Journal of Engineering and Applied Sciences 14 (Special Issue 1): 3866-3876, 2019
ISSN: 1816-949X

New Block Cipher Key with RADG Automata

Salah A. Albermany and Fatima Radi Hamade
Department of Computer, College of Computer Science and Mathematics, University of Kufa, Kufa, Iraq

Abstract: With the development of electronic communications RADG devices it has become increasingly important to protect data from violation, especially, large data. The main way to achieve this protection is using cryptography algorithms. Cryptography is the science of data encryption with the goal of reducing the availability of information to an analyst's cipher. In this study, we propose a novel design, i.e., block cipher keys by attempting to develop one such algorithm, i.e., develop the RADG method to the BRADG block cipher key while keeping the RADG attributes. BRADG (Block cipher Reaction Automata Direct Graph) is used in protecting wireless networks. BRADG processes data blocks of B bits with key length of B bits and gives cipher text of size B bits where B is 64, 128, 512, … , bits. BRADG is based on the unbalanced feistel structure in both encryption and decryption. In comparison to the previous design, ciphering of a new design is a faster and more efficient way to encrypt large data.

Key words: RADG automata, block cipher, S-boxes, unbalanced Feistel network, key schedule, encryption, decryption

INTRODUCTION

Cryptography is a science of converting text or messages into unreadable form and subsequently transmitting it across insecure networks. Cryptography can achieve this via. algorithms. Cryptography algorithms can be classified into symmetric key algorithms, e.g., AES, DES, Blowfish and Twofish and a symmetric key algorithms, e.g., RSA and ElGamal (Neelima and Mandal, 2015; Stinson, 2006; Coron, 2006; Alshahrani and Walker, 2015; Sharma, 2012; Pradesh, 2015). There is more than structure for a block cipher; In this study, we are concerned with the unbalanced feistel structure which was proposed by Schneier and Kelsey (1996), that is similar to the conventional feistel network. An unbalanced feistel network divides a block into two parts that are not equal in size. This change of the feistel cipher has interesting implications for designing ciphers that are secure against linear and differential attacks such as the block cipher algorithm CLEFIA and MacGuffin algorithm (Schneier and Kelsey, 1996; Blaze and Schneier, 1994). Wireless networks are some of the most important parts used in communication systems where the wireless network enables one or more users to communicate without physical connections. Wireless communication has become very popular in our business and personal lives. Because of the increasing number of wireless technologies users use in their environments such as wireless e-mail, web browsing and internet access it is necessary to reduce the security risks associated with wireless technologies. The loss of privacy and integrity are risks associated with wireless communications (Karygiannis and Owens, 2002; Huang *et al*., 2009). In this study, we propose a new secure communication for wireless networks by attempting to develop an already existing algorithm, i.e., develop the RADG method to a new block cipher key called BRADG (Block cipher Reaction Automata Direct Graph). RADG schema were presented by Albermany and Safda (2014) where the researchers presented a new different type of ciphering based on the original message to achieve the security operations with integrity, privacy, authentication and non-repudiation.

RADG mathematical modeling is influenced by graph theory and proves the soundness of ciphering where the process of breaking the code within a large system requires significant effort compared to the schemes of classical cryptography. Because the focal points in the science of data encryption reduce the availability of information to the analyst cipher, we use a secret key in the proposal design. A new design will keep the same number of bits in the encryption and decryption because the new block cipher will be dependent on using the round function of an unbalanced feistel cipher after each cipher process of the RADG method. The novel design

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

depends on the use of substitution boxes because the use of S-boxes affects the cryptography strength where the S-boxes are without backtracking and the key is used to decrypt the cipher text. The RADG method provides features to the proposed block cipher, i.e., the same plaintext gives more than one different cipher text where RADG ciphering will be random.

MATERIALS AND METHODS

BRADG mathematical model: The BRADG design is based on the RADG mathematical model (Albermany and Safda, 2014) it keeps the same features and characteristics of the RADG model with a block cipher key. A basic design of BRADG consists of (Q, J, R, H) where Q is a set of standard states with $|Q|\geq 2$ states, J represents a subset of Q called jump states with $J \subset Q$ such that $|J| \leq |Q|/2$, R represents a finite set of reaction states and H represents hidden states where $H\cap (Q \cup J) = \varnothing$ with $|Q \cup J| |H|$ (Theorem 1). Each state in Q/J and R have λ distinct values where $\lambda = 2^x$ with x representing the size of the input block such that $x \geq 2$ where x = 1 represents the RADG modeling by Albermany and Walker (2015). Value of state also has a bit size dependent on n (standard states), m (reaction states), k (jump states) and x (size of input block) this means that n = |Q|, m = |R| and k = |J| such that the k that appears in Eq. 2 for this in the proposed design can define the value V, denoted as the number of values in the design that can be calculated by solving $\lambda (n+m-k) = 2^x (n+m-k)$. The reason for making k negative is that doesn't does not have any values then the number of bits in the design is calculated as follows:

$$\log_2\left(2^x(n+m-k)\right) = \log_2 2^x + \log_2(n+m+k) = x+\log_2(n+m-k) \quad (1)$$

Because $\log_2(2^x(n+m-k))$ is a non-integer value $||$ to make the an value integer number then the number of bits in the value of state, denoted as V can be calculated by the solving following Eq. 2:

$$V = x+\left\lceil \log_2(n+m-k)\right\rceil k \leq \lfloor n/2 \rfloor \quad (2)$$

where, $\lfloor\ \rfloor$ deneoted denotes the maximum integer number, $\lceil\ \rceil$ denoted denotes the minimum integer number n = |Q|, m = |R| and k = |J| such that $k \leq \lceil n/2 \rceil$. Then, from Eq. 1 and 2, suppose x (size of the input block) can be determined as solving the following Eq. 3:

$$x = 2\times\left\lceil \log_2(n+m-k)\right\rceil \quad (3)$$

Because $\lambda = 2^x$ then $x = \log_2\lambda$. For this reason, we can prove the equation of x as follows:

$$\lambda = 2^{2\log_2(n+m-k)} = 2^{(\log_2(n+m-k))^2} \cong (n+m-k)^2$$

Then:

$$2\left[\log_2(n+m-k)\right]$$

Where:

$$\log_2 2^x \cong 2\left[\log_2(n+m-k)\right] \rightarrow x = 2\left[\log_2(n+m-k)\right]$$

Theorem 1: $|Q\cup J|\geq|H|$

Proof: Because: $|Q\cup J| = n+k$ and $\log_2(n+k)\leq\lceil\log_2(n+k)\rceil$

Then:

$$1+\log_2(n+k)\leq\log_2(n+k)$$
$$\log_2 2+\log_2(n+k)\geq\lceil\log_2(n+k)\rceil$$
$$\log_2(2(n+k))\geq\lceil\log_2(n+k)\rceil$$
$$2(n+k)\geq 2^{\lceil\log_2(n+k)\rceil}$$
$$(n+k)\geq 2^{\lceil\log_2(n+k)\rceil}-(n+k)$$

From the above, we prove that: $|Q\cup J|\geq|H|$

The BRADG operations are performed on a sub block of x bits. The BRADG algorithm divides block message M of size B bits into |M|/x sub messages ($m_0, \ldots, m_{|M|/x}$) where each sub block has size x bits. The BRADG algorithm encrypts each x bit using a random RADG state selection where we obtain ciphering values V. These values are used as input in the round function to obtain cipher text C of size x bit. Transitions for ciphering all sub blocks will depend on two addresses: one for standard states Q, denoted by the symbol ad and the other for random state R (Reaction states), denoted by the symbol adr. The reason for using two addresses instead of one is to ensure that each address in Q (Standard states) moves to R (Reaction states) only through J (Jump states) to keep randomness where the address of Q states ad can be calculated by solving the following equation:

$$ad = \left\lceil \log_2(n+k)\right\rceil \quad (4)$$

The reason to use hidden states |H| in the new design is to ensure obtaining all probabilities of ad-bit, e.g., if ad consists of 4 bit and $|Q\cup J| = 10$ states, note that the number of probabilities for ad-bit is $2^4 = 16$ probabilities. For this, there are 6 states that will be hiding states called|

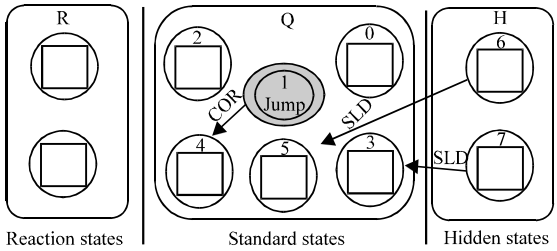

Fig. 1: BRADG design example

H| such that if ad has bits they are not in |Q| and should be in |H| and each state in |H| has a transition called a slide, denoted as SLD, used to return the address ad to a certain state according to the SLD transition of |H| states in the design the number of hidden states |H| can be calculated by solving the following equation:

$$H| = 2^{ad}-(n+k) \quad (5)$$

where, 2^{ad} the total number of states in (H ∪ Q ∪ J).

Theorem 2: ad < x such that 0< k ≤ ⌊n/2⌋.

Proof: Because the maximum value of k is ⌊n/2⌋,. Suppose that k = ⌊n/2⌋ because the minimum value of m (reaction States R) is zero. Suppose that m = 0.
Let $n = 2^z$, where z >0 then:
Using Eq. 3, $x = 2\lceil \log_2(2^z-2^{z-1})\rceil$
$2\lceil \log_2(2^{z-1})\rceil = 2(z-1)$
Using Eq. 4, $ad = 2\lceil \log_2(2^z-2^{z-1})\rceil =$
$\lceil \log_2(3\times(2^{z-1}))\rceil = \lceil \log_2 3+\log_2 2^{z-1}\rceil = \lceil 1.584+(z-1)\rceil = z$

Then, from the above equations, we prove that 2(z-1)>z where z≥3. This means that ad < x. There is another transition from the J-Q state called corresponding, denoted by COR this transition is used when the R state moves randomly to the Jump state and in this case, must be returned to the transition to a certain state in Q (Standard state) (because the R state must make the transition to the Q state and not J state for this reason, that corresponding to the Q state is used).

Example 1: If = 5, m = 2, k = 1 then the BRADG design is as shown in Fig. 1. where ad = 3 Eq. 4 and |H| = 2 Eq. 5. During the encryption process for the state address, two bits must be added to the address of state these two bits represent the composition of the address location. Then the new address is encrypted with the key bit (the number of key bits depends on the number of address bits). Two bits added to the address of state take the following form:

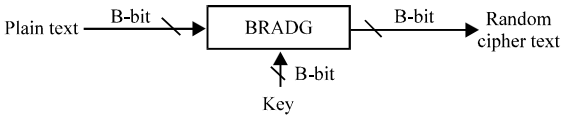

Fig. 2: BRADG algorithm

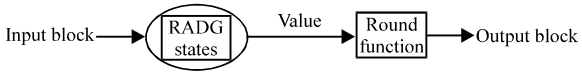

Fig. 3: General design of BRADG encryption

- Q state takes 00 bit as location
- R state takes 01 bit as location
- J state takes 10 bit as location
- H state takes 11 bit as location

BRADG algorithm: A novel design of the private key block cipher BRADG (Block Cipher Reaction Automata Direct Graph) processes data blocks of size B bits with key length of size B bits where B = 64, 128, 512, … , bits as shown in Fig. 2. The BRADG algorithm is a strength because using the RADG method makes the encryption operation random, depending on the original message and the same plain text can give more than one different cipher texts. The new design uses substitution boxes that give the cryptography strength because S-boxes without backtracking and substitution boxes are a nonlinear transformation that performs the confusion of bits.

BRADG design: A new design of the RADG block cipher key, shown in Fig. 3, consists of a round function based on the unbalanced feistel structure which divides the input block into two halves (L_0 and R_0) and that are not equal in size where the size of Left half L_0 and Right half R_0 will be: L_0 = x/2 bits, R_0 = x bits (Fig. 3). Operations of each round in the ciphering process are explained in Fig. 4 where the number of round encryptions depends on the size of input block x, e.g.: if x = 8 bits (size of input block) and M = 64 bits (size of data block) then the number of round encryptions will be 8 according to the M/x number of rounds in the design also if M = 128 bits, note that the number of rounds will be 16 and so on. The reason for using the M/x number of rounds goes back to the original design of RADG by Albermany and Safda (2014) where the RADG design uses = 1 we use M rounds because M/1 = M.

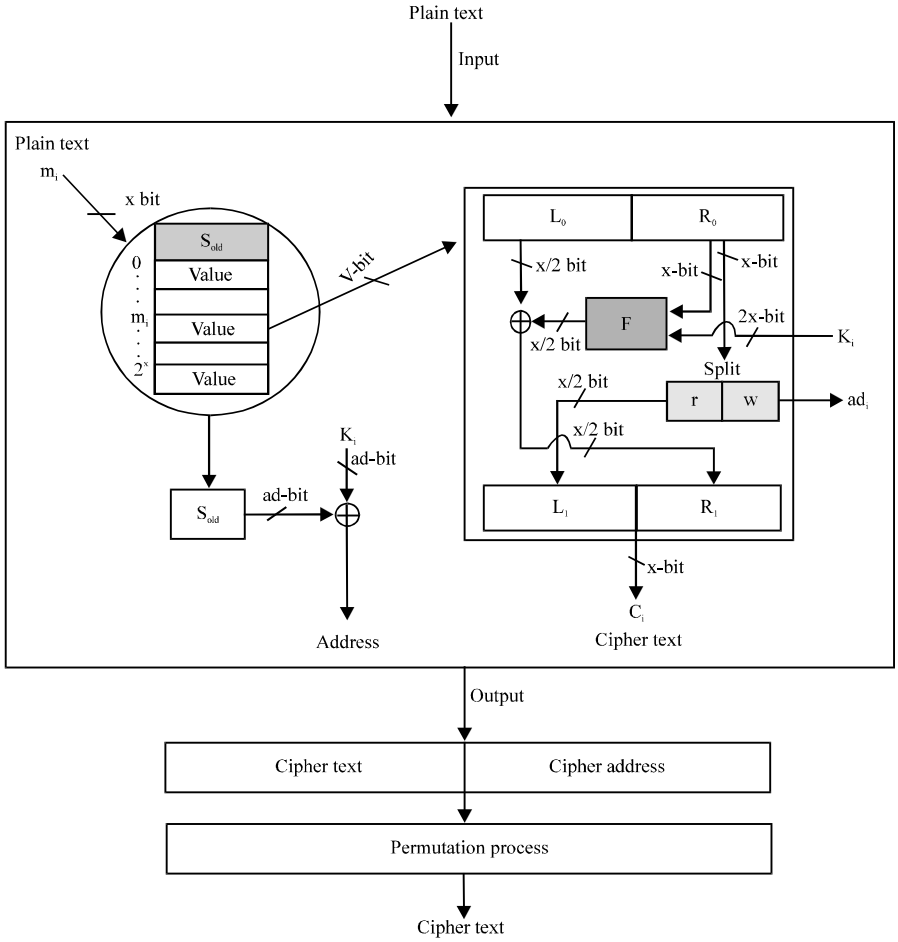

Fig. 4: Round encryption process

Table 1: Round encryption notations

Notations	Details
S_{old}	The previous state generated randomly where Sold∈QR
Value	The value of the RADG state where the size of the value V-bits
x	Size of input block m_i of plain text where I = 0, …. , |M|/x
L_0	The Left half of the round function
R_0	The Right half of the round function
2	Number of value in each state where
m_i	Input sub block of plain text where i = 0, … , |M|/x
F	Substitution Function (F Function) where F = F(R0, ki) and ki is the sub key
r	The Left half of R_0
w	The Right half of R_0 where the size of w is ad bits
K_i	The sub key of size 2x-bit
ad	The address of the next state where ad = w
L_1	The Left half of the cipher text where the size of L1 is x/2 bits and L_1 = r
R_1	The Right half of the cipher text where the size of R_1 is x/2 bits

All notations of Fig. 4 are explained in Table 1. From Fig. 4, note that the output of the encryption process using the BRADG design consists of two parts. The first part represents the cipher text of input block, denoted by cipher text C while the second part represents the cipher text of address state, denoted by cipher address where the address of the RADG state is ciphered by adding it to the same number of key bits (ad bits) taken from the left half of the sub key to obtain the address cipher. This address cipher is used in the decryption process to obtain the address number of states which is used to encrypt a certain input block such that if the address is in the R set (Reaction state) in this case, we must encrypt the address of the J state which causes the random case in those states of the R set to decrypt the cipher address if the address belongs to the J state, this means that the address returns to the state in the R set in this case, a search in R states must be performed to know the correct state in the R set.

BRADG state: The state of the BRADG design consists of a number of value denoted by λ and each state has a number that represents the address of the state, denoted

by S_{old} where this address is encrypted by XORed with a sub key generated previously to obtain the cipher value, called the cipher address. The states of design are denoted as reaction states |R|, standard states |Q| and Jump states |J| which is a subset of |Q| and Hidden states |H|. Both |J| and |H| do not have any value |J| is used to generate a random operation and |H| is used to return the address to standard states when the address is out of range. The encryption process starts by selecting a state randomly from standard states |Q| such that it does not start with the jump state and encrypts each x bit according to the indexed x to obtain the value V which is used as the input in the round function.

RESULTS AND DISCUSSION

BRADG round function: The round function of the BRADG algorithm design is based on the unbalanced feistel cipher which divides the value of the RADG state, denoted by the symbol V into two halves not equal in size (Schneier and Kelsey, 1996). The left half, denoted by L_0 is of size x/2 bits while the right half, denoted by R_0 is of size x bit then the right half will pass into two operations: the F Function and the split operation where the supported key length of size 2x bits represents the sub key generated previously from the master key. Operations of each round in the ciphering process are explained in Fig. 4 where the output of the F Function is XORed with Left half L_0 to obtain the right half of the cipher text, denoted by R_1 of size x/2 bits the left half from the split operation will represent the left half of the cipher text also of size x/2 bit, denoted by L_1. Finally, the output of each round will consist of two parts: the first part will be the cipher text of input block x while the second part will be the address cipher which is used in the decryption process to decipher the same cipher text that appeared with it during the encryption process.

BRADG function: The Round function divides the value obtained from the RADG state into two halves, i.e., left half R_0 according to the unbalanced feistel structure rule. The Right half R_0 is used as the input block of the F Function which takes two input Right half R_0 of size x bit and the sub key of size 2x bit generated previously. The operations of the F Function are shown in Fig. 5 where we note that the F Function performs the Expansion process on the Right half R_0 of with x bit block size to obtain bit as the output of the expansion function. The reason for expanding input block R_0 is to obtain the same number of

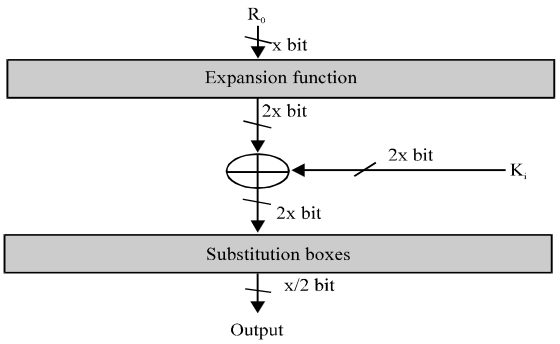

Fig. 5: F Function

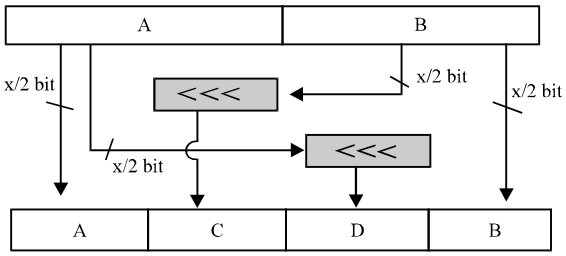

Fig. 6: Expansion function

bits as key length 2x bit to perform the XORed operations between them. The output from the XORed operations between the sub key and output from the expansion operation will be 2x bit and should shrink to shrink tox/2 bit for this reason, the output of the XORed operation is used as the input for the Substitution function (S-boxes). The substitution function (F Function) is calculated only in one direction such that it makes the use of the operations of the F Function inefficient in the inverse direction (i.e., it cannot invert the operations of the F Function to start from S-boxes to the expansion process to obtain the original value of R_0).

Expansion function: The expansion function expends x bit into 2x bits where x bit is divided into two halves A and B of x/2 bits then it rotates to the left by one bit both A and B to generate D and C, respectively which are also of size x/2 bit. It also uses A and B to generate 2x bits from A, C, D, B, respectively, as show in Fig. 6.

Substitution function: The fixed structure of S-boxes elements can be implemented in any block cipher system. S-boxes are considered a critical step in any block cipher system where substitution of S-boxes is responsible for confusion in the encryption process. Many secure block cipher algorithms use S-boxes in their design such as

DES, Twofish, Lucifer, Blowfish and other encryption algorithms (Schneier, 1996; El-Ramly *et al*., 2001). Matt Blaze and Bruce Schneier presented by Blaze and Schneier (1994) a new block cipher called the MacGuffin block cipher algorithm. The MacGuffin algorithm uses 8 S-boxes S_1-S_8. The researchers generated MacGuffin S-boxes directly from adopted DES S-boxes in a certain way to use in MacGuffin S-boxes where the latter takes 6 bits of input and returns two bits as output. Because MacGuffin needs only 2 bits as output, the researchers use only the outer two output bits from each S-box. Biham *et al*. (1998) presented a new block cipher algorithm called serpent to satisfy the Advanced Encryption Standard (AES) requirements. The researchers used in this algorithm 32 rounds where each round uses only one S-box whereas the Serpent algorithm uses 32 S-boxes 4→4 mapping where four input bits are mapped to four output bits. The 32 S-boxes are chosen from DES S-boxes as a separate line where S_0 of the serpent algorithm represents the first line of DES S-box S_1, S_1, represents the second line of S_1 and, so on until S31 of the serpent algorithm which represents the last line of DES S-box S8 (Biham *et al*., 1998). In the BRADG algorithm, we will apply the same principle by using DES S-boxes that will generate two BRADG S-boxes S1 and S2 of 8→4 mapping where each of the four DES Substitution boxes will be used to generate one S-box of BRADG such that S1-S4 of the DES S-boxes will be used to generate S1of the BRADG algorithm where the first column of each DES S-box will represent the first line of S1 for the BRADG algorithm, respectively, each second column of each DES S-box will represent the second line of S1 and so on until S1 of the BRADG algorithm is generated. In the same way it will generate S2 of the BRADG algorithm but by using the other four DES S-boxes S5-S8.

Split operation: The split operation on the Right half R_0 of the round function divides x bit of R into two parts: the first part, denoted by r and the second part, denoted by w where both r are w of size x/2 bit. This means divide the size of R_0 by 2 to obtain the size of both parts. The first part r will be used as the left half of cipher text L_1 the second part w will be used as address ad of the next state to encrypt the following sub message (next input block) in the same way as explained above.

BRADG permutation process: The final step in the encryption process is the permutation process where the ciphering will represent two parts of the cipher text, denoted by C and cipher address. To hide the bits of address within the cipher text, Permutation is performed after ending the encryption process for all plaintext. The permutation process for bits will be as in the layout in Fig. 7. The permutation in Fig. 7 is an example of a 16 bit cipher output where the index of Table 2 ranges from 0-15 as follows:

Table 2: Permutation process

Index of cipher output	Index after permutation
0	15
1	8
2	1
3	2
4	3
5	4
6	5
7	6
8	9
9	10
10	11
11	12
12	13
13	14
14	7
15	0

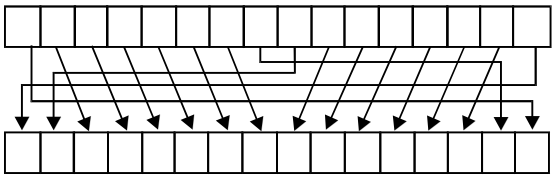

Fig. 7: Permutation process

BRADG key schedule: The BRADG algorithm uses a master key of size B bits to generate x sub keys of size 2x bit. An algorithm key schedule performs some of the algebraic operations such as the bitwise XOR, addition OR and rotate left (Shift left cycle) in computing the secret sub keys. In the generation, we achieve two general principles of block ciphers: confusion and diffusion. The mixing between operation of the algebraic groups helps to achieve the confusion principle while diffusion achieved by rotating left (shift left cycle) each sub key according to the number of shifts in Table 3 where rotating left rearranges the positions of key bits (permutation/transposition). The idea is to spread the influence of a single key bit over many bits of output (Masuda *et al*., 2006). The encryption process uses only unique keys to encrypt each x bits of plaintext to ensure a secure cipher against attacks. Figure 8 explains the

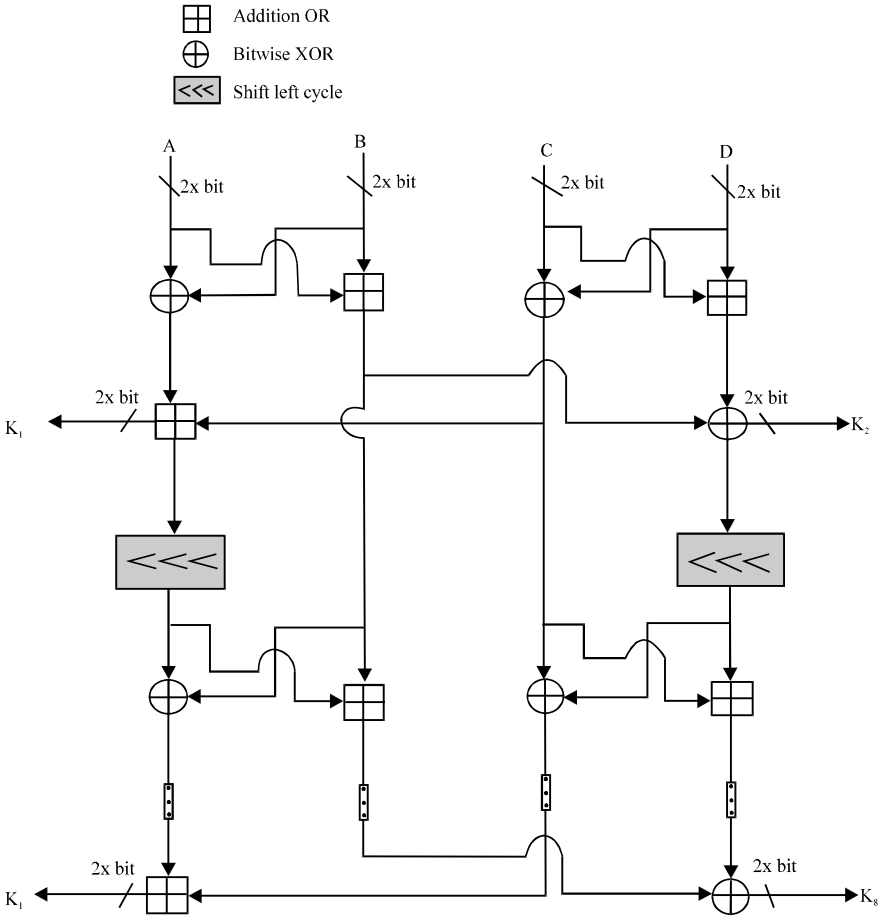

Fig. 8: BRADG key schedule

Table 3: Shifting table

Key number	Shifting number
1	1
2	3
3	3
4	3
5	3
6	3

design of generating the secret sub keys. Figure 8 is an example of generating eight secret sub keys, each of which has size 16 bit where a master key of size 64 bit is used. The key schedule design splits the master key of size 64 bit into four Parts: A-D all of size 16 bit and uses these parts to compute the secret keys according to Fig. 8.

Key schedule notations:

- A-D←k copies data bit from 64 bit k into 4 registers A-D of size 16 bit
- ⊕ the bitwise exclusive-or bit by bit module 2
- ⊕ Addition or operation bit by bit module 2
- <<< rotate left number of bit according to shift Table SH
- ←assignment Operator
- Key←0 initialization String Array Key used to store sub keys of size 16 bit in each position of array Key

Algorithm 1; Key schedule algorithm:

Input: the key K of size 64 bit, where SH means shifting table
Output: the sequence of secret sub keys ki of size 8 bits, where i = 0, … , 8
Key ← 0; n ← 1; A, B, C, D ← K0, … , 63
for i =1-8 do
for j =1-16 do
t [a, b, c, d] ← (A⊕B, A⊕B, C⊕D, C⊕D
L← (a⊕c); R ← (b⊕d); end
Key[i] ← L; Key[i+1] ← R; A ← (L<<< SH[n])
D ← (R <<< SH [n+1]); B ← b; C ← c; i ← i + 2
n ← n + 1
end
return K

Implementation: This study provides algorithms of BRADG encryption and decryption, as well as of the F Function. All algorithms employ the notations outlined in Table 1 and 4.

Table 4: BARDG implementation notations

Notations	Details
q_{rand1}	State of rand1 generated randomly between {0,…, |Q|-1}
q_{rand2}	State of rand2 generated randomly between {|Q|, … , |Q|+|R|-1}
Permutation (C)	Permutation step to change the index of bits in cipher text
Ctxt	Variabe used to hold the cipher text after separating it from the address where C_L involves cipher text of size x-bit and address
Add (C, B_L)	Add B_L to string of data C
Search(C_L,Q)	Search in Q to determine whether the state has the value C_L
F(R_0,K_L)	Function F performs some operations in one direction on the Right half R_0 of the round function
T (S_{old}, m_L)	Transformation T to find the corresponding value of input block m_L in state S_{old}
R	Reaction states
ad	The address of the state
L	Number of sub blocks of input Message M
|M|/x	Number of input sub blocks of Message M
T^{-1}	Inverse Transformation of T
Short Path (Sold, q, J)	A random short path between Sold and q, where q ∈ J
C_F	The final cipher text
S	Hidden States in Q
Search path (BL, S)	Search about path between Q state and the hidden states
M	Reverse sequences of P from p_0, p_1, … , $p_{|J|}$ to $p_{|J|}$,$p_{|J|-1}$, …, p_1,p_0 then saved in M

Algorithm 2; F Function algorithm:

Input: Left half R_0 where R_0 is of size x-bit and the Key K is of size 2x bits
Output: Y of size x/2 bit
Step 1: A, B←R_0; C←(B <<< 1); D←(A <<< 1) E←A, C, D, B; Step2: F←ki⊕E
Step 3: Write F in the form B1, B2, ... where each B is a group of 8 bits
Step 4: X←S1(B1); Y←S2(B2)
Step 5: R←X⊕Y; Step 6: Return R

Algorithm 3; BRADG encryption algorithm:

Input: Message M is a sequence of block {m_0, m_1, …, $m_{|M|/x}$}, where the block size is x bits and the Key is a sequence of block {k_0, k_1, …, $k_{|M|/x}$}
Output: The sequence of blocks C←{C_0, C_1 , ...,}
Step 1: L ← 0; S_{old}←qrand1; B←φ; C←φ
Step 2: if (L > |M|/x), then go to step 6 ; T (S_{old}, m_L)→C_L
Step 3: L_0, R_0←C_L; R_1←L_0⊕F (R_0, K_L) r, w←R_0, r←L_1, w←S_{new}; k_{LL}, k_{LR} ← k_L
if (S_{old} ∈ R) {Lo = 01; S_{old} = merge (S_{old}, Lo)}
elseif(S_{old} ∈ J){Lo = 10; S_{old} = merge (S_{old}, Lo)}
elseif(S_{old} ∈ H); {Lo = 11; S_{old} = merge (S_{old}, Lo)}
else {Lo = 00; S_{old} = merge (S_{old}, Lo)}
address ← S_{old} ⊕ k_{LL}; R_1, L_1, address → B_L
Step 4: Add (C, B_L); L→L+1; goto step 2
Step 5: if (S_{new}∈J &&L ! = |M|/x){Sold←q_{rand2};goto step 2}
else if (S_{new} ∈ H){S_{old}←path (S_{new}, q)}
else {S_{old}∈ S_{new}}
Step 6: if (S_{old}∈J) then goto step 7
SP←ShortPath(S_{old},q,J); M'←a1,a2, ... ,a|SP|
for (I = 1; I< = |SP|; ++i) T (S_{old}, ai)→(S_{new}, Si)
if (I! = 1) then Add (C, Si); Sold←S_{new}
Step 7: C_F←Permutation (C)
Step 8: return C_F

Algorithm 4; BRADG decryption algorithm:

Input: the block sequence C←{C_0, C_1 ,..., $C_{|C|/x}$}, the key is a sequence of block {k_0, k_1, ..., $k_{|M|/x}$} where the block size is 16 bits. Sa is the address
Output: the block sequence M←{m_0, m_1, ...,}
Step 1: L←|C|; M←φ; P←φ; S_{new}←Search (C_L, Q)
Step 2: if (L <0) then goto step6
C_{txt}, ad←C_L; L_1, R_1←C_{txt}; w S_{new}, r←L_1, r
w→R_0; L_0← R_1⊕F (R_0, K_L); S_{oldL}←ad⊕K_L
Lo, Sold←S_{oldL}, L_0, R_0←B_L; (S_{old}, B_L)→(S_{new}, m_L)
Step 3: Add (P, m_L); L←L-1;
Step 4: if (S_{new}∈R); S_{new}←Search (C_L, J); goto Step 2
Step 5: if ((S_{old}, ∈B_L) = ∅) {S_{new}←SearchPath (B_L, H) T^{-1}(S_{old}, B_L) (S_{new}, m_L); Add (P, m_L) }
Step 6: M = reverse Sequence (p_0, p_1, ..., $p_{|J|}$); return M

Application of BRADG-SECURE: The BRADG algorithm (block cipher reaction automata direct graph) converts the wireless network from the personal wireless communication applied in the RADG method (Albermany and Safda, 2014) into an extensive wireless communication network where more than two users can have the RADG design on the same network but with different keys this provides security against possible attacks and can be applied to achieve secure wireless communication networks between any two users connected in the network where for any two users to communicate there must be first an agreement regarding the use of the key which is then communicated on the network such that a third user or parity that also has the RADG design cannot communicate the discovery of the message sent because the key will be unknown. Figure 9 explain show to connect three nodes in a wireless network where each two nodes can be connected securely and a third node cannot know the message because all nodes in the network have the RADG design but different keys.

Performance and security analysis: The performance and security analysis of the BRADG algorithm (RADG block cipher key) is explained in example 2.

Example 2: If n = 10, m = 5 and k = 2 then calculated the following:

$$x = 8 \text{ bit}, v = 12 \text{ bit} = 2x = 28, L_0 = 4 \text{ bit}$$
$$R_0 = 8 \text{ bit}, ad = 4 \text{ bit}, |H| = 4 \text{ state}$$

Note that, $\lambda = 2^8$ means that each state of the BRADG design has 256 values. The first value in each state represents the state number and the other 255 values are used in data encryption where the second value in each state has zero index three have one index and, so, on until the last value in each state is 255.

Security analysis

Confidentiality: Data confidentiality means ensuring that information cannot be made available to unauthorized individuals (Stallings, 2011). The confidentiality of data can be proven by considering example 4: if assume the input M (the Message) and K (the Key) where both M and K are of size 64 bits where:

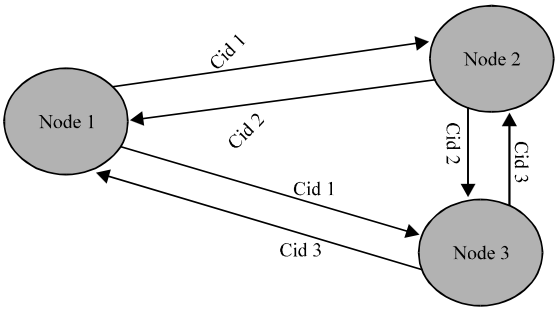

Fig. 9: Wireless network communication

M = 01010101110010011100001101110010011000101001111001001001111111111

K = 00110001001100100011001100110100001100100011001100100100001001010

The encryption process includes two parts:

Key generation: The steps for generating 8 sub keys of size 16 bits are:

Step 1: Split the Key into 4 parts and sort them into four registers A-D:

A = 0011000100110010, B = 0011001100110100
C = 0011010100110110, D = 0100100001001010

Step 2: Perform several operations bitwise XOR (⊕) and Addition OR (⊕) (bit by bit module 2):

a_0 = 0000001000000110, b_0 = 0011001100110110
c_0 = 0111110101111100, d_0 = 0111110101111110
k_1 = 0111111101111110, k_2 = 0100111001001000

Step 3: Rotate left (Shift left cycle) both k_1 and k_2 according to the shift table (Table 3) where k_1 is rotated by 1 position, k_2, by 3 positions and, so on, until all secret keys which will be used in data encryption are generated. Repeat all steps above to generate all sub keys needed in the data encryption process:

k3 = 1100111111111110, k4 = 1000000010000000
k5 = 1000101100111010, k6 = 1111000011000000
k7 = 1010111100111111, k8 = 0111000011000001

Data encryption and decryption

Encryption: The encryption process starts randomly to select the start state in encryption data these states should be in the Q set. During the encryption process if the state is in J (Jump state) it goes to the R set (Reaction states). The encryption process always finishes in the Q set, where the cipher text is returned. If the state is final and it is jump in this case it will take a short path between the last state in the Q set and the jump state. Table 5 shows the results of the encryption process of message M and key K according to Fig. 4 and the F Function in Fig. 5. The output of the cipher text with the address is:

C = 1000100011100011111111000011011101001000101110001010011110001001001101100011011110000011010110001 0

The final step in the encryption process of data is permutation of the cipher according to the permutation layout in Fig. 7 mentioned previously.

Decryption: In the decryption process, the receiver will reverse all operations of encryption, except that the short path will be deleted before the decryption process. In the decryption process, the same key used in the encryption process is used. The first step in decryption is searching to find the new state that represents the address of the correct value in the design. Table 6 shows the results of the decryption process of cipher text C and the key K.

Data integrity: Integrity means that data are not modified by malicious nodes when sent from one node to another (Wang *et al*., 2006). The most efficient way to ensure integrity of data is by using a session key or secret key where two users agree on the use of a common key, thus, key exchange between the sender and receiver to encrypt data by sender and sent to receiver uses the same key to decrypt data. Use of a secret key in ciphering data ensures the integrity of the data where the data of the cipher text cannot be modified by unauthorized users because the key will be unknown our algorithm uses the transitions function which cannot respond to any modification of data of the cipher text.

Authentication: Authentication is the investigation of the identity of the party who generated data (Neuman and Ts'o, 1994). Cryptographic block cipher algorithms rely on keys during the authentication phase which ensure that the authentication of data is CBC-MAC whose method is to generate a message authentication code of a block cipher by using the following equation (Petrank and Rackoff, 2000).

$$f_k^{(L)}(m) = f_k\left(f_k\left(,..., f_k\left(f_k(m_1)\oplus m_2\right)\oplus,..., \oplus m_{L-1}\right)\oplus m_L\right)$$

where message m = (m_1, m_2, ... , m_L) and k is the key. To ensure authentication of message ciphering using the BRADG method, we will depend on using the CBC-MAC method (Petrank and Rackoff, 2000) which encrypts a

Table 5: Encrypt the message

C	Address	S	S_{new}	J	Value	S_{old}	Key	M
10001000	111000		2		110110000010	9	0111111101111110	01010101
00111111	1100001		13	J	000000111101	17	0100111001001000	11001001
01101110	100111	S	2		000001100010	5	1100111111111110	11000011
00010111	0001010		10	J	000100011010	18	1000000010000000	01110010
10011110	001000		7		101110010111	10	1000101100111010	01100101
01001101	100000		10		110101001010	7	1111000011000000	01001111
11011110	000000		6		110111010110	10	1010111100111111	00100100
11010110	001011		8	SP	011111011000	5	0111000011000001	11111111

Table 6: Decerypt the cipher text

M	S_{old}	T^{-1} or search ()	S_{new}	Key	Address	C	Values
11111111	5	Search	8	0111000011000001	001011	11010110	0
00100100	10	T^{-1} is false then search in Q	6	1010111100111111	000000	11011110	1
01001111	7	T^{-1} is True	10	1111000011000000	100000	01001101	2
01100101	10	T^{-1} is True	7	1000101100111010	001000	10011110	3
01110010	18	T^{-1} is false then search in R	10	1000000010000000	00010	00010111	4
11000011	5	T^{-1} is false then search in S	2	1100111111111110	100111	01101110	5
11001001	17	T^{-1} is false then search in R	13	0100111001001000	1100001	00111111	6
01010101	9	T^{-1} is True	2	0111111101111110	111000	10001000	7

message using the BRADG block cipher algorithm in CBC mode with a secret key to generate a sequence of blocks, each of which is dependent on the previous encrypt block in this way ensuring that the key cannot be predicted. This sequence of blocks is called MAC the sender generates MAC of the message, encrypts the message and then sends both MAC and the encrypted message to the receiver. On the other side, the receiver will decrypt the ciphering of the message, generate MAC for the message after decryption then compare between the received MAC and MAC after decryption to ensure authentication of message where if two MACs are equal this means that the authenticity and integrity of the message are ensured.

Non-repudiation: The term non-repudiation refers to preventing the sender of a message from denying the sending of the message or the authenticity of his signature upon the sending of the message (Oppliger, 2007). The basic non-repudiation services are Non Repudiation of Origin (NRO) provides the receiver of a message with evidence to prevent the riginator (sender) from denying sending the message and Non-Repudiation of Receipt (NRR) provides the sender a message with evidence to prevent the receiver (recipient) from denying receival of the message (Zhou and Gollmann, 1997). The BRADG algorithm refers to Non-Repudiation of Origin (NRO) where the sender (origin) cannot deny sending the message.

Performance analysis: Hamming distance is used in information systems to measure and investigate the performance of cryptosystems (Milenkovic, 2001). In the BRADG method, hamming distance is used to show the difference between different cipher texts for the same messages run 100 times. Increasing the difference between cipher texts is performed to make breaking of the code very difficult compared toot her schemes in cryptography. Figure 10 shows the hamming distance of different cipher texts for the same message (Adill, 2015; Ali *et al*., 2015; Hamid *et al*., 2012; Khan *et al*., 2016; Tayel and Shawkey, 2014).

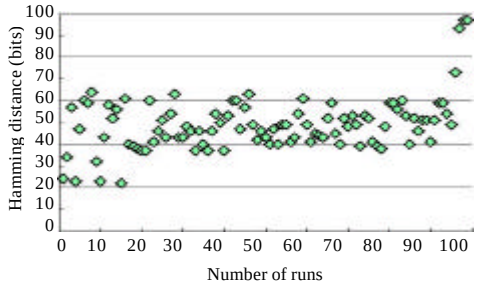

Fig. 10: Hamming distance for messages run 100 times

CONCLUSION

This study presented a novel design of a block cipher key called BRADG. BRADG is a new Block Cipher Reaction Automata Direct Graph with B bit block size (B-bits plaintext), B-bit cipher text and supported key length of B-bits. The round structure of the BRADG algorithm design is based on the unbalanced feistel cipher. The BRADG algorithm is used in protecting a wireless network, enabling two users to communicate across an insecure channel to transmit data from one node to another. The BRADG algorithm is dependent on using cryptographic keys to perform security requirements such as confidentiality, data integrity, authentication and non-repudiation.

IMPLEMENTATIONS

In implementation of the BRADG algorithm, the same plaintext can yield different cipher texts and the encryption of plaintext will be randomly. The new design of the block cipher key is efficient enough to encrypt large data.

REFERENCES

Adil, A.R., 2015. Text steganography to border image using novel method. Appl. Math. Sci., 9: 3087-3096.

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

Ali, N.H.M., A.M.S. Rahma and A.S. Jamil, 2015. Text hiding in color images using the secret key transformation function in GF (2 n). Iraqi J. Sci., 56: 3240-3245.

Alshahrani, A.M. and S. Walker, 2015. New approach in symmetric block cipher security using a new cubical technique. Intl. J. Comput. Sci. Inf. Technol., 7: 69-75.

Biham, E., R. Anderson and L. Knudsen, 1998. Serpent: A New Block Cipher Proposal. In: Fast Software Encryption, Vaudenay, S. (Ed.). Springer, Berlin, Germany, pp: 222-238.

Blaze, M. and B. Schneier, 1994. The MacGuffin block cipher algorithm. Proceedings of the 2nd International Workshop on Fast Software Encryption, December 14-16, 1994, Springer, Berlin, Germany, pp: 97-110.

Coron, J.S., 2006. What is cryptography ?. IEEE. Secur. Privacy, 4: 70-73.

El-Ramly, S. H., T. El-Garf and A.H. Soliman, 2001. Dynamic generation of S-boxes in block cipher systems. Proceedings of the IEEE Eighteenth National Radio Science Conference, March 27-29, 2001, Mansoura, pp: 389-397.

Hamid, N., A. Yahya, R.B. Ahmad and O.M. Al-Qershi, 2012. Image steganography techniques: An overview. Intl. J. Comput. Sci. Secur., 6: 168-187.

Huang, X., S. Wijesekera and D. Sharma, 2009. Quantum cryptography for wireless network communications. Proceedings of the 4th International Symposium on Wireless Pervasive Computing (ISWPC 2009), February 11-13, 2009, IEEE, Melbourne, Victoria, Australia, ISBN:978-1-4244-2965-3, pp: 1-5.

Karygiannis, T. and L. Owens, 2002. Wireless Network Security, 802.11, Bluetooth and Handheld Devices. NIST Special Publication, Hamburg, Germany, Pages: 119.

Khan, I., S. Gupta and S. Singh, 2016. A new data hiding approach in images for secret data communication with Steganography. Intl. J. Comput. Appl., 135: 9-14.

Masuda, N., G. Jakimoski, K. Aihara and L. Kocarev, 2006. Chaotic block ciphers: From theory to practical algorithms. IEEE. Trans. Circuits Syst., 53: 1341-1352.

Milenkovic, O., 2001. On the generalized hamming weight enumerators and coset weight distributions of even isodual codes. Proceedings of the IEEE International Symposium on Information Theory, June 29, 2001, IEEE, Washington, DC., USA., ISBN:0-7803-7123-2, pp: 62-62.

Neelima, S. and S. Mandal, 2015. Review paper on cryptography. Intl. J. Res., 2: 45-49.

Neuman, C. and T. Ts'o, 1994. Kerberos: An authentication service for computer networks. IEEE Commun. Maga., 32: 33-38.

Oppliger, R., 2007. Providing certified mail services on the internet. IEEE. Secur. Privacy, 5: 16-22.

Petrank, E. and C. Rackoff, 2000. CBC MAC for real-time data sources. J. Cryptology, 13: 315-338.

Pradesh, U., 2015. The RC7 encryption algorithm. Intl. J. Secur. Appl., 9: 55-59.

Schneier, B. and J. Kelsey, 1996. Unbalanced feistel networks and block cipher design. Proceedings of the 3rd International Workshop on Fast Software Encryption, February 21-23, 1996, Springer, Cambridge, UK, pp: 121-144.

Schneier, B., 1996. Applied Cryptography: Protocols, Algorithms and Source Code in C. 2nd Edn., John Wiley and Sons, New York, USA., ISBN-13: 978-0471117094, pp: 758.

Sharma, R., 2012. A novel approach to combine public-key encryption with symmetric-key encryption. Intl. J. Comput. Sci. Appl., 1: 8-15.

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

Stinson, R., 2006. Cryptography: Theory and Practice. 3rd Edn., CRC Press, London, ISBN: 1584885084, pp: 593.

Tayel, M. and H. Shawky, 2014. A proposed assessment metrics for image steganography. Intl. J. Cryptography Inf. Secur., 4: 1-11.

Wang, Y., G. Attebury and B. Ramamurthy, 2006. A survey of security issues in wireless sensor networks. IEEE Communi. Surveys Tutorials, 8: 2-23.

Zhou, J. and D. Gollmann, 1997. An efficient non-repudiation protocol. Proceedings of the 10th Workshop on Computer Security Foundations, June 10-12, 1997, IEEE, Rockport, Massachusetts, USA., ISBN:0-8186-7990-5, pp: 126-132.