International Journal of Advanced Computer Research
ISSN (Print): 2249-7277 ISSN (Online): 2277-7970
Volume-5 Issue-19 June-2015

Design and Implementation of Enhanced version of MRC6 algorithm for data security

Nanda Hanamant Khanapur[1*] and Arun Patro[2]
PG student, VTU Extension Centre, UTL Technologies Limited, Bengaluru, India[1]
Lecturer, VTU Extension Centre, UTL Technologies Limited, Bengaluru, India[2]

Received: 24-May-2015; Revised: 20-June-2015; Accepted: 21-June-2015

Abstract

In today's world data transmission through channel requires more security, since there is chance of hacking the data while transmitting through a channel. Hence gaining security is very important and challenging. This paper introduces the Enhanced Modified version of RC6 (EMRC6) block cipher. The design is implemented to meet the requirement of the Advanced Encryption Standard (AES). New features of EMRC6 include the use of 32 working registers instead of two or four, and the inclusion of integer multiplication as an additional primitive operation. EMRC6 makes essential use of data-dependent rotations like RC6. The use of multiplication greatly increases the diffusion achieved per round, allowing greater security, fewer rounds, and increased throughput. The proposed design is described with verilog, HDL and synthesized using cadence tool.

Keywords

AES,Encryption,RC5,RC6,ERC6,MRC6 and EMRC6.

1. Introduction

Cryptography is an art of securing of data transmission by protecting the information. It transforms the data into an unreadable format in which a message can be hidden from reader and only the intended recipient will be able to convert it into original text. Its main goal is to keep the data secure from unauthorized access. Plain text is a data that can be read and understood without any measure.

*Author for correspondence

This work was supported in part by the Department of Digital Electronics at VTU Extension Centre, UTL Technologies Ltd, Bangalore, Karnataka, India.

The method of converting plaintext in such a way as to hide its substances is called encryption. Encrypting plaintext results in unreadable form called cipher text.

The process of reverting cipher text to its original plaintext is called decryption. System that provides encryption and decryption is called as cryptosystem. Cryptography provides number of security goals to ensure the privacy of data. Due to the great security advantages of cryptography it is widely used today [3] [4].

2. Literature survey

In 1994, the National Institute of Standards and Technology (NIST) initiated a process to specify a new symmetric key encryption algorithm capable of protecting sensitive data. Conventional RC algorithms explained in literature survey.

1. RC5 algorithm [1] [5] [6][7][8].
2. RC6 algorithm [2] [9] [10] [11].
3. ERC6 algorithm [12].
4. MRC6 algorithm [13].

The RC5 (represented by RC5 w/r/b (32/12/16)) cipher was developed by Ron Rivest. The cipher is block based and symmetric. The advantage of the RC5 cipher over other ciphers is its simplicity of implementation and its flexibility. The algorithm uses a series data dependent rotations heavily changes the data during encryption [5] [6] [7][8].

The decryption stage performs the inverse of the operations performed in the encryption stage to obtain the original data or plain text. Both the encryption and decryption stages use the expanded version of the key called as S array for their operations. The flexibility of the algorithm is due to the fact that the word length (W), key size (b) and the number of rounds(R), are variable [1] [8].

RC6 is an evolutionary improvement of RC5.Main drawback in RC5 is increasing number of rounds for security, it will consume more memory and speed is also decreases. Nominated value of RC6 is (32/20/16). RC6 include the use of four working registers instead of two, inclusion of integer Multiplication as additional primitive operation and fixed bit rotation. The use of multiplication greatly increases the diffusion achieved per round, allowing for greater security and fewer rounds [2] [9] [10] [11]. Increasing no 'of rounds for more security, it will consume more memory and speed is also decreases like RC5, RC6 makes essential use of data dependent rotations. Increasing no' of rounds for security, it will consume more memory and speed is also decreases.

RC6 algorithm supports the following operations [2]:

1. Integer addition.
2. Integer subtraction.
3. Integer multiplication.
4. Performs EXOR operation in bitwise.
5. Rotational left shift.
6. Rotational right shift.

Enhancement for RC6(ERC6) can be represented by ERC6(32/16/16).It uses 8 (32-bit) working registers and it acts on 256-bits input/output blocks.ERC6 encrypts at about 17.3 MB/sec making it about 1.7 times faster than RC6 [12].The use of multiplication greatly increases the diffusion achieved per round, allowing for greater increases the security. By increasing size of' b', key size will also increases so that memory requirement for the operation is more.
Modification of RC6 (MRC6) be represented by (32/16/16) that act on 512-bits input/output blocks [13].It uses 16 (32-bit) registers rather than four (128-bit) registers [13].This modification has the advantage of increasing the number of rotation per round, and using more bits of data to determine rotation amounts in each rounds. But Throughput is less as compared with EMRC6.

3. Module for network security

Figure1 shows the model for network security. At the transmitter side plaintext applied as input to the encryption algorithm and other input is secret key which is shared between sender and recipient .Encryption algorithm converts plaintext to cipher text and then cipher text is transmitted along the channel. At the recipient side, the received cipher text act as an input to decryption algorithm with symmetric secret key. Decryption algorithm converts cipher text to plaintext hence receiver will receive the actual information of that sender.

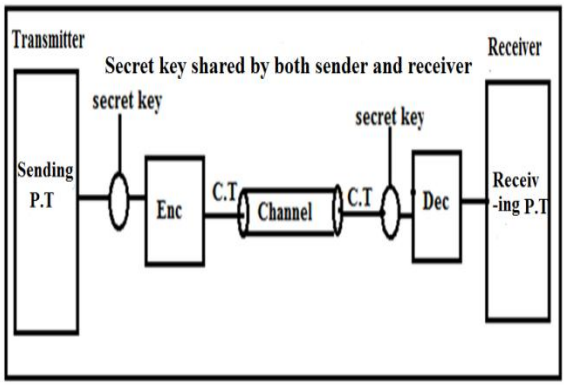

Figure 1: Model for network security

As shown in Figure2, cryptography can be classified based on type of operation used, based on key used and the way in which plaintext will process. Based on key used, the cryptography again can be classified as symmetric cipher and asymmetric cipher [4].

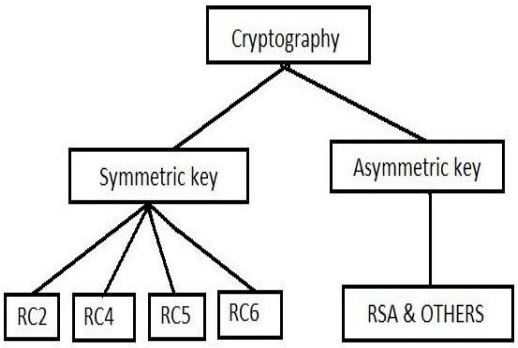

Figure 2: Classification of cryptography

Symmetric block cipher uses the same key for both encryption and decryption, but Asymmetric key uses different keys for encryption and decryption. As Shown in Figure2 RC2, RC4, RC5 and RC6 are symmetric block ciphers. RSA and others will come under asymmetric key [4].

4. EMRC6 algorithm

This section describes architecture of EMRC6 algorithm module. The module architecture of EMRC6 algorithm is shown in Figure3.

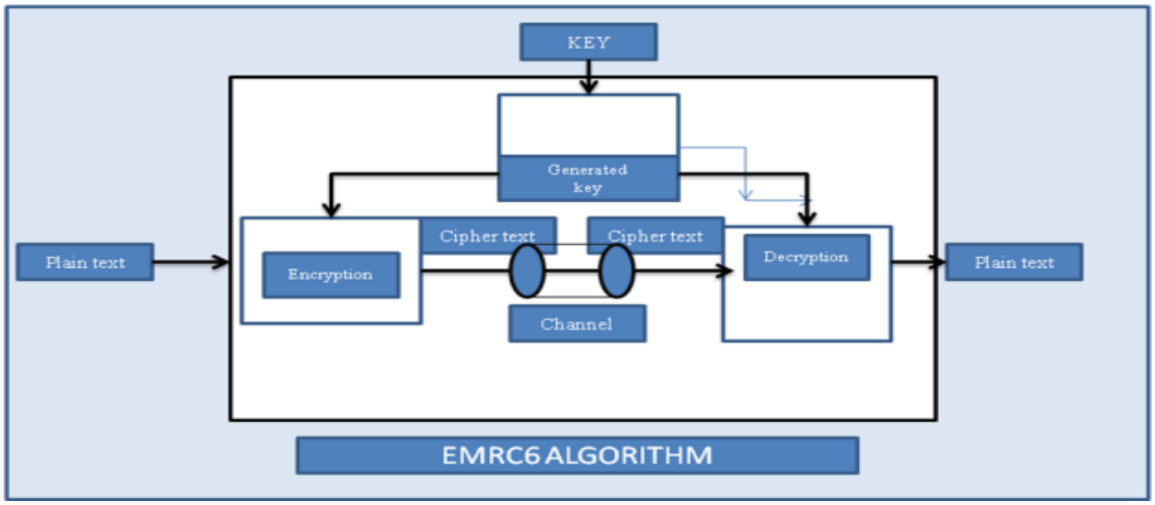

Figure 3: Block diagram of EMRC6

The complete system contains the following module section

1) Key generator
2) Encryption
3) Decryption
4) Transmitter
5) Receiver

Transmitter module will provide secret key for the key generation module and provide data for encryption module. In Key generation module, more words are derived from the user-defined key for the use for encryption and decryption. The user supplies a key of b bytes. The key is derived using two Magic constants P and Q. The value of P is B7E15163 and is derived from the binary expansion of e-2, where e is base of the natural logarithm function. The value of Q is 9E3779B9 and is derived from (phi)-1, where (phi) is the Golden ratio.

As shown in Figure4.generated S array values are generated key values, generated key values will use for both encryption and decryption operation.

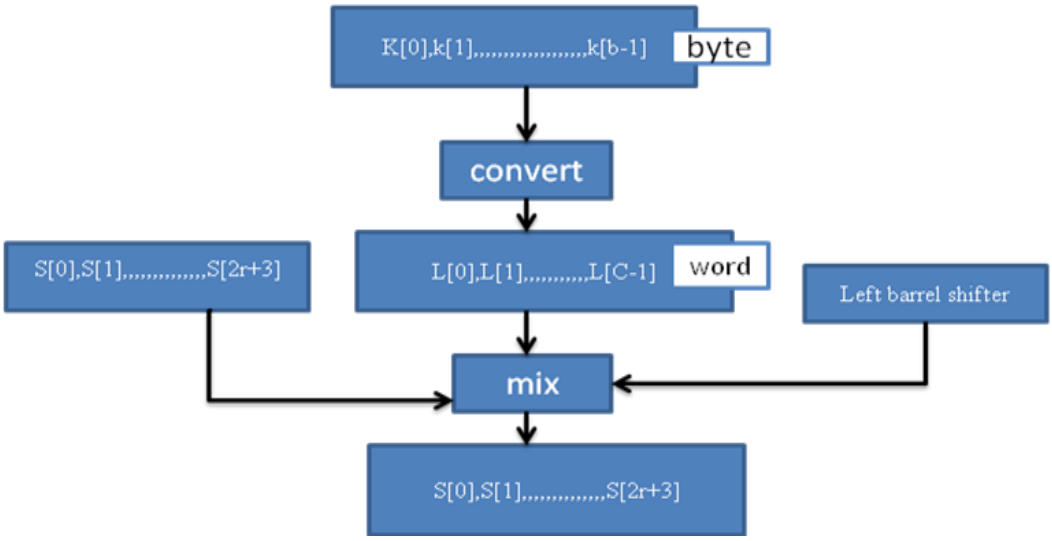

Figure 4: Key expansion module

Encryption module uses generated key for converting data into cipher text. The Cipher text is converted to plain text again using the 16r+31subkeys. Encryption algorithm for RC5,RC6, ERC6 and MRC6 is different because of bits used in input /output blocks are differ and different number of working registers are used for the operation. First byte of data is stored in LSB of R1 and last byte of data stored in MSB of R16 each of 32 bit. The 16r+31 sub keys are used to encrypt the plain text to give cipher text. The decryption process is converting cipher text into plain text which is done by using Decryption module. It is a just reverse of encryption process.

EMRC6 is same as RC6 but in this using 32 working registers instead of 4 working registers.EMRC6 algorithm is fully parameterized algorithm. It can be represented as EMRC6 w/r/b where 'w' is word size, 'r' is no' of rounds and 'b' secret key in a byte. Even it can be specify as EMRC6 w/r. It is easy to implement in both software and hardware, and even there is lot of speed variation between RC algorithms. According to AES submission w is varies from 16, 32 and 64,r is varies from 0-255 ,key length can be varied from 0-256 in bytes .

EMRC6-32/18/16 is the best version of RC6. It is more secure. It has more throughputs compared to RC5, RC6, ERC6 and MRC6. Minimum encryption /decryption time and maximum speed of operation can be achieved using this algorithm. Increasing number of rounds 'r' increases security but rotations requires more memory for storage, even some times it over kill the performance of the application.

For r=0, there is no security i.e. even RC6 algorithm can use in system where there is no requirement of security of the data.

For r=1, security is hack able by third person.

For b=0, no need of encryption operation.

EMRC6 has 18 rounds for more security and performance of the system. EMRC6-32/18/16 uses 32(32-bit) working registers and will increases diffusion.

"Diffusion" means that any change of bits in a plaintext will affect many bits in cipher text to enhance complexity between the plaintext and the cipher text. In a block encryption/decryption system, diffusion can be achieved by repeatedly implementing a specific permutation and then execute a functional operation.

Whitening is part of encryption and decryption: whitening is two type pre and post whitening. Without these steps, plaintext reveals part of input to first round of encryption. As well as in cipher text, cipher text reveals part of input to last round of encryption [2].

Rotation process: two types of rotation

1. Fixed rotation, where a 32-bit word is rotated to the left 5 times. This process is the same for encryption and decryption [2].
2. Data dependent rotation, where a 32-bit word is rotated to the left by the amount given by the least significant 5-bits of another 32-bit word, encryption. For the case of decryption the rotation is to the right [2].

4.1 Key expansion algorithm

Input: user supplied b byte key pre-loaded into the c-word, Array L[0,,,c-1], r number of rounds**.**
Output: w-bit round keys S[0,,,,16r+31].
Procedure:
S [0] = Pw
For i=1 to 16r+31 do
S[i] = S [i-1] +Qw
R1=R2=i=j=0
V=3*max{c, 16r+31}
For i=1 to 16r*31 do
{
R1=S[i] = (S[i] +R1+R2) <<<3
R2=L[j] = (L[j] +R1+R2) <<< (R1+R2)
i= (i+1) mod (16r+31)
j= (j+1) mod c
}

Figure 5: Key expansion of EMRC6

4.2 Encryption algorithm

The algorithm can be mathematically represented as follows:

Input: plaintext stored in 16 w-bit input register Ri (i=1 to 32). 'r' is number of rounds. W-bit round keys S[0 to 16r+31].
Output: cipher text stored in register Ri.
Procedure:
R2=R2+ S [0] , R4=R4+S[1],
R6=R6+ S [2], R8=R8+S [3],
R10=R10+ S [4], R12=R12+S [5],
R14=R14+ S [6], R16=R16+S [7],
R18=R18+S [8], R20=R20+S [9],
R22=R22+S [10], R24=R24+S [11],

R26=R26+S [12], R28=R28+S [13],
R30=R30+S [14], R32=R32+S [15],
For i=1 to r do
{
k1= (R2*(2R2+1)) <<<lg w
l1= (R4*(2R4+1)) <<<lg w
m1= (R6*(2R6+1)) <<<lg w
n1= (R8*(2R8+1)) <<<lg w
t1= (R10*(2R10+1)) <<<lg w
u1= (R12*(2R12+1)) <<<lg w
v1= (R14*(2R14+1)) <<<lg w
z1=(R16*(2R16+1))<<<lgw
k=(R18*(2R18+1))<<<lg w
l= (R20*(2R20+1)) <<<lg w
m= (R22*(2R22+1)) <<<lg w
n= (R24*(2R24+1)) <<<lg w
t= (R26*(2R26+1)) <<<lg w
u= (R28*(2R28+1)) <<<lg w
v= (R30*(2R30+1)) <<<lg w
z= (R32*(2R32+1)) <<<lg w
R1= ((R1 exor k1)) <<<l1) +S [16i]
R3= ((R3 exor 1l)) <<<k1) +S [16i+1]
R5= ((R5 exor m1)) <<<n1)+S [16i+2]
R7= ((R7 exor n1)) <<<m1)+S[16i+3]
R9= ((R9 exor t1))<<<u1)+S[16i+4]
R11= ((R11 exor u1)) <<<t1) +S [16i+5]
R13= ((R13 exor v1)) <<<z1) +S [16i+6]
R15= ((R15 exor z1)) <<<v1)+S [16i+7]
R17= ((R17 exor k)) <<<l) +S [16i+8]
R19= ((R19 exor l)) <<<k) +S [16i+9]
R21= ((R21 exor m)) <<<n) +S [16i+10]
R23= ((R23 exor n)) <<<m) +S [16i+11]
R25= ((R25 exor t)) <<<u) +S [16i+12]
R27= ((R27 exor u)) <<<t) +S [16i+13]
R29= ((R29 exor v)) <<<z) +S [16i+15]
R31= ((R31 exor z)) <<<v) +S [16i+16]
(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,R17,R18,R19,R20,R21,R22,R23,R24,R25,R26,R27,R28,R29R30,R31,R32)=(R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,R17,R18,R19,R20,R21,R22,R23,R24,R25,R26,R27,R28,R29 R30,R31,R32,R1)
}
R1=R1+S [16r+16], R3=R3+S[16r+17],
R5=R1+S [16r+18], R7=R3+S [16r+19],
R9=R1+S [16r+20], R11=R3+S [16r+21],
R13=R1+S [16r+22], R15=R3+S [16r+23],
R17=R1+S [16r+24], R19=R3+S [16r+25],
R21=R1+S [16r+26], R23=R3+S [16r+27],
R25=R1+S [16r+28], R27=R3+S [16r+29],
R29=R1+S [16r+30], R31=R3+S [16r+31].

Figure 6: Encryption of EMRC6

4.3 Key expansion algorithm

The algorithm can be mathematically represented as follows:

Input: cipher text stored in 16 w-bit input register Ri (i=1 to 32). 'r'is number of rounds. W-bit round keys S [0 to 16r+31].
Output: plaintext stored in register Ri.
Procedure:
R31=R31-S [16r+31], R29=R29-S[16r+30]
R27=R27-S [16r+29], R25=R25-S[16r+28]
R23=R23.-S [16r+27], R21=R21-S[16r+26]
R19=R19-S [16r+25], R17=R17-S[16r+24]
R15=R15.-S [16r+23], R13=R13-S[16r+22]
R11=R11-S [16r+21], R9=R9-S[16r+20]
R7=R7-S [16r+19], R5=R5-S[16r+18]
R3=R3-S [16r+17], R1=R1-S[16r+16]
For i=r down to 1 do
{
(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,R17,R18,R19,R20,R21,R22,R23,R24,R25,R26,R27,R28,R29R30,R31,R32)=(R32,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,R17,R18,R19,R20,R21,R22,R23,R24,R25,R26,R27,R28,R29 R30,R31)
z=(R32*(2R32+1))<<<lg w
v=(R30*(2R30+1))<<<lg w
u=(R28*(2R28+1))<<<lg w
t=(R26*(2R26+1))<<<lg w
n=(R24*(2R24+1))<<<lg w
m=(R22*(2R22+1))<<<lg w
l=(R20*(2R20+1))<<<lg w
k=(R18*(2R18+1))<<<lg w
z1=(R16*(2R16+1))<<<lg w
v1=(R14*(2R14+1))<<<lg w
u1= (R12*(2R12+1))<<<lg w
t1= (R10*(2R10+1))<<<lg w
n1=(R8*(2R8+1))<<<lg w
m1= (R6*(2R6+1))<<<lg w
l1= (R4*(2R4+1))<<<lg w
k1= (R2*(2R2+1))<<<lg w
R31=((R31- S[16i+15]))>>>z) exor v
R29=((R29- S[16i+14]))>>>v) exor z
R27=((R27- S[16i+13]))>>>u) exor t
R25=((R25- S[16i+12]))>>>t) exor u
R23=((R23- S[16i+11]))>>>n) exor m
R21=((R21- S[16i+10]))>>>m) exor n
R19=((R19- S[16i+9]))>>>l) exor k
R17=((R17- S[16i+8]))>>>k) exor l
R15=((R15- S[16i+7]))>>>z1) exor v1
R13=((R13- S[16i+6]))>>>v1) exor z1
R11=((R11- S[16i+5]))<<<u1) exor t1
R9=((R9- S[16i+4]))<<<t1) exor u1
R7=((R7- S[16i+3]))<<<n1) exor m1
R5=((R5- S[16i+2]))<<<m1) exor n1
R3=((R3- S[16i+1]))<<<l1) exor k1
R1=((R1- S[16i]))<<<k1) exor l1
}
R32=R32-S[15], R30=R30-S[14]
R28=R28-S[13], R26=R26-S[12]
R24=R24-S[11], R22=R22-S[10]
R20=R20-S[9], R18=R18-S[8]
R16=R16-S[7], R14=R14-S[6]

R12=R12-S[5],	R10=R10-S[4]
R8=R8-S[3],	R6=R6-S[2]
R4=R4-S[1],	R2=R2-S[0]

Figure 7: Decryption of EMRC6

5. Experimental results

Table 1: Comparison Chart

S.No	EMRC6 Algorithm (Rounds)	Clock cycles for encryption and decryption	Throughput
1	8	140ns	7342Mb/sec
2	12	204ns	5019Mb/sec
3	16	268ns	3820Mb/sec
4	18	300ns	3413Mb/sec
5	20	332ns	3084Mb/sec
6	24	396ns	2585Mb/sec

There is tradeoff between throughput and security. As number of round increases security will increase at the same time throughput decreases. Also increasing 'b' secret key length value contributes to increase in the security of the data.

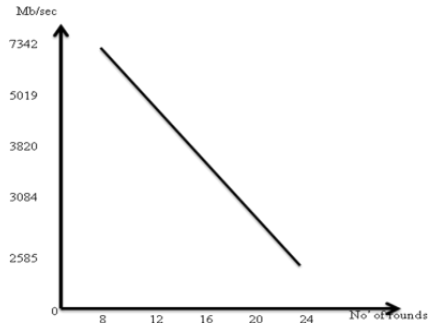

Figure 8: Throughput of EMRC6

Throughput is a rate of time at which data transmit from input to output, measured in terms of M bits per sec. Significant throughput value and security can be achieved with EMRC6 as compared to other block ciphers. Table1 shows comparative analysis with round, clock cycles and throughput

Table 2: Comparison Chart

S.No	Parameters	MRC6	EMRC6
1	Working resistors	16	32
2	Key size	0-255	0-255
3	Throughput	24.34MB/sec	34.13MB/sec
4	Plaintext size	512	1024
5	S-array size	8r+15	16r+31
6	Rotations/round	8	16
7	Security	More	More than MRC6
8	No' of rounds	16	18
9	Complexity	More	More than MRC6

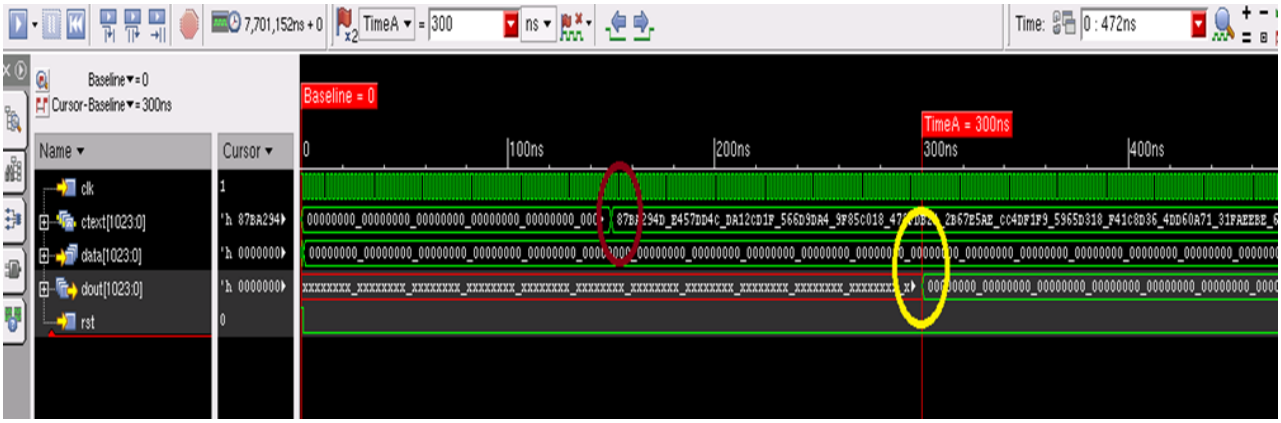

Figure 9: Simulation result of EMRC6

6. Simulation result

Figure9 shows the simulation result of EMRC6 algorithm, in which 16r+31 sub keys are used to encrypt the plain text to give cipher text.The Cipher text is converted to plain text again using the [16r +31] sub keys. The decryption process is just the reverse of encryption process. Table2 shows ccomparisons of RC algorithm with security, complexity and throughput values of algorithm.

7. Conclusion and Future Work

Enhanced Modified version of RC6 (EMRC6) 32/18/16 is the best version of RC6. It has more complexity, security and throughput. Throughput value of EMRC6 algorithm is 34.13MB/sec.

During transmission of data in wireless communication and long distance communication there are chances to hacking information by unauthenticated person, EMRC6 algorithm for increase security, performance and throughput for such communication applications.

Acknowledgment

I would like to take this opportunity to express my deep sense of gratitude to Dr.Venkataratanam, Principal, VTU Extension Centre, UTL Technologies Limited, for his invaluable inspiration and guidance without which the project work would not have progressed to its present state.

I cannot forget the constant encouragement and help provided by my guide Mr. Arun Patro, Lecturer, VTU Extension Centre, UTL Technologies Limited, who is considered me like a friend and made me feel at ease in times of difficulty during my project work. I express my sincere gratitude to him.

References

[1] Rivest, Ronald L. "The RC5 encryption algorithm." Fast Software Encryption. Springer Berlin Heidelberg, 1995.

[2] Shareef, Faez F., Ashwaq Talib Hashim, and Waleed F. Shareef. "Compact hardware implementation of FPGA based RC6 block cipher." J Eng Appl Sci 3 (2008): 598-601.

[3] Hura, Gurdeep S., and Mukesh Singhal. Data and computer communications: networking and internetworking. CRC Press, 2001.

[4] Gunasundari, T., and Dr K. Elangovan. "A Comparative Survey on Symmetric Key Encryption Algorithms." International Journal of Computer Science and Mobile Applications, ISSN (2014): 2321-8363.

[5] Yin, Y. L. "The RC5 encryption algorithm: Two years on." RSA Laboratories' Cryptobytes 2 (1997): 14-15.

[6] Tummalapalli, V. Chaitanya, and MD Khwaja Muinnuddin Chisti. "Implementation of Low Power RC5 Algorithm in XILINX FPGA." International Journal of Engineering Research and Applications (IJERA) 2.3 (2012): 924-928.

[7] Elkeelany, Omar, and Adegoke Olabisi. "Performance Comparisons, Design, and Implementation of RC5 Symmetric Encryption Core using Reconfigurable Hardware." Journal of Computers 3.3 (2008): 48-55.

[8] Raghunatha and Sowmya Sunkara, "ASIC Implementation of 128 Bit Key RC5 Cipher", International Journal of Advanced Computer Engineering and Communication Technology (IJACECT), Volume-2,Issue-2, 2013.

[9] Robshaw, M. J. B. "RC6 and the AES." January 2001Available at http://www. rsasecurity. com/rsalabs/rc6 (2001).

[10] Rivest, Ronald L., Matthew JB Robshaw, and Yiqun Lisa Yin. "RC6 as the AES." AES Candidate Conference. 2000.

[11] Harsh Kumar Verma, Ravindra Kumar Singh, "Performance Analysis of RC6, Twofish and Rijndael Block Cipher Algorithms" International Journal of Computer Applications. Volume-42, No.16, March 2012.

[12] Abdel Hamid M. Ragab, Nabil A. Ismail and Osama S. Faragallah, "Enhancement and Implementation of RC6 Block cipher for Data Security ", published in Electronic Engineering Bulletin, No. 21, January 2001.

[13] Nawal A. El-Fishawy I, Talat E.El-Danaf, and Osama M Abou Zaid, "A modification of RC6 block cipher algorithm for data security (MRC6)", International Conference on Electrical, Electronic and Computer Engineering, 2004, pp:222–226.

Nanda Hanamant Khanapur pursuing Master Degree (final year) in Digital Electronics branch at VTU Extension Centre, UTL Technologies Limited, Bangalore, Karnataka, India. Her area of interest is Verification, Cryptography, Embedded Hardware Designs and Digital design.
Email:nanda.khanapur@gmail.com

Arun Patro is finished his M.Tech in (VLSI) and currently he is a lecturer in VLSI Department at VTU Extension Centre, UTL Technologies Limited, Bangalore, Karnataka, India. His area of an interest is Verilog, VHDL, Digital design and Verification, Static timing analysis, and VLSI.