Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 48 (2015) 158 – 165

Procedia Computer Science

International Conference on Intelligent Computing, Communication & Convergence

(ICCC-2015)

Conference Organized by Interscience Institute of Management and Technology,

Bhubaneswar, Odisha, India

Secured Network on Chip (NoC) Architecture and Routing with Modified TACIT Cryptographic Technique

Adesh Kumar[a]*, Piyush Kuchhal[b] , Sonal Singhal[c]

[a.] *Department of Electronics, Instrumentation and Control Engineering, University of Petroleum & Energy Studies, Dehradun India*
[b.] *College of Engineering, University of Petroleum & Energy Studies, Dehradun India*
[c.] *Department of Electrical Engineering, Shiv Nadar University, NCR G.B Nagar India*

Abstract

Network on Chip (NoC) architecture needed secured data processing and routing in multicore system on Chip (SoC). Sometime it becomes very difficult to provide secured network routing for physically access network. The performance of NoC architecture depends on switching techniques, routing scheme and topological structure. The paper proposed the chip implementation of the new technique of securing data in NoC routers. Many algorithms have been anticipated already for secured NoC routing but limited to their key size and block size. In the paper, NoC architecture is integrated with modified TACIT security algorithm on Virtex-5 FPGA. The key generation scheme is considered based on Hash function and distributed under 4 Hash function (4H) scheme. The greatest advantage of TACIT security algorithm is that the block size and key size both can be of 'n' bit. The design is developed for 'n' bit with the help of VHDL programming language in Xilinx ISE 14.2 and Modelsim 10.1 b software and synthesized for 512 and 1024 bit of block size on Virtex-5 FPGA. The design is optimized with the help of device utilization summary, timing parameters, maximum frequency and memory support.

Keywords: Very Large Scale of Integration (VLSI), Field Programmable Gate Array (FPGA), Hardware Description Language (HDL), Network on Chip(NoC), System on Chip (SoC)

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: adeshmanav@gmail.com

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)
doi:10.1016/j.procs.2015.04.165

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)

1. Introduction

In Cryptography[5, 6], the original message or data is called plain text which is encoded with key, called cipher text and transmitted over a channel. The process is called encryption. The reverse process of encryption is decryption, in which the plain text is decoded from the cipher text. It takes secret key and cipher text and produces the original plain text. Cryptography involves encryption and decryption with the sharing of same key at both end or the different key on both ends, called symmetric and asymmetric encryption respectively. The model of symmetric key is shown fig.1 in which plaintext (A) is encrypted with key value (K) and transmitted cipher text is B = E [K, A], the same text is extracted with decryption algorithm[6] A = E [K, B], and same key (K).

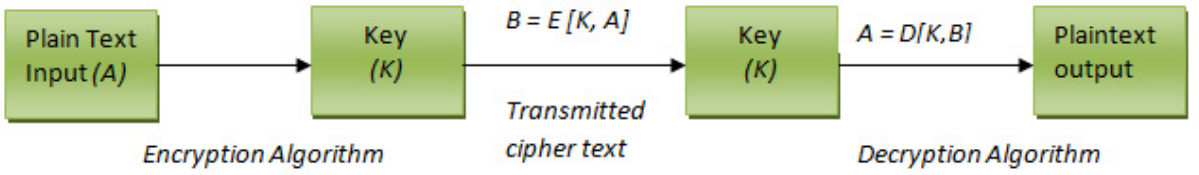

Fig.1 Model of symmetric encryption and decryption

The secret key and key size[6] is very important input to encryption algorithm. The algorithm encrypted output is changed on the key value. The algorithms can produce the different outputs based on specific key value and exact substitutions in the algorithms are dependent on key. The channel security is an issue especially in a Multiprocessor System on Chip (MPSoC) and Network on Chip (NoC).

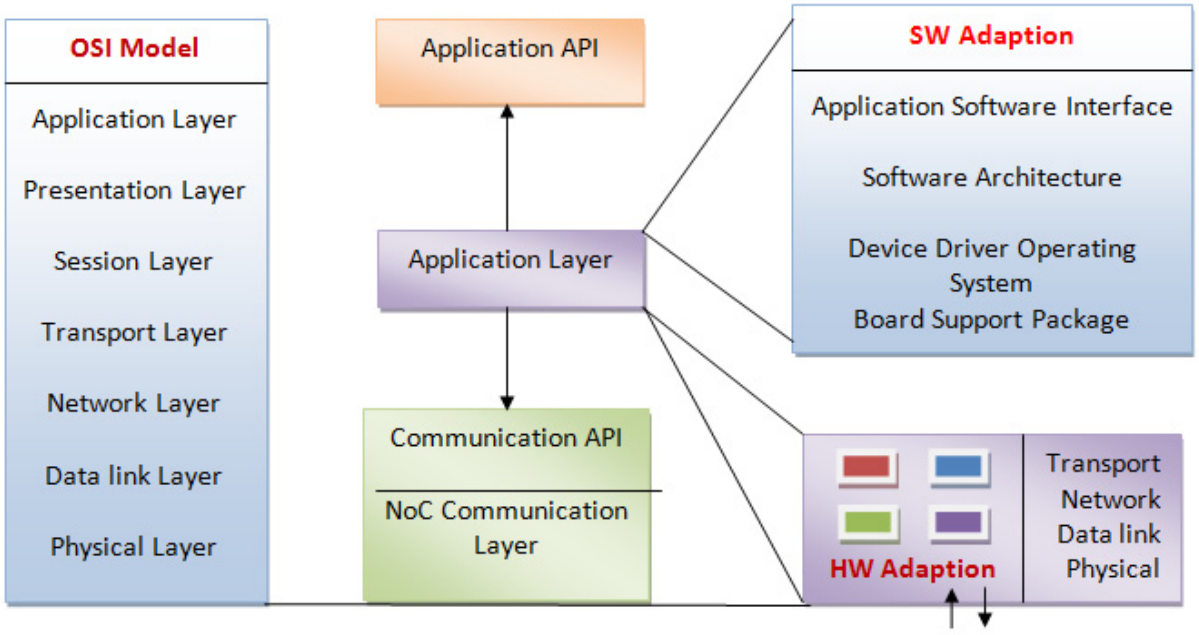

Fig.2 NoC Architecture

Network-on-Chip (NoC) [1, 2] is an approach for designing any network subsystem between IP cores[2] in a System on Chip (SoC)[2]. In NOC communication stack [2], the very critical aspect is software and application layer. A NOC template[4] consists of chip regions which are continuous areas of the chip. Regions can communicate with each other, and are isolated physically. The computing and other resources are embedded in the slots within the switches [2, 3] .A resource can be a microprocessor, memory, FPGA or an I/O resource. In OSI layer model resources are being implemented by the application layer. All resources are connected to networks which consist of network interface and switches [2]. The network interface provides networking services to a resource. Network Interface is being implemented by presentation, session and transport layers. Each network Interface is connected to a switch, which itself is connected to a number of other switches. Switches deliver packets from the source to destination. The switches and the metal wires that interconnect them represent the network, data link and physical layer. Fig. 2 shows the layered NOC structure [1, 4] for data transmission. A resource may have different representations for numbers e. g. floating point or fixed type; there must be some process to convert in the same type. This functionality is provided by the presentation layer. Connections between resources are established by session layer. Transport layer provides a mechanism which checks that no packets are lost in the lower layers Switches are implemented in the network layer. Network layer also deals with the network topology [2, 4]. Addressing scheme is closely related with the topology [1, 2] and is again dealt with network layer. Data link layer [1, 2] ideally passes data from one point to another. Physical layer deals with the electrical properties.

2. TACIT Encryption and decryption Algorithm

TACIT Encryption Algorithm: Fig. 3 shows the flow of the encryption algorithms
Step 1: First, read the text file and apply the concept of initial permutation approach to shuffle the position of each character with the help of key value [5].
Step 2: Read the character from the text file corresponding to the text and get the ASCII value of that character [5].
Step 3: specific n-bit key value is XORed with corresponding text [5].
Step 4: Apply TACIT Logic which is n^k xor k^k along with some specific operations [5].
Step 5: It is needed to convert the resulted value from step 4 into binary one [5].
Step 6: Perform reverse operation on resulted value from step 5 on the binary string [5].
Step 7: Find the corresponding decimal value [5].
Step 8: Formation of unicode character corresponds to the decimal value, which is nothing other than the cipher text [5].
Step 9: Continue all steps 1 to 7 for the next characters and complete until End of File (EoF) is achieved [5].

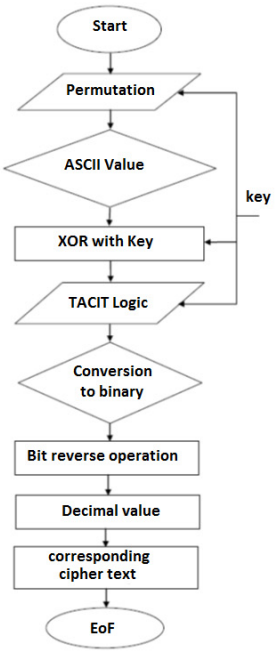

Fig.3 Encryption Algorithm

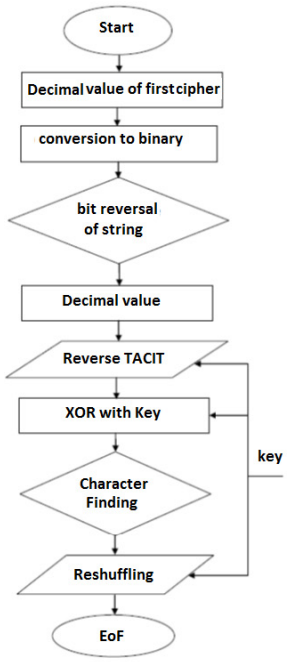

Fig. 4 Decryption Algorithm

TACIT Decryption Algorithm: Fig. 4 shows the flow of the decryption algorithms.
Step 1: Encode text in encryption algorithm is cipher text. Get the corresponding decimal value of cipher text, after reading the first character from the cipher text [5].
Step 2:.Evalute the corresponding binary value[5] and reverse it.
Step 3: Perform the inverse operation of the tacit logic[5].
Step 4: Perform XOR logical operation with next key value or n-bit key value[5].
Step 5: Determine the character corresponds to it [5].
Step 6: Now, reshuffling is needed with the help of key value [5].
Step 7: Repeat the steps (1 to 6) till EoF is achieved [2,5].

3. Key Managgment Policy

In symmetric algorithm, key generation [2,5] is a difficult task for the cryptographer. Sender and receiver both are having the same key value in symmetric encryption algorithm. The key generation can be understood with the help of hash function table 1. The diagram support to key sharing is shown in fig. 5. In the hash function[2,5] table a, b, c, d presents the no. of lower case alphabetic characters, no. of numerical characters, no. of upper case alphabetic characters and no. of special characters.

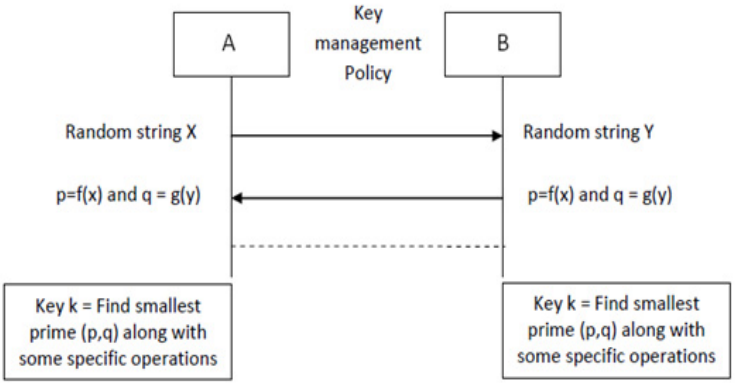

Fig 5 key distribution system[5]

Table 1 Hash Function for 4H key

a	4 H Hash Functions				Hash Function Procedure
	H_1	H_2	H_3	H_4	
	$a_g = a > (b, c, d)$	$b_g = b > (a, c, d)$	$c_g = c > (a, b, d)$	$d_g = d > (a, b, c)$	Generate the stream X and Y and exchange
0	a^b - a.b	b^c - b.c	c^d - c.d	d^a - d.a	*Case 1*: ($X = Y = a_g$) Get 'a' : p= H_1 and q = H_1
1	$a^c + (a + c)$	$b^d + (b + d)$	$c^a + (c + a)$	$d^b + (d + b)$	Where, $a_g = a > (b, c, d)$
2	$a^d – (c + d)$	$b^a – (d + a)$	$c^b – (a + b)$	$d^c – (b + c)$	
3	$b^c + (d. a)$	$c^d + (a. b)$	$d^a + (b. c)$	$a^b + (c. d)$	*Case 2*: ($X = Y = b_g$) Get 'a' : p= H_2 and q = H_2
4	$b^d + (b. a)$	$c^a + (c. b)$	$d^b + (d. c)$	$a^c + (a. d)$	Where, $b_g = b > (a, c, d)$
5	b^a - a	c^b - b	d^c - c	a^d - d	
6	c^a - a	d^b - b	a^c - c	b^d - d	*Case 3:* ($X = Y = c_g$) Get 'a' : p= H_3 and q = H_3
7	$c^b + (b + a - c)$	$d^c + (c + b - d)$	$a^d + (d + c - a)$	$b^a + (a + d - b)$	Where, $c_g = c > (a, b, d)$
8	$c^d + (b+ a+ d- c)$	$d^a + (c + b +a- d)$	$a^b + (d +c+b - a)$	$b^c + (a+d+c - b)$	
9	a.b.d + (a.c)	b.c.a + (b.d)	c.d.b + (c.a)	d.a.c + (d.b)	*Case 4:* ($X = Y = dg$) Get 'a' : p= H_4 and q = H_4
					Where, $d_g = d > (a, b, c)$

Let random string X is at transmitting end and random string Y is at receiving end. Both exchange the string to familiar with each other and the value of p and q is calculated based on hash table. Now generate a random number at sender's end within a specified range say 0 to 9 and add this number in a code sequence which signifies a specific hash. There exist four cases [2] to break the key. (i) $a_g = a > (b, c, d)$ (ii) $b_g = b > (a, c, d)$ (iii) $c_g = c > (a, b, d)$ (iv) $d_g = d > (a, b, c)$. The algorithm with possible hash functions are listed in table 1. The random sequence has more number of lowercase is alphabetic characters is denoted by a_g. The random sequence is followed at both the ends X string and Y string respectively. After exchange between X and Y, the first value denotes the value of 'a' in the random generated sequence. Based on the value of 'a', the value of p and q is calculated and key value is generated with the least prime number at both the ends with trail solution method. The actual key value is the average of least and lager prime number between 'p' and 'q'. The generated key is used to encrypt and decrypt the data.

Table 2 Example of key generation

Example:	
Let, stream X = ade3010SH#@{}%	stream Y = upes203IND#$^*t5
Here (a = 3, b = 4, c = 2, d = 5)	Here (a = 5, b = 4, c = 3, d = 4)
X supports condition dg = d > (a, b, c) and Hash function H4. Therefore, p = da - d.a = (5)3 – (5.3) = 125 -15 =110	X supports condition ag = a > (b, c, d) and Hash function H1. Therefore, q = ab - a.b = (5)4 – (5.4) = 625 -20 = 605
Key Value: Lowest prime number between 110 and 605 is 111 and largest prime number between 110 and 605 is 601. Then the key value will be the average of Lowest and Largest prime numbers. Key = (Lowest prime + Largest prime) = (111 + 601)/2 = 712/2 = 306	

4. Results & Discussion

The design is developed in VHDL. The method is used for VHDL implantation is finite state machine and behaviour style of modeling in VHDL. The Register Transfer Level (RTL) view of the developed chip is shown in fig. 6 and the description of the chip is explained in table 1.

Step input 1: reset = '1', clk is used for the synchronization and then run. The clock pulse is applied with rising edge, to check the results at 50% duty cycle.

Step input 2: reset = '0', same clk is used for synchronization. Select the input text, Model_selection input to select encryption and decryption mode, enable input to enable the particular logic. The description of the pins in listed in table 1

Mode_selection is forced to function the chip in two modes. If mode_slection = '1', data encryption logic is there and mode_selection = '0' decryption logic is there. It is used to differentiate the encryption and decryption logic from the integrated chip.

There is the need to enable the logic for encryption and decryption also. If enable = '1', it is the encryption logic, for decryption algorithm enable = '0' which disable the encryption logic. Force the mode-selection and enable with input_text <n bit>

Table 3 Pin description of RTL view of encryption and decryption logic

Pins	Functional Description
Reset	Used to reset sender and synchronized with clock of std_logic (1 bit)
Clk	Default input for sequential logic, rising edge of clock pulse of std_logic (1 bit)
input_text [N-1:0]	Input text of the encryption end it can be of 'N' bit. It is of std_logic_vector type
Decryption_text[N-1 : 0]	Decrypted text at receiving end, it is also of 'n'bit and of std_logic_vector type
Mode_Selection	1 bit input(std_logic) to select in a particular mode if mode_selection = '1' it is in encryption mode and mode_selection = '0', it is in decryption mode
Enable	1 bit input (std_logic) enable and disable the encryption logic. If enable = '1' encryption algorithms else decryption logic
Cipher_text [N-1 : 0]	Cipher text is the text which is encrypted with key at the transmitting end. It can be any garbage value and it is of std_logic_vector type

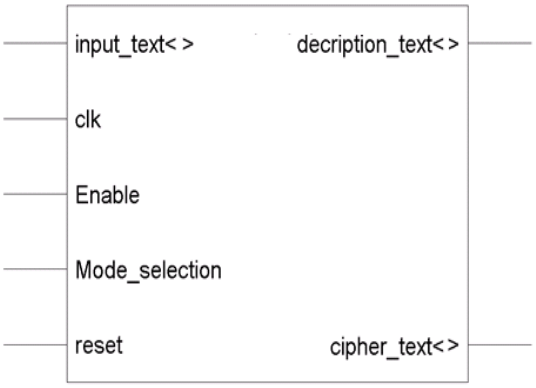

Fig. 6 RTL view of developed chip

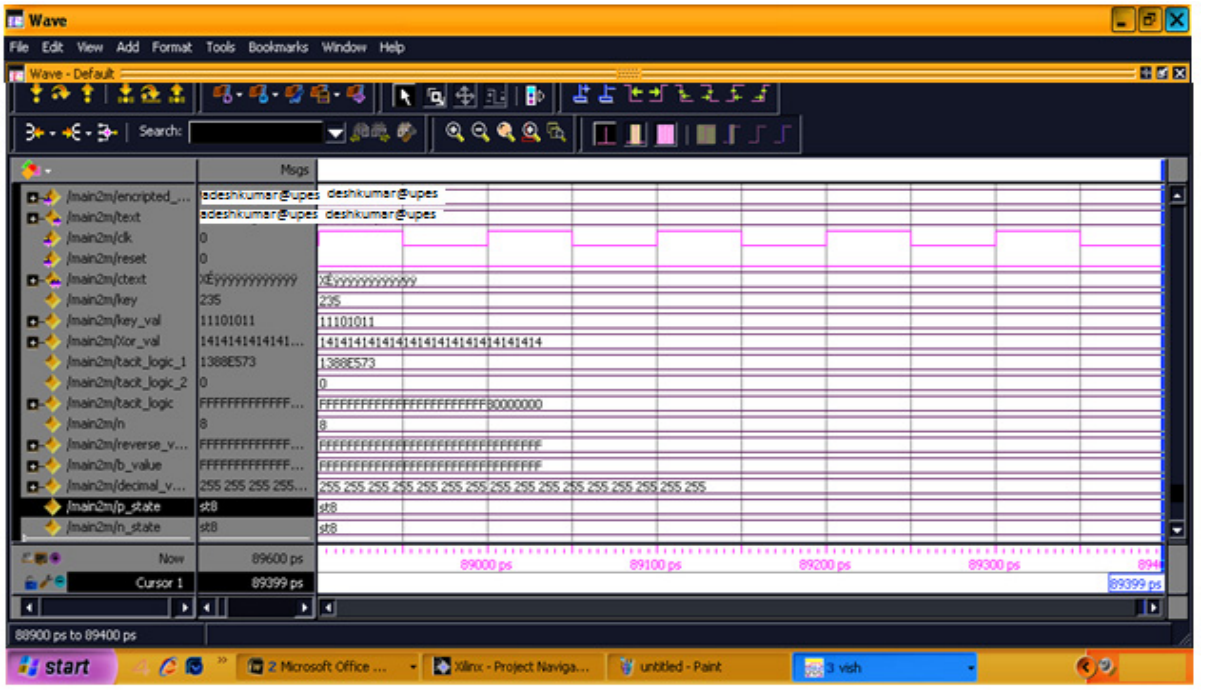

Fig. 7 Modelsim simulation of encryption and decryption

Device utilization and timing analysis

Device utilization report gives the percentage utilization of device hardware for the chip implementation. Device hardware includes No of slices, No of flip flops, No of input LUTs, No. of bounded IOBs and No of gated clocks (GCLKs) used in the implementation of design. Timing details provides the information of delay, minimum period value, maximum frequency value, minimum input arrival time before clock and maximum output required time after clock. Total memory utilization value required to complete the design. The target device is: xc5vlx20t-2-ff323 synthesized with Virtex-5 FPGA. Table 4 and Table 5 list the simulated values of the design.

Table 4 Device utilization summary

Device Part	**Encryption**		**Decryption**	
Block Size	**512 bit**	**1024 bit**	**512 bit**	**1024 bit**
Block size	125 out of 12480 1 %	245 out of 12480 2 %	121 out of 12480 1 %	224 out of 12480 2 %
Number of Slices	100 out of 12480 1%	198 out of 12480 2 %	97 out of 12480 1%	189 out of 12480 2 %
Number of Slice Flip Flops	3 out of 9 33%	3 out of 9 33%	3 out of 9 33%	3 out of 9 33%
Number of 4 input LUTs	16 out of 172 9 %	18 out of 172 10 %	16 out of 172 9 %	18 out of 172 10 %
Number of bonded IOBs	1 out of 32 3%	1 out of 32 3%	1 out of 32 3%	1 out of 32 3%

Table 5 Timing Parameters

Parameters	**Encryption**		**Decryption**	
Block Size	**512 bit**	**1024 bit**	**512 bit**	**1024 bit**
Minimum Period	1.098 ns	1.578 ns	0.918 ns	1.437 ns

Maximum Frequency	750 MHz	789 MHz	715 MHz	756 MHz
Minimum input arrival time before clock	2.732 ns	2.837 ns	2.117 ns	2.481 ns
Maximum output required time after clock	5.826 ns	5.912 ns	4.951 ns	5.124 ns
Total memory usage	102764 kB	115569 kB	92764 kB	101265 kB

In comparison to the existing work we have achieved the optimized results. Ref[5] proposed the future work as chip development of TACIT cryptographic logic. We have developed and synthesized the chip for the logic. In ref[2] the developed design was synthesized for 128 bit of block size, in our work the synthesis is carried out for 512 and 1024 bit of block size with maximum support frequency of 789 MHz in encryption.

5. Conclusions

The TACIT algorithm is simulated for the 'n' bit block size and key value. The results are synthesized on Virtex-5 FPGA for 512 bit and 1024 bit of block size successfully. The proposed TACIT algorithm has proven good results and simulated data is tested for different test cases. The greatest advantage of the proposed algorithm is that it will have key size and block size of 'n' bit. NoC security concern can be resolved using TACIT hardware FPGA chip integration embedded with NoC router. In future, the work can be done with the security with compression of data packets.

References

[1] Andreas Hansson, Kees Goossens and Andrei Radulescu "A Unified Approach to Mapping and Routing on a Network-on-Chip for Both Best-Effort and Guaranteed Service Traffic" *Hindawi Publishing Corporation VLSI Design* Volume 2007, pp (1-16).

[2] Arun Ambashenker, Planisamy Nirmal Kumar "Modified TACIT algorithm based on 4H Key distribution for secure routing in NoC architecture" *IEICE Electronic Express*, Vol 1, No. 13, 2014 pp(1-8)

[3]David Atienzaa, Federico Angiolini, Srinivasan Murali,A ntonioPullini dLucaBeninic, Giovanni De Michelia, "Network-on-Chip design and synthesis outlook" *Integration The VLSI Journal Elsevier*, Vol. 41 , pp(340-359), 2008.

[4] Ganghee Lee, Kiyoung Choi, and Nikil D. Dutt, "Mapping Multi-Domain Applications onto Coarse-Grained Reconfigurable Architectures" IEEE *Transaction on Computer Aided Design of Integrated Circuits and Systems*, Vol. 30, No. 5, pp (637-650) , May 2011.

[5]Prosanta Gope, Ashwani Sharma Ajit Singh Nikhil Pahwa "An Efficient Cryptographic Approach for Secure Policy Based Routing (TACIT Encryption Technique)", *Conference Proceedings, IEEE Xplorer,* (2011), pp (359-363)

[6] Xinmiao Zhang, and Keshab K. Parhi *"*High-Speed VLSI Architectures for the AES Algorithm*", IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* Vol. 12, No. 9, Sep. 2004, pp (957-968)