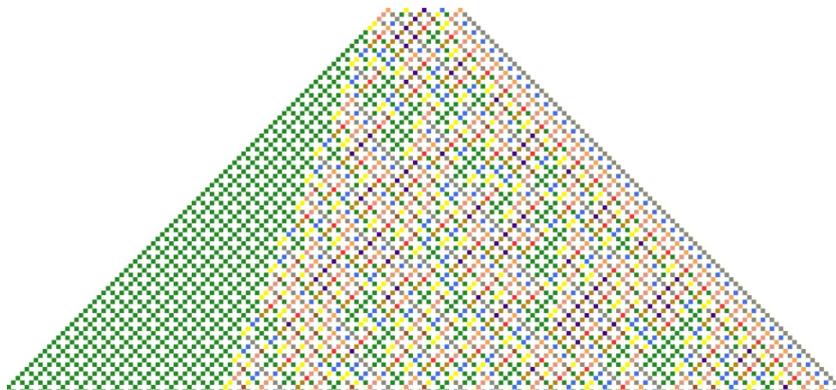


Figure 7: Bitmap model for DGBT of time 7:19:27.

**Example 3.1** Suppose that  $x = 7 : 19 : 27$  and  $y = 45218$ . By Theorem ??(iii), we have 11438 pairs until level 100, i.e., in  $G_{100}(x)$  and  $G_{100}(y)$ . We bring up some of these MS-pairs in the following:

- $\{(2, 103), (2, 116), (2, 212), (2, 238), (3, 60), (3, 91), (3, 185), (3, 239), (4, 14), (4, 149),$
- $(4, 331), (4, 360), (5, 36), (5, 257), (6, 136), (6, 248), (6, 329), (7, 118), (7, 233), (7, 260), (8, 18),$
- $(8, 72), (8, 146), (8, 213), (8, 261), (8, 361), (8, 374), (10, 168), (11, 46), (11, 104), (11, 204),$
- $(12, 47), (12, 115), (12, 153), (12, 281), \dots \}$ .

Now if we change time  $x$  to  $x_1 = 7 : 19 : 26$  we have 11136 pairs in  $G_{100}(x_1)$  and  $G_{100}(y)$ . The following are some of these MS-pairs:

- {(2, 62), (2, 116), (2, 356), (2, 390), (3, 60), (3, 63), (3, 185), (3, 195), (3, 306), (3, 391),
- (4, 14), (4, 149), (5, 36), (5, 103), (5, 385), (6, 248), (7, 4), (8, 18), (8, 72), (8, 75), (8, 146),
- (8, 331), (8, 374), (9, 332), (10, 10), (10, 17), (10, 25), (10, 34), (10, 44), (10, 55), (10, 67),
- (10, 80), (10, 94), (10, 109), (10, 125), (10, 142), (10, 60), ...}

We compare these two sequences of pairs. Only the pairs

$$\{(2, 116), (3, 60), (4, 14), (5, 36), (6, 248), (8, 18), (8, 146)\}$$

are equal from these two sequence of MS-pairs. As we see, a small change in the root number, causes very large changes in the correspond MS-pairs. This phenomenon is referred to butterfly effect in chaos theory [1]. In order to evaluate and comparing this properties, we draw scatter plot for two MS-pairs generated in Example ???. Please see Figure 8. In Figure 8, there are two figures. For generating Figure 8a), we use the pairs that produced by  $x = 7 : 19 : 27$ , and  $y = 45218$  that displayed with blue color. Figure 8b) is related to MS-pair that produced by  $x = 7 : 19 : 26$ , and  $y = 45218$  displayed with green color. Observe that the two forms in these two figures are very different.

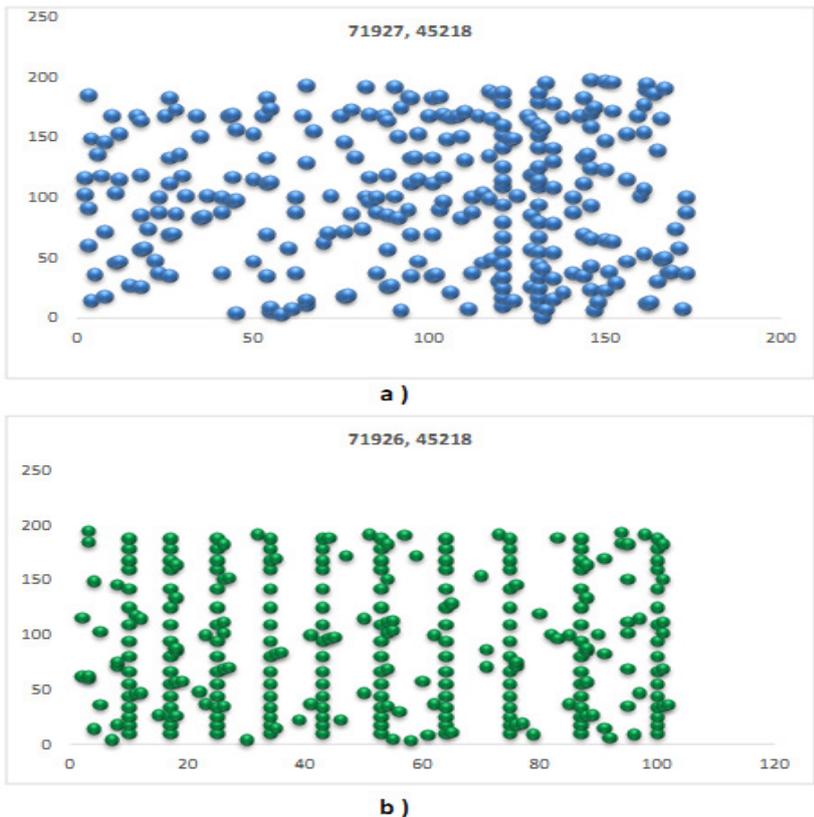


Figure 8: a) MS-pairs produced using  $x = 7 : 19 : 27$ , and  $y = 45218$ . b) Related to MS-pair that produced by  $x = 7 : 19 : 26$ , and  $y = 45218$ .

### 3.1.1 Non-repudiation system by MS-pairs

Non-repudiation is the assurance that someone cannot deny the validity of something. Non-repudiation is a legal concept that is widely used in information security and refers to a service, which provides proof of the origin of data and the integrity of the data. In other words, non-repudiation makes it very difficult to successfully deny who/where a message came from as well as the authenticity and integrity of that message.

Digital signatures (combined with other measures) can offer non-repudiation when it comes to online transactions, where it is crucial to ensure that a party to a contract or a communication cannot deny the authenticity of their signature on a document or sending the communication in the first place [2].

Here, using MS-pairs we prove the claim of one of the two parties (users or systems). As an example, we consider two persons  $A$ ,  $B$ , where  $A$  has ID-number 15963 and  $B$  has ID-number 15964. At the time  $t = 7 : 19 : 27$  person  $A$  have signed the contract  $d$ . We generate MS-pairs with  $t = 7 : 19 : 27$ , ID-number=15963 and write the produced MS-pairs on the contract  $d$ . Suppose that after signing the contract  $d$  by  $A$ , two person  $A$ ,  $B$  both claim that in the time  $t$ , signed or not-signed the contract  $d$ . To judge their claims, system or the third person  $C$  get their ID-numbers and produce MS-pairs for these ID-numbers and the time  $t$  on the contract  $d$ . If the produced MS-pairs are equal to the MS-pairs on the contract, then the claim of that person is true. For example, the MS-Pairs generated for the person  $A$  is:

$MS - Pairs(7 : 19 : 27, 15963) = [(4, 61), (5, 73), (6, 1), (7, 2), (7, 29), (8, 3), (9, 73), (10, 10), (10, 17), (10, 25), (10, 34), (10, 44), (10, 55), (10, 67), (10, 80), (10, 94), (13, 7), (13, 13), (13, 20), (14, 8), (16, 24), (16, 43), (16, 66), (16, 93), (17, 10), (17, 17), (17, 25), (17, 34), (17, 44), (17, 55), (17, 67), \dots]$  and

The MS-Pairs generated for the person  $B$  is:

$MS - Pairs(7 : 19 : 27, 15964) = [(1, 62), (2, 75), (4, 61), (5, 73), (6, 1), (6, 74), (7, 2), (7, 29), (7, 88), (9, 73), (10, 23), (10, 32), (10, 42), (10, 53), (10, 65), (10, 78), (10, 92), (12, 87), (13, 7), (13, 13), (13, 20), (16, 31), (16, 52), (16, 77), (17, 23), (17, 32), (17, 42), (17, 53), (17, 65), (17, 78), (17, 92), \dots]$

The following algorithm, gives the non-reproduction steps of the proposed algorithm in more detail.

---

#### Algorithm 1: Non-repudiation System Algorithm

---

```

input :  $ID_A, ID_B, time t, contract with d ID$ 
output: MS-pairs, claim true
1  $get(ID_A, t)$ 
2  $d \leftarrow produce\_pairs(ID_A, t)$ ;
3  $get(ID_A, ID_B, t)$ ;
4  $claim_A \leftarrow produce\_pairs(ID_A, t)$ ;
5  $claim_B \leftarrow produce\_pairs(ID_B, t)$ ;
6 if  $d == claim_A$  then
7    $print("A claim is true")$ 
8 if  $d == claim_B$  then
9    $print("B claim is true")$ 

```

---

### 3.2 Applications to Steganography and Cryptography

Stream cipher is an important branch of symmetric cryptosystems, which takes obvious advantages in speed and scale of hardware implementation. It is suitable for using in the cases of massive data transfer or resource constraints, and has always been a hot and central research topic in cryptography. A word-oriented stream cipher usually works on and outputs words of certain size, like 32, 16, 8 bits [2]. We will use MS-pairs for word-stream-cipher cryptographic application. We present a simple algorithms to show that how MS-pairs could to encrypt and decrypt data or message. Word size in this algorithm is 8 bits.

---

#### Algorithm 2: MS-Pairs encryption

---

```

1 Sender_Select :  $x_1, diamond\_number$ 
2 Sender_Send :  $x_1, diamond\_number \rightarrow receiver$ 
3 Receiver_Select :  $x_2$ 
4  $MS - pairs \leftarrow Receiver\_Compute\_algorithm1(x_2, x_1)$ 
5 while  $p$  in  $MS - pairs$  do
6    $e_i \leftarrow ((p_a + p_b) \bmod Chara\_number);$ 
7    $i = i + 1$ 
8    $en_i \leftarrow (|e_i - char\_code|);$ 
9 Output  $\leftarrow en$ ;
```

---

Decryption algorithm is as follows:

---

#### Algorithm 3: MS-Pairs decryption

---

```

1 Sender_recived :  $x_2, en$ 
2  $MS - pairs \leftarrow Receiver\_Compute\_algorithm1(x_2, x_1)$ 
3 while  $p$  in  $MS - pairs$  do
4    $d_i \leftarrow ((p_a + p_b) \bmod Chara\_number);$ 
5    $i = i + 1$ 
6    $de_i \leftarrow (|d_i - char\_code|);$ 
7 Output  $\leftarrow de$ ;
```

---

For example we encrypt and decrypt the plain-text “Graph” with arbitrary  $x_1 = 71927, x_2 = 45218$  and  $diamond\_number = 13$ . First we produce MS-Pairs for  $x_1, x_2$  then, we take the number of characters in the message from the MS-Pairs. According this algorithm since the plain-text has five characters so we can select only five pairs. Generated pairs are:

$$[(7, 118), (7, 233), (7, 260), (8, 18), (8, 72)].$$

We consider arbitrary character number, say 256. According the encryption algorithm we have:

$$\begin{aligned}
e_1 &= (7 + 118) = 125 \text{ (mode 256)}, \\
e_2 &= (7 + 233) = 240 \text{ (mode 256)}, \\
e_3 &= (7 + 260) = 11 \text{ (mode 256)}, \\
e_4 &= (8 + 18) = 26 \text{ (mode 256)}, \\
e_5 &= (8 + 72) = 80 \text{ (mode 256)}.
\end{aligned}$$

Then we discover the code of characters in “Graph”. We have:

$$\begin{aligned}
char\_code &\leftarrow \{71, 114, 97, 112, 104\} \text{ and} \\
en &= \{|71 - 125|, |114 - 240|, |97 - 11|, |112 - 26|, |104 - 80|\}.
\end{aligned}$$

Finally, the encryption of "Graph" will be:  $en = \{54, 126, 86, 86, 24\}$ . As we know that the decryption algorithm is reverse of encryption. So we can reach from  $en = \{54, 126, 86, 86, 24\}$  to "Graph" by using decryption algorithm. For decryption we can do the reverse operation on encrypted message. According Algorithm 2, we have

$$en = \{71, 114, 97, 112, 104\}.$$

## 4 Acknowledgement

The authors would like to express their gratitude to the referee for her/his careful reading and helpful comments.

## References

- [1] Edward, L., The Essence of Chaos, University of Washington Press, USA, 1993.
- [2] Jiao, L., Hao, Y., Feng, D., Stream cipher designs: a review, Sci. China Inf. Sci. 63, 131101 (2020) 1-25.
- [3] Wayner, P., Disappearing Cryptography: Information Hiding: Steganography & Watermarking, Morgan Kaufmann Publishers, Amsterdam; Boston, 2009.
- [4] Zeynali Azim, M., Jabraeil Jamali, M.A., Anari, B., Alikhani, S., Akbarpour, S., Graphs and codes produced by a new method for dividing a natural number by two, Available at <https://arxiv.org/abs/2003.01691>.