The Dynamic Dual Key Encryption Algorithm Based on joint Galois Fields

Abdul Monem S. Rahma1† and Basima Z.Yacob ††

†Computer Science Department, University of Technology, Baghdad, Iraq,
††Computer Science Department, University of Duhok, Duhok, Kurdistan Iraq.

Summary
This paper presents a novel symmetric approach (algorithm) called Dynamic Dual Key Encryption Algorithm which uses dual key for encryption with variable (dynamic) block bits size, where each block of size 3, 4, 5 and 6 bits are interpreted as an element of a finite field. The first key is called control key determines the length of bits block (3, 4, 5, or 6bits block) size to encrypt, and the second key is used for encryption by using an equation: $Y = X * A + B$ Where X is bits block, A and B are the encryption keys. The mathematical operations addition and multiplication in this equation are based on mathematical theory of Galois field GF(2^n). The proposed algorithm achieves best results, where it provides high level of security by using dual key and four dynamic tiny block cipher; to decrypt the ciphertext with 128 bits, the attacker needs $1.86285884e+204$ of possibilities of keys as minimum and $1.80032832e+399$ as maximum, In addition, it is approximately 43 times faster than AES encryption and 64 times faster than AES decryption.

Key words:
Dynamic block, Dual keys, Encryption algorithm

1. Introduction

Encryption algorithms play a main role in information security systems, various algorithms [3, 4] have been proposed till now and each has their merits and demerits. As a result researchers are working in the field of cryptography to enhance the security further. The strength of encryption depends on two elements, key and algorithm, in this paper we will focus on the algorithm.
The Dynamic Dual Key Encryption approach is considered as a stream of bits and the technique uses dual key, first key (control key) to determine the length of bits block and the second one is used for encryption a according to the equation that used addition and multiplication based on mathematical theory of Galois field GF(2^n).
Each block (3, 4, 5, or 6) bits size in this algorithm are interpreted as finite field elements using a representation
In which a 3,4,5 or 6 bits with bits $b_0\, b_1\, b_2$, $b_0\, b_1\, b_2\, b_3$, $b_0\, b_1\, b_2\, b_3\, b_4$ or $b_0\, b_1\, b_2\, b_3\, b_4\, b_5$ represents the polynomial consecutively.
The proposed algorithm has been designed with two major factors in mind; first, decreasing the time needed for encryption/decryption, second, the level of security should be high enough so attackers cannot obtain the encryption/decryption key easily.
The efficiency of this method is that using a Dual key and it supports a variable length bits block.
The file (plaintext) can be recovered using the same algorithm but with reverse mathematical operations process that in equation.

2. Base Mathematical for 3, 4, 5, And 6 bits Block Size

A field is a commutative ring (with unity) in which all nonzero elements have a multiplicative inverse [5]. For a given prime, p, the finite field of order p, GF (p) is defined as the set Z_p of integers $\{0, 1, \ldots, p-1\}$, together with the arithmetic operations modulo p [7]. The finite field of order p^n is generally written GF (p^n).
Arithmetic in a finite field is different from standard integer arithmetic. There are a limited number of elements in the finite field; all operations performed in the finite field result in an element within that field.
A polynomial f(x) in GF (2^n) is represented as:

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0$$

Can be uniquely represented by its n binary coefficients $(a_{n-1}a_{n-2}\ldots a_0)$. Thus, every polynomial in
GF (2^n) can be represented by an n-bit number. In this paper we are concerned with the finite field
GF (2^3) ,GF (2^4), GF (2^5) , and GF (2^6)

The addition of two finite field elements is achieved by adding the coefficients for Corresponding powers in their polynomial representations, this addition being performed in GF(2), that is, modulo 2, so that $1 + 1 = 0$. Consequently, addition and subtraction are both equivalent to an exclusive-or operation on the n-bits that represent the field elements of GF (2^n) .

Tables 1, 2, 3, and 4 represent the addition in GF (2^3) , GF (2^4) , GF (2^5) , and GF (2^6) consecutively.

Manuscript received August 5, 2011
Manuscript revised August 20, 2011

Tables 5, 6, 7, and 8 represent the addition inverse in GF (2^3) , GF (2^4) , GF (2^5) , and GF (2^6) consecutively

The Finite field multiplication is achieved by multiplying the two polynomials for the two elements concerned and collecting like powers of x in the result, if multiplication result in a polynomial of degree greater than n-1, then the polynomial is reduced modulo some irreducible polynomial m(x) of degree n. That is, divided by m(x) and keep the remainder. The definition of irreducible polynomial is a polynomial f(x) over a field F is called irreducible if and only if f(x) cannot be expressed as a product of two polynomials, both over F, and both of degree lower than that of f(x) [7].

Since each polynomial for 3 bits block can have powers of x up to 3, the multiplication result can have powers of x up to 6 and will no longer fit within a 3bits form, for 4, 5, and 6 bits block will be the same and powers of x will be 8, 10, 12, respectively.

This situation is handled by replacing the result with the remainder polynomial after division by a special order irreducible polynomial; irreducible polynomial of degree 3 there are only two, of degree 4, there are three, of degree 5 there are six, and of degree 6, there are nine such polynomial.

To construct the multiplication finite filed GF(2^n) requires choosing an irreducible polynomial of degree n.

The following tables 9, 10, 11, and 12 represent multiplication in finite field GF (2^3) , GF (2^4) , GF (2^5) and GF (2^6) where the chosen irreducible polynomials are m(x) = $x^3 + x + 1$, m(x) = $x^4 + x + 1$, m(x) = $x^5 + x^2 + 1$ and m(x) = $x^6 + x + 1$ consecutively.

Each element of the finite field set other than 0 has a multiplicative inverse [7]. Just as the Euclidean algorithm can be adapted to find the greatest common divisor (gcd) of two polynomials, the extended Euclidean algorithm can be adapted to find the multiplicative inverse of a polynomial. Specifically, the algorithm will find the multiplicative inverse of b(x) modulo m(x) if the degree of b(x) is less than the degree of m(x) and gcd[m(x), b(x)] = 1. If m(x) is an irreducible polynomial, then it has no factor other than itself or 1, so that gcd [m(x), b(x)] = 1. Suppose two polynomials a(x), b(x), and m(x) is an irreducible polynomial, a(x) and b(x) are mutual inverses If $a(x) * b(x)\ mod\ m(x) = 1$.

The tables 13, 14, 15, and 16 represent multiplication inverse in finite field GF (2^3) , GF (2^4) GF (2^5) and GF (2^6) with irreducible polynomials m(x) = $x^3 + x + 1$, m(x) = $x^4 + x + 1$, m(x) = $x^5 + x^2 + 1$ and m(x) = $x^6 + x + 1$ consecutively.

Table 1: Addition in GF(2^3). [7]

+		000	001	010	011	100	101	110	111
		0	1	2	3	4	5	6	7
000	0	0	1	2	3	4	5	6	7
001	1	1	0	3	2	5	4	7	6
010	2	2	3	0	1	6	7	4	5
011	3	3	2	1	0	7	6	5	4
100	4	4	5	6	7	0	1	2	3
101	5	5	4	7	6	1	0	3	2
110	6	6	7	4	5	2	3	0	1
111	7	7	6	5	4	3	2	1	0

Table 2: Addition operation in GF(2^4).

+		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0000	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0001	1	1	0	3	2	5	4	7	6	9	8	11	10	13	12	15	14
0010	2	2	3	0	1	6	7	4	5	10	11	8	9	14	15	12	13
0011	3	3	2	1	0	7	6	5	4	11	10	9	8	15	14	13	12
0100	4	4	5	6	7	0	1	2	3	12	13	14	15	8	9	10	11
0101	5	5	4	7	6	1	0	3	2	13	12	15	14	9	8	11	10
0110	6	6	7	4	5	2	3	0	1	14	15	12	13	10	11	8	9
0111	7	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
1000	8	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7
1001	9	9	8	11	10	13	12	15	14	1	0	3	2	5	4	7	6
1010	10	10	11	8	9	14	15	12	13	2	3	0	1	6	7	4	5
1011	11	11	10	9	8	15	14	13	12	3	2	1	0	7	6	5	4
1100	12	12	13	14	15	8	9	10	11	4	5	6	7	0	1	2	3
1101	13	13	12	15	14	9	8	11	10	5	4	7	6	1	0	3	2
1110	14	14	15	12	13	10	11	8	9	6	7	4	5	2	3	0	1
1111	15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 3: Addition in $\mathbf{GF}(\mathbf{2^5})$.

+		00000	00001	00010	00011	00100	00101	00110	---	---	---	11001	11010	11011	11100	11101	11110	11111
		0	1	2	3	4	5	6	---	---	----	25	26	27	28	29	30	31
00000	0	0	1	2	3	4	5	6	---	---	----	25	26	27	28	29	30	31
00001	1	1	0	3	2	5	4	7	---	---	----	24	27	26	29	28	31	30
00010	2	2	3	0	1	6	7	4	---	---	----	27	24	25	30	31	28	29
00011	3	3	2	1	0	7	6	5	---	---	----	26	25	24	31	30	29	28
00100	4	4	5	6	7	0	1	2	---	---	----	29	30	31	24	25	26	27
00101	5	5	4	7	6	1	0	3	---	---	----	28	31	30	25	24	27	26
00110	6	6	7	4	5	2	3	0	---	---	----	31	28	29	26	27	24	25
---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----
---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----
---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----
---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----
11001	25	25	24	27	26	29	28	31	---	---	----	0	3	2	5	4	7	6
11010	26	26	27	24	25	30	31	28	---	---	----	3	0	1	6	7	4	5
11011	27	27	26	25	24	31	30	29	---	---	----	2	1	0	7	6	5	4
11100	28	28	29	30	31	24	25	26	---	---	----	5	6	7	0	1	2	3
11101	29	29	28	31	30	25	24	27	---	---	----	4	7	6	1	0	3	2
11110	30	30	31	28	29	26	27	24	---	---	----	7	4	5	2	3	0	1
11111	31	31	30	29	28	27	26	25	---	---	----	6	5	4	3	2	1	0

Table 4: Addition operation in GF(2^6).

+		000000	000001	000010	000011	000100	000101	000110	---	---	---	111001	111010	111011	111100	111101	111110	111111
		0	1	2	3	4	5	6	---	---	---	57	58	59	60	61	62	63
000000	0	0	1	2	3	4	5	6	---	---	---	57	58	59	60	61	62	63
000001	1	1	0	3	2	5	4	7	---	---	---	56	59	58	61	60	63	62
000010	2	2	3	0	1	6	7	4	---	---	---	59	56	57	62	63	60	61
000011	3	3	2	1	0	7	6	5	---	---	---	58	57	56	63	62	61	60
000100	4	4	5	6	7	0	1	2	---	---	---	61	62	63	56	57	58	59
000101	5	5	4	7	6	1	0	3	---	---	---	60	63	62	57	56	59	58
000110	6	6	7	4	5	2	3	0	---	---	---	63	60	61	58	59	56	57
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
111001	57	57	56	59	58	61	60	63	---	---	---	0	3	2	5	4	7	6
111010	58	58	59	56	57	62	63	60	---	---	---	3	0	1	6	7	4	5
111011	59	59	58	57	56	63	62	61	---	---	---	2	1	0	7	6	5	4
111100	60	60	61	62	63	56	57	58	---	---	---	5	6	7	0	1	2	3
111101	61	61	60	63	62	57	56	59	---	---	---	4	7	6	1	0	3	2
111110	62	62	63	60	61	58	59	56	---	---	---	7	4	5	2	3	0	1
111111	63	63	62	61	60	59	58	57	---	---	---	6	5	4	3	2	1	0

Table 5: Addition inverse in $\mathbf{GF(2^3)}$. [7]

	W	-W
000	0	0
001	1	1
010	2	2
011	3	3
100	4	4
101	5	5
110	6	6
111	7	7

Table 6: Addition inverse in $\mathbf{GF(2^4)}$.

	W	-W
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	10
1011	11	11
1100	12	12
1101	13	13
1110	14	14
1111	15	15

Table 7: Addition inverse in $GF(2^5)$.

	W	-W
00000	0	0
00001	1	1
00010	2	2
00011	3	3
00100	4	4
00101	5	5
00110	6	6
00111	---	---
---	---	---
---	---	---
---	---	---
---	---	---
---	---	---
11001	---	---
11010	26	26
11011	27	27
11100	28	28
11101	29	29
11110	30	20
11111	31	31

Table 8: Addition inverse in $GF(2^6)$.

	W	-W
000000	0	0
000001	1	1
000010	2	2
000011	3	3
000100	4	4
000101	5	5
000110	6	6
---	---	---
---	---	---
---	---	---
---	---	---
---	---	---
111001	57	57
111010	58	58
111011	59	59
111100	60	60
111101	61	61
111110	62	62
111111	63	63

Table 9: Multiplication in $GF(2^3)$ with the irreducible polynomial m(x) $= (x^3 + x + 1)$. [7]

*		000	001	010	011	100	101	110	111
		0	1	2	3	4	5	6	7
000	0	0	0	0	0	0	0	0	7
001	1	0	1	2	3	4	5	6	7
010	2	0	2	4	6	3	1	7	5
011	3	0	3	6	5	7	4	1	2
100	4	0	4	3	7	6	2	5	1
101	5	0	5	1	4	2	7	3	2
110	6	0	6	7	1	5	3	2	4
111	7	0	7	5	2	1	6	4	3

Table 10: Multiplication in $GF(2^4)$with the irreducible polynomial m(x) $= x^4 + x + 1$

*		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0001	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0010	2	0	2	4	6	8	10	12	14	3	1	7	5	11	9	15	13
0011	3	0	3	6	5	12	15	10	9	11	8	13	14	7	4	1	2
0100	4	0	4	8	12	3	7	11	15	6	2	14	10	5	1	13	9
0101	5	0	5	10	15	7	2	13	8	14	11	4	1	9	12	3	6
0110	6	0	6	12	10	11	13	7	1	5	3	9	15	14	8	2	4
0111	7	0	7	14	9	15	8	1	6	13	10	3	4	2	5	12	11
1000	8	0	8	3	11	6	14	5	13	12	4	15	7	10	2	9	1
1001	9	0	9	1	8	2	11	3	10	4	13	5	12	6	15	7	14
1010	10	0	10	7	13	14	4	9	3	15	5	8	2	1	11	6	12
1011	11	0	11	5	14	10	1	15	4	7	12	2	9	13	6	8	3
1100	12	0	12	11	7	5	9	14	2	10	6	1	13	15	3	4	8
1101	13	0	13	9	4	1	12	8	5	2	15	11	6	3	14	10	7
1110	14	0	14	15	1	13	3	2	12	9	7	6	8	4	10	11	5
1111	15	0	15	13	2	9	6	4	11	1	14	12	3	8	7	5	10

Table 11: Multiplication in GF(2^5) with the irreducible polynomial m(x) = $x^5 + x^2 + 1$

*		00000	00001	00010	00011	00100	00101	00110	---	---	---	11001	11010	11011	11100	11101	11110	11111
		0	1	2	3	4	5	6	---	---	----	25	26	27	28	29	30	31
00000	0	0	0	0	0	0	0	0	---	---	----	0	0	0	0	0	0	0
00001	1	0	1	2	3	4	5	6	---	---	----	25	26	27	28	29	30	31
00010	2	0	2	4	6	8	10	12	---	---	----	23	17	19	29	31	25	27
00011	3	0	3	6	5	12	15	10	---	---	----	14	11	8	1	2	7	4
00100	4	0	4	8	12	16	20	24	---	---	----	11	7	3	31	27	23	19
00101	5	0	5	10	15	20	17	30	---	---	----	18	29	24	3	6	9	12
00110	6	0	6	12	10	24	30	20	---	---	----	28	22	16	2	4	14	8
---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----
---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----
---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----
---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----
11001	25	0	25	23	14	11	18	28	---	---	----	6	8	17	20	13	3	26
11010	26	0	26	17	11	7	29	22	---	---	----	8	3	25	21	15	4	30
11011	27	0	27	19	8	3	24	16	---	---	----	17	25	2	9	18	26	1
11100	28	0	28	29	1	31	3	2	---	---	----	20	21	9	23	11	10	22
11101	29	0	29	31	2	27	6	4	---	---	----	13	15	18	11	22	20	9
11110	30	0	30	25	7	23	9	14	---	---	----	3	4	26	10	20	19	13
11111	31	0	31	27	4	19	12	8	---	---	----	26	30	1	22	9	13	18

Table 12: Multiplication in GF(2^6) with the irreducible polynomial m(x)= $x^6 + x + 1$

*		000000	000001	000010	000011	000100	000101	000110	---	---	---	111001	111010	111011	111100	111101	111110	111111
		0	1	2	3	4	5	6	---	---	---	57	58	59	60	61	62	63
000000	0	0	0	0	0	0	0	0	---	---	---	0	0	0	0	0	0	0
000001	1	32	33	34	35	36	37	38	---	---	---	57	58	59	60	61	62	63
000010	2	3	1	7	5	11	9	15	---	---	---	49	55	53	59	57	63	61
000011	3	35	32	37	38	47	44	41	---	---	---	8	13	14	7	4	1	2
000100	4	6	2	14	10	22	18	30	---	---	---	33	45	41	53	49	61	57
000101	5	38	35	44	41	50	55	56	---	---	---	24	23	18	9	12	3	6
000110	6	5	3	9	15	29	27	17	---	---	---	16	26	28	14	8	2	4
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
111001	57	4	61	53	12	37	28	20	---	---	---	62	54	15	38	31	23	46
111010	58	39	29	16	42	10	48	61	---	---	---	54	59	1	33	27	22	44
111011	59	7	60	50	9	46	21	27	---	---	---	15	1	58	29	38	40	19
111100	60	34	30	25	37	23	43	44	---	---	---	38	33	29	47	19	20	40
111101	61	2	63	59	6	51	14	10	---	---	---	31	27	38	19	46	42	23
111110	62	33	31	30	32	28	34	35	---	---	---	23	22	40	20	42	43	21
111111	63	1	62	60	3	56	7	5	---	---	---	46	44	19	40	23	21	42

Table 13: Multiplication Inverse in GF(2^3) with the Irreducible polynomial m(x) = $x^3 + x + 1$. [7]

	W	W^{-1}
000	0	--
001	1	1
010	2	5
011	3	6
100	4	7
101	5	2
110	6	3
111	7	4

Table 14: Multiplication inverse in GF(2^4) with the irreducible polynomial m(x) = $x^4 + x + 1$

	W	W^{-1}
0000	0	--
0001	1	1
0010	2	9
0011	3	14
0100	4	13
0101	5	11
0110	6	7
0111	7	6
1000	8	15
1001	9	2
1010	10	12
1011	11	5
1100	12	10
1101	13	4
1110	14	3
1111	15	8

Table 15: Multiplication Inverse in GF(2^5) with the Irreducible polynomial m(x) = $x^5 + x^2 + 1$.

	W	W^{-1}
00000	0	--
00001	1	1
00010	2	18
00011	3	28
00100	4	9
00101	5	23
00110	6	14
00111	---	---
---	---	---
---	---	---
---	---	---
---	---	---
---	---	---
11001	---	---
11010	26	21
11011	27	31
11100	28	3
11101	29	19
11110	30	20
11111	31	27

Table 16: Multiplication inverse in GF(2^6)with the irreducible polynomial m(x) = $x^6 + x + 1$

	W	W^{-1}
000000	0	--
000001	1	1
000010	2	33
000011	3	62
000100	4	49
000101	5	43
000110	6	31
---	---	---
---	---	---
---	---	---
---	---	---
---	---	---
111001	57	8
111010	58	59
111011	59	58
111100	60	55
111101	61	16
111110	62	3
111111	63	32

3. The Encryption Algorithm (Methodology)

The encryption algorithm can be described by the following steps:

Input :
Plaintext , KeyOne , KeyTwo
Output :
Ciphertext
No_K //number of bits from keyOne that are used in first round
No_K2 //number of bits from keyTwo that are used in first round

Step 0:
- Round= 0
- While Round < 2 do :

Step 1: Read a portion of KeyOne (Control key)
Step 2: depending on the value of KeyOne's portion do the following:
Select the block size (3, 4, 5 or 6 bits) from plaintext.
Read from KeyTwo A and B Keys.
Perform the following Encryption Equation :
$Y = X * A + B$
Step3 : Compute the number of bits for KeyOne and KeyTwo that are used in first round
Check If Round =0 then
No_K1=No_K1 + 2
No_k2=No_K2 + block size * 2
End if
Step 4: Repeat steps 1, 2 and 3 until plaintext is finished.
Round=Round+1
End while.

The Dual key technique uses two keys, the first is called control key (keyOne) which is used to determine the size of bit block and the second one(KeyTwo) is used in the encryption equation .
The size of bit block is 3 or 4 or 5 or 6 bits. The first step of the technique is reading the two bits from the control key(KeyOne) , If the value is 0 or 1 or 2 or 3 the block size will be 3 or 4 or 5 or 6 consecutively.
Suppose the value of these two bits is 0, then 3bits from plaintext is to be read and stored in X varible. 3bits from KeyTwo is stored in A variable and the next three bits is also stored in B variable . Second step, the encryption equation is performed on the 3bits block:

$$Y = X * A + B .$$

If two bits' value of KeyOne is 1 or 2 or 3 the same steps are applied but with four or five or six bits consecutively instead of three bits. These steps are applied to the rest plaintext message.
Addition and multiplication in the encryption equation are based on a Galois field $GF(2^3)$, $GF(2^4)$, $GF(2^5)$, and $GF(2^6)$.
The following example to illustrate our technique in encryption part let us suppose the following:
Plaintext= 11101101.
keyOne =10011000.
KeyTwo=10001001011011101111011111001101110

With Irreducible polynomial chosen for each bits.
To encrypt the plaintext do the following:
Step1:
Read a portion (the first two bits) from KeyOne, the two bits are 00, the bit block size will be read from plaintext to encrypt is 3bits, it is 101.
Read two 3bits block from KeyTwo and store them into variables A, and B, A =110, B =101.
Step 2:
Perform the following equation:

$$Y = X * A + B$$

The result of multiplication according to Table 9 is 101* 110 =011
The result of addition according to Table 1 is 011 + 101=110, the ciphertext for 3bits block=110.
Step3 is executed then:

No_K1= 2
No_k2= 6

To encrypt next block, the same steps are applied, The next two bits from KeyOne is 10 , next 5bits from plaintext is 11101, The two 5bits blocks from keyTwo are A =11001, and B =11011.
Perform the equation:
$11101 * 11001 + 11011 = 10110$
Ciphertext for 5bits block=10110.
Step3 is executed then:

No_K1=4
No_k2= 16

The numbers of bits of KeyOne and KeyTwo that are used in first round are 4 and 16 consecutively.
The ciphertext 10110110 is the result of the first round of encryption plaintext 11101101.
The same steps are applied in the second round:
The next two bits from KeyOne is 01, next 4bits from ciphertext is 0110, the two 4bits blocks from keyTwo are A =1011, and B =1011.
Perform the equation:
$0110 * 1011 + 1011 = 0100$
The next two bits from KeyOne is 10, next 5bits from is 01011 (adding 0 pad at most left), the two 5bits blocks from keyTwo are A =00101, and B =10001
$01011 * 00101 + 10001 = 10011$
The ciphertext 100110100 is the result of encryption 11101101.

4. The Decryption Algorithm

The decryption algorithm can be described by the following steps:

```
Input :
        Ciphertext , KeyOne , KeyTwo
Output :
        Plaintext
Step 0 :
   -apply a circular shift of (No_K1) bits and (No_K2)
      bits for KeyOne and KeyTwo consecutively
    -Round=0
    -While Round < 2 do
Step 1 : Read a portion of KeyOne(Control key)
Step 2 : depending on the value of KeyOne's portion do
          the following
       Select the block size ( 3, 4 , 5 or 6 bits) from plaintext
       Read from KeyTwo A and B Keys
       Perform the following Decryption Equation :
     X = (Y + addtion inverse(B)) * multiplicative inverse(A)
Step 3: Repeat steps 1 and 2 until Ciphertext is finished
              Round=Round+1
               End while
```

For decryption the same steps of encryption are applied but with reverse equation's operations are performed.
The following example to illustrate our technique in decryption part. Suppose the following:
Ciphertext= 100110100
KeyOne= 10011000.
KeyTwo=10001001011011101111011111001101110.
With Irreducible polynomial chosen for each bits block.
To decrypt the ciphertext the steps are:
Step0:
Rotate left 4 bits of KeyOne and 16 bits of KeyTwo:
KeyOne =10001001
KeyTwo=1101111001101110 1000100101101110 11

Step 2:
The first two bits of KeyOne is 01 ,
The first block size is read from ciphertext is 4bits, it is 0100,
Read two 4bits blocks from KeyTwo, store them into A=1011, and B =1011.
Step 3:
Perform the decryption equation:
(0100 + addtion inverse (1011)) * multiplicative Inverse (1011) =0110
To decrypt second block, the same steps are applied,
The next two bits from KeyOne is 10, the next 5bits from ciphertext is 10011, the next 5bits from KeyTwo are A =00101, and B =10001.
Perform decryption equation: (10011 +addtion inverse (10001)) * multiplicative Inverse (00101) = 01011
The result from the first decrypt round is 010110110, (last 0 is padding zero)
The same steps are applied in the second round:
Step 1:
The next two bits from KeyOne is 00, next 3bits is 110, the two 3bits blocks from keyTwo are A =110, and B =101.
Perform decryption equation:
(110 +addtion inverse (101)) * multiplicative Inverse (110) =101
The next two bits from KeyOne is 10, next 5bits is 10110, the two 5bits blocks from keyTwo are A =11001, and B =11011.
(101+addtion inverse (11011)) * multiplicative Inverse (11001) =11101
The ciphertext 11101101 (after remove padding zero) is the result of the second round.

5. Security of Dual key Technique

Cryptography may be described as the science of secure communication over a public channel. An important area of cryptography is symmetric key cryptography. In symmetric key cryptography, the two parties share a secret piece of information, the key, and a public encryption algorithm.
The message (M) is encrypted by using the encryption half of algorithm $C=E_k(M)$. The message is decrypted by computing $D_k(C) = D_k(E_k(M)) = M$. M, K, and C may are taken to be finite sequences of bits.
Without knowing K, an attacker cannot compute M from C. A more common assumption in modern cryptography is to assume that an attacker may have several pairs M_i, C_i from which to try to recover the key (a known plaintext attack), or may even be able to pick M's or C's to have encrypted or decrypted. (a chosen plaintext or chosen ciphertext attack, or, if both are allowed, a chosen text attack). In these cases, simply by trying all possible keys an attacker will eventually recover the key K. This is known as a brute force attack, or exhaustive key search.

The proposed algorithm presents a high resistant against brute force attacks, because it employs dual key and dynamic block cipher, hence it will be very difficult to guess the keys.

The following example illustrates number of possibilities of keys that the attacker needs to decrypt the
ciphertext with 128 bits by using Dual key technique:
The control key (KeyOne) determines the block size either 3 or 4 or 5 or 6 bits block size.
To construct $GF(2^3)$, there are two an irreducible polynomials degree 3, for construct $GF(2^4)$, $GF(2^5)$ and $GF(2^6)$ there are three with degree 4, six with degree 5, nine with degree 6 consecutively.
The equation is:
$Y = X * A + B$, A and B are keys, the size of both is 3or 4 or 5 or 6 bits.
If the block size is 3 bits, the number of possible keys for each key is 2^3.
The number of possibility of keys to decrypt one 3bits block size is:
$2 * 2^3 * 2^3 = 2^7$.
The number of possibility of keys to decrypt one 4bits block size is:

$$3 * 2^4 * 2^4 = 3 * 2^8.$$

The number of possibility of keys to decrypt one 5bits block size is:

$$6 * 2^5 * 2^5 = 6 * 2^{10} = 6144.$$

The number of possibility of keys to decrypt one 6 bits block size is:

$$9 * 2^6 * 2^6 = 9 * 2^{12} = 36864.$$

To decrypt only one block, the number of possibility of keys for each round is:

$$(2^7 + 3 * 2^8 + 6 * 2^{10} + 9 * 2^{12}) = 43904.$$

It is possible to compute the minimum and maximum number of possibility of keys to decrypt 128 bits of ciphertext, first the maximum number of blocks is computed when all blocks are of 3bits size,
128 ÷3=42.6, the result with reminder will be increased by one, thus the maximum number of blocks is 43, and the minimum number of blocks is computed when all blocks are of 6bits size ,128 ÷6=21.3,the result will increased by one, therefore the minimum number of blocks is 22.
It can be computed the minimum and maximum number of possibility of keys to decrypt 128 ciphertext bits.
The maximum number of possibility of keys for each round is:

$$(2^7 + 3 * 2^8 + 6 * 2^{10} + 9 * 2^{12})^{43} = 4.243027599e + 199$$

The minimum number of possibility of keys for each round is:

$$(2^7 + 3 * 2^8 + 6 * 2^{10} + 9 * 2^{12})^{22} = 1.36486587e + 102$$

The algorithm has two rounds hence the maximum number of possibility of keys is:

$$(2^7 + 3 * 2^8 + 6 * 2^{10} + 9 * 2^{12})^{43} * (2^7 + 3 * 2^8 + 6 * 2^{10} + 9 * 2^{12})^{43} = 1.80032832e + 399$$

The minimum number of possibility of keys is:

$$(2^7 + 3 * 2^8 + 6 * 2^{10} + 9 * 2^{12})^{22} * (2^7 + 3 * 2^8 + 6 * 2^{10} + 9 * 2^{12})^{22} = 1.86285884e + 204$$

6. Experimental Results

The following tables represent the experimental results for the speed of the Dynamic Dual Key algorithm in first and second round, and the speed of AES algorithm.

Table 17: The encryption and decryption times for Dynamic Dual Key algorithm in the first round

Plaintext Size (byte)	*Encryption time(ms)*	*Decryption Time(ms)*
19000	69.86	105
20000	71	73
40000	144	178
50000	178	214

Table 18: The encryption and decryption times for Dynamic Dual Key algorithm in the second round

Plaintext Size (byte)	*Encryption time(ms)*	*Decryption Time(ms)*
19000	157.72	165
20000	158	160
40000	303	372
50000	372	428

Table 19: The encryption and decryption times for AES algorithm using key size 128bit

Plaintext Size (byte)	*Encryption time(ms)*	*Decryption Time(ms)*
19000	6500	10641
20000	6813	11188
40000	13812	22141
50000	17156	27516

According to the output results of our algorithm as shown in Tables 17, and 18 the proposed encryption algorithm is much better than all other known techniques like 3-DES, AES-Rijndael, and RSA [6].

Advanced Encryption Standard (AES) is an algorithm of the first category which provides much higher security level than DES and perform it in 3 to 10 less computational power than 3-DES [8], it has better performance than DES, 3DES, and RC2 [9],based on these facts ,AES is to be compared with proposed algorithm.

The comparison between AES and our proposed algorithm is illustrated in Tables 17, 18 and 19.

From Tables 17 and 19, we can note that our new algorithm in the first round achieves best results, where it is approximately 93 times faster than AES encryption and 100 times faster than AES decryption.

From Tables 18 and 19, can be observed that the Dynamic Dual Key algorithm is approximately 43 times faster than AES encryption and 64 times faster than AES decryption.

The designed technique and AES algorithm both has been implemented successfully using visual basic 6 Programming language and also implemented with processor of Pentium IIII (3.40 GHZ) and 3GB of RAM on windows XP.

7. Discussion

Dual key and dynamic block cipher prevent exhaustive key search and differential attacks.

Non fixed (dynamic) size block cipher avoid replaying in authentication and attacks that can happen on the fixed sized block cipher algorithms, dynamic block length in proposed algorithm lead to maximum cryptographic confusion and consequently makes it difficult for cryptanalysis.

We have also come to conclusion that the more the rounds of the proposed algorithm are increased the higher security is achieved.

Most of the available encryption/decryption techniques are not perfect for Real-Time Applications [6] since they were originally built for text data, and due to their extensive computations which result in an unacceptable processing time, from the results that we obtained in section 6 can be seen in a new encryption/decryption approach minimums the encryption and decryption time that makes it appropriate for Real-Time Applications RTA. In addition, it provides high level of security by using dual key and dynamic tiny block cipher.

References

[1] I. Tanenbaum, “Computer Networks” 2nd edition, Prentice Hall, London, 1989.

[2] D.R .Stinson, “Cryptography Theory and Practice”, CRC Press, London, 1995.

[3] W. Stallings, "Network security Essentials Applications and Standards", 4th education, Prentice Hall, 2011.
[4] NIST,"Security Requirements for Cryptographic Modules", FIPS PUB 140-1 Federal Information Processing Standard Publication 140-1, U.S. Department of Commerce, 1994 January 11.
[5] J. Nakahara Jr and E. Abrah˜ao, "A New Involutory MDS Matrix for the AES", International Journal of Network Security, Vol.9, No.2, PP.109–116, Sept. 2009.
[6] A. H. Omari, B. M. Al-Kasasbeh, R. E. Al-Qutaish and Mohammad I. Muhairat, "DEA-RTA: A Dynamic Encryption Algorithm for the Real-Time Applications", International Journal of computers, Issue 1, Volume 3, 2009.
[7] W. Stallings, "Cryptography and Network Security", 4th edition, Prentice-Hall, 2005.
[8] J. Dray, Report on the NIST Java AES Candidate Algorithm Analysis, 1999.
[9] D. S. Abd Elminaam, H. M. Abdual Kader, and M. M. Hadhoud, "Evaluating the Performance of Symmetric Encryption Algorithms", International Journal of Network Security, Vol.10, No.3,PP.216–222, May 2010.

Prof. Dr. Abdul Monem Saleh Rahma awarded his MSc from Brunel University and his PhD from Loughborough University of technology United Kingdom in 1982, 1985 respectively. He taught at Baghdad university department of computer science and the Military Collage of Engineering, computer engineering department from 1986 till 2003.He fills the position of Dean Asst. of the scientific affairs and works as a professor at the University of Technology Computer Science Department .He published 82 Papers in the field of computer science and supervised 24 PhD and 57 MSc students. His research interests include Cryptography, Computer Security, Biometrics, image processing, and Computer graphics. And he Attended and Submitted in many Scientific Global Conferences in Iraq and Many other countries.

Basima Zrkqo Yacob received the B.Sc. degree in Computer Science from Mosul University, Iraq, in 1991, The MSc. Degree in computer science from University of Duhok ,Iraq in 2005. Currently she is a PhD student at faculty of computer science at Duhok University.