Journal of Engineering and Applied Sciences 13 (Special Issue 5): 4654-4662, 2018
ISSN: 1816-949X
© Medwell Journals, 2018

Novel Design on RADG with Elliptic Curve

Salah Albermany and Ali Hasan Alwan
Department of Computer, College of Computer Science and Mathematics,
University of Kufa, Kufa, Iraq

Abstract: In the keyless Reaction Automata Direct Graph (RADG) using personal wireless network, once the sender sends a message to the receiver and there is someone (third part) on the network are spy on the communication the third part can be decrypts the cipher text easily as receiver, if the third part has the design of RADG because of RADG totally depends on the design. This study will present a development on RADG by using the theory of divided and conquer on the reaction states this produce several sets of reaction states (called multi-reaction) the search in decryption process on the separate sets much fast and easily than on set of reaction like in RADG in addition using the elliptic curve over galois field of prime number this offer smaller memory and processer requirements it's needed and small key length with the same levels of secure of other standard methods of public key cryptography.

Key words: RADG, decryption, elliptic, galois, cryptography, wireless network

INTRODUCTION

Cryptography is a science that study and deals with encryption and decryption of the data the cryptography is concerned with several objectives: confidentiality integrity, non-repudiation, authentication (Stallings, 2011). There are two types of cryptography symmetric and asymmetric, the last one is brought by "Diffie" and "Hellman" in 1976 when they published the method of public-key cryptography, the public key techniques base on pair of keys "public and private key", the public key is sheared in public channel in network and it's known by everyone unlike the private key only key owner have it (Stinson, 2005) most of public key methods based on DLP (discrete logarithm problem), Suppose there are two nodes A and B in cognitive radio network will communicate with each other at first each nodes choice private key and compute the public key then sheared it in network, finally before proceed in encryption they need to compute their shared key "K_{ab}" to compute "K_{ab}" the node A select node B pubic key and compute shared key in the node B the process vice versa (Stallings, 2010) in 1978 RSA is submitted then El-Gamal presented in 1985 (Menezes *et al*., 1996). The previous mentioned methods are generally used for public key cryptography and signature, A new methods of public key cryptography is Elliptic Curve "EC", The EC also based on discrete logarithm problem like RSA and elgamel but the elliptic curve is offer two benefit smaller key in the same levels of security small memory and processor requirements (Cohen *et al*., 2005) it's invented in 1985 by "Miller" and "Koblitz", The EC is faster than RSA and diffie-helman , far smaller in key size and more efficient (Stallings, 2011). this study was based on RADG 2014, by "Salah A. Albermany" and "Ghazanfar A. Safdar" that focuses on random cipher text and keyless (Albermany and Safdar, 2014). the mostly random methods in cryptography use to generate keys called Pseudo Random Number Generators (PRNGs) it's produce random number based on initial value "Seed" but these methods are not used for encryption data with property of randomness (Blake *et al*., 1999).

Multi-Reaction Automata Direct Graph (MADG): The Multi-reaction represented by 3 sets: reaction denoted by R, Q and jump set denoted by J and transition between them, each set have a several nodes. In each node there is two parts, number of state explain in this study the second part is the state value (as points in elliptic curve). The transition started internal from one value of the state forwarded to another state It refers externally in Fig. 1, except the Jump's state they don't have output transition, each jump state travels randomly into state in the one specific set of reaction review, the main goal of multi-reaction is to decrease the decryption (inverse) time into half or more than half while preserving the random property of produce cipher text.

Corresponding Author: Salah Albermany, Department of Computer, College of Computer Science and Mathematics, University of Kufa, Kufa, Iraq

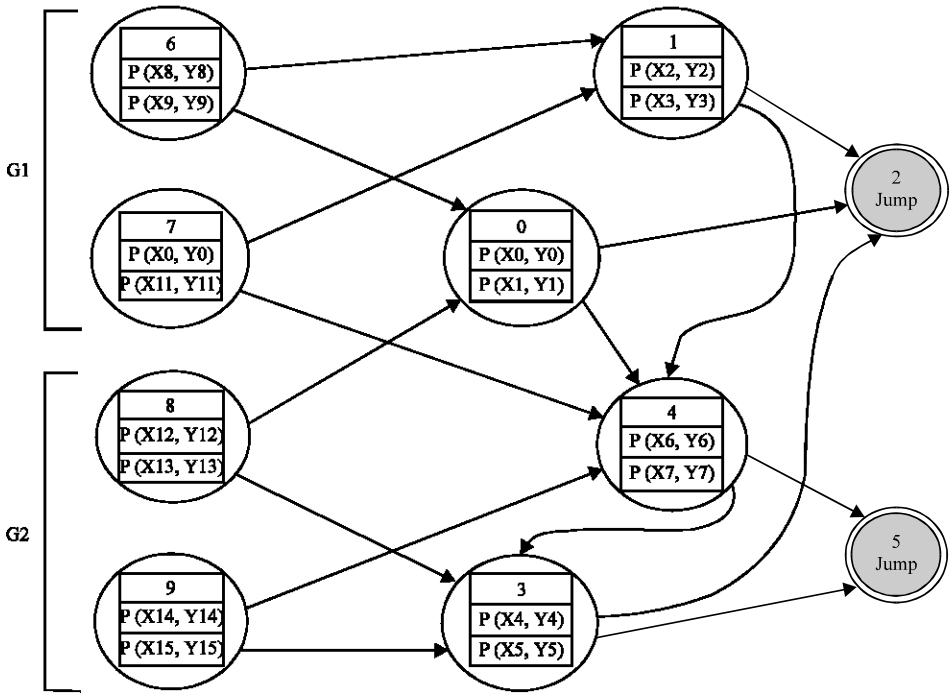


Fig. 1: MADG example with 4 reaction states divided into 2 subgroup and 6 Q states with 2 jump states

MATERIALS AND METHODS

Mathematical model of MADG: The mathematical model of MADG depends on automata direct graph and elliptic curve. MADG can be describing in 6 tuples {Q, R, Σ, Ψ, J, T} where Q stander states set , R reaction states set, Σ, input data,Ψ output data from transition T, J Jump states set and T transition function. All of above can be describing clearly in Fig. 1, (Udin *et al*., 2012) (Appendix A). The Reaction set can be divided into two subsets or more $\{G_1, G_2, ..., G_h\}$ where h is the partitions number of reaction set and G_1 is the first subset of set R, $G_1 \subset R$, G_2 is the second subset of set R,$G_2 \subset R$ and G_1 (i = 1, 2, …, h) is the i th subset of set R , $G_i \subset R$ where the union of all reaction subsets $G_1, G_2, ..., G_1$ is the set of reaction R:

$$\bigcup_{i=1}^{h} G_i = R \quad (1)$$

The intersection of any two sub sets ($G_1, G_2, ..., G_1$) of the reaction states given an empty, since, G_1, G_2, ..., G_h are partition to set R that mean:

$$G_{i \cap j G_j} = \varnothing \quad (2)$$

$$i \neq j$$

$$i = 1, 2, 3, ..., h$$

$$j = 1, 2, 3, ..., h$$

where, ∅ is an empty set. To get a random property when applying multi-reaction must verify the condition h≤k where, k is a size of set J when k = h is a normal RADG.

The number of states distributions on reaction set partitions: The proposed to design of RADG cryptography depends on Multi-Reaction partitions. In order to calculate the distribution cases of reaction states for each partition or subset of R there are two situations according to whether the partition number (h) is divided the total number of reaction states (m) or not. In order to compute the distribution cases of nodes in partitions of sub sets of reaction state there is two theorems:

Definition (1): The partitions of the set R that have a minimum number of reaction states [m/h] in each partition (subset of R) is called normal multi-partition denoted by NMP.

Definition (2): The name of each partition of R is called partition-box denoted by PB.

Theorem (1): The number of all state distribution of NMPs where h|m:

$$\Omega = \prod_{i=1}^{h} \frac{(m\text{-}i\text{-}1\times u)}{u} \quad (3)$$

Where:

Ω = Total number of reaction states distributions in partitions

m = Number of reaction states

u = The minimum number of Reaction states in each partition where u [m/h] and [X] defines a maximum integer number less than or equal X where X∈R

Example: Suppose nine of reaction states (m = 9) are divided into three portions (h = 3) at first computed u the minimum number of states in each partition u = [m/h], u = [9/3] = 3 there is no remaining in division operation h|m:

$$\Omega = \binom{9\text{-}(1\text{-}1)\times 3}{3} \times \binom{9\text{-}(2\text{-}1)\times 3}{3} \times \binom{9\text{-}(3\text{-}1)\times 3}{3}$$

$$\Omega = \binom{9\text{-}0}{3} \times \binom{9\text{-}3}{3} \times \binom{9\text{-}(2)\times 3}{3}$$

$$\Omega = \binom{9}{3} \times \binom{6}{3} \times \binom{3}{3}$$

$$\Omega = \frac{9!}{3!6!} * \frac{6!}{3!3!} * \frac{3!}{3!}$$

Ω = 1680 ways to distribute 9 states of reaction set into 3 partitions.

Theorem (2): If m divided into h partition where PBi (PBi Partition Box of Index i) have exactly u_i states and $m = \sum_{i=1}^{h} u_i$ with define the $u_0 = 0$ then the all cases to distributed m in into h partitions is:

$$\Omega = \prod_{i=1}^{h} \binom{m\text{-}\sum_{j=0}^{i-1} u_j}{u_i} \quad (4)$$

Theorem (3):

$$\Omega = \sum_{i=1}^{c} \Omega_i \quad (5)$$

$$\Omega_i = \prod_{j=1}^{h} \binom{m\text{-}\sum_{r=1}^{i} L_{r-1}}{L_{ij}} \quad (6)$$

Where:

ℸ = Not divided

C = Number of ways to write m as a sum of h non-negative integers of at least u

Ω_i = All cases of the i 'Th way of the C

L_{ij} = The size of Pb_{ij} distribution box of order j in the I The way of the C with define Li0 = 0

Example: Suppose we have m = 8, h = 3 then each state have the minimum number u such that u = [8/3]⇒⌊2.6667⌋ = 2 to represent the ways distributions of states into h partitions there are two ways only (that mean C = 2) the first way is 3, 3, 2 and the second way is 4, 2, 2. The first way is L_{11}, = 3, L_{12}, = 3, L_{13} = 2 and:

$$\Omega_1 = \prod_{j=1}^{3} \binom{m\text{-}\sum_{r=1}^{i} L_{r-1}}{L_{ij}} \quad (7)$$

$$\Omega_1 = \binom{8}{3}\binom{8\text{-}3}{3}\binom{8\text{-}(3+3)}{2}$$

Ω_1 = 8!/3!5! 5!/3! 2! 2!/2! = 560 cases to distribution 8 states in first way. The second way is:

$$L_{21} = 4, L_{22} = 2, L_{23} = 2$$

$$\Omega_1 = \prod_{j=1}^{3} \binom{m\text{-}\sum_{r=1}^{i} L_{r-1}}{L_{ij}} \quad (8)$$

$$\Omega_1 = \binom{8}{4}\binom{8\text{-}4}{2}\binom{8\text{-}(4+2)}{2}$$

$$\frac{8!}{4!4!}; \frac{4!}{2!\,2!}; \frac{2!}{2!} = 5040$$

Cases to distribution 8 states in second way. Then Ω = Ω1+Ω

The number of ways to write m as a sum of h integers: To compute partition of m numbers into L_{ij} ,the number of ways a positive integer n can be written as the sum of positive integers less than or equal to n (Table 1). The integer partition function is often denoted by c (n) For example, c (5) = 7 because there are 7 ways to write 5 as a sum and c (0) as 1. While there is no simple explicit formula for computing c (n) there is a recursion formula that allows you to calculate c (n) as long as you know the values of c (I) for i<n (Table 2). The recursive Eq. 9 is:

Table 1: Partitions cases

n	C(n)
0	1
1	1
2	2
3	3
4	5
5	7
6	11
7	15
8	22
9	30
--	--
50	204226

Table 2: Example of MADG state with 16 values as elliptic curve points

Values	Nodes numbers
0	P (X_0, Y_0)
1	P (X_1, Y_1)
2	P (X_2, Y_2)
3	P (X_3, Y_3)
4	P (X_4, Y_4)
5	P (X_5, Y_5)
6	P (X_6, Y_6)
7	P (X_7, Y_7)
8	P (X_8, Y_8)
9	P (X_{10}, Y_{10})
10	P (X_{11}, Y_{11})
11	P (X_{13}, Y_{13})
12	P (X_{14}, Y_{14})
13	P (X_{15}, Y_{15})
14	P (X_{12}, Y_{12})
15	P (X_9, Y_9)

$$c(n)=\sum_{k=1}^{n}(-1)^{k+1}\left[c\left(n-\frac{k(3k-1)}{2}\right)+c\left(n-\frac{k(3k+1)}{2}\right)\right] \quad (9)$$

where, k from 1 to. P (b) = 0 for all negative integers b.

Example: To compute the partitions cases of n = 5:

$$C(5)=c(5-1)+c(5-2)-c(0)-c(-1)$$

$$C(5)=c(4)+c(3)-c(0)$$

$$C(5)=5+3-1=7$$

$$\begin{cases} 5 \\ 4+1 \\ 3+2 \\ 3+1+1 \\ 2+2+1 \\ 2+1+1+1 \\ 1+1+1+1+1 \end{cases}$$

Bits of states address: Shown each node in design has a number; it's saved and consumed size of memory to compute the size needed to addresses by using the Eq:

$$Ad=\lceil \log_2 (m+n+k) \rceil$$

As an example in m = 4, n = 4, k = 2 where m = |R|, n = |Q| and k = |J|:

$$Ad=\lceil \log_2 (m+n+k) \rceil$$

$$Ad=\lceil 3.3219 \rceil$$

Ad = 4 bit for represent the nodes address in the memory. The minimum address is 0000 and maximum address is up to 1111 in Fig. 1 the min = 0 (0000) and max = 9 (1001) there is 6 places not used in the addressing from (1010)-(1111).

RESULTS AND DISCUSSION

Implementation: Implementation is the phase in which the whole design process is converted into working software, whereas testing ensures the validation and verification of the proposed work. Different techniques for both software implementation and testing are used in the proposed work.

MADG method is dependent on automate direct graph the graph consists of two parts: the states and transitions between them. The transition shall be in two forms either transported out of the state is called Output transition or input to the state (the input transition is referring on the node) each value inside state has own output transition but for the entire transition is not refers to Single value (point) but it's refers to the entire state then the value of input message or part of message that enter to the state is determine the chosen point.

Key generation: The key in MADG basically content from two parts private and public part illustrated in the private key "Pr" is the part held by the owner and it shall be without deployment in the network the node selects a word or sentence to be a private key ex:" moon", the character is converted into number where "A" = 0 and "B" = 1 and so on and numbers "0" = 52, "1" = 53 where, [A, ..., Z, a, …, z, 0, ..., 9] = [0, ..., 25, 26, ..., 51, 52, ..., 61] or using the ASCII code of characters where, "0" = 48, "A" = 65 and "a" = 97 the general scheme to generate the key shows through the following shown:

$$\text{Public key} = \text{Private} \times G \quad (10)$$

where, G is the generation point of selected elliptic curve after public key exchange between communication nodes in the network. Finally each private and public keys are

Table 3: Key generation notations

Notations	Details
n1	Refers to first node in network and It is part of secure commination by MADG
n2	Refers to second node in network and It is part of secure commination by MADG
Pn1	Public key of first node
Pn2	Public key of second node
Length ()	Function return numbers of given parameter e.g: the result is 4
K_1	Private key for first node
K_2	Private key for second node
K_{ab}	Shared key between the communications nodes

not used to encrypt and decrypt, the communication nodes compute the shared key "K_{ab}" in each nodes this key will be used in encrypted and decrypted node n_i select private stream key that length L_1 and compute public key Pb1 that composed of stream of points the node n_2 make the same operations and produce Pr2 and Pb2, respectively.

The K_{ab} = Pb2×Pr1 and vice versa in node n2 where, K_{ab} = Pb1×Pr2 is the shared key is must be equivalent in the two nodes. Key generation in the two communications parts (Table 3).

Algorithm 1; Key generation:

By n1

Step 1: n1 select Pn2
Step 2: Declare i = 0;
Step 3: P_L = length (P_{n2}), K_L= length (K_1)
Step 4: While (large (K_1 |?| P_{n2}))
Step 5: K_{ab} (i) = P_{n2} (i % P_L)×K_1 (i % K_L)
Step 6: i = i+1;
Step 7: End While

By n2

Step 1: n2 select Pn1
Step 2: Declare i = 0;
Step 3: P_L = length (P_{n1}), K_L = length (K_2)
Step 4: While (large (K_2 |□| P_{n1}))
Step 5: K_{ab} (i) = P_{n1} (i %P_L)×K_2 (i% K_L)
Step 6: i = i+1
Step 7: End While

Encryption:
Step 1: Declare i= 0, P←Ø, M = {M0 ,....., Mi}
Step 2: Old_{index}←random (Q_{set})
Step 3: While (i = |M|)
Step 4: {p (i), new_{index}}←f (Mi, K_{ab} (I), Old_{index}}
Step 5: If (I = m && Old_{index}∈ J)
Step 6:
New_{index} ←$close_{state}$ (old_{index},Q_{set})
Step 7: Add (P,p (i))
Step 8: If (new_{index}∈ J_r)
Step 9: New_{index} ←random (R_r set)
Step 10: Old_{index}←new_{index} index
Step 11: I = i+1
Step 12: End while
Step 13: Return P
Step 14:

Decryption:
Step 1: Declare i←|P| - 1, P← {P0, ..., Pi} M←⌀
Step 2: □Old □ı (index)←Search in Q (P_i |K ı ab (i % K_L)
Step 3: While (i = 0)
Step 4: If (Old_{index}) in Q_{set})
Step 5: {M (I), New_{index}, Bool}←F^{-1} (P_i K_{ab}, Old_{index})
Step 6: If (Bool = false)
Step 7: If (I = |M|-2)
Step 8: {m (I), New_{index}}←search in Q (P_i, K_{ab} (I% K_L))
Step 9: Else
Step 10: {M (I), New_{index}}←search in R (P_i, K_{ab} (I% K_L)
Step 11: If (Old_{index} in R set)
Step 12:{m(i),New_{index}), Bool}←F^{-1}) (P_i, K_{ab},Old_{index}),J_r)
Step 13: Add (M,m (i))
Step 14: I = i+1
Step 15: End While
Step 16: Return M

Performance and analysis example: The performance of MAGD, it can be shown in the application example with 15 states (Fig. 2) with Ψ = 2 means that each state has four values without include the numbers on the top of state it's represent the index if state. Message length taken 16 bits (2 hexadecimal number), n1, n2 agree on MADG design shown in Fig. 2 and elliptic curve Eq. y2 = x3-x+188 mod 751 and G point (0.376), n1, n2 select random stream key "Key", "System", respectively n1 sends message 'hi'

Confidentiality

Encryption: Before encryption process the sheared key must by compute the progress of computing key described in study 3 "multi-reaction automata direct graph". In Table 3, the sheared key as a points on the Elliptic curve = {(414, 663), (126, 476), (508.83), (280.247), (742, 74), (41, 477)}. The second step is the encryption by Sprite the Message into sets of 2 bits = {01, 10, 10, 00, 01, 10, 10, 01}. Start random in Q set in this example start at index = 2 as shown in Table 4 select (689,399) point according to value in column M then add with key point to produce cipher point, the function T is transition into state number 3 and repeat this process. In the index 2 in Table 4 after selected point the T function is transition into Jump state where it's select randomly from R-set then repeat this processes again until finish the message. Shaded rows are produced from jump. The cipher text of M is 8 points = {(609,125), (634, 657), (649, 57), (241,230), (295, 641), (94, 73), (375, 65), (245, 509)} (Table 5 and 6).

Decryption: Node n2 compute the shared key. The binary M = {011010001101001} where each 8 bits represent ASCII for one character:

$$01101000 \rightarrow 104 \rightarrow h \rightarrow 01101001 \rightarrow 105 \rightarrow i$$

Time complexity of MADG algorithm: The time complexity of encryption algorithm is calculate as O (n) depends on algorithm in this study with calculating the number of loops, taking into account

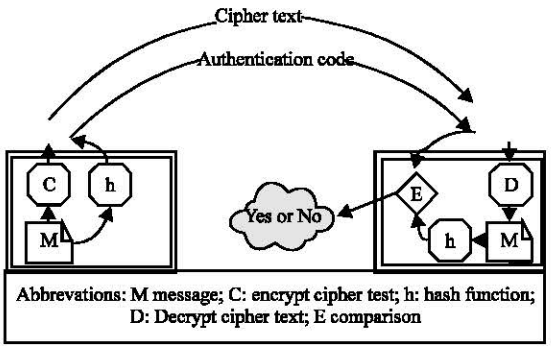


Fig. 2: Authentication between two users on network using MADG and MD5

Table 4: Phases of encryption

i	M	Index	p-values	K	C points
0	01	2	(689.399)	(414.663)	(609.125)
1	10	3	(17.419)	(126.476)	(634.657)
2	10	4	(30.515)	(508.830)	(649.570)
3	**00**	**13**	**(624.295)**	**(280.247)**	**(241.230)**
4	01	3	(565.557)	(742.740)	(295.641)
5	10	2	(663.257)	(41.477)	(94.730)
6	**10**	**10**	**(227.677)**	**(414.663)**	**(375.650)**
7	01	1	(624.456)	(126.476)	(245.509)

Shaded rows are produced from jump

Table 5: Shared key compute by node n2

n2 (ASCII)	Public key n1	Shared key
s (107)	(131.340)	(414.663)
y (101)	(432.441)	(126.476)
s (121)	(581.200)	(508.830)
t (107)	(131.340)	(280.247)
e (101)	(432.441)	(742.740)

Table 6: Decryption process

C points	K-1	Index	P-values	M
(245.509)	(126.476)	1	(624.456)	01
(375.65)	(414.663)	10	(227.677)	10
(94.73)	(41.477)	2	(663.257)	10
(295.641)	(742.740)	3	(565.557)	00
(241.230)	(280.247)	13	(624.295)	01
(649.57)	(508.830)	4	(30.515)	10
(634.657)	(126.476)	3	(17.419)	10
(609.125)	(414.663)	2	(689.399)	01

the conditions and the rest of steps considered a fixed variable in the time complexity so it is not count, the worst case and best case are the same in the encryption as shown in the following Table 7. In decryption process the time complexity of the algorithm is calculate (Table 8).

O (nk λ) bits input that will be decrypted, k is the number of inverse states that direct attach with the current state and λ is the values count inside each state and it is a constant value then the time complexity O (nk). The best case when k = 1→O (n).

Table 7: Encryption time complexity of MADG

	Encryption time complexity	
Best case	Worst case	Average case
O (n)	O (n)	O (n)

Table 8: Decryption time complexity of MADG

	Encryption time complexity	
Best case	Worst case	Average case
O (n)	O (n)	O (n)

Authentication: Authentication is a technique that provide ways to verify from the sender of the message by using hash value. The Authentication mechanism is extremely joined with data integrity, the data aforementioned to be licensed and integrated if they don't modification through transmitted channel (Kou, 2012; Boyd and Mathuria, 2013). In MADG the Authentication process do not need server on network, the whole can be describe in Fig. 3 in three steps.

Each point in massage points {P1, P2, …, Pn} where the X and Y the axis of point are contact as X|Y ex: P (9, 8) then the X|Y = 98 and so on to other points then combine the values into one value PA, this value are entering into "h" hash function (using one of hashing algorithm like MD5). In MD5 the input length string is 512 bits if PA less than 512 bits padding with zeros on the left, the sender sends the cipher points and hash value as authentication code.

The receiver will be decoding by the cipher points "D" function in Fig. 4 to get plaintext then compute the hash value like step 1 from the message points (plaintext). The final step is compare between the authentication code sends from the sender and hash value calculate by receiver if they equal the receiver authentication the sender if the values is not equal the receiver reject the sender.

Integrity: It's a protocol used to ensure that data received without any modification on it and detected any changes on data. There are several ways to achieved integrity similar to ways used in authentication (Stallings, 2011; Kou, 2012; Boyd and Mathuria, 2013). In MADG any modify on any cipher point value then the decipher process can't be resume as shown in Table 9 the point marked with underline change be an authorized node (user) on network the process of decrypt is failed because the search operation failed to find the point (4th point in Table 9) by using T-1 in another case if the point produce from cipher points plus key point are also find in the same state as unchanged point in this case (the property of happens are so week) the producer of decrypt are failed in 2 step at most.

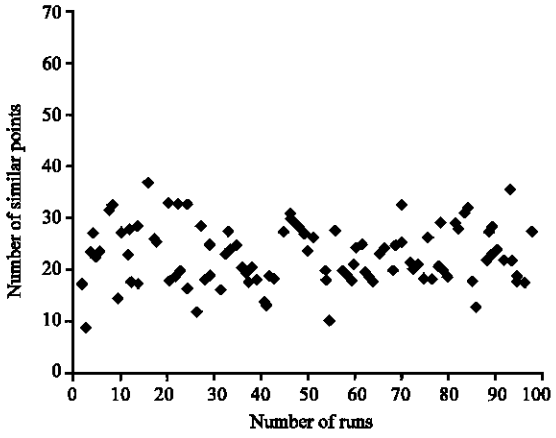


Fig. 3: Hamming distance between the different ciphertext for same plaintext result from MADG algorithm

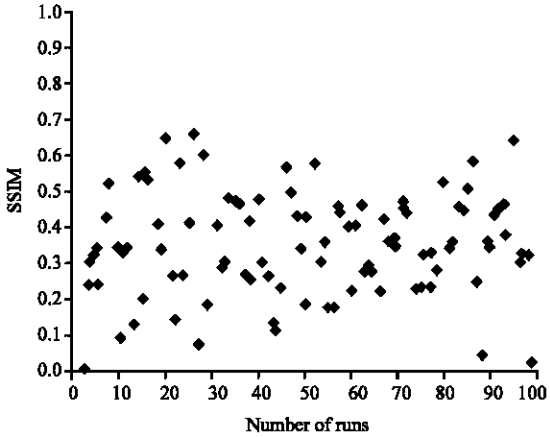


Fig. 4: SSIM between the different ciphertext for same plaintext result from MADG algorithm

Table 9: Authentication steps

C points	K-1	Index	Search	p-values	M
(245.509)	(126.476)	1	Search (Q)	(624.456)	01
(375.65)	(414.663)	10	F-1 = true	(227.677)	10
(94.73)	(41.477)	2	Search (J5)	(663.257)	10
(30.515)	(742.740)	--	--	--	--
(663.257)	(280.247)	--	--	--	--
(649.57)	(508.830)	--	--	--	--
(634.657)	(126.476)	--	--	--	--
(609.125)	(414.663)	--	--	--	--

Non-repudiation: It is new technique of authentication that the message attached with NRV "Non-Repudiation Vector" its strong evidence to proof sending the message from originator to receptor. The non-repudiation classification into two part NRO non-repudiation it's protected the receipt from attempt the origin to denial sending the message and NRR non-repudiation receipt it's protected the origin from the attempting of the receipt to denial receiving message, non-repudiation can be done by end-to-end communication.

Performance analysis: MADG function runs 100 time on the same plaintext (the length of plaintext is 76 points on the EC) to show the difference between the first cipher text and the next 99 cipher text, it is shown in Fig. 3 the maximum similarity of points is about 40 and minimum is less than 10 points. Because of the difference of cipher text, the statistical attack is very difficult compared with other methods. Figure 3 is shown the measure of similarity of output result between 100 different runs in MADG on the same message.

The Hamming distance between two strings of point of equal length is the number of positions at which the corresponding symbols are different. Or it measures the minimum number of substitutions required to change one string into the other or the minimum number of errors that could have transformed one string into the other.

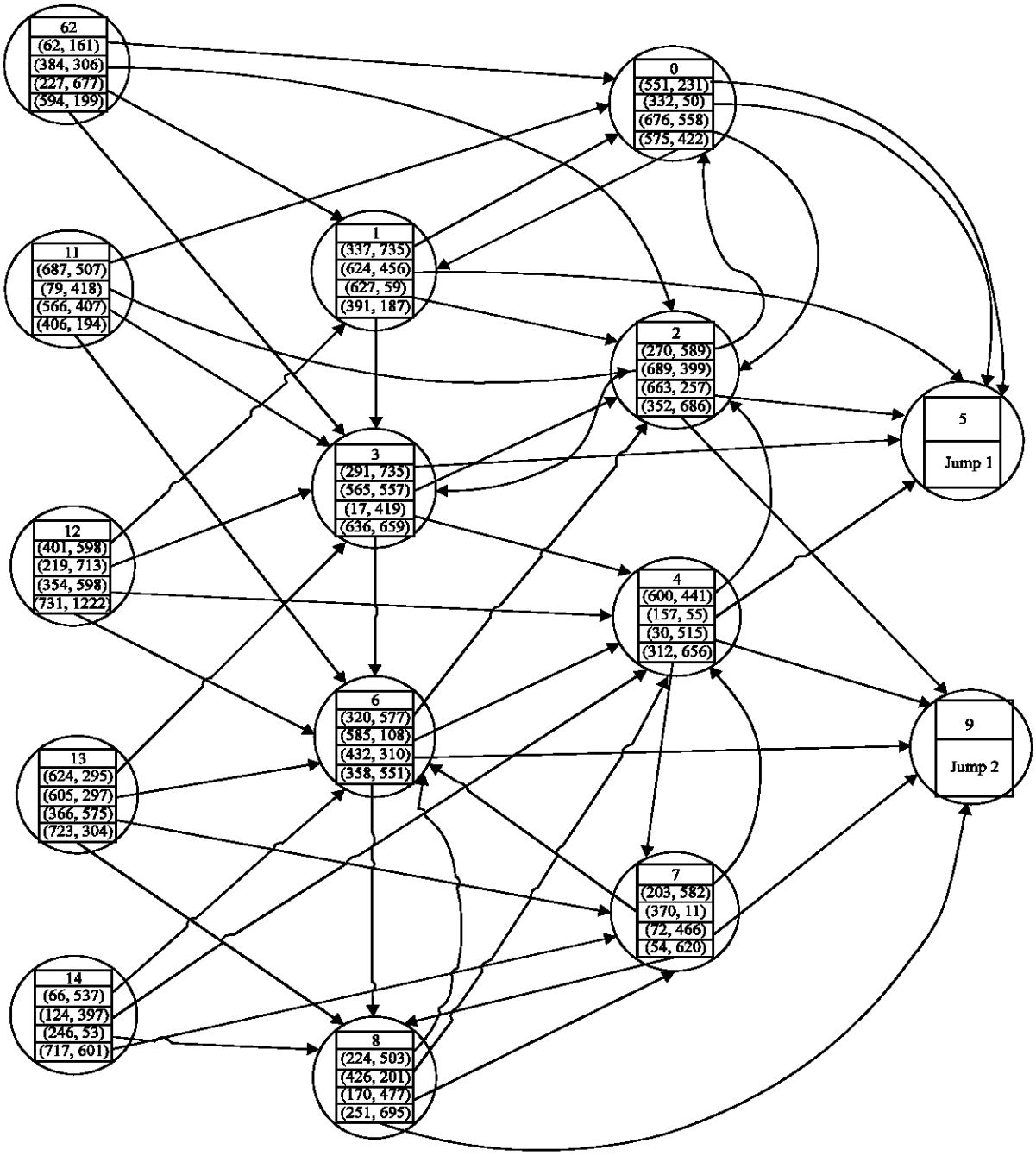
In deferent way to measure the difference between points on range of many runs can be use SSIM a method to measure similarity was introduced by Wang and Bovik where a structural similarity has been designed and called "SSIM" this measure gives near to one in the maximum (if the two victors are similarity) and gives near to zero in the minimum (if the two victors are distinct) (11).

CONCLUSION

The strong point of previous algorithm design RADG is the random characteristic on the produce ciphertext, the improvement in the proposed design is to design the RADG with elliptic curve to economize the memory and processer consuming and also the increasing of Ø values Leads to reduce the ciphertext length, there is Inverse relationship between them:

$$\text{Cipher text length} = \text{Plain text}\frac{\text{length}}{\square}\psi$$

According to the performance analysis the proposed design is strongly against the statistical attack, increasing the range of networks that RADG implementation in the RADG which applied on personal networks and the developed method MADG apply on wider networks including intelligent networks.



Appendix A: MADG (Multi-Reaction Automata Dircet Graph) example of MADG design

REFERENCES

Albermany, S.A. and G.A. Safda, 2014. Keyless security in wireless networks. Wirel. Pers. Commun., 79: 1713-1731.

Blake, I., G. Seroussi and N. Smart, 1999. Elliptic Curves Cryptography. Cambridge University Press, New York.

Boyd, C. and A. Mathuria, 2013. Protocols for Authentication and Key Establishment. Springer, Berlin, Germany, ISBN:978-3-642-07716-6, Pages: 321.

Cohen, H., G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen and F. Vercauteren, 2005. Handbook of Elliptic and Hyperelliptic Curve Cryptography. Taylor and Francis Group, London, ISBN-13: 9781420034981, Pages: 848.

Kou, W., 2012. Networking Security and Standards. Vol. 394, Springer, Berlin, Germany, ISBN: 978-1-4613-7820-4, Pages: 203.

Menezes, A.J., C.V.O. Pual and A.V. Scott, 1996. Handbook and Applied Cryptography. CRC Press, Boca Raton, Florida, USA., ISBN: 13-978-0-84-938523-0, Pages: 755.

Stallings, W., 2011. Cryptography and Network Security: Principles and Practice. 5th Edn., Prentice Hall, USA., ISBN: 9780136097044, Pages: 719.

Stinson, D.R., 2005. Cryptography: Theory and Practice. 3rd Edn., CRC Press, Boca Raton, Florida, USA., ISBN:978-1-58488-508-5, Pages: 583.

Udin, M.N., S.A. Halim, M.I. Jayes and H. Kamarulhaili, 2012. Application of message embedding technique in el-gamal elliptic curve cryptosystem. Proceedings of the International Conference on Statistics in Science, Business and Engineering (ICSSBE), September 10-12, 2012, IEEE, Langkawi, Malaysia, ISBN:978-1-4673-1581-4, pp: 1-6.