I J C T A, 9(17) 2016, pp. 8549-8558
© International Science Press

Secure Private Key exchange in Symmetric Cryptography using Finite Field approach GF(2^n)

Rama Devi K.* **and Prabakaran S.****

ABSTRACT

Information security is one of the crucial issues in data transmission through networks. Networks are extremely exposed to security attacks and consign a great challenge today. The development of cryptography and cryptanalysis are considered as the areas of on-going research. Some researchers use biological techniques to hide secret data. DNA Cryptography is a new and potentially excellent area that enhances data security. In this paper we host a twofold encryption method to share the private key among the sender and receiver which is exponent values of an polynomial equation using mathematical concept Galois field GF(2^n). The first key determines the length of block and the second key is used for encryption purpose. The proposed algorithm provides high level security by using twofold keys that uses mathematical operations addition and multiplication of polynomial equation based on GF.

Keywords: DNA Cryptography, polynomial, calculus, Galois Field.

I. INTRODUCTION

DNA cryptography is a novel cryptographic domain transpired with the research of DNA computing. In this area DNA strands are used as information carrier. Recent biological technologies are used as implementation tool.DNA cryptography differs from conventional cryptography in which the keys and cipher text are biological molecules. In our method the cipher text will be a DNA strands that includes four nucleotides adenine (A), guanine (G), thymine (T) and cytosine (C).

Cryptographic algorithms play a vital role in information security, various algorithms have been proposed till now and each has their own pros and cons. The security intensity of encryption depends on two factors Key and algorithm. In our previous work [2] we have proposed symmetric cryptographic algorithm using differential and integral calculus. The algorithm uses DNA code as a private key which is shared by sender and receiver. We generate a polynomial equation with random exponent values, for instance T-4,G-3,C-2,A-1 and this exponent values are the private keys used for encryption and decryption algorithm.

In this paper this private key will be considered as a plaintext which is a binary value of the DNA sequence. Now we propose a novel twofold encryption algorithm that uses polynomial addition and multiplication based on Galois field GF(p^n). The Cipher text generated by previous algorithm[2] is used to generate two keys that are used in this paper to encrypt the private key which is a DNA strands. Key-I will be a eight bit value which is length of cipher text and Key-II will be the binary value of the cipher text generated in previous algorithm.

Key-I is used to determine the length of the bits in a block and Key-II is used for encryption based on the equation B=A*X+Y where A is plaintext block, X and Y are encryption keys. The block size can be

* Research Scholar, Department of Computer Science & Engineering, SRM University, Chennai, Tamil Nadu, India, *E-mail: ramadevi.sarav@gmail.com*

** Department of Computer Science & Engineering, SRM University, Chennai, Tamil Nadu, India, *E-mail: prabakaran.mani@gmail.com*

3,4,5 or 6 that are interpreted based on the value of bits in Key-I. If two bits in Key-I is 00 then the block is 3, if 01 then the size is 4, if 10 the size is 5 and if 11 then the size is 6. The proposed algorithm is designed with two major concerns in mind; firstly, decreasing time required for encryption and decryption and secondly, level of security is high as the attackers cannot trap the key easily.

II. FINITE FIELD ARITHMETIC

Arithmetic operations like addition (and subtraction) and multiplication in Galois Field requires additional steps.

2.1. Addition and Subtraction

An addition in Galois Field is pretty straightforward. Suppose f (p) and g(p) are polynomials in $GF(p^n)$. Let $A= a_{n-1},a_{n-2},\ldots..a_1,a_0$ be the coefficients of f(p) , $B=b_{n-1},b_{n-2},\ldots..b_1,b_0$ be the coefficients of g(p) and $C=c_{n-1},c_{n-2},\ldots.c_1,c_0$ be the coefficients of h(p) where

$$h(p)=f(p)+g(p)$$

If a_k, b_k and c_k are coefficients of p^k in f(p), g(p) and h(p) respectively then

$$c_k=a_k+b_k \pmod p$$

Similarly for subtraction

$$h(p)=f(p)-g(p)$$

If a_k, b_k and c_k are coefficients of p^k in f(p), g(p) and h(p) respectively then

$$c_k=a_k - b_k \pmod p$$

Since computer works in $GF(2^8)$, if a_k and b_k refer to the k^{th} bit in the bytes we wish to add then c_k, the k^{th} bit in the resulting 4 byte, is given by

$$c_k=a_k+b_k \pmod 2$$

Since $0 + 1 = 1 + 0 = 1 \pmod 2 = 1$ and $0 + 0 = 0 \pmod 2 = 1 + 1 = 2 \pmod 2 = 0$, we can think of addition as exclusive-or operation which is also known as XOR operation. That is, XOR operation returns 0 if both entries are equal and returns 1 otherwise which also means that subtraction and addition is the same in Galois Field whose characteristic is 2. Due to the nature of Galois Field, addition and subtraction of two bytes will not go any bigger than 11111111 = 255, the biggest value one byte can store, and is therefore a safe operation.

2.2. Multiplication and Multiplicative Inverse

Suppose f (p) and g(p) are polynomials in $GF(p^n)$. Let m(p) be irreducible polynomial(i.e polynomial that cannot be factored) of degree at least n, so that the product of two polynomials f(p) and g(p) does not exceed 255 (i.e 11111111)

$$h(p)=(f(p).g(p)) \pmod{m(p)}$$

The Multiplicative inverse of f(p) is a(p) then

$$f(p).a(p) \bmod m(p) =1$$

III. MATHEMATICAL BASICS FOR 2,3,4 AND 5 BLOCK SIZE

For a given prime p, finite field of order p, GF(p) is defined as set of integers. Any polynomial f(x) in $GF(2^n)$ is represented as

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

Can be uniquely represented by its n binary coefficients (a_{n-1},a_{n-2},….a_0). Thus every polynomial in GF(2^n). can be represented by a binary number. Here we consider GF(2^2), GF(2^3), GF(2^4) and GF(2^5).

Tables 1, 2, 3, and 4 represent the addition in GF(2^2), GF(2^3), GF(2^4) and GF(2^5) respectively. Tables 5, 6, 7, and 8 represent the addition inverse in GF(2^2), GF(2^3), GF(2^4) and GF(2^5).

Table 1
Addition in GF(2^2)

+		00	01	10	11
		0	1	2	3
00	0	0	1	2	3
01	1	1	2	3	0
10	2	2	3	0	1
11	3	3	0	1	2

Table 2
Addition in GF(2^3)

+		*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	4	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 3
Addition 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 4
Addition in GF(2^5)

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

Table 5
Addition inverse GF(2^2)

	W	*-W*
00	0	0
01	1	1
10	2	2
11	3	3

Table 6
Addition inverse GF(2^3)

	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 7
Addition inverse 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 8
Addition inverse 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	—	—
———	—	—
———	—	—
———	—	—
———	—	—
———	—	—
———	—	—
11010	26	26
11011	27	27
11100	28	28
11101	29	29
11110	30	30
11111	31	31

Tables 9,10, 11, and 12 represent the multiplication in GF(2^2), GF(2^3), GF(2^4) and GF(2^5) respectively. Tables 13, 14, 15, and 16 represent the multiplicative inverse in GF(2^2), GF(2^3), GF(2^4) and GF(2^5) .

Table 9
Multiplication in GF(2^2)with the irreducible polynomial m(x)=(x^2 +x+1)

*		*00*	*01*	*10*	*11*
		0	1	2	3
00	0	0	0	0	3
01	1	0	1	2	0
10	2	0	2		1
11	3	0	3	1	2

Table 10
Multiplication in GF(2^3)with the irreducible polynomial m(x)=(x^3+x+1)

*		*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 11
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 12
Multiplication in GF(2^5) with the irreducible polynomial m(x)=($x^5 + x^2 + 1$)

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

Table 13
Multiplication Inverse in GF(2^2) with the Irreducible polynomial m(x)=$x^2 + x+1$

	W	W^{-1}
	00	0	0
	01	1	1
	10	2	2
	11	3	3

Table 14
Multiplication Inverse in GF(2^3)with the Irreducible polynomial m(x) = $x^3 +x +1$.

	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 15: 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 16
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	—	—
———	—	—
———	—	—
———	—	—
———	—	—
———	—	—
———	—	—
11010	26	21
11011	27	31
11100	28	3
11101	29	19
11110	30	20
11111	31	27

IV. ENCRYPTION ALGORITHM (METHODOLOGY)

Input: Plaintext, Key-I, Key-II
Output: Cipher Text, k1, k2
//k1-Number of bits used from Key-I in round I
// k2-Number of bits used from Key-II in round I
Step 0: —Round =0
While round < 2
Step 1: Read two bits from Key-I
Step 2: Depending on value of Key-I the block size is selected
(i.e., if the two bits are 00, then the block size is 2, if the bits are 01, then the block size is 3, if the bits are 10, then the block size is 4 and if the two bits are 11 then the block size is 5)
Step 3: Read the block from Key-II as X and Y perform the encryption operation using the formula
B=A * X +Y
Step 4: If round=0 then
k1=k1*2
k2=k2+block size *2
endif
Step 5: Repeat steps 1 to 4 until plaintext is finished
Round=Round+1
End of While

V. DECRYPTION ALGORITHM (METHODOLOGY)

Input: Cipher text, Key-I, Key-II,k1,k2
Output: Plain Text
Step 0:
Apply circular shift of k1 bits and k2 bits for Key-I and Key-II respectively.
—Round =0
While round < 2
Step 1: Read two bits from Key-I
Step 2: Depending on two bits of Key-I, select the block size as 2, 3, 4 or 5 bits from cipher text as B.
Step 3: Read the block from Key-II as X and Y perform the decryption operation using the formula
A=(B + additive inverse(Y)) * Multiplicative inverse(X)
Step 4: Repeat steps 1 to 3 until cipher text is finished.
Round=Round+1
End of While

VI. EXPERIMENTAL RESULTS

The following tables represent the experimental results for the speed of the twofold key algorithm in two rounds.

Table 17
The encryption and decryption times 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 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

IV. CONCLUSION AND FUTURE WORK

In this paper, we designed an encryption and decryption algorithm with the use of Galois Field GF(2^n). We come to conclusion that the more the rounds of the proposed algorithm are increased the higher security is achieved. Twin key encryption and dynamic block size 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. In future the algorithm can be enhanced by increasing the key size and number of rounds to secure the information that resists all kinds of attcks.

REFERENCES

[1] M. Adleman “Molecular Computation of solution to combinatorial problems” Science, New Series, Vol. 2 Leonard 66, No. 5187. Pp.1021-1024 Nov. 11, 1994.

[2] K. Rama Devi, Dr. S.Prabakaran “Cryptographic Method For Hiding Data in Genomic DNA Using Calculus” Global Journal of Pure and Applied Mathematics. ISSN 0973-1768 Volume 11, Number 6 (2015), pp. 4541-4546.

[3] W. Stallings, “Network security Essentials Applications and Standards”, 4th education, Prentice Hall, 2011.

[4] Hayam Mousa, Kamel Moustafa, Waiel Abdel-Wahed, and Mohiy Hadhoud “Data Hiding Based on Contrast Mapping Using DNA Medium” The International Arab Journal of Information Technology, Vol. 8, No. 2, April 2011.

[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] LAI XueJia, LU MingXin, QIN Lei, HAN JunSong& FANG XiWen : Asymmetric encryption and signature method with DNA technology Science China Press and Springer-Verlag Berlin Heidelberg 2010.

[7] Wong PC, Wong KK, Foote H: Organic data memory using the DNA approach. Communications of the ACM 2003, 46:

[8] B. Corona, M. Nakano, H. Pérez, “Adaptive Watermarking Algorithm for Binary Image Watermarks”, *Lecture Notes in Computer Science, Springer, pp. 207-215, 2004*.

[9] A. A. Reddy and B. N. Chatterji, “A new wavelet based logo-watermarking scheme,” Pattern Recognition Letters, vol. 26, pp. 1019-1027, 2005.

[10] P. S. Huang, C. S. Chiang, C. P. Chang, and T. M. Tu, “Robust spatial watermarking technique for colour images via direct saturation adjustment,” Vision, Image and Signal Processing, IEE Proceedings -, vol. 152, pp. 561-574, 2005.