Cyber security in Cloud system using VHDL based ASIC design of superscalar Homomorphic Encryption

[1]Chandra kant Maurya, [2]Abhishek Singh

[1]M.Tech Scholar, [2]Assistant Professor
Department of Electronics and Communication,
Gyan Ganga Institute of Technology and Science, Jabalpur, MP

Abstract: **Cyber Network Security and cryptography in high speed Cyber Networks demands for specific hardware in order to match up with fast cloud systems. In cryptographic module, Dual key Homomorphic, a symmetric key block cipher is been designed as algorithm for implementation. design goal is to increase data encryption rate i.e. throughput to a substantial value so that design may be used as a cryptographic processor in high speed Cyber Network applications. $2^{\wedge n}+1$ modulo multiplier is a main element in Homomorphic encryption of cloud system, hence Paper work presented a new $2^{\wedge n}+1$ modulo multiplier in design which generates less number of partial products (≤ n 2) and less area at very high speed. Multiplication is based on Wallace tree along with specialized shifting. Presented work is mainly based on designing an optimized architecture for a cryptographic block and a Cyber Network intrusion detection system for a very high speed Cyber Network. All designs are coded using HDL language VHDL and are synthesized using Xilinx VIVADO 18.2i for verifying its functionality and working. Zynq-7000 pro FPGA is chosen as target platform device for realization of presented design.**

Index Terms: Cloud System, FPGA, Homomorphic Encryption, VHDL, Cyber Security, Modulo Multiplier

I. INTRODUCTION

Module multiplication is a essential Need of cloud security based Homomorphic Encryption. As per earlier work done on modulo $(2^N + 1)$ multiplier for Dual key Homomorphic cipher system, there was no enough space for very high speed operation or area reduction up to mark. Also previous work discussed that there were no clear theory for power optimization which is important area in current situation. problem areas for my research work are hardware implementation of cipher system on FPGA to reduce number of logic gates and provide faster efficiency. The main objective of presented work is implementation and optimization of FPGA based of unique new Modulo Multiplier $(2^N + 1)$ Design for High Speed Secured Cyber Network system by application of symmetric dual key dual key cryptography. Multiplier is most time and space consuming operation any computation. research work is implementation with help of Xilinx VIVADO software FPGA Zynq-7000.

Table 1 Literature Summary

Author	Brief	Journal	Results
Sujoy Sinha Roy et al [1]	They design an new Parallel Architecture for Homomorphic Encryption for could based secure data storage. they implemented the design on Zynq-7 FPGA and used Xilinx Vivado for design and simulations.	IEEE proceeding in 2019	0.362 Second for Send cipher text and Receive result cipher text in 0.18 sec. 200 MHz speed of encryption achieved. 63522 LUT's of Zynq-7 FPGA for Encryption engine
Zhe Liu et al [2]	They design a Ring-LWE Encryption engine on IoT ARM-NEON processors and used high level language for design Efficient Software system.	IEEE transactions 2017	Encryption cycles require are 149,400 with 32-bit ARM-NEON processors
Yang Su et al [3]	They design Ring-LWE Fully Homomorphic Encryption for cloud system secure data storage and fast access. they use FPGA-based Hardware Accelerator and design a Leveled encryption module.	IEEE Access 2020	95854 LUT's of Virtex UltraScale FPGA with 150MHz

II. PROPOSED METHODOLOGY

The paper work is an new approach in encryption area for cloud system secure storage, motivation behind work is that Encryption and decryption is an very important requirement now a day's specially in cloud computing ware client data need to be secure at any situation and it must be done with using some highly secure data security mechanism, as in proposed work a modified Homomorphic encryption is used. however it is not compulsory requirement for data communication it is just a important need,

work done in area till now is itself an achievement and very robust, however it is also an overhead for system and hardware and time necessary for encryption and decryption is just a overhead for system, presented work is an highly secure encryption techniques for data communication with less amount of hardware and less time, presented work is encryption technique which is less complicated and uses presented new unique transform encoding.

Cryptography is technique of keeping communication secure, so that hikers cannot decipher transmitted messages. Transmission speeds of core Cyber Networks require hardware based cryptographic modules, since software-based dual key cryptography cannot meet required throughput requirements. Field programmable gate arrays (FPGAs) are ideal components for fast cryptographic algorithms. Large capacities of FPGAs enable fitting of fully pipelined algorithms on a single chip. Reprogram ability of FPGAs enables using same hardware platform as a cryptographic engine for a multitude of communications protocols. Cryptographic algorithms are divided into public-key and secret-key algorithms. In public-key algorithms both public and private keys are used, with private key computed from public key. Secret-key algorithms rely on secure distribution and management of session key, which is used for encrypting and decrypting all messages. When it comes to both software- and hardware-based implementations, secret-key algorithms are 100 to 1000 times faster than public key algorithms. For this reason, dual-key sessions use a secret-key algorithm for bulk of communication, whereas session specific secret keys are agreed on and distributed with a public key algorithm.

Homomorphic Symmetric Key Encryption

Coding with various combinations of eight rounds is been done at gate level i.e. fully dataflow modeling style for high throughput. New modulo multiplication is been presented in which multiple patterns may be done with less area. string matching module is coded and functionally verified using VHDL language targeting Zynq-7 pro FPGA and performance measures in terms of speed and resource utilization. Our work is mainly based on designing an efficient architecture (IP) for a cryptographic module for secure data trafficking and a Cyber Network intrusion detection system for a high speed Cyber Network. complete designs are coded using VHDL language and are verified using Xilinx-VIVADO simulator for verifying its functionality.

The current era has seen an explosive growth in communications. Applications such as online banking, personal digital assistants, mobile communication, smartcards etc. have emphasized need to security in resource constrained environments. International Data Encryption Algorithm (IDEA) [6] cryptography serves as a perfect Cyber Network intrusion detection system (NIDS) tool due to its 128 bits key sizes and high security comparable to that to other algorithms. However, to match ever increasing requirement to speed in today's applications, hardware acceleration to cryptographic algorithms is a necessity. As a further challenge, designs have to be robust against side channel attacks. Our work is mainly based on designing an efficient architecture to a Cyber Network intrusion detection system to a high speed Cyber Network. Homomorphic is a symmetric, secret-key block cipher. Keys to both encryption and decryption must be kept secret from unauthorized persons. Since two keys are symmetric, one may divide decryption key from encryption one or vice versa.

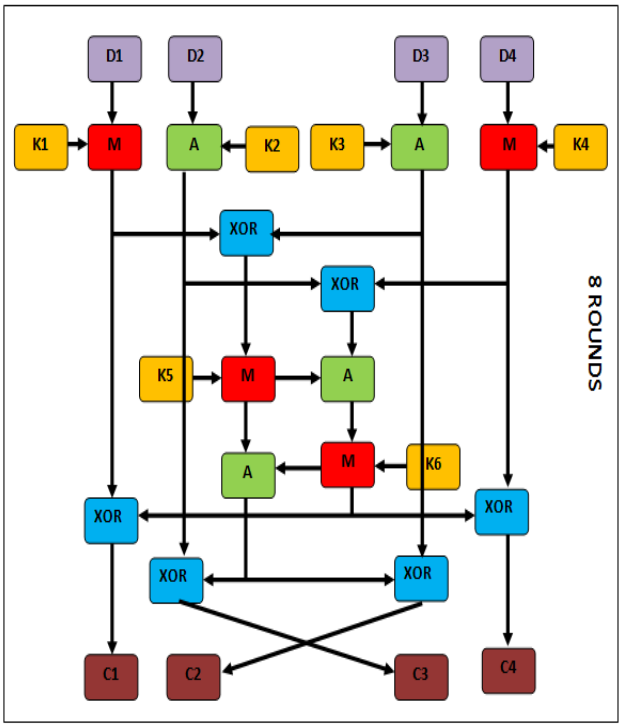

Figure 2 Homomorphic Symmetric Key Encryption in Cloud storage

The size to key is fixed to be 128 bits and size to data block which may be handled in one encryption or decryption procedure is fixed to 64 bits. All data operations in Homomorphic cipher are in 16-bit unsigned integers. When processing data which is not an integer multiple to 64-bit block, padding is required. Security to Homomorphic algorithm [14] is based on mixing to three various kinds to algebraic operations: EX-OR, addition and modular multiplication. Homomorphic is based upon a basic function, which is iterated eight times. First iteration operates on input 64-bit plain text block and successive iterations operate on 64-bit block from previous iteration. After last iteration, a final transform step produces 64-bit cipher block. Algorithm structure has been chosen such that, with exception that various key sub-blocks are used, encryption procedure is identical to decryption process. Homomorphic

uses both confusion and diffusion to encrypt data. Three algebraic groups, EX-OR, addition modulo 2^{16} and multiplication modulo $(2^{16} + 1)$ are mixed and they are all easily implemented in both hardware and software. All these operations operate on 16-bit sub-blocks. Figure 2 above shows working flow of Homomorphic cipher encryption here M is modulo multiplier, 'A' is modulo adder K1-K6 are 16 bit part to Key. In each round to 8 rounds to algorithm, following sequences to events [7] are performed:

1. modulo Multiply D1 and K1
2. Modulo Add D2 and K2
3. Modulo Add D3 and K3
4. modulo Multiply D4 and K4
5. XOR results to step 1 and step 3
6. XOR results to step 2 and step 4
7. Modulo Multiply results to step 5 with K5
8. Modulo Add results to step 6 and step 7
9. Modulo Multiply results to step 8 with K6
10. Modulo Add results to step 7 and step 9
11. XOR results to step 1 and step 9
12. XOR results to step 3 and step 9
13. XOR results to step 2 and step 10
14. XOR results to step 4 and step 10

key Generation

Original Key = K8 K7 K6 K5 K4 K3 K2 K1
Rotate left by 25 bit = K16 K15 K14 K13 K12 K11 K10 K9
Rotate left by 25 bit = K24 K23 K22 K21 K20 K19 K18 K17
Rotate left by 25 bit = K32 K31 K30 K29 K28 K27 K26 K25
Rotate left by 25 bit = K40 K39 K38 K37 K36 K35 K34 K33
Rotate left by 25 bit = K48 K47 K46 K45 K44 K43 K42 K41
Rotate left by 25 bit = K56 K55 K54 K53 K52 K51 K50 K49
K1-K6 to round two
K7-K12 to round three
K13-K18 to round four
K19-K24 to round five
K25-K30 to round six
K31-K36 to round seven
K37-K42 to round eight
K43-K48 to round nine
K49-K52 to round ten

Each to eight complete rounds necessary six sub keys and final transformation "half round" necessary four sub keys. So entire procedure necessary 52 sub keys. 128-bit key is split into eight 16-bit sub keys. Then bits are shifted to left 25 bits. Resulting 128-bit string is split into eight 16-bit blocks that become next eight sub keys. Shifting and splitting procedure is repeated until 52 sub keys are generated. Shifts to 25 bits ensure that repetition does not occur in sub keys. Six sub keys are used in each to 8 rounds. Final 4 sub keys are used in ninth "half round" final transformation. Six 16-bit key sub-blocks from 128-bit key. Since a further four 16-bit key-sub-blocks are required to subsequent output transformation, a total to 52 (= 8 x 6 + 4). First, 128-bit key is partitioned into eight 16-bit sub-blocks which are then directly used as first eight key sub-blocks. The 128-bit key is then cyclically shifted to left by 25 positions, after which resulting 128-bit block is again partitioned into eight 16-bit sub-blocks to be directly used as next eight key sub-blocks. The cyclic shift procedure described above is repeated until all to require 52 16-bit key sub-blocks have been generated.

Modulo 2^n Adders

Modulo 2^N adders [6] are important for several applications including residue number system, digital signal processors and dual key cryptography algorithms. Modular characteristic of Residue Number System (RNS) offers potential for high-speed and parallel arithmetic. In RNS logic, each operand is represented by its residues with respect to a set of numbers comprising base. Addition, subtraction and multiplication are performed in parallel on residues in distinct design units (often called channels) avoiding carry propagation among residues. So, arithmetic operations, e.g. addition, subtraction and multiplication may be carried out lot efficiently in RNS than in conventional two's complement systems. That makes RNS a good candidate for implementing a lot of application fields. Typical applications of RNS may be found in Digital Signal Processing (DSP) for filtering, convolutions, correlations, FFT computation, fault-tolerant computer systems, communication, and cryptography. In Dual key Homomorphic 2^16 modulo adder in use equation below show operation. x, y are inputs z is output

$z= (x + y) \bmod 2^{16}$

Modulo $(2^N + 1)$ multiplier

Binary numbers with n bits are denoted as

$A= an\text{-}1an\text{-}2.....a0$
in following text, where
n-1
$A= \Sigma\, 2^i a_i$

i=0

Reduction of a number A modulo a number M ("A mod M") may be accomplished by a division (with remainder as result) or by iteratively subtracting modulus until A < M. For modulo multiplication,

$P = X.Y \bmod (2^n + 1)$

The reduction modulo (2n+1) may be computed as:

$$A \bmod (2^n+1) = (A \bmod 2^n - A \operatorname{div} 2^n) \bmod (2^n + 1)$$

Where modulo operation on right hand side is used for final correction if subtraction Yields a negative result. Thus, modulo $(2^n + 1)$ reduction is computed by subtracting high n bit word from low n bit word and then conditionally adding $(2^n + 1)$.Modulo $(2^n + 1)$ multiplication is considered here for application in Dual key HOMOMORPHIC cipher. That is, n bit numbers in normal representation are used for operands and result, where value 0 is not used and value 2n is represented by "00…0". represented algorithm may easily be adapted for number representations with value 0 included and value 2n indicated by a separate bit. Modulo $(2^n + 1)$ multiplication using normal number representation may be formulated as:

$$X.Y \bmod (2^n+1) = (X.Y \bmod 2^n - X.Y \operatorname{div} 2^n) \bmod (2^n + 1)$$

III. IMPLEMENTATION

Proposed Modulo Multiplier: As observed from Dual key Homomorphic cipher algorithm there is four types of computation $(2^n + 1)$ modulo multiplier, (2^n) modulo adder, XOR and shifter in key generation. Out of these ability to perform fast modulo 2^n+1 multiplication is then still a major challenge, particularly from a hardware point of view. Even though a modulo $2^n + 1$ multiplier may be implemented using look-up tables, memory requirements are a big constraint for large values of n. Hence, to avoid exponential growth of memory requirements several implementations based on combinational arithmetic circuits have been proposed. Figure 4.3 below explains working. First let for n=4, $(2^n + 1)$ may be factorized in four forms below:-

R1=10001000
R2=01000100
R3=00100010
R4=00010001

R1, R2, R3 and R4 are possible four various factors of 2^4+1

Let if X=0110 and Y= 0101
Than XY = 011110

Possible solution with presented procedure is =>

00011110 – 00010001 => 0000111

Observed ANS in only one step

Let if X=1110 and Y= 1101
Than XY= 10110110

Possible solution with presented procedure is =>

10110110 – 10001000 => 00101110 - 00100010 => 00001100

Observed ANS in only two steps

Let if X=1010 and Y= 1101
Than XY= 10000010

Possible solution with presented procedure is =>

10000010 – 01000100 => 00111110 - 00100010 => 00011100-00010001=>00001001

Observed cipher only in three steps it is MAX steps required with presented architecture. The same approach is been used for 2^8+1 and $2^{16}+1$ modulo multiplier. Figures of presented $(2^n + 1)$ architecture are shown in figures next pages for n=4

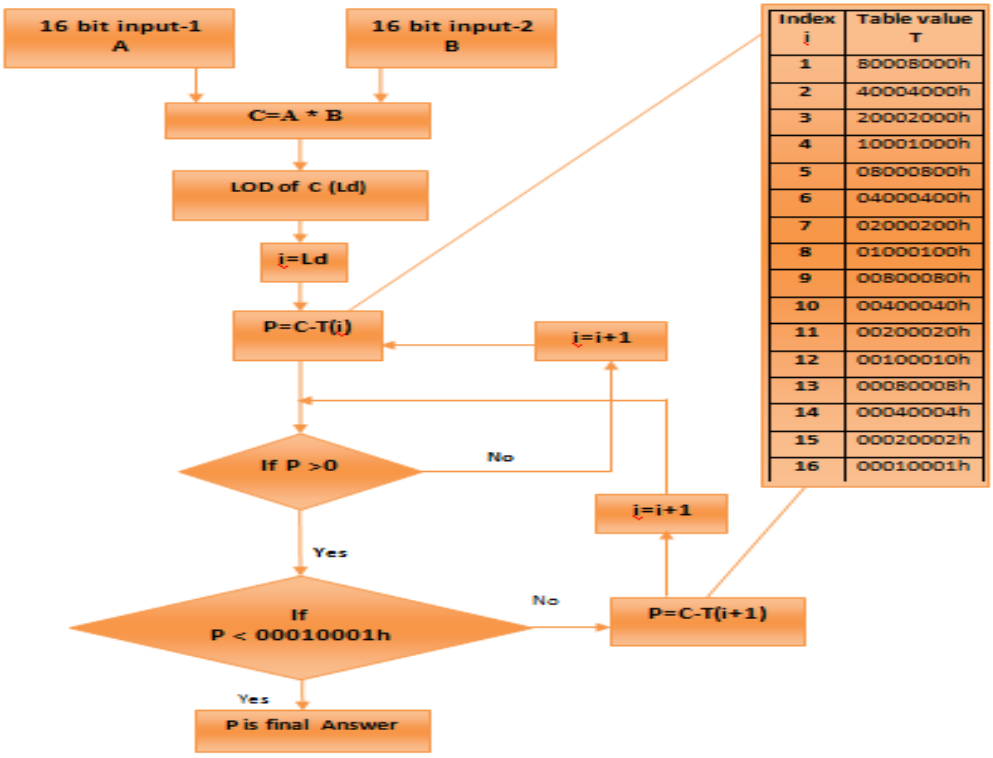

Figure 3 presented new modulo ($2^{16}+1$) Multiplier

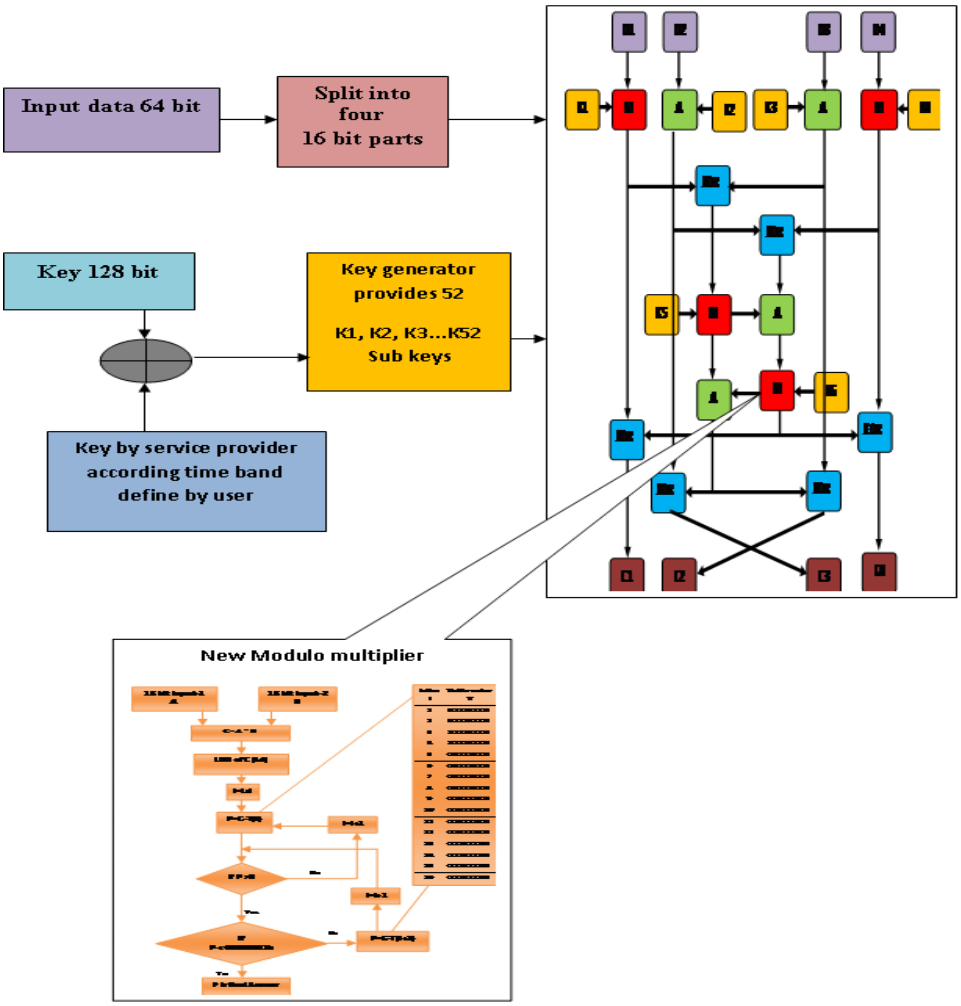

Figure 4 Block diagram of presented work

Figure 3 shown above is overall working of presented work here size of initial data is of 64 bit then there is One single 128 bit Key1 provided by user and one 128 bit Key2 provided by service provider which will be based on time of encryption and tentative time of decryption Final KEY is XOR of Key1 and Key2 With help of final KEY and Key-generator 52 sub-keys developed as k1,

k2, k3.....k52. presented work did all this for dual key based security which simply squared encryption security. After all that this keys and data provided to Homomorphic encryption engine which developed 64 bit output cipher , however there is one change that modulo multiplier in Homomorphic encryption engine is new presented by us, and this new modulo multiplier will helps to reduce overall area and enhance speed of cipher generation.

IV. RESULTS

The aim id this paper is to reduce chip are and so power. area in any digital design always required to be reduce as much as possible because area directly related to overall cost of system, size of system, power of system and as we know power matters in all battery based devices. In FPGA implementation number of slices represents area. also this work is aims to reduce the Time delay it is extreme frequency another important parameters in VLSI deigns because any system performance mainly measure in terms of speed and if it is fast enough that system will consider better. Table 2 below shows the synthesis results obtains for the proposed work when implemented on Xilinx Vivado EDA tool.

Table 2Synthesis Results obtained

Target device Zynq-7000 FPGA	
Parameters	**Results**
Slices	25630
LUT	48319
IOB	256
Time delay	1831.26 ns
Max Freq.	192.13 Mhz
Numbers of cycles for complete encryption	One Round 950 machine cycles, 8 rounds 8x950=6800, one sub-round 475 Total 7275 machine cycles and 87300 clock cycles.
Latency (time need for one data to get cipher start to end)	0.354 seconds

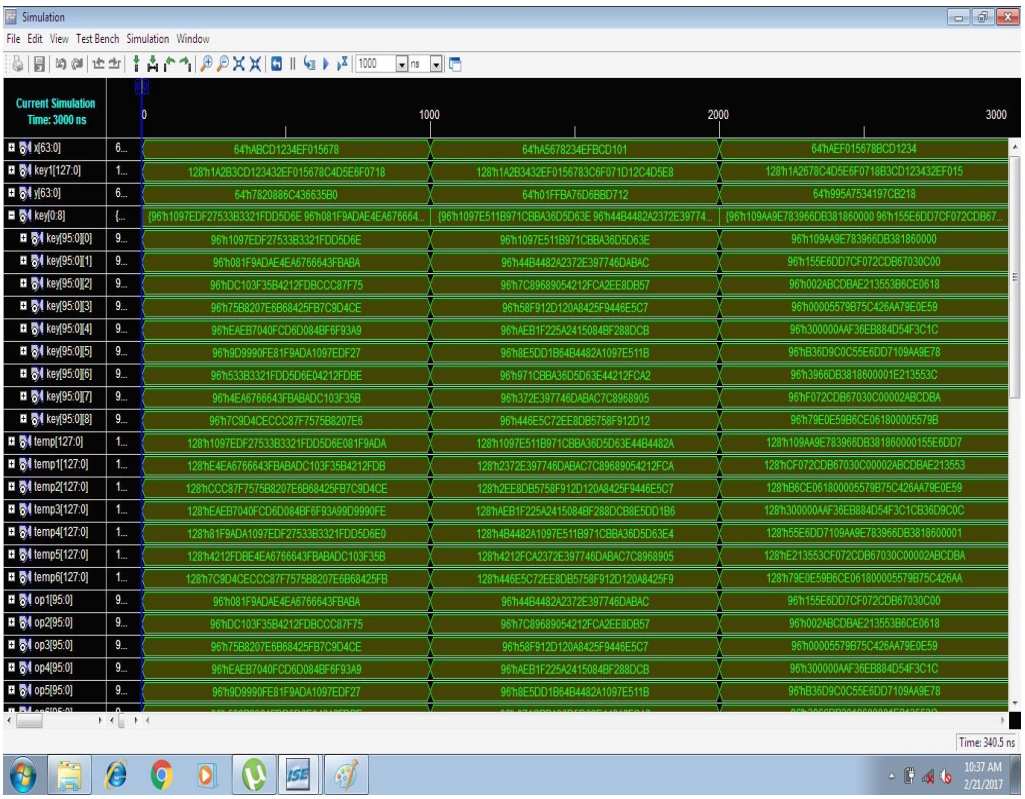

Figure 5.1 Simulation Obtained for Homomorphic cipher algorithm Round 1-4

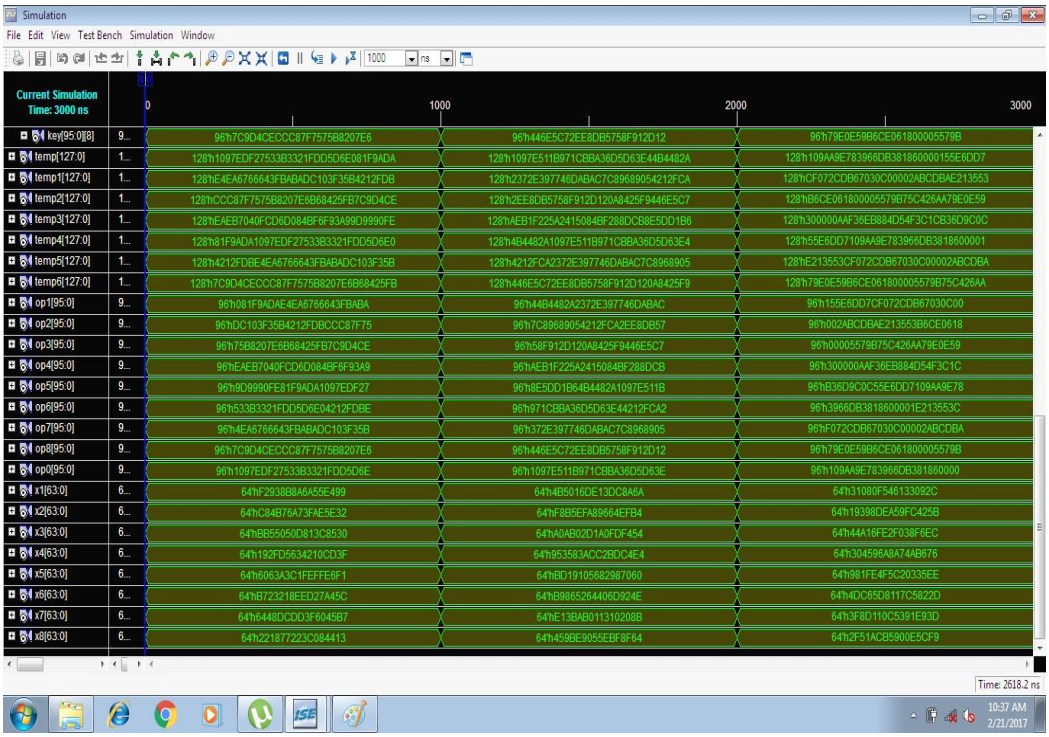

Figure 5.2 Simulation Obtained for Homomorphic cipher algorithm Round 5-8 and Sub-round

Figure 5.1 and 5.2 above shows simulation results where we may monitor output for three deferent Keys, and various inputs. In simulation we may monitor inputs and output after each rounds and also we may monitor he method of Key generation and key to each round for all three cases.

Comparative Results

Table 3 below shows the comparative results with the proposed Homomorphic cipher module implemented using Xilinx VIVADO on Zynq-7000 FPGA SoC kit and Viretx-7 FPGA kit respectively.

Table 3 Comparative Results

Author	**Work**	**LUT's (chip Area)**	**Max Freq. (Mhz)**	**Latency (One 128 bit Data encryption time)**	**Clock cycle of processor needed to complete Encryption**
Sujoy Sinha Roy et al [1]	Parallel Architecture for Homomorphic Encryption.	63522 when Zynq-7 FPGA	200	0.362 Sec	
Zhe Liu et al [2]	Ring-LWE Encryption engine on IoT on ARM-NEON processors.				149400 of ARM Neon
Yang Su et al [3]	Ring-LWE Fully Homomorphic Encryption	95854 when Virtex UltraScale FPGA	150		
Proposed work	superscalar Homomorphic Encryption	48319 when Zynq-7000 FPGA selected as target device 78567 LUT's when Vertex-7 selected as target device	192.13	0.354 Sec	87300 of ARM Cortex SoC

Form the table 3 above it may concluded that proposed work is using very less chip area in compared with other cloud secure encrypted data storage methods. also the numbers od clock cycles are required less which further reduce the dynamic power consumption of proposed design. Also the latency time is less which signifies that proposed work will encrypt single data fast in compare with other methods. Only the maximum operating frequency of proposed work is 7.87 MHz less then work in [1] though the maximum operating frequency in proposed work is 42.13 MHz higher then worn in [3].

V. CONCLUSION

One may conclude on behalf of literature survey for which we have gone through many research papers, books, Datasheets of EDA tools and references mansion in this paper that presented work is a better cloud system encryption procedure in terms of area and throughput, as known dual key cloud system encryption is just a overhead for any system and it should not took many of area or time so presented work may be solution for same as presented work necessary very less area and time as compare to other existing work in same research area. As known dual key cryptography is just an overhead for any system and it should not took many of time so presented work is a solution for same as presented necessary very less time and highly security (i.e. high avalanche effect) as compare to other existing work in same category. One may conclude that presented encryption procedure has fastest among available methods such as AES, DES and RSA. presented technique is also faster than procedure developed by researchers of [2], [5] and [1]. total avalanche observed for presented technique is 76% which is best among all procedure available hence presented work is a better cryptograph procedure in terms of throughput and security level. In this paper work, we have discussed about functionality of Cyber Network components when implemented in hardware. So FPGA is ultimate choice for such practice and out work is just gives support to this argument. In this thesis, we have presented improved modulo multiplier for Cyber Network processing applications and they achieved a substantial high throughput. Homomorphic encryption used as Cyber Network Intrusion Detection System architecture for verifying its functionality in FPGA. However, there have been few constraints associated with our work which we want to sort out in future. Firstly, our work has been realized in Vertex 5vlx330ff1760-2FPGA. In future, we want to verify our design in other High Speed FPGAs. Moreover, we have not used internal Block RAMs of FPGA. In future, we may extend our work by using Xilinx Core generator which may reduce time. This paper gives a clear statement that FPGAs are a good candidate for efficient implementation of Cyber Network processing applications. We hope that there will be few lot proposals and implementations of few other algorithms in near future

REFERENCES

[1] S. Sinha Roy, F. Turan, K. Jarvinen, F. Vercauteren and I. Verbauwhede, "FPGA-Based High-Performance Parallel Architecture for Homomorphic Computing on Encrypted Data," 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), Washington, DC, USA, 2019, pp. 387-398, doi: 10.1109/HPCA.2019.00052.

[2] Z. Liu, R. Azarderakhsh, H. Kim and H. Seo, "Efficient Software Implementation of Ring-LWE Encryption on IoT Processors," in IEEE Transactions on Computers, vol. 69, no. 10, pp. 1424-1433, 1 Oct. 2020, doi: 10.1109/TC.2017.2750146.

[3] Y. Su, B. Yang, C. Yang and L. Tian, "FPGA-Based Hardware Accelerator for Leveled Ring-LWE Fully Homomorphic Encryption," in IEEE Access, vol. 8, pp. 168008-168025, 2020, doi: 10.1109/ACCESS.2020.3023255.

[4] Y. Fan, G. Zhao, W. Shang, J. Shang, W. Lin and Z. Wang, "A Preliminary Design for Authenticity of IoT Big Data in Cloud Computing," 2020 29th International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, 2020, pp. 1-2, doi: 10.1109/ICCCN49398.2020.9209646.

[5] Zhongyuan Hao, Wei Guo, Jizeng Wei, Dual Processing Engine Architecture to Speed Up Optimal Ate Pairing on FPGA Platform, 2016 IEEE/Trustcom/BigDataSE/ISPA, DOI: 10.1109/TrustCom.2016.0113, ISSN: 2324-9013

[6] LI Wei , ZENG Xiaoyang , NAN Longmei , CHEN Tao , DAI Zibin , A reconfigurable block cryptographic processor based on VLIW architecture, China Communications (Volume: 13, Issue: 1, Jan. 2016), DOI: 10.1109/CC.2016.7405707, Page(s): 91 – 99, ISSN: 1673-5447

[7] Harivans Pratap Singh, Shweta Verma, Shailendra Mishra, Secure-International Data Encryption Algorithm, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, problem 2, February 2013, ISSN (Online): 2278 – 8875

[8] nick hoffman, a simplified Homomorphic algorithm, online documents, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.501.2662&rep=rep1&type=pdf

[9] Sandipan Basu, international data encryption algorithm (idea) a typical illustration, Volume 2, No. 7, July 2011 Journal of Global Research in Computer Science REVIEW ARTICLE Available Online at www.jgrcs.info, JGRCS 2010

[10] Oleg Vyshnyvetshkey, sebistian gulloex, Homomorphic block cipher final presentation, RIT cryptographic course, 2012

[11] Xilinx documents, https://www.xilinx.com/products/design-tools/Vivado-design-suite.html

[12] Vertex-7 FPGA datasheet, https://www.xilinx.com/support/documentation /datasheets/ds112.pdf

[13] NPTL lectures on VHDL, http://nptel.ac.in/courses/117108040

[14] Swapna kumari , Dasari. Subbarao, Implementation of AES-256 Encryption Algorithm on FPGA, International Journal of Emerging Engineering Research and Technology Volume 3, problem 4, April 2015, PP 104-108 ISSN 2349-4395 (Print) and ISSN 2349-4409 (Online) Address for correspondence subbaraod15@gmail.com International Journal of Emerging Engineering Research and Technology V3 I4 April 2015 104

[15] Kyle Wilkinson, Using Encryption to Secure a 7 Series FPGA Bitstream, Application Note: 7 Series FPGAs, XAPP1239 (v1.0) April 15, 2015 www.xilinx.com