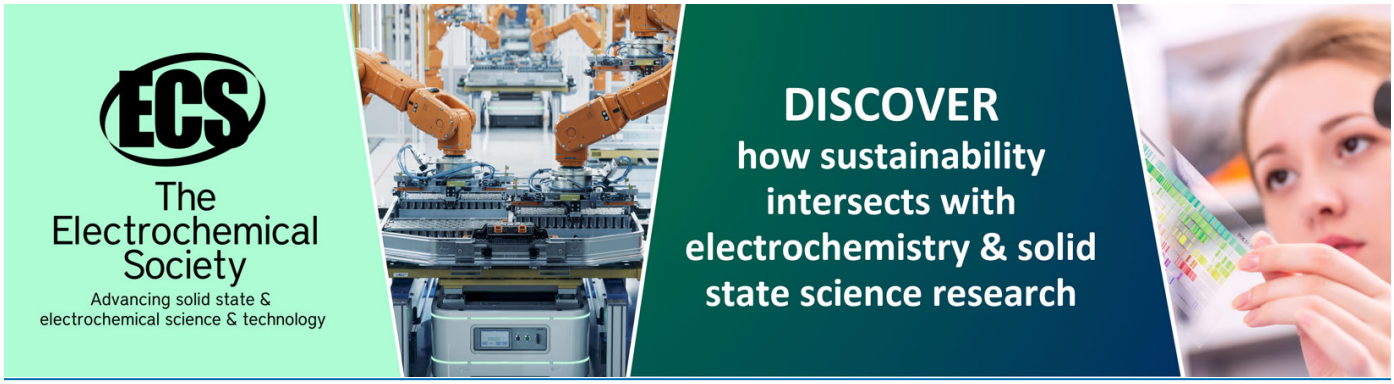
Improving the robustness of the affine cipher by using a rainbow antimagic coloring



Improving the robustness of the affine cipher by using a rainbow antimagic coloring

R Nisviasari[1], Dafik[1,2], I H Agustin[1,3], E Y Kurniawati[1], I N Maylisa[1] and B J Septory[1,4]

[1]CGANT Research Group, University of Jember, Jember, Indonesia
[2]Department of Mathematics Education, University of Jember, Jember, Indonesia
[3]Department of Mathematics, University of Jember, Jember, Indonesia
[4]Department of Mathematics, University of Airlangga, Surabaya, Indonesia

*Corresponding author

E-mail: d.dafik@unej.ac.id

Abstract. Nowadays, cryptosystems can be applied in several areas in life. One of them is in transaction data. In transaction data, a very strong cryptosystem is needed so that the transaction data is safe. Cryptosystems are better with a strong keystream. In this case, we use rainbow antimagic as a cryptosystem key to improve the robustness of the keystream by using affine cipher. The algorithm uses the edge weights of rainbow antimagic vertex labeling of graphs as a key for encryption and decryption. In this paper, we found the rainbow antimagic connection number of tadpole graphs and two algorithms to straighten affine ciphers.

1. Introduction

Nowadays, cryptosystems can be applied in several areas in life. One of them is in transaction data. In transaction data, a very strong cryptosystem is needed so that the transaction data is safe. Cryptosystems are better with a strong keystream. In this case, we use rainbow antimagic as a cryptosystem key to improve the robustness of the keystream.

Cryptosystem consists of two processes, namely the encryption process and the description process. Each process requires a key. The process is intended to make it difficult for others to read the secret message. We can see some of the results of other researchers' research in [2, 5, 6]. There are several types of encryption and decryption processes in cryptosystems, one of them is an affine cipher. Affine cipher is a technique that uses modulo 26 processes in encryption and decryption processes based on the number of letters in the letter [8].

A graph labeling is an assignment of integers to the vertices or edges, or both, subject to certain conditions [4]. Based on [4], there are many kinds of labeling. In [3], they

introduce rainbow antimagic coloring of graphs. The rainbow antimagic coloring of graphs is one type of labeling in which assignment of integers to the vertices of graphs to certain conditions for every edge weights becomes a rainbow coloring path.

First of all, we have to understand about labeling of graph which have bijective function f from vertex set $V(G)$ to the integers number 1 until the cardinality of vertex set $|V(G)|$. We can call the bijective function f as rainbow antimagic vertex labeling if for any two vertices have different edge weights as rainbow path at least one. It means that for every $k, m \in V(G)$ have rainbow path $k-m$ for any edges as uv and u^*v^* in path $k-m$ with edge weights $w(uv) = f(u) + f(v)$ and $w(u^*v^*) = f(u^*) + f(v^*)$ are different $w(uv) \neq w(u^*v^*)$ [3].

All Proposition, Lemma and Theorem that we used to prove the theorem in this paper are as follows.

Proposition 1. *[1] Let G be a nontrivial connected graph of size m. Then*

(a) $src(G) = 1$ if and only if G is a complete graph,

(b) $rc(G) = 2$ if and only if $src(G) = 2$,

(c) $rc(G) = m$ if and only if G is a tree.

Proposition 2. *[1] For each integer $n \geq 4$, $rc(C_n) = src(C_n) = \lceil \frac{n}{2} \rceil$.*

Lemma 1. *[7] Let G be any connected graph. Let $rc(G)$ and $\Delta(G)$ be the rainbow connection number of G and the maximum degree of G, respectively. Then, $rac(G) \geq \max\{rc(G), \Delta(G)\}$.*

Theorem 1. *[3] For $\forall n \geq 3$ where $n \in \mathcal{N}$, then*

$$rac(C_n) = \begin{cases} 3, & \text{if } n = 4, \\ \lceil \frac{n}{2} \rceil, & \text{if } n \equiv 1, 2 \pmod 4, \end{cases}$$

and

$$\left\lceil \frac{n}{2} \right\rceil \leq rac(C_n) \leq \left\lceil \frac{n}{2} \right\rceil + 1, \quad \text{if } n \equiv 0, 3 \pmod 4, n \neq 4.$$

2. Results

Theorem 2. *Let $T_{\alpha,\beta}$ for $\alpha \geq 3$ and $\beta \geq 1$ where α, β is natural number,*

$$rac(T_{\alpha,\beta}) = \begin{cases} 3 + \beta, & \text{if } \alpha = 4, \\ \lceil \frac{\alpha}{2} \rceil + \beta, & \text{if } \alpha \equiv 1, 2 \pmod 4, \end{cases}$$

and $\lceil \frac{\alpha}{2} \rceil + \beta \leq rac(T_{\alpha,\beta}) \leq \lceil \frac{\alpha}{2} \rceil + \beta + 1, \quad \text{if } \alpha \equiv 0, 3 \bmod 4, \alpha \neq 4.$

Proof. The graph $T_{\alpha,\beta}$ is a tadpole graph with vertices $V(T_{\alpha,\beta}) = x_1, x_2, \ldots, x_\alpha, z_1, z_2, \ldots, z_\beta$ and edges $E(T_{\alpha,\beta}) = x_1x_2, x_2x_3, \ldots, x_{\alpha-1}x_\alpha, x_\alpha x_1, x_\alpha z_1, z_1z_2, z_2z_3, \ldots, z_{\beta-1}z_\beta$. We know that the tadpole graph contains C_α and P_β. According to Proposition 1, Proposition 2 and Lemma 1, $rac(T_{\alpha,\beta}) \geq \lceil \frac{\alpha}{2} \rceil + \beta$.

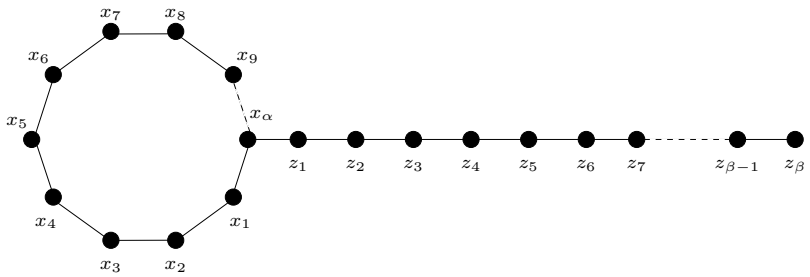


Figure 1. A tadpole graph.

To realize the equality, let us define a vertex labeling $f : V(T_{\alpha,\beta}) \to \{1, 2, 3, \ldots, \alpha+\beta\}$ such that

$$f(x_i) = \begin{cases} \frac{5+i}{2}, & \text{if } \alpha = 4, \text{for } i \text{ is odd} \\ \frac{6-i}{2}, & \text{if } \alpha = 4, \text{for } i \text{ is even} \\ 2i, & \text{if } \alpha \equiv 1 \pmod 4, \text{ for } 1 \le i \le \frac{\alpha-3}{2}, i \text{ is odd or} \\ & \text{if } \alpha \equiv 2 \pmod 4, \text{ for } 1 \le i \le \frac{\alpha}{2}, i \text{ is odd or} \\ & \text{if } \alpha \equiv 3 \pmod 4, \text{ for } 1 \le i \le \frac{\alpha-1}{2}, i \text{ is odd or} \\ & \text{if } \alpha \equiv 0 \pmod 4, \text{ for } 1 \le i \le \frac{\alpha-2}{2}, i \text{ is odd} \\ 2i-1, & \text{if } \alpha \equiv 1 \pmod 4, \text{ for } 2 \le i \le \frac{\alpha-1}{2}, i \text{ is even or} \\ & \text{if } \alpha \equiv 2 \pmod 4, \text{ for } 2 \le i \le \frac{\alpha-2}{2}, i \text{ is even or} \\ & \text{if } \alpha \equiv 3 \pmod 4, \text{ for } 2 \le i \le \frac{\alpha+1}{2}, i \text{ is even or} \\ & \text{if } \alpha \equiv 0 \pmod 4, \text{ for } 2 \le i \le \frac{\alpha}{2}, i \text{ is even} \\ 2i-\alpha-2, & \text{if } \alpha \equiv 1 \pmod 4, \text{ for } \frac{\alpha+3}{2} \le i \le \alpha-1, i \text{ is even or} \\ & \text{if } \alpha \equiv 3 \pmod 4, \text{ for } \frac{\alpha+3}{2} \le i \le \alpha, i \text{ is odd} \\ 2i-\alpha-1, & \text{if } \alpha \equiv 1 \pmod 4, \text{ for } \frac{\alpha+5}{2} \le i \le \alpha, i \text{ is odd or} \\ & \text{if } \alpha \equiv 2 \pmod 4, \text{ for } \frac{\alpha+2}{2} \le i \le \alpha, i \text{ is even or} \\ & \text{if } \alpha \equiv 3 \pmod 4, \text{ for } \frac{\alpha+5}{2} \le i \le \alpha-1, i \text{ is even or} \\ & \text{if } \alpha \equiv 0 \pmod 4, \text{ for } \frac{\alpha+2}{2} \le i \le \alpha-1, i \text{ is odd} \\ 2i-\alpha, & \text{if } \alpha \equiv 2 \pmod 4, \text{ for } \frac{\alpha+4}{2} \le i \le \alpha-1, i \text{ is odd} \\ & \text{if } \alpha \equiv 0 \pmod 4, \text{ for } \frac{\alpha+4}{2} \le i \le \alpha, i \text{ is even.} \end{cases}$$

$$f(x_{\frac{\alpha+1}{2}}) = \alpha, \text{ if } \alpha \equiv 1 \pmod 4 \qquad f(z_k) = \alpha + k, \text{ for } 1 \le k \le \beta.$$

Then, the labeling f provides the vertex weights as follows.

$$w_f(x_ix_{i+1}) = \begin{cases} 4i+1, & \text{if } \alpha \equiv 1 \pmod 4, \text{ for } 1 \le i \le \frac{\alpha-3}{2}, \text{ or} \\ & \text{if } \alpha \equiv 0,2 \pmod 4, \text{ for } 1 \le i \le \frac{\alpha-2}{2}, \text{ or} \\ & \text{if } \alpha \equiv 3 \pmod 4, \text{ for } 1 \le i \le \frac{\alpha-1}{2}, \\ \alpha+1, & \text{if } \alpha \equiv 1 \pmod 4, \text{ for } i = \frac{\alpha+1}{2}, \text{ or} \\ & \text{if } \alpha \equiv 2 \pmod 4, \text{ for } i = \frac{\alpha}{2}, \text{ or} \\ 4i-2\alpha-1, & \text{if } \alpha \equiv 1 \pmod 4, \text{ for } \frac{\alpha+3}{2} \le i \le \alpha-1, \\ & \text{if } \alpha \equiv 3 \pmod 4, \text{ for } \frac{\alpha+3}{2} \le i \le \alpha-1, \\ 4i-2\alpha+1, & \text{if } \alpha \equiv 2 \pmod 4, \text{ for } \frac{\alpha+2}{2} \le i \le \alpha-1, \\ & \text{if } \alpha \equiv 0 \pmod 4, \text{ for } \frac{\alpha+2}{2} \le i \le \alpha-1, \end{cases}$$

$$w_f(x_\alpha x_1) = \begin{cases} \alpha+1, & \text{if } \alpha \equiv 1,2 \pmod 4, \\ \alpha+2, & \text{if } \alpha \equiv 0 \pmod 4, \\ \alpha, & \text{if } \alpha \equiv 3 \pmod 4. \end{cases}$$

$$\begin{aligned} w_f(x_{\frac{\alpha-1}{2}}x_{\frac{\alpha+1}{2}}) &= 2\alpha-2, && \text{if } \alpha \equiv 1 \pmod 4, \\ w_f(x_{\frac{\alpha}{2}}x_{\frac{\alpha+2}{2}}) &= \alpha, && \text{if } \alpha \equiv 0 \pmod 4, \\ w_f(x_{\frac{\alpha+1}{2}}x_{\frac{\alpha+3}{2}}) &= \alpha+1, && \text{if } \alpha \equiv 3 \pmod 4, \\ w_f(z_kz_{k+1}) &= 2\alpha+2k+1, && \text{for } 1 \le k \le \beta-1. \end{aligned}$$

Table 1. $k-m$ rainbow path in tadpole graph $T_{\alpha,\beta}$.

Case	k	m	rainbow path	condition
1	x_i	z_k	$x_i, x_{i-1}, x_{i-2}, \ldots, x_1, z_1, z_2, \ldots, z_{k-1}, z_k$	$i = 1, 2, \ldots, \lceil \frac{\alpha}{2} \rceil$
2	x_i	z_k	$x_i, x_{i+1}, x_{i+2}, \ldots, x_\alpha, z_1, z_2, \ldots, z_{k-1}, z_k$	$i = \lceil \frac{\alpha}{2} \rceil + 1, \lceil \frac{\alpha}{2} \rceil + 2, \ldots, \alpha$
3	z_k	z_l	$z_k, z_{k+1}, z_{k+2}, \ldots, z_{l-1}, z_l$	$k < l$, $k, l = 1, 2, \ldots, \beta$
4	x_i	x_j	$x_i, x_{i+1}, x_{i+2}, \ldots, x_{j-1}, x_j$	$i < j,\ j - i \le \lceil \frac{\alpha}{2} \rceil$ $i, j = 1, 2, \ldots, \alpha$
5	x_i	x_j	$x_j, x_{j+1}, x_{j+2}, \ldots, x_\alpha, x_1, x_2, \ldots, x_{i-1}, x_i$	$i < j,\ j - i \ge \lceil \frac{\alpha}{2} \rceil$ $i, j = 1, 2, \ldots, \alpha$

We evaluate the tadpole graph edge coloring is a rainbow antimagic connection with shown in Table 1 that every $k, m \in V(G)$ have rainbow path $k-m$ for any edges as uv and u^*v^* in path $k-m$ with edge weights $w(uv) = f(u) + f(v)$ and $w(u^*v^*) = f(u^*) + f(v^*)$

are different $w(uv) \neq w(u^*v^*)$. Therefore, the $\lceil \frac{\alpha}{2} \rceil + \beta + 1$-edge coloring of $T_{\alpha,\beta}$ is a rainbow antimagic connection, so we get $rac(T_{\alpha,\beta}) \leq \lceil \frac{\alpha}{2} \rceil + \beta + 1$. Based on the description above, it is found that $\lceil \frac{\alpha}{2} \rceil + \beta \leq rac(T_{\alpha,\beta}) \leq \lceil \frac{\alpha}{2} \rceil + \beta + 1$.

□

3. Inculcating a affine cipher

After getting rainbow antimagic coloring, we can use it to develop an affine cipher by inculcating it. In the affine cipher, the entire method relies on working for *mod* m (the length of the alphabet used in the Affine cipher). We define the steps of the affine cipher cryptosystem as follows.

- The key system source is collected from edge weights of rainbow antimagic vertex labeling of graphs
- The key length regards to edge element of graphs

3.1. Role of keystream

We use the rainbow antimagic coloring algorithm for the key To develop an affine cipher. This algorithm uses the edge weights of rainbow antimagic vertex labeling of graphs. The sequence is used as a key for encryption and decryption. By using 26 English language alphabets, the following algorithm gives a procedure of the construction.

Algorithm 1. **Role of Keystream**

1. To use the graph elements, define f
2. If f is a bijection, do 3, and bring it back to 1 otherwise
    3. To use the sequence for edge weight for every edge, define z_j where $j =$ the number of edge and $1 \leq j \leq |E(G)|$
    4. Add z_j and arrange the sequence according to the smaller vertex weights
    5. Set k as element of the z_j sequence

3.2. Encryption and Decryption Algorithm

The keystream generated by algorithm 1 is used to determine encryption and decryption of Affine cipher. The method of encryption and decryption are performed by using the following algorithm.

Algorithm 2. **Affine Cipher**

1. Given that the plaintext $P = (p_i), 1 \leq i \leq q$
2. Compute the ciphertext using Equation 1 and compute the plaintext blocks using Equation 2.

$$C_n = (P_n + K) \ mod \ 26 \quad (1)$$

$$P_n = (C_n - K) \ mod \ 26 \quad (2)$$

where P_n, K, and C_n are the n-th of plaintext, key sequence, and ciphertext, respectively.

For an illustration of how the algorithms are working, we give the following examples. Given that a plaintext $P = UNIVERSITASJEMBER$, by means of the two

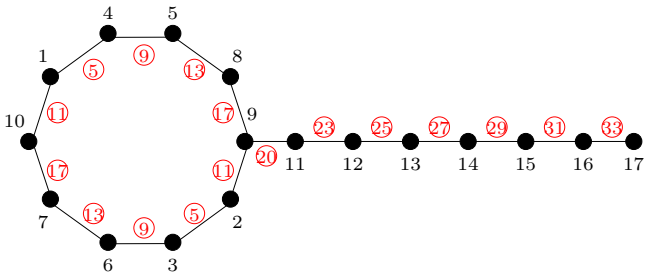


Figure 2. A tadpole graph.

algorithms above we have a ciphertext $C = ZSREPCFVKRMGDNEJY$. The cryptosystem process can be described in the following tables.

Table 2. Encryption process.

P	U	N	I	V	E	R	S	I	T	A	S	J	E	M	B	E	R
P_i	20	13	8	21	4	17	18	8	19	0	18	9	4	12	1	4	17
K_i	5	5	9	9	11	11	13	13	17	17	20	23	25	27	29	31	33
$P_i + K_i$	25	18	17	30	15	28	31	21	36	17	38	32	29	39	30	35	50
C_i	25	18	17	4	15	2	5	21	10	17	12	6	3	13	4	9	24
C	Z	S	R	E	P	C	F	V	K	R	M	G	D	N	E	J	Y

Table 3. Decryption process.

C	Z	S	R	E	P	C	F	V	K	R	M	G	D	N	E	J	Y
C_i	25	18	17	4	15	2	5	21	10	17	12	6	3	13	4	9	24
K_i	5	5	9	9	11	11	13	13	17	17	20	23	25	27	29	31	33
$C_i - K_i$	20	13	8	-5	4	-9	-8	8	-7	0	-8	-17	-22	-14	-25	-22	-9
P_i	20	13	8	21	4	17	18	8	19	0	18	9	4	12	1	4	17
P	U	N	I	V	E	R	S	I	T	A	S	J	E	M	B	E	R

4. Concluding remarks

In this paper, we have presented the result of the rainbow antimagic coloring of the tadpole graph. In addition, by using Affine cipher and inculcating affine cipher, we can develop the key efficiently. Since the decryption process will pass the keystream, i.e. by

using the rainbow antimagic coloring, we ensure that the key will be obviously hard to be revealed by any intruder. However, we need more work to make those algorithms applicable in real life, especially for the use of IOT, thus we propose the following open problems.

Open Problem 1. *Obtain programming based on the two algorithms to create a GUI program for the purpose of handling encryption and decryption process by means of rainbow antimagic coloring.*

Acknowledgement
We gratefully acknowledge the support from CGANT Research Group, and LP2M University of Jember of the year 2022.

References

[1] Chartrand G, Johns G L, McKeon K A, and Zhang P 2008 Rainbow connection in graphs *Math. Bohem.* **133**(1) pp 85-98
[2] Dafik, Nisviasari R, Maryati T K, Agustin I H, and Venkatachalam M 2021 On local super antimagic total face coloring and the application in developing a cipher block chaining key *Journal of Discrete Mathematical Sciences Cryptography* **(24)**4 1101-1111
[3] Dafik, Susanto F, Alfarisi R, Septory B J, Agustin I H, and Venkatachalam M 2021 On rainbow antimagic coloring of graphs [Submitted]
[4] Gallian J A 2019 A Dynamic Survey of Graph Labeling *The Electronic Journal of Combinatorics* pp 1-553
[5] Prihandoko A C, Dafik, Agustin I H, Susanto D, Kristiana A I, and Slamin 2016 The Construction of Encryption Key by Using a Super H-antimagic Total Graph *Program and Abstract the Asian Mathematical Conference AMC* **408** ISBN 978-602-74668-0-7
[6] Prihandoko A C, Dafik, and Agustin I H 2019 Implementation of Super H-antimagic Total Graph on Establishing Stream Cipher *Indonesian Journal of Combinatorics* **(3)**1 14-23
[7] Septory B J, Utoyo M I, Dafik, Sulistiyono B, and Agustin I H 2021 On rainbow antimagic coloring of special graphs *Journal of Physics: Conference Series* **1836** 012016
[8] Sriramoju A B 2017 Modification Affine Ciphers Algorithm for Cryptography Password *International Journal of Research In Science and Engineering* **(3)**2 346-351