RADG Modification with Different Values of States

Mudher Muhammad Salih1*, Yahya Mourad Abdul – Abbass2
1 Ministry of Education, Educational Directorate Najaf AlAshraf, Iraq
2 Ministry of Education Educational Directorate Babylon, Iraq
1 modhern@gmail.com
2 yihiamor@yahoo.com

Abstract.
Cryptography may be a science of changing messages into unclear kind through the employment of encoding algorithms that is an algorithm that converts the text to a cypher and increased by arithmetic that shield information in powerful ways. A novel keyless security theme Reaction Automata Direct Graph (RADG), that relies on automata direct graph and reaction states. The novelty of RADG lies within the incontrovertible fact that it doesn’t need any key to perform the cryptographically operations therefore creating it a possible scheme for big wireless systems. Chaotic maps represent the backgrounds for chaos based mostly cryptography thanks to the actual fact that chaotic transformations are sensitive to changes within the initial conditions and generate solutions similar to random ones. In this paper a proposed modification to RADG will be performed. The main problem of the RADG is that the values in the states are static. In this paper this problem will be solved using chaos logistic map. The RADG which is used in this paper is key not keyless (by using chaos logistic map).

1. Introduction
Spread of digital info forms dramatically, utilized by several sectors similar to banks, financial markets or E-Commerce so this spreading need high security level against malicious activities, the knowledge technology a rise dramatically everywhere the world. The knowledge sent from one device to another across sure network. This network could also be insecure wireless network and this result in the requirement to shield wireless networks from violation. The main points achieve this, cryptography algorithms [1]. Cryptography could be a research field of arithmetic and engineering science involved with techniques to secure the transfer of messages between 2 parties. It's perpetually assumed that a third party, usually known as adversary, will hear the channel carrying such a message and consequently will acquire the message. Preventing such opponent from understanding the message is achieved by applying a cryptographic mechanism known as secret writing algorithmic rule employing a piece of information that's solely understand to the act parties and expressly unknown to the adversary. This piece of data is termed the key of the encryption algorithmic rule and also the algorithm alongside the key's known as a cryptosystem [2]. Chaos could be a quite science that deals with elements of the planet that are unpredictable, apparently random, not
essentially random, disorderly aural and irregular misbehaved. A class of models is known which will represent the two-phase microfluidic flow in numerous experimental conditions [3] [4] [5].The logistic map is a one-dimensional discrete-time nonlinear system exhibiting quadratic non-linearity. The logistic map is given by the function $f$: $[0, 1] \rightarrow \mathbb{R}$ defined by [6]:

$$f(x) = \mu x (1 - x) \quad (1-1)$$

which is expressed in state equation form as:

$$x_{n+1} = f(x_n) = \mu x_n (1 - x_n) \quad n=0,1,2, \ldots, \quad (1-2)$$

where $x_n \in (0, 1)$ and $\mu \in (0, 4)$. $\mu$ is known as the control parameter or bifurcation parameter. Here $x_n$ is the state of the system at time $n$. $x_{n+1}$ denotes the next state and $n$ denotes the discrete time. Repeated iteration of $f$ gives rise to a sequence of points $\{x_n\} \rightarrow [6]$.

RADG (reaction automata direct graph) is a combination of automata direct graph and reaction states, RADG doesn’t need key exchange or agreement between users. RADG can be represented by a sextuple as $(Q; R; \Sigma; \Psi; J; T)$, such that $Q$ stands for a set of standard states, $R$ stands for a set of reaction states, $\Sigma$ stands for a set of input data, $\Psi$ stands for a set of output transitions, $J$ stands for a set (which is subset of $Q$ called jump states) and $T$ represents transition function.

Each state has $\lambda$ values. RADG ciphering depends on relation between states. On the other hand, the design of RADG depends on $m, n, k,$ and $\lambda$ where $n = |Q|, m = |R|, k$ is the number of jump states and $\lambda$ representing the number of values in each state. Encryption of the message starts with a random state in the $Q$ cross referencing the message incoming with the value from state chosen continuing until a jump state occurs moving to $R$ set and choosing a random state from it then go back to the $Q$ set using the fixed transitions in the design. The example below helps understanding RADG Design.

**Example:** If $n = 5; k = 1; m = 2$ and $\lambda = 2$ number of value in each state then the transition design is in the figure below, if $\lambda = 2$, then $\Sigma = \{0, 1\}$, number of states in $R$ (reaction states) is only one state according to $m = 2$, number of jump states is $1$ according $k = 1$ and number of states in $Q$ (standard states) is $5$ according to $n = 5$, and $\Psi = \{11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24\}$.

Suppose the original message to be encrypted using RADG is 1110. Transition function as $T$ (address of state, bit of message). Thus we have $T (0,1) = (3,14), T (3,1) = (2,16), T (2,1) = (1,18), T (1,0) = (3,11)$. The corresponding output is 14, 16, 18, 11 respectively [2].
The following steps are to explain the encryption using RADG.

Step1: $T(0,1) = (3,14)$
- $T(0,1)$: transition function containing state 0 to encrypt message 1
- $(3,14)$: this means message 1 gives 14 as cipher and moves to state 3 by transition drawing from 14 in state 0 to state 3

Step2: $T(3,1) = (2,16)$, in step1 the next state after encryption process is 3, and the next bit in message is 1. Then:
- $T(3,1)$: transition function contain state 3 to encrypt message 1
- $(2,16)$: this means message 1 gives 16 as cipher and move to state 2 by transition drawing from 16 in state 3 to state 2.

Step3: $T(2,1) = (1,18)$, in step2 the next state after encryption process is 2, and the next bit in message is 1. Then:
- $T(2,1)$: transition function contains state 2 to encrypt message 1
- $(1,18)$: this means message 1 gives 18 as cipher and move to state 1 by transition drawing from 16 in state 3 to state 2.

Step4: $T(1,0) = (3,11)$. in step3 the next state after encryption process is 1, and the next bit in message is 0. Then:
- $T(1,0)$: transition function contain state 1 to encrypt message 0
- $(3,11)$: this means message 1 gives 18 as cipher and move to state 1 by transition drawing from 11 in state 1 to state 3.
2. Related Work

In [7] they illustrate different types of ciphering based on the original message to achieve the security operation as integrity, confidentiality, authentication and nonrepudiation. Mathematical modeling of RADG scheme is based on Automata Graph (ADG) and reaction states (R). RADG method influenced by graph theory where ADG is the main states for automata encryption, while reaction state is to forward increase and reduce the random expectation. RADG proves soundness of the ciphering where the process of breaking the code within large system requires significant effort compared to the schemes of classical cryptography.

In [8] RADG algorithm is improved using RSA algorithm as a transition function and changing the ciphertext into a point on the elliptic curve with the applied pseudorandom generated key.

In [9] they present a new Random block cipher method based on develop the keyless security of RADG Method called BRADG (Block Cipher Reaction Automata Direct Graph) algorithm. BRADG method based on the same properties of the RADG method, 64-bit key length and 64-bits plain text and Cipher text. BRADG method depend on develop the same concepts of DES algorithm where based on bijective Function, generates a new s-box from DES s-box and bitwise operations. The researchers presented the new algorithm to develop the wireless network where the RADG method apply on peer to peer network and the proposed method enabled more than one user to connect on the wireless network at the same time.

In [2] they proposed work presents a novel chaos-based design on the concept of automata direct graph (ADG) and reaction (R) states. The focal points in the science of data encryption reduce the availability of information to the cryptanalyst that’s why a secret key was used in both proposed methods CRADG (Chaotic reaction automata direct graph) and Improved CRADG.

3. RADG modification with different value in states

The classical RADG (Reaction Automata Direct Graph) design depends on the constant value in each state in $Q/J$ and $R$ present by $\lambda$, but where distinct values of states, must $\lambda=2^x$, with $x$ represent the size of input block such that $x \geq 1$, if $x = 1$ for each states represents classical RADG model design. The number of values in RADG design, it’s summation of variant $\lambda$’s of state, it’s can be calculated by

$$\tau = \sum_{i=0}^{n+m-k-1} \lambda_i$$

(3-1)

where $\lambda_i$ the number of values in state of address $i$, $n$ (size of standard states set $Q$), $m$ (size reaction states set $R$), and $k$ (size jump states set $J$), this means $n = |Q|$, $m = |R|$ and $k = |J|$. in this model suppose that $\lambda_i = 2^x$, $x_i \geq 1, x_i \in \{1,2,\ldots,d\}$, $i = 0,1,\ldots,n+m-k-1$ $\mu = n + m - k$ $\rho = \left\lfloor \frac{\mu}{d} \right\rfloor$

Where $d$ the number types values in states, and
\[ x_i = 1 + i \mod (d) \]  \hspace{1cm} (3-2)

Where \( i \mod (d) \) is a remainder of divided \( i \) over \( d \), then from (3-1) and (3-2), the number of values in design is

\[
\tau = \sum_{i=0}^{\mu-1} 2^{1+i \mod (d)} = 2 \sum_{i=0}^{\mu-1} 2^{i \mod (d)}
\]

Then

\[
\tau = 2 \sum_{i=0}^{\mu-1} 2^{i \mod (d)} = 2 \{ 2^0 + 2^1 + \ldots + 2^{(\mu-1) \mod (d)} \} = 2 \{ \mu (2^d - 1) + T(\mu) \}
\]

Where

\[
T(\mu) = \begin{cases} 
0 & \mu \mod (d) = 0 \\
\sum_{l=1}^{\mu \mod (d)} 2^{l-1} & \mu \mod (d) > 0 
\end{cases}
\]

\[
\tau = \begin{cases} 
\rho 2^{d+1} - 2\rho & \mu \mod (d) = 0 \\
\rho 2^{d+1} + 2v+1 - 2(\rho + 1) & \mu \mod (d) > 0 
\end{cases}
\]  \hspace{1cm} (3-3)

Where \( \nu = \mu \mod (d) \)

Then number of bits in RADG design is (denoted by \( \tau_B \)):

\[
\tau_B = \lceil \log_2 \tau \rceil
\]  \hspace{1cm} (3-4)

**Example (3-1):**

Let \( n=7, m=5, k=2, \) and \( d=3 \), then \( \mu=10, \rho=3, \nu=0, \tau=44, \) and \( \tau_B=6 \).

The values of states in RADG design are selected from the chaos sequence, that convert to distinct integer sequence \( v_0, v_1, \ldots, v_{\tau-1} \) between 0, and \( \tau-1 \), with initial value \( x_0 \) of chaos function map (logistic map, tent map, ...).

From above example when \( x_0=0.62, \) and \( r=3.71 \), for logistic map

\[
x_j = r x_{j-1} (1 - x_{j-1}),
\]

\[
d_j = \lceil \tau x_j \rceil
\]

Then creation distinct integer sequence \( d_j \) from the sequence \( d_j' \) where \( 0 \leq j \leq \tau \).

Then the values are

\[
\begin{align*}
27 & \quad 38 & \quad 17 & \quad 39 & \quad 15 & \quad 36 & \quad 22 & \quad 40 & \quad 10 & \quad 30 & \quad 34 & \quad 3 & \quad 20 & \quad 18 & \quad 33 & \quad 14 & \quad 19 & \quad 23 & \quad 42 \\
11 & \quad 32 & \quad 31 & \quad 7 & \quad 5 & \quad 8 & \quad 28 & \quad 37 & \quad 21 & \quad 43 & \quad 0 & \quad 6 & \quad 9 & \quad 4 & \quad 24 & \quad 1 & \quad 35 & \quad 13 & \quad 12 & \quad 26 \\
25 & \quad 16 & \quad 29 & \quad 2 & \quad 41
\end{align*}
\]
4. **Encryption/Decryption Algorithms**

In this modification of RADG the keys of encryption method are design of RADG, and \(n, m, k, d, x_0, \) and \(r\) to generated the sequence of values in states of RADG, The RADG method is a symmetric block cipher with variable size of block to create a multi-random ciphertext.

**4.1 Encryption Algorithm.**

The encryption algorithm by modification RADG method. Algorithm (4.1) explain encryption algorithm of modification RADG method.

![Algorithm (4.1): Encryption of the Modification-RADG Method.](image)

**Input:** Message (M) of binary digits. with key \(n, m, k, d, x_0, \) and \(j\) (start state where \(0 \leq j \leq n-k-1\)).

**Output:** The ciphertext (c) in binary digits

1. Start from the state \(S_0\), and read message from index id
2. \(L= j \text{ mod}(d)\).
3. Select binary sub message \(B_i\) from M of length \(L+1\),
   - If \(B_i\) is empty
     \[C=C_1C_2\ldots\]
   - else
     start from id ( index reading message M) to id+L+1 ,
4. Encryption \((B_i)\) by corresponding order value \(C_i = V_{B_i}\) in state \(j\).
5. Next state where \(j=\text{address of next state}.
   - If \(S_j \in \text{jump}\)
6. Go to step 2

**Figure (2): Encryption Algorithm.**

**Example (4.1):**

Suppose the message \(hi\), and the key is \(n=7, m=5, k=2, \) and \(d=3\), then \(\mu=10, \rho=3, \upsilon=0, \tau=44,\) and \(\tau_0=6\). The first step convert to binary by ASCII code, or any coding,

\[M=\text{10100010101}\]

By using table(1) the above binary message will be processed as illustrated in table (2). In table (1) every cell \(i, j\), the \(i\) represent the row and the \(j\) represent the column. In every cell there are two values, the top value represent the value of the state \(i\), in the order \(j\), and the bottom value represents the next state from value of order \(j\).
Table (1): The states transitions.

| State | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-------|---|---|---|---|---|---|---|---|
| 0     | 27 | 38 | 3 |   |   |   |   |   |
|       | 1  | 3 |   |   |   |   |   |   |
| 1     | 17 | 39 | 15 | 36 | 10 | 4 |   |   |
|       | 0  | 3 |   |   |   |   |   |   |
| 2     | 22 | 40 | 10 | 30 | 34 | 3 | 20 | 18 |
|       | 4  | 10 |   | 3 |   |   |   |   |
| 3     | 33 | 14 |   |   |   |   |   |   |
|       | 4  | 2 |   |   |   |   |   |   |
| 4     | 19 | 23 | 42 | 11 |   |   |   |   |
|       | 11 | 3 |   |   |   |   |   |   |
| 5     | 32 | 31 | 7  | 5  | 8  | 28 | 37 | 21 |
|       | 3  | 0  | 2  | 3  | 4  | 1  | 0  | 1  |
| 6     | 43 | 0  | 1  |   |   |   |   |   |
|       | 2  |   |   |   |   |   |   |   |
| 7     | 6  | 9  | 4  | 14 |   |   |   |   |
|       | 0  | 1  | 3  |   |   |   |   |   |
| 8     | 1  | 35 | 13 | 12 | 26 | 25 | 16 | 29 |
|       | 4  | 0  | 2  |   |   |   |   |   |
| 9     | 2  | 41 |   |   |   |   |   |   |
|       | 1  | 2  |   |   |   |   |   |   |

Table (2): The Encryption Processing.

| Secret Binary Bits | Value | Values in binary | Next State |
|--------------------|-------|------------------|------------|
| 01                 | 39    | 100111           | 3          |
| 0                 | 33    | 100001           | 4          |
| 01                 | 23    | 010111           | 3          |
| 1                 | 14    | 001110           | 2          |
| 0001               | 40    | 101000           | 10 Jump to State 7 |
| 10                | 9     | 001001           | 1          |
| 10                | 15    | 001111           | 10 Jump to State 9 |
| 1                | 41    | 100101           | 2          |
| 0                | 22    | 011010           | Stop       |

After the converting the ciphertext to the binary values then the ciphertext is
$$C=0110101001011110010011010000011110010111100001100111$$

4.2 Decryption Algorithm.

The decryption algorithm by modification RADG method. Algorithm (4.2) explain decryption algorithm of modification RADG method.
**Algorithm (4.2): Decryption of the Modification-RADG Method.**

**Input:** Ciphertext (C) of binary digits. with key n,m,k,d, x0, and r

**Output:** The plaintext (M) in binary digits

1. Start from the state $S_j$, and read ciphertext from index id
2. $L= j \mod (d)+1$.
3. Select binary sub message $B_i$ from C,
   - If C, is empty
     - $M=M_1M_2…$
   - else
     - start from id ( index reading ciphertextC) to id+L+1
4. Decryption ($B_i$) by corresponding order value $M_i = V_B_i$ in state $j$.
5. Next state where $j=address$ of next back value state in C of size $\tau$.
   - If $S_j \in jump$
     - $S_j = the \ state \ search \ value \ in \ R\_states$
6. Go to step 2

---

**Figure (3): Decryption Algorithm.**

**Example (4-2):**

As illustrated in algorithm (4.2) after the ciphertext C receiving, it will be converted to decimal values (the values of the states in RADG design), then searching about these values in the states will be performed in order to return the corresponding secret binary bits see table (1). After this processing then the binary bits will be converted to decimal values which corresponding to the ASCII values of the plaintext.

5. **Conclusions**

   In the original version of the RADG the encryption is performed without key (keyless) with the using of static design (constant states and values in the graph). In this paper the constant values in the graph are designed in automatic way using the chaos logistic map. In the proposed work the values of states in RADG design are selected from the chaos sequence , that convert to distinct integer sequence $v_0,v_1, \ldots , v_{\tau-1}$ between 0, and $\tau-1$, with initial value $x_0$ of chaos function map.

6. **References**

   [1] Salah A.Albermany, et al,”Novel Design of Block Cipher in RADG Automata with Wireless Network”, Master Thesis, Computer Science and Mathematics, University of Kufa, 2016.

   [2] Salah A.Albermany, et al,”RADG Cryptography Construction Using Chaos Generators”, Master Thesis, Computer Science and Mathematics, University of Kufa, 2017.

   [3] Vieira, R.S.S.; Michtchenko, T.A. Relativistic chaos in the anisotropic harmonic oscillator. Chaos Solitons Fractals 2018, 117, 276–282.

   [4] Alves, P.R.L.; Duarte, L.G.S.; da Mota, L.A.C.P. Detecting chaos and predicting in Dow Jones Index. Chaos Solitons Fractals 2018, 110, 232–238.
[5] Cairone, F.; Anandan, P.; Bucolo, M. Nonlinear systems synchronization for modeling two-phase microfluidics flows. Nonlinear Dyn. 2018, 92, 75–84

[6] V. H. Mankar1, et al, “Discrete Chaotic Sequence based on Logistic Map in Digital Communications”, National Conference on “Emerging Trends in Electronics Engineering & Computing” (E3C 2010).

[7] Oabid A.J., AlBermany S., Alkaam N.O. (2020) Enhancement in S-Box of BRADG Algorithm. In: Solanki V., Hoang M., Lu Z., Pattnaik P. (eds) Intelligent Computing in Engineering. Advances in Intelligent Systems and Computing, vol 1125. Springer, Singapore. https://doi.org/10.1007/978-981-15-2780-7_80.

[8] A. H. Salah A. Albermany, MRADG design on Elliptic Curve Cryptography, London UK: American Research Foundation, ICIIIDT, October 2016.

[9] F. R. H. Salah A. Albermany, "New Block Cipher Key with RADG Automata," Asian Journal of Information Technology, vol. 16, no. 5, 2017.