Auto Transition Function in RADG

Sundus. Q. Haeeb\(^1\), Salah A. K. Albermany\(^2\)

Computer science dept., University of Kufa, Iraq, Najaf \(^1,2\)

sundus2392@gmail.com \(^1\)
Salah.albermany@uokufa.edu.iq \(^2\)

Abstract:

The main problem of RADG (Reaction Automata Directed Graph) is the fixed design of the graph. The purpose of this paper is to develop the transition function from fixed design to dynamic design, the dynamic design based on a key between sender and receiver in addition to some rules, the new design is more efficient and fairly distributed between standard states and reaction states, that design keeps the main aspect of RADG which gives more than one ciphertext to the same plaintext and adds high level of security to the wireless network.

Keywords: RADG, Cryptography, Network and Transition function

I. Introduction:

Cryptography is the science which protect network and readable data into a form which cannot be understood when transmission through wireless. The high growth in the networking technology leads a mutual civilization for interchanging of the data very radically. Cryptography is used to ensure that the contents of a message should have confidentiality, authentication, integrity, non-repudiation and access control. The Sensitive information need to protect while transmitting it, like credit cards and social security numbers [1-6].

Network Security is the most important component in security because it generally concentrates on protection of information passing through insecure network and prevents unauthorized users from accessing to confidential information, which plays important rule to provide the security of the wireless network [5,7-10], Wireless communications are becoming widespread in offices, home and enterprises, that provide high speed and quality information exchange [11]

Reaction Automata Direct Graph(RADG), which is based on automata directed graph and reaction state, the main design of RADG depended on fixed design between sender and receiver and that can be considered as a main problem of RADG [12-14].

This paper is using a novel keyless security schema RADG which is based on Reaction Automata Direct Graph [12], new trend in security that creates auto transition function to move between different states of RADG. The problem discussed in this paper is changing the graph from static design into dynamic design using key between sender and receiver in addition to some different rules.
II. Reaction Automata Directed Graph (RADG)

Reaction Automata Directed Graph (RADG), depends on automated directed graph and reaction state. The novelty fact of the first RADG that does not require any key to perform encryption and decryption operation, there is no argument between user to communicate. Modeling of the first RADG is influenced by graph theory, where ADG is denoted for automata encryption and R (reaction states), RADG design can be represented by a sextuple as \( \{Q, R, \Psi, \Sigma, J, T, \lambda\} \), such that Q represents standard states, R represents reaction states, \( \Sigma \) stands for non-empty finite set of input data, \( \Psi \) represents output transitions, J represents non-empty finite set called jump is subset of Q , T represents transition function and \( \lambda \) represents the number of value in each state, except jump states which do not take any value . RADG encryption and decryption are based on transition between state the RADG design depended on the value of \( n, m, k \) and \( \lambda \), where \( n = |Q|, m = |R| \) and \( k = |J| \). The encryption will start from Q state if Q state goes to J state then J is given random address to any state in R to encrypt the input then return back to Q state to continue in encryption process until all values in message is used [12].

Example (1):

If \( n = 6, m=2, k=2 \) and suppose \( \lambda=3 \) number of value in each state then the transition among states as shown in Figure (1). \( \lambda=3 \) this means each state has three values, and the first value of each state is address of a state, then parameters of R, Q, J, \( \Sigma \), and \( \Psi \) are set as \( R = \{4,5\} \), \( Q = \{0,1,2,3\} \), \( J = \{6,7\} \). Input data set as \( \Sigma = \{0,1,2\} \), and output transitions \( \Psi = \{20,11,12,13,14,15,16,\ldots,37\} \).

Suppose the original message to be encrypted using RADG is [011002]. Transition function as T (address of state , bit of message )

![Figure (1) RADG Design in Example (1)](image-url)
III. Auto-Transition function

The main purpose of auto transition function is to generate dynamic design for the RADG (Reaction Automata Directed Graph), the transition between states in the dynamic design based on mathematical rules and key of size 128 bit between sender and receiver, {n, m, k, λ, w}, can be considered as a key of dynamic design divided as {42 bit, 34 bit, 24 bit, 4bit and 24 bit}, Auto-Design can be represented by sextuple as \{Q, R, Ψ, Σ, J, T, λ, d, w, s_i, v_i\} such that d represents the index value of state, w represents the number of state move to jump, \(s_i\) represents the address of states and \(v_i\) represents values of states, see algorithm(1) can explain how the Auto function works.

Algorithm (1): Auto-Transition Design Between State
Input: key of size 128 bit can be divided as below
n: Size of the set Q, 42 bit
m: Size of the set R, 34 bit
k: Size of jump state, 24 bit
λ: The number of value in each state, 4 bit
w: The number of transition to jump state, 24 bit
Output: RADG Design \(f_i\)
Begin

1- Compute the number of value for all state except jump state \(τ=λ(n+m-k)\).
2- Compute the index value of state \(d_i = λs_i + v_i\), where \(i = 1,2,...,τ\), \(s_i\) is the address of state it value 0,1,...,n+m-k-1 and \(v_i\) is the value of state its value 0,1,...,λ.
3- Calculate the index of state that move jump state

\[
J_j = \left\lfloor \frac{j(λ(n-k))}{w+1} \right\rfloor . \text{where } j = 1.2,...,w
\]
4- Compute $f_i(s_i, v_i)$ is the transition between state by using the rules below to get RADG design

$$f_i(s_i, v_i) =$$

$$n + m - k + j - 1$$

$$\begin{cases} 
\left(2s_i + v_i + \left\lfloor \frac{s_i}{\lambda} \right\rfloor + \left\lfloor \frac{d_i}{(n-k)} \right\rfloor + \lambda \right) \mod (n-k) \\
\text{where } \left(2s_i + v_i + \left\lfloor \frac{s_i}{\lambda} \right\rfloor + \left\lfloor \frac{d_i}{(n-k)} \right\rfloor \right) \mod (n-k) = s_i \text{ and } s_i \leq (n-k) \\
\end{cases}$$

$$\begin{cases} 
\left(2s_i + v_i + \left\lceil \frac{s_i}{\lambda} \right\rceil + \left\lceil \frac{d_i}{(n-k)} \right\rceil \right) \mod (n-k) \\
\text{other wise} \\
\end{cases}$$

End

Note: $[x]$ mean round the number to nearest integer number if $x=1.6$ then $[x]=2$, $x=1.2$ then $[x]=1$

Example (2):

If $n = 14$, $m = 4$, $k = 3$, $w = 3$ and suppose $\lambda = 3$ mean that every state has three value then the transition among states by using the above algorithm shown in Table (2) and Figure (2). Then the first component of the function is address of a state and second component of the function is the values of states.

1- $\tau = 45$

2- $J_f = \{8, 16, 24\}$, $J_1 = 8$, $J_2 = 16$, $J_3 = 24$.

3- Compute transition to get Auto-Design of RADG

| $d_i$ | $f_i(s_i, v_i)$ |
|---|---|
| $d_1 = 0$ | $f_1(0.0) = \left(2 \ast 0 + 0 + \left\lfloor \frac{0}{3} \right\rfloor + \left\lfloor \frac{0}{11} \right\rfloor \right) \mod 11 = 0 + 3 = 3$ , second component of $f(s_i, v_i)$ |
| $d_2 = 1$ | $f_2(0.1) = \left(2 \ast 0 + 1 + \left\lfloor \frac{0}{3} \right\rfloor + \left\lceil \frac{1}{11} \right\rceil \right) \mod 11 = 1$ , third component of $f(s_i, v_i)$ |
| $d_3 = 2$ | $f_3(0.2) = \left(2 \ast 0 + 2 + \left\lfloor \frac{0}{3} \right\rfloor + \left\lceil \frac{2}{11} \right\rceil \right) \mod 11 = 2$ , third component of $f(s_i, v_i)$ |
| $d_4 = 3$ | $f_4(1.0) = \left(2 \ast 1 + 0 + \left\lceil \frac{1}{3} \right\rceil + \left\lceil \frac{2}{11} \right\rceil \right) \mod 11 = 2$ , third component of $f(s_i, v_i)$ |
| $d_5 = 4$ | $f_5(1.1) = \left(2 \ast 1 + 1 + \left\lceil \frac{1}{3} \right\rceil + \left\lceil \frac{4}{11} \right\rceil \right) \mod 11 = 3$ , third component of $f(s_i, v_i)$ |
\[
\begin{array}{|c|c|}
\hline
i & f_i(s_i, v_i) \\
\hline
1 & f_1(3.1) = \left( 2 \times 3 + 0 + \frac{3}{3} + \frac{2}{11} \right) \mod 11 = 9, \text{ first component of } f_i(s_i, v_i) \\
2 & f_2(1.2) = \left( 2 \times 1 + 2 + \frac{1}{3} + \frac{5}{11} \right) \mod 11 = 4, \text{ third component of } f_i(s_i, v_i) \\
3 & f_3(2.0) = \left( 2 \times 2 + 0 + \frac{2}{3} + \frac{6}{11} \right) \mod 11 = 6, \text{ third component of } f_i(s_i, v_i) \\
4 & f_4(2.1) = \left( 2 \times 2 + 1 + \frac{2}{3} + \frac{7}{11} \right) \mod 11 = 7, \text{ third component of } f_i(s_i, v_i) \\
5 & f_5(2.2) = 14 + 4 - 3 + 1 - 1 = 15, \text{ first component of } f_i(s_i, v_i) \\
6 & d_6 = 5 \\
7 & d_7 = 6 \\
8 & d_8 = 7 \\
9 & d_9 = 8 \\
10 & f_9(3.2) = \left( 2 \times 3 + 3 + \frac{3}{3} + \frac{11}{11} \right) \mod 11 = 10, \text{ third component of } f_i(s_i, v_i) \\
11 & f_10(3.1) = \left( 2 \times 3 + 1 + \frac{3}{3} + \frac{10}{11} \right) \mod 11 = 9, \text{ third component of } f_i(s_i, v_i) \\
12 & d_{12} = 11 \\
13 & f_12(3.2) = \left( 2 \times 3 + 3 + \frac{3}{3} + \frac{11}{11} \right) \mod 11 = 10, \text{ third component of } f_i(s_i, v_i) \\
14 & f_13(4.0) = \left( 2 \times 4 + 0 + \frac{4}{3} + \frac{12}{11} \right) \mod 11 = 10, \text{ third component of } f_i(s_i, v_i) \\
15 & f_14(4.1) = \left( 2 \times 4 + 1 + \frac{4}{3} + \frac{13}{11} \right) \mod 11 = 0, \text{ third component of } f_i(s_i, v_i) \\
16 & f_15(4.2) = \left( 2 \times 4 + 2 + \frac{4}{3} + \frac{14}{11} \right) \mod 11 = 1, \text{ third component of } f_i(s_i, v_i) \\
17 & f_16(5.0) = \left( 2 \times 5 + 0 + \frac{5}{3} + \frac{15}{11} \right) \mod 11 = 2, \text{ third component of } f_i(s_i, v_i) \\
18 & d_{18} = 17 \\
19 & f_18(5.2) = \left( 2 \times 5 + 2 + \frac{5}{3} + \frac{17}{11} \right) \mod 11 = 5 + 3 = 8, \text{ second component of } f_i(s_i, v_i) \\
20 & f_19(6.0) = \left( 2 \times 6 + 0 + \frac{6}{3} + \frac{18}{11} \right) \mod 11 = 5, \text{ third component of } f_i(s_i, v_i) \\
21 & f_20(6.1) = \left( 2 \times 6 + 1 + \frac{6}{3} + \frac{19}{11} \right) \mod 11 = 6 + 3 = 9, \text{ second component of } f_i(s_i, v_i) \\
22 & f_21(6.2) = \left( 2 \times 6 + 2 + \frac{6}{3} + \frac{20}{11} \right) \mod 11 = 7, \text{ third component of } f_i(s_i, v_i) \\
23 & f_22(7.0) = \left( 2 \times 7 + 0 + \frac{7}{3} + \frac{21}{11} \right) \mod 11 = 7 + 3 = 10, \text{ second component of } f_i(s_i, v_i) \\
24 & f_23(7.1) = \left( 2 \times 7 + 1 + \frac{7}{3} + \frac{22}{11} \right) \mod 11 = 8, \text{ third component of } f_i(s_i, v_i) \\
25 & f_24(7.2) = \left( 2 \times 7 + 2 + \frac{7}{3} + \frac{23}{11} \right) \mod 11 = 9, \text{ third component of } f_i(s_i, v_i) \\
26 & d_{25} = 24 \\
27 & f_25(8.0) = 14 + 4 - 3 + 3 - 1 = 17, \text{ first component of } f_i(s_i, v_i) \\
28 & f_26(8.1) = \left( 2 \times 8 + 1 + \frac{8}{3} + \frac{25}{11} \right) \mod 11 = 0, \text{ third component of } f_i(s_i, v_i) \\
29 & f_27(8.2) = \left( 2 \times 8 + 2 + \frac{8}{3} + \frac{26}{11} \right) \mod 11 = 1, \text{ third component of } f_i(s_i, v_i) \\
30 & f_28(9.0) = \left( 2 \times 9 + 0 + \frac{9}{3} + \frac{27}{11} \right) \mod 11 = 1, \text{ third component of } f_i(s_i, v_i) \\
\hline
\end{array}
\]
\[
\begin{array}{|c|c|}
\hline
d_{29} = 28 & f_{29}(9.1) = \left(2 * 9 + 1 + \frac{9}{3} + \frac{28}{11}\right) \mod 11 = 3, \text{ third component of } f_i(s_i, v_i) \\
d_{30} = 29 & f_{30}(9.2) = \left(2 * 9 + 2 + \frac{9}{3} + \frac{29}{11}\right) \mod 11 = 4, \text{ third component of } f_i(s_i, v_i) \\
d_{31} = 30 & f_{31}(10.0) = \left(2 * 10 + 0 + \frac{10}{3} + \frac{30}{11}\right) \mod 11 = 4, \text{ third component of } f_i(s_i, v_i) \\
d_{32} = 31 & f_{32}(10.1) = \left(2 * 10 + 1 + \frac{10}{3} + \frac{31}{11}\right) \mod 11 = 5, \text{ third component of } f_i(s_i, v_i) \\
d_{33} = 32 & f_{33}(10.2) = \left(2 * 10 + 2 + \frac{10}{3} + \frac{32}{11}\right) \mod 11 = 6, \text{ third component of } f_i(s_i, v_i) \\
d_{34} = 33 & f_{34}(11.0) = \left(2 * 11 + 0 + \frac{11}{3} + \frac{33}{11}\right) \mod 11 = 7, \text{ third component of } f_i(s_i, v_i) \\
d_{35} = 34 & f_{35}(11.1) = \left(2 * 11 + 1 + \frac{11}{3} + \frac{34}{11}\right) \mod 11 = 8, \text{ third component of } f_i(s_i, v_i) \\
d_{36} = 35 & f_{36}(11.2) = \left(2 * 11 + 2 + \frac{11}{3} + \frac{35}{11}\right) \mod 11 = 9, \text{ third component of } f_i(s_i, v_i) \\
d_{37} = 36 & f_{37}(12.0) = \left(2 * 12 + 0 + \frac{12}{3} + \frac{36}{11}\right) \mod 11 = 9, \text{ third component of } f_i(s_i, v_i) \\
d_{38} = 37 & f_{38}(12.1) = \left(2 * 12 + 1 + \frac{12}{3} + \frac{37}{11}\right) \mod 11 = 10, \text{ third component of } f_i(s_i, v_i) \\
d_{39} = 38 & f_{39}(12.2) = \left(2 * 12 + 2 + \frac{12}{3} + \frac{38}{11}\right) \mod 11 = 0, \text{ third component of } f_i(s_i, v_i) \\
d_{40} = 39 & f_{40}(13.0) = \left(2 * 13 + 0 + \frac{13}{3} + \frac{39}{11}\right) \mod 11 = 1, \text{ third component of } f_i(s_i, v_i) \\
d_{41} = 40 & f_{41}(13.1) = \left(2 * 13 + 1 + \frac{13}{3} + \frac{40}{11}\right) \mod 11 = 2, \text{ third component of } f_i(s_i, v_i) \\
d_{42} = 41 & f_{42}(13.2) = \left(2 * 13 + 2 + \frac{13}{3} + \frac{41}{11}\right) \mod 11 = 3, \text{ third component of } f_i(s_i, v_i) \\
d_{43} = 42 & f_{43}(14.0) = \left(2 * 14 + 0 + \frac{14}{3} + \frac{42}{11}\right) \mod 11 = 4, \text{ third component of } f_i(s_i, v_i) \\
d_{44} = 43 & f_{44}(14.1) = \left(2 * 14 + 1 + \frac{14}{3} + \frac{43}{11}\right) \mod 11 = 5, \text{ third component of } f_i(s_i, v_i) \\
d_{45} = 44 & f_{45}(14.2) = \left(2 * 14 + 2 + \frac{14}{3} + \frac{44}{11}\right) \mod 11 = 6, \text{ third component of } f_i(s_i, v_i) \\
\hline
\end{array}
\]
Figure (2): Auto Transition Between State of RADG
IV. Measure the Efficiency of Auto Transition Function

The following algorithm can be used to measure the efficiency of Auto-Transition function (Auto Design of RADG) to ensure that it achieves the fair distribution among states based on ratio balance, that will give more than one method to Auto transition but choose the more efficient based on proposed efficiency algorithm. See algorithm (2).

Algorithm (2): Compute Efficiency of Auto-Transition Function

**Input:**
- $n$: Size of the set $Q$, 42 bit
- $m$: Size of the set $R$, 34 bit
- $k$: Size of jump state, 24 bit
- $\lambda$: The number of value in each state, 4 bit
- $w$: The number of transition to jump state, 24 bit

**Output:** Efficiency of the Auto Design of RADG

**Begin**

1. $f_i$ : Auto Design of RADG State see algorithm (1)
2. Compute $c_i$ is the number of internal transition to each state (incoming)
3. Compute the ratio balance value $r_b$ is the number of transition that each state must receive $r_b = \frac{\tau - w}{(n-k)}$. where $\tau = (\lambda(n+m-k))$
4. Compute ceiling and floor $c_u$ and $c_f$ for the $r_b$
5. Compute the two vector $k_u$ and $k_f$ by using the absolute different between step3 and step6
6. Find the smallest value between the two vector $k_u$ and $k_f$ and save in vector $s_v$
7. Compute $E$ efficiency of auto transition function
   \[
   E = \frac{\sum_{i=1}^{n-k} s_v}{n-k} \]

**End**

Example (3):

Take the result of example (2) to measure efficiency it achieves a fair distribution among states where the result of auto design is nearest to zero, it is more efficient than another Design. See figure below which shows different efficiencies for different auto transition functions.

1. $f_i = \{3,1,2,2,3,4,6,7,15,8,9,10,10,0,1,2,16,8,6,9,7,10,8,9,17,0,1,1,3,4,4,5,6,7,8,9,9,10,0,1,2,3,4,5,6\}$
2. $c_i = \{3,5,4,4,4,3,3,4,5,4\}$, transition to each state.
3- $r_b = 3.8182$
4- $c_u = 4$
5- $c_f = 3$
6- $k_u = \{1,1,0,0,0,1,1,0,1,0\}$
7- $k_f = \{0,2,1,1,0,0,0,1,2,1\}$
8- $s_v = \{0,1,0,0,0,0,0,0,1,2\}$
9- $E = 0.1818$

When take the key $n = 14$, $m=4$, $k=3$, $w=3$ and $\lambda=3$ and apply auto transition function figure (3), this will show the number of transition of each state received when ratio balance=3.81818 which means some state must have 3 transitions and other have 4 transitions, except some states which may have less or more determined ratio. The efficiency of the above mentioned key is 0.1818.

![Figure (3): Show Efficiency when n = 14, m=4, k=3, w=3 and \lambda=3](image)

When take the key $n = 100$, $m=20$, $k=15$, $w=3$ and $\lambda=15$ and apply auto transition function figure (4), this will show the number of transition of each state received when ratio balance= 3.5294 which means some states must have 3 transitions and other have 4 transitions, except some states which may have less or more than determined ratio. The efficiency of the above mentioned key is 0.4588.

![Figure (4): Show Efficiency when n = 100, m=20, k=15, w=3 and \lambda=15](image)
When take the key $n = 250$, $m=50$, $k=30$, $w=3$ and $\lambda=30$ and apply auto transition function figure (5), this will show the number of transitions of each state received when ratio balance= 3.5455 which means some states must have 3 transitions and other have 4 transitions and must state between 3 and 4, except some states which may have less or more than determine ratio. The efficiency of the above mentioned key is 0.2727.

Figure (5): Show Efficiency when $n = 250$, $m=50$, $k=30$, $w=3$ and $\lambda=30$

When take the key $n = 500$, $m=100$, $k=50$, $w=3$ and $\lambda=50$ and apply auto transition function figure (6), this will show the number of transitions of each state received when ratio balance= 3.5556 which means some states must have 3 transitions and other have 4 transitions and must state between 3 and 4, except some states which may have less or more than determine ratio. The efficiency of the above mentioned key is 0.2822.

Figure (6): Show Efficiency when $n = 500$, $m=100$, $k=50$, $w=3$ and $\lambda=50$

When take the key $n = 1000$, $m=200$, $k=100$, $w=3$ and $\lambda=100$ and apply auto transition function figure (7), this will show the number of transitions of each state received when ratio balance=3.5556 which means some states must have 3 transitions and other have 4 transitions and must states between 3 and 4, except some states which may have less or more than determine ratio. The efficiency of the above mentioned key is 0.2289.
When take the key \( n = 2000, m=500, k=200, w=3 \) and \( \lambda=200 \) and apply auto transition function figure (8), this will show the number of transitions of each state received when ratio balance=3.7222 which means some states must have 3 transitions and other have 4 transitions and must states between 3 and 4, except some states which may have less or more than determine ratio. The efficiency of the above mentioned key is 0.3944.

Another example can be taken to see efficiency when the Auto-transition function algorithm and the key are changed to ensure that proposed algorithm is efficiency. there are many auto transition functions which are shown in the table below but the best one is chosen. The figure (9) shows clearly that the nearest function to zero is the most efficient one and it is chosen.

| \( n, m, k, \lambda \) and \( \omega \) | \( (s_i + v_i + \lambda) \mod (n-k) \) | \( (s_i + v_i + v_i) \mod (n-k) \) | \( (s_i + v_i + s_i) \mod (n-k) \) |
|---|---|---|---|
| 14,4,3,3,3 | E=1.1111 | E=0.6667 | E=0.6667 |
Table (2): Efficiency of Different Example and Function

| n, m, k, λ and ω | (s_i + v_i + \frac{v_{i+1}}{λ}) \mod (n-k) | (s_i + v_i + d_i) \mod (n-k) | (s_i + v_i + \frac{d_i}{n-k}) \mod (n-k) |
|-----------------|---------------------------------|-----------------------------|---------------------------------|
| 14,4,3,3,3      | E=1.1111                        | E=0.6667                    | E=0.4444                        |
| 100,20,15,3,15  | E=0.5529                        | E=0.3176                    | E=0.6000                        |
| 250,50,30,3,30  | E=0.5227                        | E=3.0227                    | E=0.5545                        |
| 500,100,50,3,50 | E=0.5022                        | E=3.0511                    | E=0.5844                        |
| 1000,200,100,3,100 | E=0.5044                     | E=3.0533                    | E=0.5956                        |
| 2000,500,200,3,200 | E=0.6044                     | E=3.2211                    | E=0.5644                        |

Table (3): Efficiency of Different Example and Function

| n, m, k, λ and ω | \left( s_i + v_i + \left\lfloor \frac{v_{i+2}}{λ} \right\rfloor \right) \mod (n-k) | \left( s_i + v_i + \left\lfloor \frac{v_{i+2}}{λ} \right\rfloor \right) \mod (n-k) | \left( s_i + v_i + \left\lfloor \frac{d_i}{n-k} \right\rfloor \mod (n-k) |
|-----------------|-------------------------------------------------|-------------------------------------------------|---------------------------------|
| 14,4,3,3,3      | E=1.1111                        | E=0.8889                        | E=0.4444                        |
| 100,20,15,3,15  | E=1.1529                        | E=0.5294                        | E=0.6235                        |
| 250,50,30,3,30  | E=2.0591                        | E=0.5182                        | E=0.5545                        |
| 500,100,50,3,50 | E=2.1511                        | E=0.5000                        | E=0.5200                        |
| 1000,200,100,3,100 | E=2.1300                     | E=0.5033                        | E=0.5133                        |
| 2000,500,200,3,200 | E=2.1872                     | E=0.6033                        | E=0.6078                        |

Table (4): Efficiency of Different Example and Function

| n, m, k, λ and ω | \left( 2 \cdot s_i + v_i + \left\lfloor \frac{s_i}{λ} \right\rfloor + \frac{d_i}{(n-k)} \right) \mod (n-k) | \left( s_i + v_i + \left\lfloor \frac{s_i}{λ} \right\rfloor + λ + \frac{d_i}{(n-k)} \right) \mod (n-k) |
|-----------------|-----------------------------------------------------------------|-----------------------------------------------------------------|
| 14,4,3,3,3      | E=0.1818                                                        | E=0.3636                                                        |
| 100,20,15,3,15  | E=0.4588                                                        | E=0.5294                                                        |
| States          | Efficiency 1 | Efficiency 2 |
|-----------------|--------------|--------------|
| 250,50,30,3,30  | E=0.2727     | E=0.5500     |
| 500,100,50,3,50 | E=0.2822     | E=0.5933     |
| 1000,200,100,3,100 | E=0.2289   | E=0.5789     |
| 2000,500,200,3,200 | E=0.3944   | E=0.5539     |

Figure (9): Show Efficiency of all function with different number of states

Conclusion:

The aim of this paper is to generate a dynamic design of the RADG, a set of functions is suggested and the most efficient function is chosen based on the efficiency algorithm when closes to zero. Some functions cannot achieve fair distribution among states of RADG when the number of states increases and other functions are nearest to fair distribution which is the best one and chosen. the aim of this paper is achieved.
Reference

1- Atul Kahate “Cryptography and Network Security”, Tata McGraw-Hill Companies, 2008.
2- William Stallings “Network Security Essentials (Applications and Standards)”, Pearson Education, 2004.
3- AbuTaha, M., Farajallah, M., Tahboub, R., & Odeh, M. (2011). Survey paper: cryptography is the science of information security.
4- J.R Childs: " General Solution of the ADFGVX Cipher System ". Aegean Park Press, (2000), USA.
5- Joshi, Mukund R., and Renuka Avinash Karkade. "Network security with cryptography." International Journal of Computer Science and Mobile Computing 4.1 (2015): 201-204.
6- Goyal, S. (2012). A Survey on the Applications of Cryptography. International Journal of Science and Technology, I(3).
7- Russell, S. F. (2001, April). Wireless network security for users. In Proceedings International Conference on Information Technology: Coding and Computing (pp. 172-177). IEEE.
8- Williams, T. C. (2001). U.S. Patent No. 6,304,973. Washington, DC: U.S. Patent and Trademark Office.
9- Stilp, L. A. (2006). U.S. Patent No. 7,084,756. Washington, DC: U.S. Patent and Trademark Office.
10- Mitali, V. K., & Sharma, A. (2014). A survey on various cryptography techniques. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 3(4), 307-312.
11- Singh, R. K., & Tiwari, N. (2015). An investigation on wireless mobile network and wireless lan (wi-fi) for performance evaluation. International Journal of Computer Applications, 126(6).
12- Albermany, S. A., & Safdar, G. A. (2014). Keyless security in wireless networks. Wireless personal communications, 79(3), 1713-1731.
13- Albermany, Salah A., and Ali Hasan Alwan. "RADG design on Elliptic Curve Cryptography." ICCIIDT Confer UK, 11-12 October 2016.
14- Albermany, Salah AK, Fatima Radi Hamade, and Ghazanfar Ali Safdar. "New random block cipher algorithm." 2017 International Conference on Current Research in Computer Science and Information Technology (ICCIT). IEEE, 2017.