RADG Design on Elliptic Curve Cryptography

Salah A. Albermany [1,*], Ali Hasan Alwan [2,*]
[1,2] University of Kufa, college of computer science and mathematics, computer department
*Contact: salah.albermany@beds.ac.uk, alih.alashour@student.uokufa.edu.iq

Abstract— The main problem in Reaction Automata Direct Graph (RADG) is the static design the purpose of this paper is to develops RADG algorithm to become more efficient by enhancing characteristics to cover large networks, use of the key in the encryption process increase the complexity against the attacker on networks and improved the single Reaction states in RADG to Multi-Reaction states which is up the speed of the algorithm finally the internal design become changeable which it was static in RADG so the RADG with our enhancement become Multi- Reaction Automata Direct Graph (MRADG), The output of the algorithm in our study result is changed from binary to points in (EC).

1. INTRODUCTION

Cryptography is coming from two a Greek words "kryptos- graphein" means hidden – writing or the science of encrypting and decrypting text [1]. Where the text or Message is transfer in unreadable formatted (usually called cipher text) done by encryption process. If we are divided the Cryptography into sections according to the encryption key there are two types, Symmetric key cryptography and Asymmetric key cryptography. Symmetric cryptography (referred as Secret-key ciphers) using the same key in encrypt and decrypt [2]. Asymmetric cryptography (referred as public-key ciphers) required public and private key to encryption/decryption and it's included Digital Signature which used to authorize the message's sender. The symmetric cryptography is much faster than Asymmetric cryptography but from the point of safety is a public key safer; it's using arithmetic operation like modular multiplication and required exponentiation time when attempted to encryption. From the first public key algorithms is RSA, The name came from the first latter of three scientists Ron Rivest, Adi Shamir, and Leonard Adleman, they published their method in 1977 Is one of the first public key cryptography using public key to encryption and deferent key (private key) to decryption it's based on two primes number and its products before encryption in RSA [3],and the modern public key Algorithm is the Elliptic Curve "EC", there are many fields of EC such as Galois Field of large Prime number "GF_p",over Binary filed 2^m where m is an integer number , EC over Z where Z is the integer number, and etc. in this paper the main matter and discuss about EC over GF_p, the general equation of EC is $y^2 = x^3 + \alpha x + \beta \bmod p$ $y^2 = x^3 + ax + b \bmod p$ Denoted by $E_p(\alpha, \beta)$ $Ep(a, b)$ where the coefficients α, β is integer in the Galois field and p is prime number (where α, β must satisfy the equation $4\alpha^3 + 27\beta^2 \neq 0$). There are two operations in the field FPF_p define on EC addition operation and multiplication operation (multiplication is repeated addition ex: 2P=P+P) [4] [5] [6].

2. RELATED WORK RADG

Reaction Automata Direct Graph (RADG) that depends on random characteristics, keyless, static internal design and using in personal network. And it's useful against statistical attack, which based on Automata Direct Graph and Reaction states. And it's applied encryption and decryption process without any key and produces several cipher text from one plaintext. The main design is contain R states, Q states and jump where jump doesn't has transitions as illustrated in Figure 1.

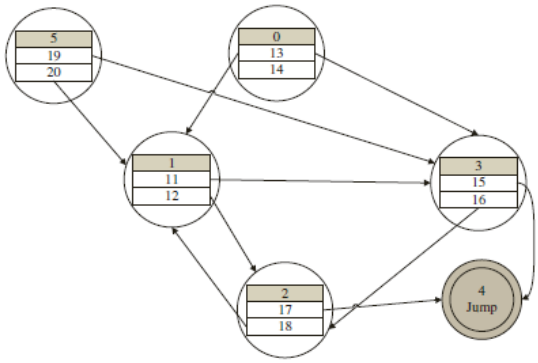

Figure 1 Example implementation RADG design

To explain the encryption process: Suppose the original message to be encrypted using RADG is 1010. Thus we have: T (0, 1) = (3, 14), T (3, 0) = (5, 15), T (5, 1) = (1, 20), and T (1, 0) = (3, 11). The corresponding output is 14, 15, 20, and 11 respectively, where the jump state redirect from state 4 to state 5 (state 5 including in Reaction states) [7].

3. MULTI-REACTION AUTOMATA DIRECT GRAPH (MRADG):

The main problem in RADG is the static design, if any one on the network gets the design with cipher text then he/she can be decrypt the cipher text effortlessly and get information from it , the purpose of this paper is to develop the RADG design to be more secure in Wide area networks such as cognitive Radio Network "CRN". By using RSA scheme instant of Transition function in RADG and convert the cipher text into point in specific EC. The new algorithm based on RADG design that contains three parts: Reaction states, Q states and Jump states. Each state has λ of values except the jump state that just refers to another state in Reaction states; the proposed algorithm used function f instead of the static transition function in RADG [7] , as shown in Figure 2.

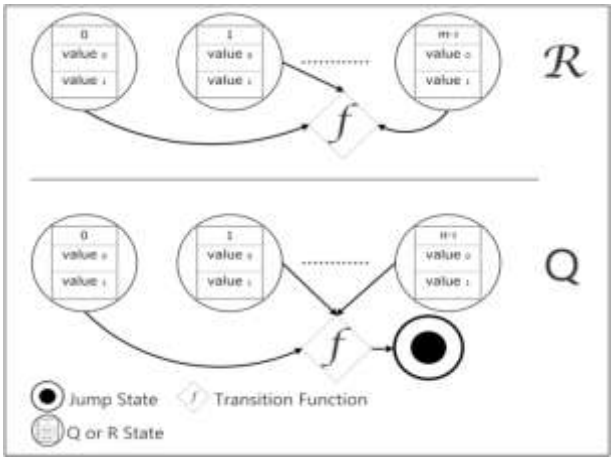

Figure 2 MRADG Design

4. THE NUMBER OF VALUES IN THE DESIGN

In general each state either it's in Q states or R states it's have λ of values in single state expect the jump states it's don't have any values then to compute the number of values in the entire design:

$$\Lambda = \lambda * (m + n - k)$$

Where m: number of R states, n: number of Q states , k: number of $Jump$ states.

5. TRANSITION FUNCTION:

At the beginning the communications parts are agreed on large number n that consist of two prime number p and q where $n = p * q$. The numbers of Q states is started from 0 to $n - 1$, also sender who's ciphering choose a random number e between $(1, \Phi(n))$ where the $\Phi(n)$ is the Euler's totient function, the e number is represented the public key for transition function on condition $\gcd(e, \Phi(n)) = 1$. There are two type of transition function f_Q and f_R there is no difference in the internal architecture of these functions, the result for them is refers to the state within the Q set but the input of f_Q is from Q states and for f_R is from R states.

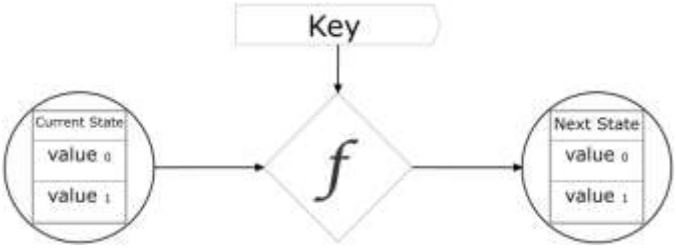

Figure 3 Transition function scheme

To determine the next state as illustrated in Figure3 where it's totally depending on the current state where the cipher progress are located in this , besides to make sure the distribution of states no. is not repeated circularly to avoid that this situation must have a secondary key (the message index represent this key) . Suppose the current state number is '1' as known the '1' raised to any power remains '1' then if the cipher process started from state '1' surely all the next states will become '1' (without secondary key). Example the current state is "11", and the message index is '8' compute the next state:

$$temp = CurrentStateNo + key \bmod n$$
$$NextState = temp^{e} \bmod n$$

Then the next state is number "28" in the Q states, and so on to the rest of states. The deference of two functions f_Q and f_R is the input state either from R state or Q state, the output always in Q state.

In the backward process from the last example we have located in the state number "28" and the message index is "8" after calculate the plaintext description in section 3 then calculate the previous state :

$$temp = (CurrentStateNo.)^{d} \bmod n$$
$$PreviousState = temp - M_{index} \bmod n$$

The previous state is "11" briefly explained in Figure4 , and so on for remains states.

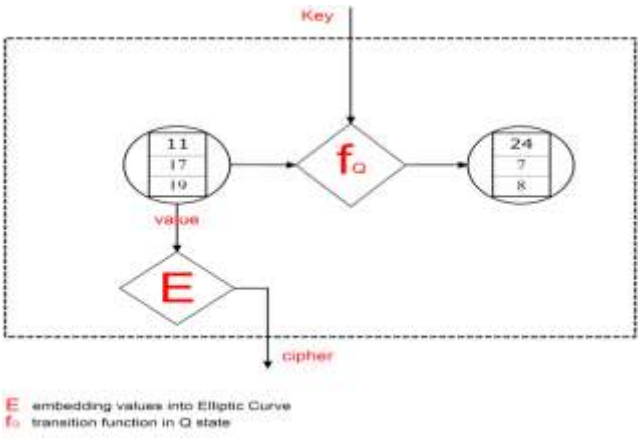

Figure4 Internal structure of MRADG

6. EMBEDDING

To represented date into the EC there is several ways, consider a curve $y^2 = x^3 + ax + b \bmod p$ and the message is content number and alphabet characters (0-35).to encode char 'A' as $m = 10$ within a public variable, n=20.

1. Compute, $ma = m * k + i$, where value of i is ranged between $\{1, .., k-1\}$ and try to get integral value of y
2. Thus ma encoding as point (x, y)
3. The decoding is simple $m = \text{floor}\left(\frac{x-1}{k}\right)$ [5] [6].

7. ALGORITHMS

Using algorithms to described clearly and understandable the encryption and decryption processes, the following table explains ambiguities abbreviations and functions in the algorithms.

TABLE 1 RADG IMPLEMENTATION NOTATIONS

Notations	Details
random(0,n)	Generate random integer number between 0 and n
IN_R,IN_Q	In R states , in Q states
getValue[Message]	Get the first or second value from the state
$next_{state}()$	Function take several parameters : previous $state_{no'}$, index as secondary key , e as transition key and n number of finite field and return number of next state

$jump_i$	Refers to one of jump states
$random_{G_i}$	In multi Reaction it's refers to a random number in one subgroup of reaction
Embadding()	Embedding the cipher value into specific Elliptic curve
$perviouse_{state}$	Inverse of $next_{state}$ function

- Key for transition function

step 1. $P \leftarrow Prime_{NO}$, $q \leftarrow Prime_{NO}$

step 2. $n \leftarrow p * q$

step 3. $\Phi(n) \leftarrow (p-1) * (q-1)$

step 4. $e \leftarrow gcd\big(random(0, n-1), \Phi(n)\big) = 1$

step 5. $d \leftarrow e^{-1}$

- Encryption

step 1. $State_{no}. \leftarrow random(0, R_{length})$

step 2. $status \leftarrow IN_R$

step 3. $while\ (index < Message_{length})$

step 4. $if\ status = IN_R$

step 5. $cipher[index] = R[State_{no}].getValue[Message]$

step 6. $State_{no}. = next\_state\,(State_{no}., index, e, n)$

step 7. $end\ if$

step 8. $if\ (State_{no}. = jump_i)$

step 9. $State_{no}. = random_{G_i}$

step 10. $enf\ if$

step 11. $if\ Status = IN_Q$

step 12. $cipher[index] = R[State_{no}].getValue[Message]$

step 13. $State_{no}. = next\_state\,(State_{no}., index, e, n)$

step 14. $end\ if$

step 15. $Embadding\,(cipher)$

step 16. $End\ Wile$

- Decryption

step 1. $Embadding^{-1}()$

step 2. $[status, State_{No}] \leftarrow perviouse_{state}\,(State_{no}, Message_{length-1}, d, n)$

step 3. $while\,(index \geq 0)$

step 4. $if\ Status = IN_Q$

step 5. $decipher[index] = Q[State_{no}].getValue\,[Message]$

step 6. $State_{no}. = perviouse_{state}\,(State_{no}., index - 1, d, n)$

step 7. $if\ (value\ not\ found\ in\ Q)$

step 8. $status \leftarrow IN_R$

step 9. $end\ if$

step 10. $end\ if$
step 11. $if\ Status = IN_R$
step 12. $decipher[index] = Q[State_{no}].getValue[Message]$
step 13. $State_{No} = jump_i(R_i)$
step 14. $State_{No} = pervious e_{state}(State_{no}., index - 1, d, n)$
step 15. $status \leftarrow IN_Q$
step 16. $end\ if$
step 17. $end\ while$

8. EXAMPLE:

This section clarification the ' Algorithms' via numbers in details.

Encryption:

Before starting in encryption process there is several things that the communications part must be agree on it, two prime number p and q in this example $= 11$ and $q = 3$.
$n = p * q = 33$
$\Phi(n) = (p - 1) * (q - 1)$
$10 * 2 = 20$
Each part on the communication has its own public key e (random number where : $0 < e < \Phi(n)$) and $(e, \Phi(n)) = 1$. Finally the cipher values are embedding in this example will used the elliptic curve with the equation: $y^2 = x^3 - x + 188 \bmod 751$,and using the design of MRADG in Appendix Table 4 Values of R set and Table 5 Values of Q set, The encryption process as shown in Table 2 .

TABLE 2 ENCRYPTION PROCESS

i	$message$	$State_{no}$	$Status$	$value$	$Point\ on\ EC$
0	0	2	IN R	19	(135,198)
1	1	29	IN Q	76	(533,74)
2	1	24	IN Q	48	(337,178)
3	0	4	IN R	64	(452,278)
4	1	28	IN Q	8	(57,332)
5	0	32	IN Q	2	(17,332)
6	1	16	IN Q	4	(30,236)
7	1	22	IN Q	75	(529,254)

The cipher is: {(50,136),(352,65),(291,16),(190,196),
(391,187),(131,34),(1,375),(170,274)}
And send the last $state_{no}.$ encryption by e $10^e \bmod n = 24$.

Decryption:

To explain the decryption process its start backward, the receiver was receives data was embedding into specific elliptic curve and cipher of last state number in this example : $State_{no}. = 28$, $= 17$, $n = 33$, $k = 7$.

Firstly compute the last state number that the decryption process is start from it, $State_{no.} = (28^{17} \bmod 33) - 8 = 11$, and the next phases of decrypt explain on Table 3

TABLE 3 DECRYPTION PROCESS

search	-	**True**	**True**	**True**	**True**	**False**	**True**	**True**	**False**
Status	-	IN R	IN Q	IN Q	IN R	IN Q	IN Q	IN Q	IN Q
Stat	17	22	16	32	28	4	24	29	2
message	-	1	1	0	1	0	1	1	0
value	-	75	4	2	8	64	48	76	19
Point on	-	(529,25	(30,236	(17,332	(57,332	(452,27	(337,17	(533,74	(283,54
index	-	7	6	5	4	3	2	1	0

9. AUTHENTICATION

Authentication is the method of proving one's identity to somebody else. And it's the most important topic in the security, it's provided by public key techniques in the following equation:

$M = D(E(M))$ Or $M = E(D(M))$

Where M is refers to Message, E refers to public Key and D refers to private key [1] [8].

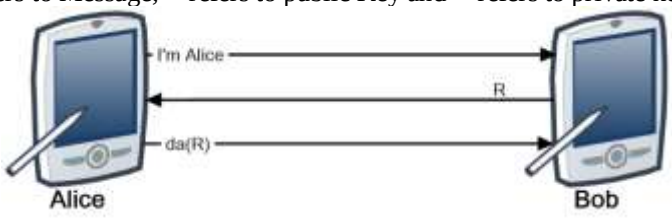

Figure 5 Authentication between two users

As illustrated in Figure 5 first Alice tells Bob he want to communication with her, Bob sends a number (R) she chosen at random, Alice encrypt the number R with his private key and send to Bob.at the final step Bob decrypt the encrypted value by Alice public key, if she have get the same value of R then she'll be sure of the identity of user who communicate with, and vice versa for authentication of Bob. Also the public key is provided the Digital Signature to get ensure for non-repudiation, the receiver verify the sender identity because message only decrypt be sender's public key, and it's only encrypted private key then the sender can't longer be able to repudiation his message [9].

10. INTEGRITY

To get the integrity of the message was sent the more efficient way to use one-way hash function $h = H(M)$, where h is more smaller than message M , H is the hash function and the stander on the internet is MD5 algorithm it's give 16 byte as output. It's impossible to find two message have the same hash value, the h value can be describe as flag to check the message was alter or not.

11. ANALYSIS

The RADG algorithm and MRADG have an important characteristic, they produces random cipher text in several exaction to same plaintext, to find the relational between the variance cipher texts as illustrated in Figure 6 by used Hamming distance [10] the algorithm applied eighty times on the same 100-bits plaintext the results is different in each execute this make the algorithm stand against the statistical attack compared to general encryption methods the Stander RADG is more efficient with Hamming Distance [7] but in term of speed as shown in Figure 7 developed algorithm MRADG is faster than RADG more than half, the black bars refers to the length of message in this example applied 20 sample the message length is duplicated in each time, the Red bars refers to the time executed by MRADG and obviously the time required implementing of encryption and decryption takes less than it takes in the RADG.

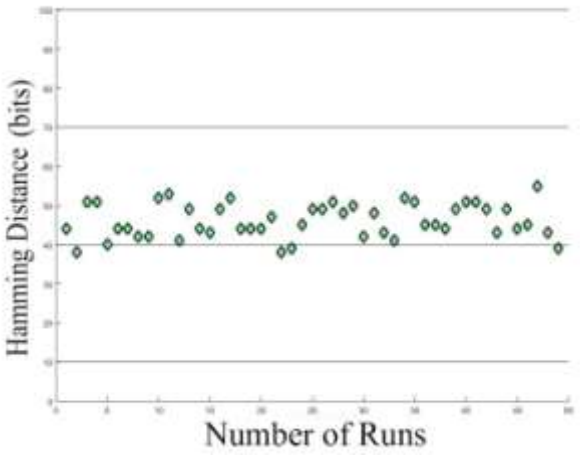

Figure 6 hamming weight

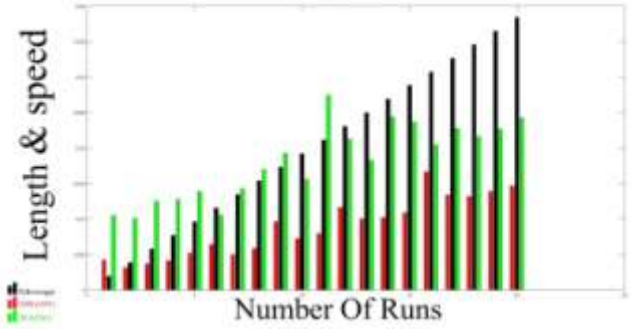

Figure 7 Run time of RADG and MRADG

12. CONCLUSION

In summary the proposed algorithm Increasing the range of networks that RADG implementation in, the stander RADG which applied on personal networks and the develop method MRADG apply on wider networks including intelligent networks CNRs by using key, Partition the Reaction states from single Reaction into Multi-Reaction states led to significantly increase the speed of preference of the algorithm with preserving the random property, The whole enhancement works to speed up the algorithm three times higher than it was in RADG, as disadvantages the algorithm is the length of the output cipher text considered in comparative relation to the input plaintext because of the encryption process take the plaintext in Binary form and each bit represented one point on EC.

REFERENCES

[1] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1997. Kaufman, Charlie and Perlman, Radia and Speciner, Mike, " Network security: private communication in a public world ",2002, Prentice Hall Press.

[2] Menezes, Alfred J and Van Oorschot, Paul C and Vanstone, Scott A, "Handbook of applied cryptography", 1996, CRC press.

[3] Kaliski, Burt, "The Mathematics of the RSA Public-Key Cryptosystem", 2006, RSA Laboratories.

[4] Stallings, William, "Cryptography and network security", 2010, Fifth Edition.
[5] Cohen, Henri and Frey, Gerhard and Avanzi, Roberto and Doche, Christophe and Lange, Tanja and Nguyen, Kim and Vercauteren, Frederik, " Handbook of elliptic and hyperelliptic curve cryptography ",2005, CRC press.
[6] Md Nizam Udin, Suhaila Abd Halim, Mohd Idris Jayes and Hailiza Kamarulhaili, " Application of message embedding technique in ElGamal Elliptic Curve Cryptosystem ",2012, pp. 1—6, IEEE .
[7] Albermany, Salah A and Safdar, Ghazanfar A, " Keyless Security in Wireless Networks "79, 3, 2014, pp.1713—1731, Springer.
[8] Boyd, Colin and Mathuria, Anish, " Protocols for authentication and key establishment ", 2013, Springer Science & Business Media.
[9] Rivest, Ronald L and Shamir, Adi and Adleman, Leonard, "A method for obtaining digital signatures and public-key cryptosystems ", 21, 2, 1978, pp. 120—126, Communications of the ACM.
[10] Olgica, Milenkovic," On the generalized Hamming weight enumerators and coset weight distributions of even isodual codes " , 2001, pp. 62, In IEEE international symposium on information theory.

Appendix: States values

TABLE 4 VALUES OF R SET

R set		
Number of state	*First value*	*Second value*
0	38	55
1	58	52
2	45	26
3	1	12
4	10	34
5	-	-
6	57	25
7	31	20
8	63	72

9	-	-
10	78	83
11	66	50
12	37	5
13	93	80
14	35	27
15	36	43
16	56	4
17	68	60
18	7	28
19	11	59
20	77	74
21	62	22
22	30	75
23	32	39
24	15	48
25	16	70
26	67	54
27	47	21
28	73	8
29	61	76
30	14	18
31	51	17
32	2	9

TABLE 5 VALUES OF Q SET

Q set		
Number of state	*First value*	*Second value*
0	40	71
1	33	3
2	19	29
3	46	65

4	46	65
5	23	53
6	84	81
7	85	91
8	90	86
9	89	92
10	44	69
11	88	79
12	87	82
13	49	24

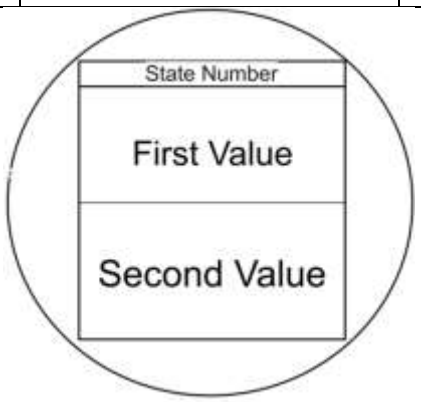

Figure 8 Example of State