International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 2, Issue 5, May 2015)

Optimized hardware implementation of RC6 Algorithm on FPGA for Cloud Security

Smitha Rose Varghese
Electronics and communication
PACE, Mangalore, India
Smithu12155@gmail.com

Nafeesath T P
Associate Prof.
Electronics and communication
PACE, Mangalore, India
nafi.tp@gmail.com

Dr.Jose Alex Mathew
Director, ECE, PACE
Mangalore, India.
ayamanamkuzhy@gmail.com

***Abstract* - RC6 is one of the finalists for AES. It is considered to be in par with various other algorithms like Twofish, Rijndael, etc.,. In this research work, investigation is done to implement secure data transfer of 192 bits using RC6 algorithm and on improving the hardware performance of the RC6 algorithm by identifying various operations that are present and those which can be optimized further. It is implemented on FPGA so as to get the hardware proof and also, its performance on FPGA is also discussed.**

I. INTRODUCTION

In cryptography, RC6 is a symmetric key block cipher derived from RC5. It was designed by Ron Rivest, Matt Robshaw, Ray Sidney, and Yiqun Lisa Yin to meet the requirements of the Advanced Encryption Standard (AES) competition. The algorithm was one of the five finalists, and was also submitted to the NESSIE and CRYPTREC projects. It is a proprietary algorithm, patented by RSA Security.RC6 cipher and decipher unit has a block size of 128 bits and supports key sizes of 128, 192 and 256 bits, but, like RC5, it can be parameterized to support a wide variety of word-lengths, key sizes and number of rounds. RC6 is very similar to RC5 in structure, using data-dependent rotations, modular addition and XOR operations; in fact, RC6 could be viewed as interweaving two parallel RC5 encryption processes.

II. LITERATURE SURVEY

RC6 was designed by Rivest, Robshaw, Sidney, and Yin and submitted by RSA Laboratories as a candidate for the AES. Adhering closely to the philosophy that an algorithm should be simple so that it can be analyzed, RC6 was based around RC5 [13] which was published in December 1994. Since the time for security assessment of the AES was anticipated to be short, an early design decision was to build as closely as possible on RC5 and to try and re-use much of the security analysis and independent cryptanalysis that had already taken place over the intervening four years. The simplicity of RC5 made it an attractive object for research.

In [8], they say, during the design of RC6 our pragmatic aim was to satisfy as many goals as possible while keeping the cipher simple. Only by keeping a cipher simple can one achieve a well-understood level of security, good performance, and a versatility of design that makes the cipher highly adaptable to future demands. It also says the three most important attributes of the final AES are security, performance, and versatility. With RC6 we achieve all three goals. RC6 is so simple that the full details of the cipher can be recalled at will. Through simplicity we have developed a truly versatile cipher. We have also developed a cipher that offers exceptional performance, and gives the best all-round suitability in Java with all the implications this holds for future applications. Most importantly, though, existing analysis on RC6 is not only by far the most extensive of any of the finalists, it is also the most accurate and the most detailed.

RC6 offers excellent performance both in raw encryption speed and in the amount of memory required. Although the least time was spent on optimizing RC6 it still comes out as the fastest algorithm on almost all platforms. Code size and memory requirements. RC6 has exceptionally compact code and requires little additional working memory. Sometimes the good performance of Rijndael is attained by the use of look-up tables. In 'C' implementations, and hand-optimized assembly on these processors, RC6 generally out-performs Rijndael. At times the performance figures are roughly comparable, but the difference in performance can sometimes amount to a factor of two or more. When we look to future 64-bit architectures the situation becomes muddled. On some processors RC6 appears to be penalized, in this particular case due to how the multiplication operation is supported. On others, such as the SGI R12000, RC6 performs at up to a factor of two faster than Rijndael. Support for the 32-bit multiplication seems to most determine the relative performance of RC6 and Rijndael.

III. RESEARCH APPROACH

RC6 does use an extra multiplication operation not present in RC5 in order to make the rotation dependent on every bit in a word, and not just the least significant few bits. In this project, we propose to apply the Vedic multiplication technique, to improve the performance RC6 hardware implementation on FPGA and also study the performance parameters in various FPGA families like Spartan, Virtex, Cyclone, Stratix, ECP3, Kintex, etc.,. However, hardware

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 2, Issue 5, May 2015)

implementation will be done on Spartan 3A FPGA using its evaluation board.

Algorithm:
Encryption/Decryption with RC6-w/r/b
Input: Plaintext stored in four w-bit input registers A, B, C & D
r is the number of rounds
w-bit round keys S[0, ... , 2r + 3]
Output: Ciphertext stored in A, B, C, D

"Encryption Procedure:"
B = B + S[0]
D = D + S[1]
for i = 1 to r do
{
t = (B*(2B + 1)) <<< lg w
u = (D*(2D + 1)) <<< lg w
A = ((A ⊕ t) <<< u) + S[2i]
C = ((C ⊕ u) <<< t) + S[2i + 1]
(A, B, C, D) = (B, C, D, A)

}
A = A + S[2r + 2]
C = C + S[2r + 3]

"'Decryption Procedure:"'
C = C - S[2r + 3]
A = A - S[2r + 2]
for i = r downto 1 do
{
(A, B, C, D) = (D, A, B, C)
u = (D*(2D + 1)) <<< lg w
t = (B*(2B + 1)) <<< lg w
C = ((C - S[2i + 1]) >>> t) ⊕ u
A = ((A - S[2i]) >>> u) ⊕ t
}
D = D - S[1]
B = B - S[0]

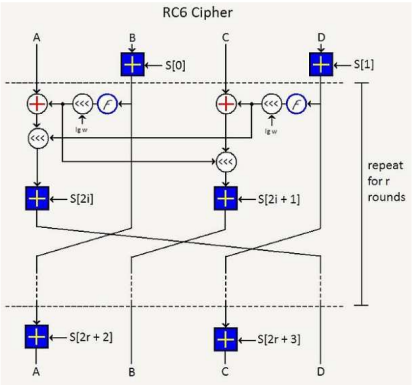

Fig 1: Basic RC6 Structure

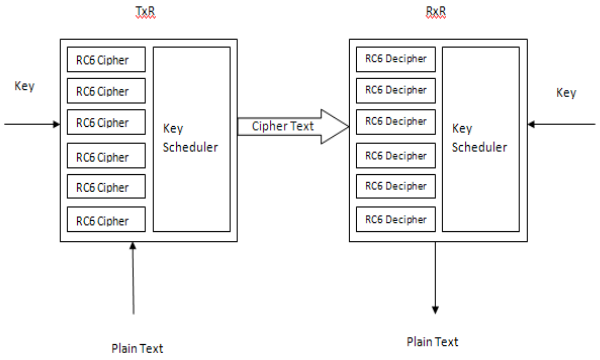

Fig 2: General setup of the RC6 system

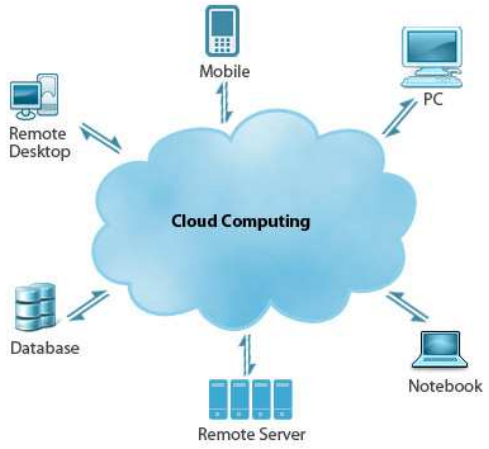

Fig 3: Typical links to Cloud which might require secure links

IV. EXPECTED RESULT

Experimental results should show reduction in the number of clock cycles required to perform the operation and also, reduction in the FPGA resources compared with conventional implementation.

V. PHASES OF EXECUTION

Phase 1	Literature survey, Scope definition, Design plan and methodology, Initial HDL Design and Testing
Phase 2	Completing HDL Design and Verification
Phase 3	Synthesis for FPGA, Synthesis for ASIC, analysis of results, improvement strategy development and execution
Phase 4	Dissertation

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 2, Issue 5, May 2015)

VI. CONCLUSION

It can be concluded that RC6 can substantially compensate for AES and other security algorithms. Merits of project are that it can be used in Network security systems, Data (Image, Sound, Text) cryptography. More extensive research can be done to improve it further.

REFERENCES

[1]. Creation of Secure Cloud Environment using RC6, Narendra Chandel, Sanjay Mishra, Neetesh Gupta, Amit Sinhal, 2013 International Conference on Intelligent Systems and Signal Processing (ISSP), 978-1-4799-0317-7/13/2013 IEEE.
[2]. Performance evaluation of scalable encryption algorithm for wireless sensor networks, Murat Çakırolu, Cüneyt Bayilmi, Ahmet Turan Özcerit and Özdemir Çetin, Scientific Research and Essays Vol. 5(9), pp. 856-861, 4 May, 2010, ISSN 1992-2248 © 2010 Academic Journals.
[3]. A new version of the RC6 algorithm, stronger against $\chi 2$ cryptanalysis,Routo Terada, Eduardo T. Ueda, Australasian Information Security Conference(AISC2009), Wellington, New Zealand, January 2009.
[4]. Enhanced Security Architecture for Cloud Data Security Dr. Chander Kant, Yogesh Sharma , Volume 3, Issue 5, May 2013, ISSN: 2277 128X, International Journal of Advanced Research in Computer Science and Software Engineering.
[5]. FPGA Implementations of the RC6 Block Cipher,Jean-Luc Beuchat, P.Y.K. Cheung et al. (Eds.): FPL 2003, LNCS 2778, pp. 101–110, 2003.c Springer-Verlag Berlin Heidelberg 2003.
[6]. Enhancing Cloud Computing Security using AES Algorithm, Abha Sachdev , Mohit Bhansali, International Journal of Computer Applications (0975 – 8887) Volume 67– No.9, April 2013 .
[7]. RC6 and the AES, M.J.B. Robshaw, 16d Stowe Rd, London, W12 8BN, UK., mrobshaw@supanet.com, January 9, 2001.
[8]. RC6 as the AES, Ronald L. Rivest1, M.J.B. Robshaw2, and Yiqun Lisa Yin3.
[9]. J. Daemen and V. Rijmen. AES Proposal: Rijndael. June 11, 1998.
[10]. R.L. Rivest. The RC5 encryption algorithm. In B. Preneel, editor, Fast Software Encryption, Lecture Notes in Computer Science Volume 1008, pages 86-96, Springer Verlag, 1995.
Available at theory.lcs.mit.edu/~rivest/
[11]. R.L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L. Yin. The RC6 Block Cipher. v1.1, August 20, 1998. Available at www.rsalabs.com/rc6/
[12]. R.L. Rivest, M.J.B. Robshaw, and Y.L. Yin. RC6 as the AES. Provided at 3rd AES conference, New York, April 2000. Available at www.rsalabs.com/rc6/
[13]. R.L. Rivest, M.J.B. Robshaw, and Y.L. Yin. The Case for RC6 as the AES. May 15, 2000. Available at www.rsalabs.com/rc6/