International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 3, Issue 7, July 2014

Design and implementation of Symmetric Cryptographic Algorithm Using 5x5 Square Matrix with Single Point Crossover

SK. Myrunnisa Begum (PG Scholar)[1] **B.Sarala (Associate professor)[2]**

Abstract:-**Thispaper introduces modified cryptographic technique, which uses two keys for encryption and same two keys are used for Decryption. This technique is generated by an intermediate cipher followed by genetic function singlepoint crossover to produce final cipher. This paper uses 5x5 squarematrix and the input data stream is put into the square matrix. Sum of the left diagonal positional values will be key 1 and with that key1 intermediate cipher text will be produced. Key 1 is variable and its value depends on the plain text.Single point crossover, is applied on the binary field of intermediate cipher text. Thefivedigit randomnumber will generate as key 2, and this key 2 is randomly generated, and fixed by the user before doing the crossover.According to the digits of key 2 block division process and crossover point is finalized to produce final cipher. Reverse procedure with the same keys will generate as plain text.**

Index Terms:**Plain text, cipher text, key, crossover, encryption, decryption, symmetric algorithm**.

IIntroduction

If an air interface is used as a communication channel, the information of the communication may be exposed to an eavesdropper or system services can be fraudulently altered. In order to have good security over wired/wireless channel, it is implemented using advanced cryptography through encryption & decryption for secured communication. There are many aspects to security applications, ranging from secure commerce and payments to private communications and protecting passwords. One essential aspect for secure communications is that of cryptography. The demand for effective internet security is increasing exponentially day by day [1].So for high protection, to maintain integrity of the data a robust and secure security is needed. With wide application of internet,especially the increasing adoption of the cloud computing paradigm, storing sensitive user data to un-trusted remote hosts on internet has been popular.Cryptography is the art of science to keep messages secure.It provides confidentiality in addition to that it provides authentication, integrity, non-repudiation[2].

This modified cryptographic algorithm, describes the encryption and decryptionprocesses. This algorithm is based on the process of substitution and genetic function[3]. Genetic function is used in the binary form of intermediate cipher using another key which is randomly generated.

Authors:
1. SK MyrunnisaBegum (PG Scholar), *ECE, MVSR Engineering College.Hyderabad, India.*
2. B.Sarala(Associate professor), *ECE, MVSR Engineering college, Hyderabad,India.*

Finally cipher text is obtained, to retrieve it back to getplain text using decryption. The block size and key length are variable and can be fixed by the user at the beginning of the ciphering. Crossover operator has the significance as that of crossover in natural genetic process. In this operation two blocks are taken and a new block is generated by taking some attributes of first block and rest from second block. In Genetic algorithms, a cross can be single point, double point, and uniform crossover[3].

Network Security is to provide security of the data which is to be transmitted to the network, which is done by cryptography [4]. Cryptography means hiding the data, for hiding the information encryption is done by encryption algorithm, to retrieve it back decryption is done by decryption algorithm. The word encryption has been coined from the word “cryptography” which was derived from the ancient Greek words “kryptos” (hidden) and “graphia” (writing). Encryption is the process of transforming text or data into an unintelligible form called cipher. Reversing the process of encryption and transforming the cipher back into its original form is called decryption. Encryption and decryption comprise the science of cryptography as it is applied to the modern computer. Encryption is the most effective way to achieve data security. To read an encrypted file, we must have access to a secret key that enables you to decrypt it.Many encryptionalgorithms are widely available and used in information security. Security depends on block size and length of the key. They can be categorized intoSymmetric (private) and Asymmetric (public) key encryption. In Symmetric keys encryption or secret keyencryption, only one key is used to encrypt and decrypt data. The key should be distributed before transmissionbetween entities. Keys play an important role. If weak key is used in algorithm then everyone may decrypt the data.Strength of Symmetric key encryption depends on the sizeof key used. For the same algorithm, encryption using longer key is harder to break than the one done using smaller key. There are many examples of strong and weak keys of cryptography algorithms like RC2, DES, 3DES,RC6, Blowfish, and AES. RC2 uses one 64-bit key.DES uses one 64-bits key. Triple DES(3DES) uses three 64-bits keys while AES uses various sizes of 128, 192 and 256 bit keys.

Blowfish uses various (32-448), default 128bits while RC6 is used various (128,192,256) bits keys [1-5]. Asymmetric key encryption or public key encryption is used to solve the problem of key distribution. In Asymmetric keys, two keys are used; private and public keys. Public key is used for encryption and private key is used for decryption (E.g. RSA and Digital Signatures). Because users tend to use two keys: public key, which is known to the public and

ISSN: 2278 – 909X

private key which is known only to the user. There is no need for distributing them prior to transmission. However, public key encryption is based on mathematical functions, computationally intensive and is not very efficient for small mobile devices [6]. Asymmetric encryption techniques are almost 1000 times slower than Symmetric techniques, because they require more computational processing power [7]. So we are implementing 5*5 matrix using symmetric encryption algorithm.

Data Encryption Standard, was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology). DES is (64 bits key size with 64 bits block size) . Since that time, many attacks and methods recorded the weaknesses of DES, which made it an insecure block cipher [8],[9]. *3DES* is an enhancement of DES; it is 64 bit block sizewith 192 bits key size. In this standard the encryption method is similar to the one in the original DES but applied 3 times to increase the encryption level and the average safe time. It is a known fact that 3DES is slower than other block cipher methods [8]. *RC2* is a block cipher with a 64-bits block cipher with a variable key size that range from 8 to128 bits. RC2 is vulnerable to a related-key attack using 234 chosen plaintexts [8]. Blowfishis block cipher 64-bit block - can be used as a replacement for the DES algorithm. It takes a variable length key, ranging from 32 bits to 448 bits; default 128 bits. Blowfish is unpatented, license-free, and is available free for all uses. Blowfish has variants of 14 rounds or less. Blowfish is successor to two fish [10]. AES is a block cipher .It has variable key length of 128,192, or 256 bits; default 256. It encrypts data blocks of 128 bits in 10, 12 and 14 round depending on the key size. AES encryption is fast and flexible; it can be implemented on various platforms especially in small devices. Also, AES has been carefully tested for many security applications [8], [11]. *RC6* is block cipher derived from RC5. It was designed to meet the requirements of the Advanced Encryption Standard competition. RC6 proper has a block size of 128 bits and supports key sizes of 128, 192 and 256 bits. Some references consider RC6 as Advanced Encryption Standard [8].DES and 3DES are affected by brute force attack. For the reasons of more security and efficiency we are implementing a new modified cryptographic algorithm using 5*5 matrix.

This modified cryptographic symmetric algorithm using square matrix single point crossover on binary field is the new technique, it uses more number of bits i.e.1600 bits of data compared with previous cryptographic algorithms. It is designed by using 5X5 matrix. It generates complex cipher text, and uses 1600 bits. So, it is more secure compared with other cryptographic algorithms. The symmetric algorithm encryption is considered to be efficient both for hardware and software implementations. Compared to software, hardware implementation is more reliable, provides increased throughput and offers better security. The security of a symmetric cryptosystem depends on the strength of the algorithm and length of the key. In this paper compares the time delay and synthesis report for existing algorithms and proposed algorithms. The rest of the paper organized as follows.

In section II related work is presented. In section III modified cryptography encryption and decryption algorithms are discussed. In section IV crossover genetic function is represented and numerical results also presented. In section V results and synthesis report computer simulation results are presented. In section VI conclusion and future works are listed.

II Related work

SubhranilSom proposed cryptographic technique by square matrix and single point crossover on binary field.This existing methodprovides security for less number of bits that is 1024 bits that is fixed and it uses 4*4 square matrix and by applying single point cross over to produce final cipher text little bit complex.

This proposed modified cryptographic algorithm consists 1600 bits of data. It uses more number of bits compared with existing algorithms, so it provides more security. It uses 5*5 square matrix, key 1 is variable and key 2 is automatically generated and fixed by the user before ciphering and applying single point crossover to produce final cipher text very complex. In this paper we design and implemented symmetric algorithm using 5 by 5 square matrix.

III Modified cryptography

In this modified cryptographic algorithm first count the number of letters present in the plain text and each letter is placed in the square matrix in a specific manner (Diagonally in the matrix). Square matrix is selected according to the size of the input plain text. Key 1 will be produced by adding the left diagonals positional values of the matrix. This key 1 will added with each letters position value of matrix to generate the intermediate cipher text. This text is arranged in the square matrix, each letter of the matrix is represented as 64 bits binary code of ASCII values. So that total number bits of intermediate cipher text will be 25*64 bits =1600 bits, next five digit random numbers will generate as key 2 that is 54825. Depending on the key 2 total 1600 bits are divided into five sections, because the first digit of key 2 is five and each section is again divided into 4, 8, 2,5 blocks respectively because they are the next digits of key 2. 1600 bits are exactly divided by five so there is no remainder. In this paper, modified cryptographic algorithm using 5*5 square matrixsingle point crossover is implemented. In single point crossover, if the number of blocks are even crossover operator is applied in between block1 and block 2, block 3 and block 4 etc. If the number of blocks are odd crossover operator is applied in between first block and last block, second block and before block of last block, and middle block as it is. In encryption process the crossover is operated using left shift operation. By applying singlepoint crossover in between two blocks it generates two child blocks that are of same size of parent blocks& that are having the characteristics of both parent blocks. In single point crossover, one crossover point is selected and the crossover operator is applied to generate the child blocks. In double point crossover, two crossover points are selected and the crossover operator is applied to generate the child blocks. Like this way child blocks are produced. Concatenate all child blocks section wise to produce final cipher text in binary form. Length of binary field will be 1600 bits,these bits will be divided into 25 blocks each of 64 bits. To convert final cipher text to plain text, we used decryption. In decryption process the crossover is operated using right shift operation. It's also based on substitution and genetic function.

ISSN: 2278 – 909X

IVRepresentation of crossover in genetic function

This section refers to the representation of single point crossover when even and odd number of blocks is present. At the time of encryption or decryption total number of blocks of each section is even, block diagram of crossover is shown in Figure 4.1. Each block is denoted by a number, N=even number.

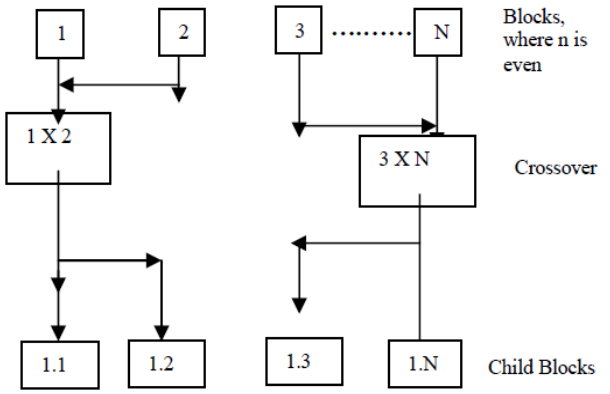

Figure 4.1 represents crossover for even number of blocks

At the time of encryption or decryption total number of blocks of each section isodd, block diagram of crossover is shown in Figure 4.2. Each block is denoted by a number. N=odd number.

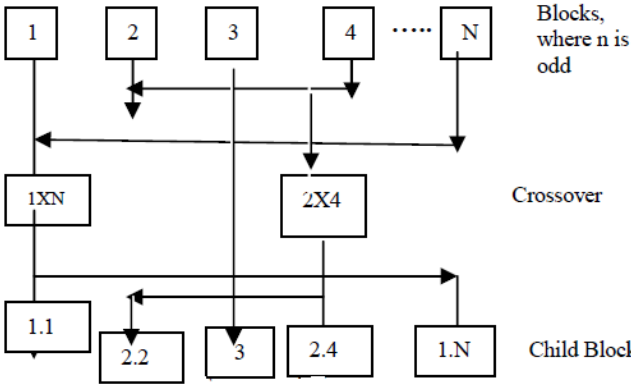

Figure 4.2 represents crossover for odd number of blocks

Crossover is a process of taking more than one parent solutions and producing a child solution from them. Many crossover techniques exist for organisms which use different data structures to store themselves. Crossover is a genetic operator that combines two chromosomes to produce a new chromosome. Crossover can be single, double, and uniform crossover. In double point crossover, two crossover points are selected on the parent organism strings. Everything between two points is swapped between the parent organisms, rendering two child organisms. Uniform crossover uses a fixed mixing ratio between two parents. Unlike one and two point crossover, the uniform crossover enables the parent chromosomes to contribute the gene level rather than segment level.

Modified cryptographic algorithm using single point crossover:

Encryption: The translation of data into a secret code is called encryption. Encryption is the most effective way to achieve data security. To read an encrypted file, we must have access to a secret key or password that enables us to decrypt it. Unencrypted data is called plain text; encrypted data is referred as cipher text. It's the conversion of data into a form called a cipher text, that can't be easily understood by unauthorized people. Decryption is process of converting encrypted data back into its original form, it can be understood. The use of encryption or decryption is as old as the art of communication. In war time, a cipher often incorrectly called a code can be employed to keep the enemy from obtaining the contents of transmissions. (Technically a code is a means of representing a signal without the intent of keeping it secret. (EX: ASCII & Morse code).

Simple ciphers include the substitution of letters for numbers, the rotation of letters in alphabet, and scrambling of voice signals by inverting the sideband frequencies. More complex ciphers work according to sophisticated computer algorithm that rearranges the data bits in digital signals. In order to easily recover the contents of an encrypted signal, the correct decryption key is required. The key is an algorithm that undergoes the work of the encryption algorithm, alternatively a computer can be used in an attempt to break the cipher. The more complex the encryption algorithm, the more difficult it becomes to eavesdrop on communications without access to the key. A type of encryption where the same key is used to encrypt and decrypt the message, it is known as symmetric encryption. This differs from asymmetric encryption, which uses one key to encrypt a message and another to decrypt the message. Let the plain text is: PULLAREDDYMEMORIALSCHOOLS. Size of plain text is **25**.So square matrix has been taken as 5*5 and all letters of plaintext is placed in the box of the matrix diagonally as shown below.

P	U	L	E	M
L	A	D	E	I
R	D	M	A	C
Y	O	L	H	O
R	S	O	L	S

Key1= positional value of 'P'+positional value of 'A'+positional value of 'M'+positional value of 'H'+positional value of'S'=16+1+13+8+19=57[where A=1……z=26].

Now the key1 will be added with each letters positional value of matrix to generate the intermediate cipher text.

P=57+16=73=U;U=57+21=78=Z;L=57+12=69=Q;L=57+12=69=Q;A=57+1=58=F;R=57+18=75=W;E=57+5=62=J;D=57+4=61=I;D=57+4=61=I;Y=57+25=82=D;M=57+13=70=R;E=57+5=62=J;M=57+13=70=R;O=57+15=72=T;R=57+18=75=W;I=57+9=66=N;A=57+1=58=F;L=57+12=69=Q;S=57+19=76=X;C=57+3=60=H;H=57+8=65=M;O=57+15=72=T;O=57+15=72=T;L=57+12=69=Q;S=57+19=76=X;

Intermediate cipher text:

UZQJRQFIJNWIRFHDTQMTWXTQX

U	Z	Q	J	R
Q	F	I	J	N
W	I	R	F	H
D	T	Q	M	T
W	X	T	Q	X

ISSN: 2278 – 909X

Each letter of matrix is represented as 64bits of binary code of ASCII value. So that total bits of intermediate cipher text will be 25*64=1600bits this is given below.

Letter	Binary code(64bit)
U	0000000000000000000000000000000000000000000000000000000001010101
Z	0000000000000000000000000000000000000000000000000000000001011010
Q	0000000000000000000000000000000000000000000000000000000001010001
J	0000000000000000000000000000000000000000000000000000000001001010
R	0000000000000000000000000000000000000000000000000000000001010010
Q	0000000000000000000000000000000000000000000000000000000001010001
F	0000000000000000000000000000000000000000000000000000000001000110
I	0000000000000000000000000000000000000000000000000000000001001001
J	0000000000000000000000000000000000000000000000000000000001001010
N	0000000000000000000000000000000000000000000000000000000001001110
W	0000000000000000000000000000000000000000000000000000000001010111
I	0000000000000000000000000000000000000000000000000000000001001001
R	0000000000000000000000000000000000000000000000000000000001010010
F	0000000000000000000000000000000000000000000000000000000001000110
H	0000000000000000000000000000000000000000000000000000000001001000
D	0000000000000000000000000000000000000000000000000000000001000100
T	0000000000000000000000000000000000000000000000000000000001010100
Q	0000000000000000000000000000000000000000000000000000000001010001
M	0000000000000000000000000000000000000000000000000000000001001101
T	0000000000000000000000000000000000000000000000000000000001010100
W	0000000000000000000000000000000000000000000000000000000001010111
X	0000000000000000000000000000000000000000000000000000000001011000
T	0000000000000000000000000000000000000000000000000000000001010100
Q	0000000000000000000000000000000000000000000000000000000001010001
X	0000000000000000000000000000000000000000000000000000000001011000

Table I represents the conversion of alphabet to ASCII values

Binary representation of intermediate cipher is:
0000000000000000000000000000000000000000000000000000000000000000
0001010101000000000000000000000000000000000000000000000000000000
0000000000001011010000000000000000000000000000000000000000000000
0000000000000000000001010001000000000000000000000000000000000000
0000000000000000000000000000000100101000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
1000100000000000000000000000000000000000000000000000000000000000
0000000010001100000000000000000000000000000000000000000000000000
0000000000000000000100100100000000000000000000000000000000000000
0000000000000000000000000000000100101000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000100100100000000000000000000000000000000000000000000000000000
0000000000000010100100000000000000000000000000000000000000000000
0000000000000000000000010001100000000000000000000000000000000000
0000000000000000000000000000000010010000000000000000000000000000
0000000000000000000000000000000000000000000000000001000100000000
0000000000000000000000000000000000000000000000000000000000000000
1010100000000000000000000000000000000000000000000000000000000000
0000000000101000100000000000000000000000000000000000000000000000
0000000000000000000010011010000000000000000000000000000000000000
0000000000000000000000000000000101010000000000000000000000000000
0000000000000000000000000000000000000000101011100000000000000000
0000000000000000000000000000000000000000000000000000000101011000
0000000000000000000000000000000000000000000000000000000000000000
0000010101000000000000000000000000000000000000000000000000000000
0000000000000000101000100000000000000000000000000000000000000000
0000000000000000000000000001011000.

Before ciphering, 5digit random number has been generated as key2.The randomly generated number key 2 is 54825. Depending on the key2, 1600bits will divided into 5 sections, because the 1st digit of key 2 is 5 and each section again divided into 4,8,2,5 blocks respectively as because 4,8,2,5 are the next digits of key 2.

Section1 is divided into 4blocks,section 2 is divided into 8blocks, section3 is divided into 2blocks, section 4 is divided into 5blocksand section5 is divided into 4blocks.

Each section contains (1600/5)bits=320bits.

Section1(each block contains 80bits (320/4))

Block1	Block2	Block3	Block4
0000000000 00 0000000000 00 00000000	0000000000 00 0000000000 00 00000000	0000000000 00 0000000000 00 01010000	0000000001 00 1010000000 00 0000000000
0000000000 00 0000000001 01 01010000 00 00000000	0000010110 10 0000000000 00 0000000000 00 00000000	0000000000 00 0000000000 00 0000000000 00 01010010	0000000000 00 0000000000 00 0000000000 00

Section2(each block contains 40bits (320/8))

Block1	Block2	Block3	Block4
0000000000 0000000000 0000000000 0000000000	0000000000 0000000101 0001000000 0000000000	0000000000 0000000000 0000000000 0000000000	0100011000 0000000000 0000000000 0000000000

Block5	Block6	Block7	Block8
0000000000 0000000000 0000010010 0100000000	0000000000 0000000000 0000000000 0000000000	0000000001 0010100000 0000000000 0000000000	0000000000 0000000000 0000000000 0001001110

Section3(each block contains 160bits (320/2))

Block1	Block2
00000000000000000000 00000000000000000000 00000000000000101011100	00000000000000000000 00010100100000000000 00000000000000000000

ISSN: 2278 – 909X

0000000000000000000000 0000000000000000000000 0000000000010010010000 0000000000000000000000 000000	0000000000000000000000 0100011000000000000000 0000000000000000000000 0000000000000000000001 001000

Section4(each block contains 64bits (320/5))

Block1	Block2	Block3	Block4	Block5
00000000 00000000 00000000 00000000 00000000 00000000 00000000 01000100	00000000 00000000 00000000 00000000 00000000 00000000 00000000 01010100	00000000 00000000 00000000 00000000 00000000 00000000 00000000 01010001	00000000 00000000 00000000 00000000 00000000 00000000 00000000 01001101	00000000 00000000 00000000 00000000 00000000 00000000 00000000 01010100

Section5(each block contains 80bits(320/4))

Block1	Block2	Block3	Block4
00000000000 0 00000000000 0 00000000000 0 00000000000 0 00000000010 1 01110000000 0 00000000	00000000000 0 00000000000 0 00000000000 0 00000101100 0 00000000000 0 00000000000 0 00000000	00000000000 0 00000000000 0 01010100000 0 00000000000 0 00000000000 0 00000000000 0 00000000	0000000 0 0101000 1 0000000 0 0000000 0 0000000 0 0000000 0 0000000 0 0000000 0 0000000 0 0101100 0

Single point crossover is to apply to each section of odd and even blocks to generate the child blocks.

Section 1: Number of blocks are 4(Even)

Block 1 X Block 2

000**0**000000000000000000000000000000000000000000
00000000101010100000000000000000

X

000**0**00000000000000000000000000000000000010110100
00000000000000000000000000000000

0000000000000000000000000000000000000000000000000
000001010101000000000000000000000 (Block1.1)

00000000000000000000000000000000000010110100000
00000000000000000000000000000000 (Block 1.2)

Block 3 X Block 4

000**0**0000000000000000000001010001000000000000000000
00000000000000000000000000000000

X

000**0**00000100101000000000000000000000000000000000
00000000000000000000000001010010

000000000000000000000001010001000000000000000000000
00000000000000000000000000000000 (Block 1.3)

0000000100101000000000000000000000000000000000000
0000000000000000000001010010000(Block1.4)

Section 2: Number of blocks 8 (Even)

Block 1 X Block 2

0000000**0**00000000000000000000000000000000

X

0000000**0**00000000010100010000000000000000

0000000000000000000000000000000000000000
(Block2.1)

0000000000101000100000000000000000000000
(Block2.2)

Block 3 X Block 4

0000000**0**00000000000000000000000000000000

X

0100011**0**00000000000000000000000000000000

0000000000000000000000000000000000100011(Block2.3)

0000000000000000000000000000000000000000(Block2.4)

Block 5 X Block 6

0000000**0**00000000000000000100100100000000

X

0000000**0**00000000000000000000000000000000

0000000000000000001001001000000000000000000(Block2.5)

0000000000000000000000000000000000000000(Block2.6)

Section 3: Number of blocks 2(Even)

Block 1 X Block 2

0**0**00000000000000000000000000000000000000000000000
000000010101110000000000000000000000000000000000
0000000000000000000001001001000000000000000000000
000000000000

X

0**0**0000000000000000000000010100100000000000000000000
000000000000000000000000000000000000000000100011000
000000000000000000000000000000000000000000000000000
0000001001000

0**0**00000000000000000000000000000000000000000000000
000000010101110000000000000000000000000000000000
0000000000000000000001001001000000000000000000000
0000000000000 (Block 3.1)

0000000000000000000000010100100000000000000000000
0000000000000000000000000000000000000001000110000
000000000000000000000000000000000000000000000000000
0000010010000 (Block 3.2)

Section 4: Number of blocks 5(Odd)

Block1 X Block5

0000**0**000000000000000000000000000000000000000000000
000000001000100X

0000**0**000000000000000000000000000000000000000000000
000000001010100

000000000000000000000000000000000000000000000000000
0000010001000000 (Block4.1)

000000000000000000000000000000000000000000000000000
0000010101000000 (Block4.5)

Block 2 X Block 4

0000**0**000000000000000000000000000000000000000000000
000000001010100 X

0000**0**000000000000000000000000000000000000000000000
000000001001101

0000000000000000000000000000000000000000000000000000
000010101000000 (Block 4.2)

0000000000000000000000000000000000000000000000000000
000010011010000 (Block 4.4)

Block 3 remains same as it is

000000000000000000000000000000000000000000000000000
000000001010001

Section 5: Number of blocks 4(Even)

ISSN: 2278 – 909X

Block1 X Block2
0000000000000000000000000000000000000000000000000000000010101110000000000000000
X
0000000000000000000000000000000000000000010110000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000010101110000000000000000 (Block5.1)
0000000000000000000000000000000000000000010110000000000000000000000000000000(Block5.2)
Block3 X Block4
0000000000000000000000010101000000000000000000000000000000000000000000000000
X
0000000010100010000000000000000000000000000000000000000000000000001011000
0000000000000000000000010101000000000000000000000000000000000000000000000000 (Block5.3)
0000001010001000000000000000000000000000000000000000000000000000001011000000(Block5.4)

The resultant blocks block 1.1, block 1.2……block 5.2 are called as child blocks. Concatenate all child Blocks sectionwise to produce the cipher text in binary form. Length of Binary field will be '1600' bits these bits will divide into 25 blocks of each 64bits.

0000000000000000000000000000000000000000000000000000000101010101000 = 680 = 2a8h
0000000000000000000000000000000000000000000000000000000101101000000 = 720 = 2d0h
0000000000000000000000000000000000000000000000000000000100010001000 = 648 = 288h
0000000000000000000000000000000000000000000000000000000100101000000 = 592 = 250h
0000000000000000000000000000000000000000000000000000000101001000000 = 656 = 290h
0000000000000000000000000000000000000000000000000000010100000000000 = 10368 = 2880h
0000000000000000000000000000000000000000000000000000010001100000000 = 8960 = 2300h
0000000000000000000000000000000000000000000000000000010010000000000 = 9344 = 2480h
0000000000000000000000000000000000000000000000000000010010100000000 = 9472 = 2500h
0000000000000000000000000000000000000000000000000000010011100000000 = 9984 = 2700h
0000000000000000000000000000000000000000000000000000000010101110 = 174 = aeh
0000000000000000000000000000000000000000000000000000000010010010 = 146 = 92h
0000000000000000000000000000000000000000000000000000000010100100 = 164 = a4h
0000000000000000000000000000000000000000000000000000000000000000 = 140 = 8ch
0000000000000000000000000000000000000000000000000000000010010000 = 144 = 90h
0000000000000000000000000000000000000000000000000000000100010000000 = 1088 = 440h
0000000000000000000000000000000000000000000000000000000101010000000 = 1344 = 540h
0000000000000000000000000000000000000000000000000000000001010001 = 81 = 51h
0000000000000000000000000000000000000000000000000000000100111010000 = 1232 = 4d0h
0000000000000000000000000000000000000000000000000000000101010000000 = 1344 = 540h
0000000000000000000000000000000000000000000000000000000010101110000 = 696 = 2b8h
0000000000000000000000000000000000000000000000000000000000101100000000 = 704 = 2c0h
0000000000000000000000000000000000000000000000000000000010101000000 = 672 = 2a0h
0000000000000000000000000000000000000000000000000000000010001000000 = 648 = 288h
0000000000000000000000000000000000000000000000000000000010110000000 = 704 = 2c0h

Final cipher text in hexadecimal format is 2a8h 2d0h 288h 250h 290h 2880h 2300h 2480h 2500h 2700h aeh 92h a4h 8ch 90h 440h 540h51h 4d0h 540h 2b8h 2c0h 2a0h 288h 2c0h.

Decryption: Decryption is a reverse process of encryption**.** It converts intermediate cipher text into plain text. In this process crossover operation is operated using right shift operation.

Cipher text: 2a8h 2d0h 288h 250h 290h 2880h 2300h 2480h 2500h 2700h aeh 92h a4h 8ch 90h 440h 540h 51h 4d0h 540h 2b8h 2c0h 2a0h 288h 2c0h.

Binary equivalent each number of cipher text:

0000000000000000000000000000000000000000000000000000000101010101000 = 680 = 2a8h
0000000000000000000000000000000000000000000000000000000101101000000 = 720 = 2d0
0000000000000000000000000000000000000000000000000000000100010001000 = 648 = 288h
0000000000000000000000000000000000000000000000000000000100101000000 = 592 = 250h
0000000000000000000000000000000000000000000000000000000101001000000 = 656 = 290h
0000000000000000000000000000000000000000000000000000010100000000000 = 10368 = 2880h
0000000000000000000000000000000000000000000000000000010001100000000 = 8960 = 2300h
0000000000000000000000000000000000000000000000000000010010000000000 = 9344 = 2480h
0000000000000000000000000000000000000000000000000000010010100000000 = 9472 = 2500h
0000000000000000000000000000000000000000000000000000010011100000000 = 9984 = 2700h
0000000000000000000000000000000000000000000000000000000010101110 = 174 = aeh
0000000000000000000000000000000000000000000000000000000010010010 = 146 = 92h
0000000000000000000000000000000000000000000000000000000010100100 = 164 = a4h
0000000000000000000000000000000000000000000000000000000000000000 = 140 = 8ch
0000000000000000000000000000000000000000000000000000000010010000 = 144 = 90h
0000000000000000000000000000000000000000000000000000000100010000000 = 1088 = 440h
0000000000000000000000000000000000000000000000000000000101010000000 = 1344 = 540h
0000000000000000000000000000000000000000000000000000000001010001 = 81 = 51h
0000000000000000000000000000000000000000000000000000000100111010000 = 1232 = 4d0h
0000000000000000000000000000000000000000000000000000000101010000000 = 1344 = 540h
0000000000000000000000000000000000000000000000000000000010101110000 = 696 = 2b8h

ISSN: 2278 – 909X

00000000000000000000000000000000000000000000000000
000010110000000 = 704 = 2c0h
00000000000000000000000000000000000000000000000000
000001010100000 = 672 = 2a0h
00000000000000000000000000000000000000000000000000
000001010001000 = 648 = 288h
00000000000000000000000000000000000000000000000000
000001011000000 = 704 = 2c0h
Total numbers of bits are 1600, Key 1 = 57 and key 2 = 54825
(1600/5) bits = 320 bits. According to the key 2, total 1600 bits will divide into 5 sections. Partitions are made in the similar order as in encryption process and the decryption algorithm is applied of genetic function to generate the intermediate cipher text.

Section1(each block contains 80bits(320/4))

Block1	Block2	Block3	Block4
0000000000 00	0000000000 00	0000000000 00	0000001001 01
0000000000 00	0000000000 00	0000000000 10	0000000000 00
0000000000 00	0000000000 00	1000100000 00	0000000000 00
0000000000 00	0010110100 00	0000000000 00	0000000000 00
0000001010 10	0000000000 00	0000000000 00	0000000000 00
1000000000 00	0000000000 00	0000000000 00	0000000000 10
00000000	00000000	00000000	10010000

Section2(each block contains 40bits (320/8))

Block1	Block2	Block3	Block4
0000000000	0000000000	0000000000	0000000000
0000000000	1010001000	0000000000	0000000000
0000000000	0000000000	0000000000	0000000000
0000000000	0000000000	0000100011	0000000000

Block5	Block6	Block7	Block8
0000000000	0000000000	0010010100	0000000000
0000000010	0000000000	0000000000	0000000000
0100100000	0000000000	0000000000	0000001001
0000000000	0000000000	0000000000	1100000000

Section3(each block contains 160bits (320/2))

Block1	Block2
00000000000000000000	00000000000000000000
00000000000000000000	00101001000000000000
00000000001010111000	00000000000000000000
00000000000000000000	00000000000000000000
00000000000000000000	10001100000000000000
00000000001001001000	00000000000000000000
00000000000000000000	00000000000000000010
000000	010000

Section 4(each block contains 64bits (320/5))

Block1	Block2	Block3	Block4	Block5
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000100	00000101	00000000	00000100	00000101
01000000	01000000	01010001	11010000	01000000

Section 5(each block contains 80bits (320/4))

Block1	Block2	Block3	Block4
0000000000 00	0000000000 00	0000000000 00	0000001010 00
0000000000 00	0000000000 00	0000000000 10	1000000000 00
0000000000 00	0000000000 00	1010000000 00	0000000000 00
0000000000 00	0010110000 00	0000000000 00	0000000000 00
0000001010 11	0000000000 00	0000000000 00	0000000000 00
1000000000 00	0000000000 00	0000000000 00	0000000000 10
00000000	00000000	00000000	11000000

Single point crossover is applied to each section of the blocks, to generate the child blocks.

Section1: Number of blocks 4 (even)

Block1 X Block2

00000000000000000000000000000000000000000000000000
0000010101010000000000000000
X
00000000000000000000000000000000000010110100000
00000000000000000000000000000
00000000000000000000000000000000000000000000000000
0000000010101010000000000000000(Block1.1)
00000000000000000000000000000000000010110100
00000000000000000000000000000000(Block1.2)

Block 3 X Block 4

00000000000000000000010100010000000000000000000000
00000000000000000000000000000000
X
00000010010100000000000000000000000000000000000000
00000000000000000000010100100000
00000000000000000000000010100010000000000000000000
00000000000000000000000000000000(Block1.3)
00000000010010100000000000000000000000000000000000
00000000000000000000000010100100(Block1.4)

Section 2: Number of blocks 8 (even)

Block 1 X Block 2

00000000000000000000000000000000000000000000000000
X
00000000001010001000000000000000000000000
00000000000000000000000000000000000000000(Block2.1)
00000000000000001010001000000000000000000(Block2.2)

Block 3 X Block 4

00000000000000000000000000000000000100011
X
00000000000000000000000000000000000000000
00000000000000000000000000000000000000000
(Block2.3)
01000110000000000000000000000000000000000
(Block2.4)

Block 5 X Block 6

00000000000000000010010010000000000000000
X
00000000000000000000000000000000000000000

ISSN: 2278 – 909X

00000000000000000000000001001001000000000**0**
(Block2.5)
000000000000000000000000000000000000000**0**
(Block2.6)
Block 7 X Block 8
00100101000000000000000000000000**0**0000000
X
0000000000000000000000000100111**0**00000000
000000000100101000000000000000000000000**0**
(Block2.7)
000000000000000000000000000000000100111**0**
(Block2.8)

Section 3:Number of **blocks 2(Even)**
Block1 X Block2
0000000000000000000000000000000000000000000000000
0000000101011100000000000000000000000000000000000
0000000000000000000000001001001000000000000000000
00000000000**0**0
X
0000000000000000000000001010010000000000000000000
0000000000000000000000000000000000000001000110000
0000000000000000000000000000000000000000000000000
00000100100**0**0
0000000000000000000000000000000000000000000000000
0000000010101110000000000000000000000000000000000
0000000000000000000000001001001000000000000000000
000000000000**0**(Block3.1)
0000000000000000000000000101001000000000000000000
0000000000000000000000000000000000000000100011000
0000000000000000000000000000000000000000000000000
000000100100**0** (Block3.2)

Section 4:Number of blocks 5 (Odd)
Block 1 X Block 5
0000000000000000000000000000000000000000000000000
0000100010**0**0000 X
0000000000000000000000000000000000000000000000000
0000101010**0**0000
0000000000000000000000000000000000000000000000000
00000000100010**0**(Block4.1)
0000000000000000000000000000000000000000000000000
00000000101010**0**(Block 4.5)
Block 2 X Block 4
0000000000000000000000000000000000000000000000000
0000101010**0**0000
X
0000000000000000000000000000000000000000000000000
0000100110**1**0000
0000000000000000000000000000000000000000000000000
00000000101010**0**(Block4.2)
0000000000000000000000000000000000000000000000000
00000000100110**1**(Block4.4)
Block 3is remains same
0000000000000000000000000000000000000000000000000
000000001010001

Section 5:Number of blocks 4 (Even)
Block1 X Block 2
0000000000000000000000000000000000000000000000000
00001010111000000000000000**0**000
X
0000000000000000000000000000000000000010110000000
000000000000000000000000000**0**000
0000000000000000000000000000000000000000000000000
0000000101011100000000000000**0** (Block 5.1)
0000000000000000000000000000000000000000010110000
000000000000000000000000000000**0** (Block 5.2)
Block 3 X Block 4
0000000000000000000000101010000000000000000000000
000000000000000000000000000**0**000
X
0000001010001000000000000000000000000000000000000
000000000000000000000101100**0**000
0000000000000000000000000101010000000000000000000
000000000000000000000000000000**0**(Block 5.3)
0000000001010001000000000000000000000000000000000
000000000000000000000000101100**0**(Block 5.4)

Concatenate all child blocks section wise to produce intermediate cipher text in binary form.Length of binary field will be1600 bits.1600 bits will be divided into 25 blocks of 64bits.

64 bits binary	Letter
0000000000000000000000000000000000000000000000 000000000001010101	U
0000000000000000000000000000000000000000000000 000000000001011010	Z
0000000000000000000000000000000000000000000000 000000000001010001	Q
0000000000000000000000000000000000000000000000 000000000001001010	J
0000000000000000000000000000000000000000000000 000000000001010010	R
0000000000000000000000000000000000000000000000 000000000001010001	Q
0000000000000000000000000000000000000000000000 000000000001000110	F
0000000000000000000000000000000000000000000000 000000000001001001	I
0000000000000000000000000000000000000000000000 000000000001001010	J
0000000000000000000000000000000000000000000000 000000000001001110	N
0000000000000000000000000000000000000000000000 000000000001010111	W
0000000000000000000000000000000000000000000000 000000000001001001	I
0000000000000000000000000000000000000000000000 000000000001010010	R
0000000000000000000000000000000000000000000000 000000000001000110	F
0000000000000000000000000000000000000000000000 000000000001001000	H
00000000000000000000000000000000000000000000000 000000000001000100	D
0000000000000000000000000000000000000000000000 000000000001010100	T
0000000000000000000000000000000000000000000000 000000000001010001	Q
0000000000000000000000000000000000000000000000 000000000001001101	M
0000000000000000000000000000000000000000000000 000000000001010100	T
0000000000000000000000000000000000000000000000 000000000001010111	W
0000000000000000000000000000000000000000000000 000000000001011000	X
0000000000000000000000000000000000000000000000 000000000001010100	T
0000000000000000000000000000000000000000000000 000000000001010001	Q
0000000000000000000000000000000000000000000000 000000000001011000	X

Table II represents the conversion of ASCII value to alphabet

ISSN: 2278 – 909X

Intermediate cipher textis: UZQJRQFIJNWIRFHDTQMTWXTQX.
Substitute intermediate cipher byitspositional values (where A=1, B=2 …Z=26) and key 1=57.
U=(21+52)-(57)=16=P,Z=(26+52)-(57)=21=U,Q=(17+52)-(57)=12=L,J = (10+52)-(57) = 5 = E, R = (52+18)-(57) = 13 = M,Q=(17+52)-(57)=12=L,F= (52+6)-(57)=1=A,I =(52+9)-(57) = 4 = D,J = (10+52)-(57) = 5 = E, N = (52+14)-(57) = 9 = I,W= (52+23)-(57)=18=R,I=(52+9)-(57)=4=D, R = (52+18)-(57) = 13 = M,F = (52+ 6)-(57) = 1 = A,H = (52+8)-(57) =3 = C,D = (52+4)-(57) = 25 = Y, T = (52+20)-(57) = 15 = O,Q = (17+52)-(57) = 12 = L,M = (52+13)-(57) = 8 = H,T = (52+20)-(57) = 15 = O,W = (52+ 23)-(57) = 18 = R,X = (52+24)-(57) = 19 = S, T = (52+20)-(57) = 15 = O,Q = (17+52)-(57) = 12 = L,X = (52+24)-(57) = 19 = S.
Each letter is placed into the square matrix according to the technique to get the plain text. The square matrix with the Plain text is as shown in figure below.

P	U	L	E	M
L	A	D	E	I
R	D	M	A	C
Y	O	L	H	O
R	S	O	L	S

Plain text:PULLAREDDYMEMORIALSCHOOLS.

V Results and Synthesis report

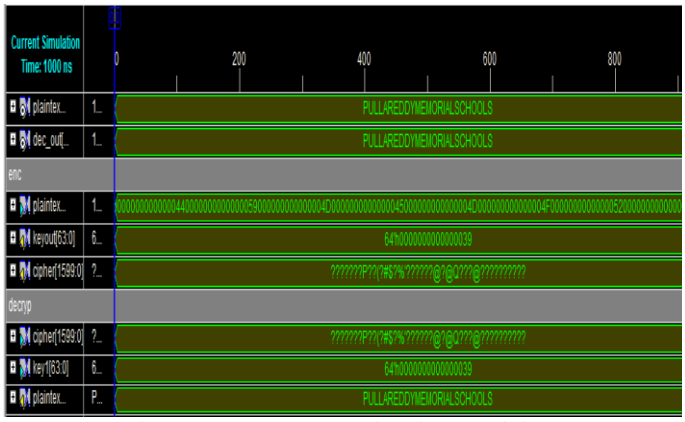

Figure 5.1 represents results of top module

The figure 5.1 top module represents the wave forms generated by the 1600 bit complete encryption and decryption processes. In encryption process, it uses 25 bit plain text as an input it will give key out size of 64 bit that is 64h0000000000000039 and cipher text outputs. In the figure cipher text is in ASCII form. In decryption process, encrypted output and key 1 will take as inputs and give plain text as output. In top module input is given as plain text and it will give output as decrypted output. Decrypted output is also called as plain text.

Synthesis report:

Synthesis report describes the time delay, and hardware used in this paper implementation.All the results are based on simulations from the Xilinx ISE simulator. All the individual transformation of both encryption and decryption are simulated using FPGA Spartan 3e families. There are 2 main routines implemented to realize this symmetric algorithm encryption and decryption.The objective of this paper design is to provide security of the data.
FPGA implementation is done using the Xilinx ISE 9.2 I design tool. In this project symmetric algorithm was developed using VHDL. The coding is tested both in simulation and synthesis. Simulation is done using Xilinx Isim simulator, for analyzing and for synthesis the Xilinx ISE tool to be used. FPGA implementation of this design has targeted to Xilinx XC3S500e-4-fg320.

Adders/ sub tractors : 79
64-bit adder : 54
64-bit sub tractors : 25
Comparators : 11050
64-bit comparator equal : 11050
Tristates : 100
64-bit tristate buffer : 100

Advanced HDL Synthesis Report:

Adders/Sub tractors: 79
64-bit adder: 54
64-bit sub tractors : 25
Comparators : 11050
64-bit comparator equal : 11050

Final report:

Design Statistics
IOs : 3200
Cell Usage:
BELS : 67470
GND : 1
INV : 2195
LUT1 : 51
LUT2 : 4644
LUT3 : 2066
LUT4 : 25020
MUXCY : 27176
MUXF5 : 1459
MUXF6 : 2
MUXF7 : 2
VCC : 1
XORCY : 4853
IO Buffers : 3200
IBUF : 1600
OBUFT : 1600

Device utilization summary:

Selected Device: 3s500efg320-5
Number of Slices: 17888 out of 4656 384% (*)
Number of 4 input LUTs: 33976 out of 9312 364% (*)
Number of IOs: 3200
Number of bonded IOBs :3200 out of 232 1379 %(*) (*)
More than 100% of Device resources are used.

Timing summary:

Speed Grade: -5
Minimum period: No path found
Minimum input arrival time before clock: No path found
Maximum output required time after clock: No path found
Maximum combinational path delay: 69.056ns

Timing Details:

All values displayed in nanoseconds (ns).
Timing constraint: Default path analysis
Total number of paths / destination ports: 74156396331151026000000 / 1600
Delay: 69.056ns (Levels of Logic = 229)
Total 69.056ns (46.591ns logic, 22.465ns (67.5% logic, 32.5% route)
CPU: 1928.58 / 1928.76 s | Elapsed: 1928.00 / 1928.00 s

ISSN: 2278 – 909X

Total memory usage is 1732148 kilobytes.

Comparisons:

Single point crossover using 4X4 matrix	Single point crossover using 5X5 matrix
It uses 1024 bits	It uses 1600bits
It provides security for less number of bits	It provides security for more number of bits
It generates less complexity cipher text	It generatescomplexity cipher text
Time delay 76.579ns	Time delay 69.056ns
Total memory usage is 542004 kilobytes.	Total memory usage is 1732148 kilobytes.
Number of IOs 2048	Number of IOs 2048
Number of slices 10726 out of 4656	Number of slices 17888 out of 4656
Number of 4 input LUTs 20383 out of 9312	Number of 4 input LUTs 33976 out of 9312
CPU utilization : 453.17/453.27 s | Elapsed : 453/453 s	CPU utilization : 1928.58 / 1928.76 s | Elapsed : 1928.00 / 1928.00 s

Table III shows comparisons of 4 by 4 and 5 by 5 matrices with single point crossover

VI Conclusion and Future scope

This section describes the conclusion and future scope of this modified cryptographic algorithm.

Conclusion: Single point crossover uses 5*5 matrix provides more security and it uses 1600 bits of data compared with single point 4*4 matrix. Time delay also reduces in 5*5 matrix compared with 4*4 matrix, so within the less time it provides security for more number of bits.

Future scope: we can design a cryptographic processor for encryption and decryption processes and also implemented for double point crossover.

Acknowledgment: This work is carried out on the basis of the paper entitled cryptographic technique by square matrix and single point crossover on binary field. We wish to offer our sincere gratitude and thanks to S. Som and M. Banerjee, for having motivated us to take up the problem of security for less number of bits that is 1024 bits that is fixed and it uses 4*4 square matrix and by applying single point crossover to produce final cipher text little bit complex.

References:

[1] S.Som, Banerjee,"cryptographic technique using substitution through circular path followed by genetic function" International conference paper, November 22, 2013.
[2] Applied cryptography by Bruce schiner.
[3] "An introduction to genetic algorithms by Metanie Mitchell
[4] Computer networking with internet protocols & techniques by William Stallings.
[5] Related work from S.Som, M.Banarjee "cryptographic technique by square matrix and single point crossover on binary field IEEE 2013.
[6] Ruangchaijatupon, P. Krishnamurthy, "Encryption and Power Consumption in Wireless LANs-N, ''The Third IEEE Workshop on Wireless LANs -September 27-28, 2001- Newton, Massachusetts.
[7] Hardjono, "Security In Wireless LANS And MANS, "Artech House Publishers 2005.
[8] W.Stallings, "Cryptography and Network Security 4thEd," Prentice Hall,PP. 58-309.
[9] Coppersmith, D. "The Data Encryption Standard (DES) Andit's Strength against Attacks.
[10] Bruce Schneier. The Blowfish Encryption Algorithm http://www.schneier.com/blowfish.html
[11] Daemen, J., and Rijmen, V. "Rijndael: The Advanced Encryption Standard."D r. Dobb's Journal, March2001, PP. 137-139.

SaralaBeeram has received her B.Tech. &M.Tech.(Digital Systems & Computer Electronics) from Jawaharlal Technological University, Hyderabad in 1993 and 1998 respectively. She is presently working as an Associate Professor in the Department of ECE in M V S R Engineering College, Hyderabad. Her areas of research include CDMA and Multi Carrier CDMA technologies & Wireless Communications. She has presented more than 10 papers in various national & international conferences. She has also published a paper in an international journal.

Sk. Myrunnisa Begum received her B.E degree in Electronics and Communication Engineering from Deccan college of Engineering and Technology. Currently she is a student of M.E in Embedded systems and VLSI design from MVSR Engineering College,Hyderabad.Areas of interesting in VLSI system design.

ISSN: 2278 – 909X