HIGH SECURED AES ENCRYPTION AND DECRYPTION ALGORITHM USING XILINX

[1]Ch. Srigiri, [2]Ch. J S Manikanta Nitish, [3]P. M S Susmitha, [4]V. Steven Viswash, [5]Ch. Uday Kishore
[1]Assistant Professor, [2]UG Student, [3]UG Student, [4]UG Student, [5]UG Student
[1,2,3,4,5]Department of Electronics and Communication Engineering,
[1,2,3,4,5]Godavari Institute of Engineering and Technology, Rajahmundry, India.

ABSTRACT

With the present omnipresence of pc networks, distributed systems generally, and also the net particularly, cryptography has become associate enabling technology to secure the data, infrastructures we tend to area unit building, using, and hoping on in way of life. In gift days, nearly each relevant communication system needs secure information transfer so as to keep up the privacy of the transmitted message. Hardware implementation on FPGA offers a faster and customizable answer. we tend to use Verilog Language for synthesizing logic style. This project uses Xilinx ISE 14.7i that is associate microcircuit development platform supported the Xilinx ISE 14.7i tool. Software package implementation on Xilinx is often advanced style and in counter mode it becomes the foremost complex drawback. we tend to customize the planning of AES formula to implement merely on Xilinx software package. This may later facilitate the developers to change and use the software package simply with none issues throughout installation of style. Some key blessings of AES encryption and coding formula are unit quick in execution and it additionally uses higher length sizes 128,192 and 256 bits for encryption which provides additional security to the information because it needs 2^{128} tries to deploy or hack the information.

I. INDRODUCTION

To demonstrate a 128-bit Advanced encryption system (AES) each regular key encryption and coding rule by developing appropriate software system style on Xilinx ISE 14.7i device, the implementation has been tested with success. The system is optimized in terms of fastness and hardware utilization. It style victimization application in Security functions, Medical field, Network Security, on-line bank security. It develop similar approaches for the implementation of AES, we are able to implement double AES for a lot of security and can less encryption speed. In today's world most of the communication is finished through victimization of electronic media. Information Security plays an important role in such communication. Hence, there's a necessity to guard information from malicious attacks. Cryptography is that the science of secret codes, enabling the confidentiality of communication through associate insecure channel. It protects against unauthorized parties by preventing unauthorized alteration of use. typically speaking, it uses a cryptanalytic system to rework a plaintext into a cipher text.

II. OBJECTIVE

The aim of this project is to speak the information in secret exploitation AES rule. Therefore we tend to initial send the information (plain text) that is of 128 bits and therefore the key which might be of 128 or 192 or 256 bits into the secret writing method. The output of this method are going to be cipher text. This cipher text is then fed into the secret writing method and so the information (plain text) as output, since we tend to add the key and shuffle the knowledge of information. It'll be terribly laborious for the unknown person to search output for the first data. Since for every key there'll be a amendment within the cipher text so the person should understand the key in order to search the first data, which is known as plaintext.
This project is all regarding providing security for the information. The information that is transmitted by sender are going to be received by the receiver. once the information encrypted and decrypted at that instant the hacker might hack the information. To avoid these kinds of the issues we tend to use some security Algorithms like AES, SHA-0, SHA-1, SHA-2, and RSA.

III. LITERATURE SURVEY

Cryptography may be a technique of securing info and communications through use of codes so solely those individuals for whom the knowledge is meant will are aware of it and method it, therefore preventing unauthorized access to info. The prefix "crypt" suggests that "hidden" and suffix graphy suggests that "writing".

it's the observe and study of techniques for secure communication within the presence of third parties referred to as adversaries. a lot of usually, cryptography is regarding constructing and analyzing protocols that stop third parties or the general public from reading non-public messages; numerous aspects in info security like knowledge confidentiality, knowledge integrity, authentication, and non-repudiation square measure central to trendy cryptography. Trendy cryptography exists at the intersection of the disciplines of arithmetic, computing, technology, communication science, and physics. Applications of cryptography embody electronic commerce, chip-based payment cards, digital currencies, pc passwords, and military communications. The expansion of cryptanalytic technology has raised variety of legal problems within the modern era. Cryptography's potential to be used as a tool for undercover work and offense has semiconductor diode several governments to classify it as a weapon and to limit or perhaps command its use and export. In some jurisdictions wherever the employment of cryptography is legal, laws allow investigators to compel the revelation of encoding keys for documents relevant to associate degree investigation. Cryptography conjointly plays a serious role in digital rights management and violation of digital media.

Effective might twenty six, 2002 the National Institute of Science and Technology (NIST) has elite a block cipher referred to as RIJNDAEL (named when its creators Vincent Rijmen and Joan Daemen) because the circulate key encryption algorithmic program to be accustomed to write in code sensitive however unclassified yankee federal info. RIJNDAEL was originally a variable block (16, 24, 32 bytes) and variable key size (16, 24, 32 bytes) encryption algorithmic program. Office has but determined to outline AES with a block size of sixteen bytes whereas keeping their choices hospitable future changes.

IV. BLOCK DIAGRAM OF PROJECT

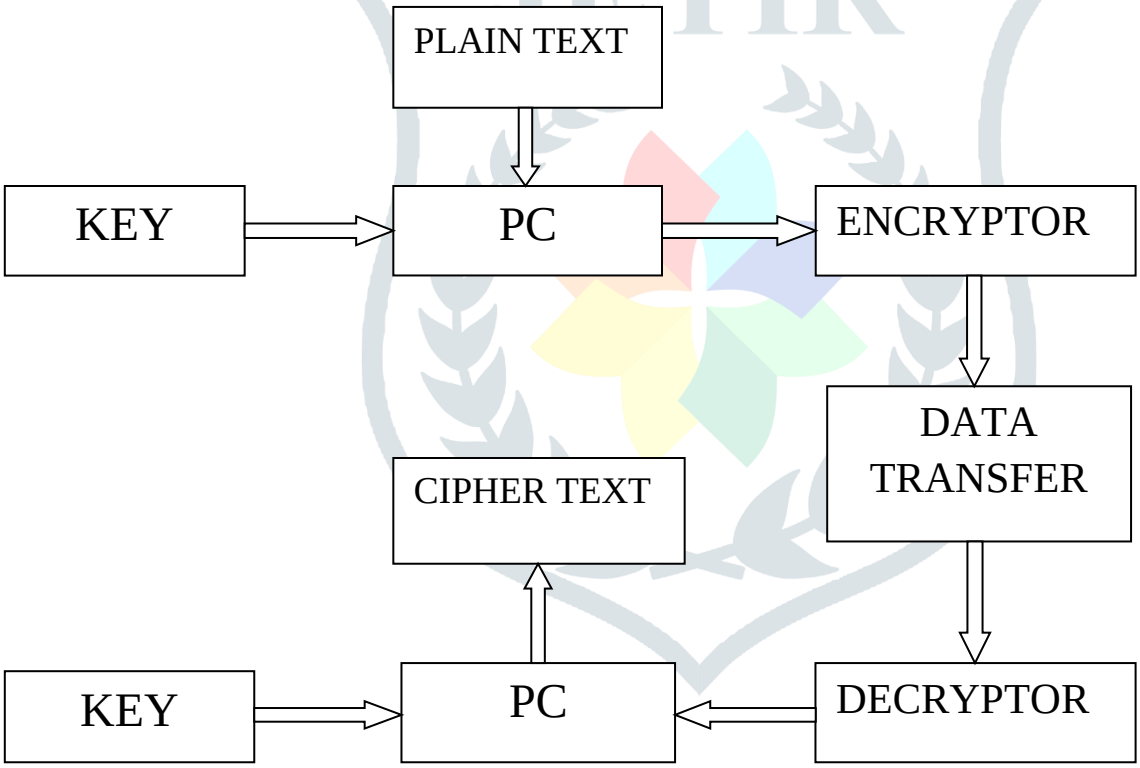

FIG 1 – BLOCK DIAGRAM

4.1 ENCRYPTION

Encryption is that the method of encoding data. This method converts the first illustration of the knowledge, referred to as plaintext, into another type referred to as cipher text. Solely licensed parties will decipher a cipher text back to plaintext and access the first data. Coding doesn't itself stop interference however denies the intelligible content to a would-be fighter aircraft. For technical reasons, a coding theme typically uses a pseudo-random coding key generated by a rule. It's attainable to decipher the message while not possessing the key, but, for a well-designed coding theme, extensive procedure resources and skills square measure needed. A certified recipient will simply decipher the message with the key provided by the mastermind to recipients however to not unauthorized users. Traditionally, numerous varieties of coding are accustomed aid in cryptography. Early coding techniques were usually utilized in military electronic messaging. Trendy

coding schemes utilize the ideas of public-key and symmetric-key. Trendy coding techniques guarantee security as a result of trendy computers square measure inefficient at cracking the coding. However, researchers at bureau and alternative cyber security specialists recommend that the event of quantum computers could threaten current coding systems.

4.2 DECRYPTION

Decryption is that the method of taking encoded or encrypted text or alternative knowledge and changing it into text that you simply or the pc will browse and perceive. This term may be wont to describe a technique of unencrypting the information manually or unencrypting the information victimization the right codes or keys. Knowledge is also encrypted to create it troublesome for somebody to steal the knowledge. Some corporations additionally cipher knowledge for general protection of company knowledge and trade secrets. If this knowledge must be seeable, it's going to need decipherment. If a decipherment pass code or secret's not out there, special computer code is also required to rewrite the information victimization algorithms to crack the decipherment and build the information decipherable.

V. ALGORITHM FLOW OF PROJECT

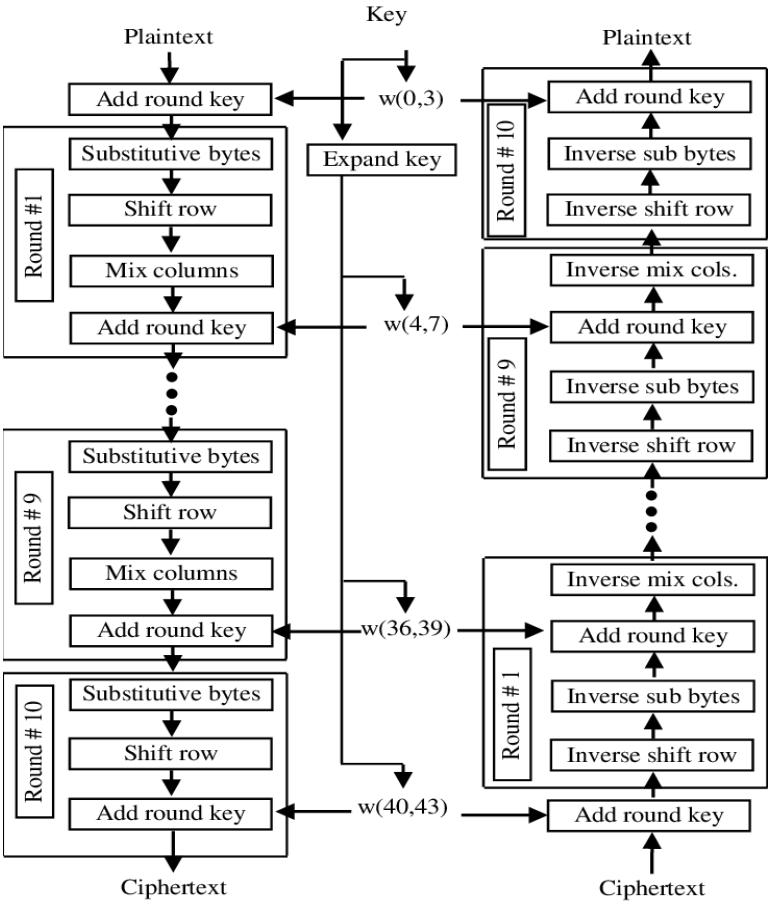

FIG 2 – ALGORITHM FLOW FOR AES ENCRYPTION AND DECRYPTION

AES is associate iterated symmetrical block cipher, which implies that:

AES similarly as most encryption algorithm is reversible. This suggests that just about identical steps square measure performed to complete each encoding and coding in reverse order. The AES formula operates on bytes, which makes it less complicated to implement and justify. This key's distended into individual sub keys, a sub keys for every operation spherical. This method is termed as KEY expansion, that is delineating at the tip of this document.

For each its Cipher and Inverse Cipher, the AES formula uses a round operation which is composed of 4 totally different byte-oriented transformations: 1) computer memory unit substitution employing a substitution table (S-box), 2) Shifting rows of the State array by totally different offsets, 3) mix the information among every column of the State array, and 4) Adding a round Key to the State. As mentioned before AES is associate iterated block cipher. All meaning is that identical operations square measure performed persistently on a hard and fast range of bytes. These operations will simply be diminished to the subsequent functions:

- ADD ROUND KEY
- SUB BYTE
- SHIFT ROW
- MIX COLUMN

An iteration of the on top of steps is termed a round. The number of rounds of the algorithm depends on the key size. The only exception being that within the last round the mix Column step isn't performed, to form the algorithm reversible throughout decryption.

Key Size (bytes)	Block Size (bytes)	Rounds
16	16	10
24	16	12
32	16	14

TABLE 1 - KEY SIZES

VI. ALGORITHM FLOW FOR CIPHER FUNCTIONS

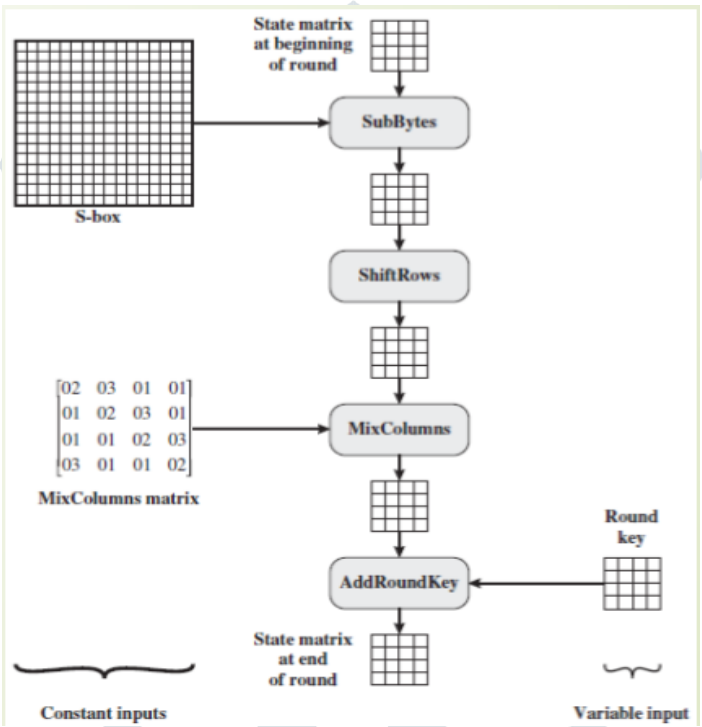

FIG 3 – ALGORITHM FLOW OF CIPHER FUNCTIONS

The algorithm flow for the operations consist the above mentioned operations in required design flow to meet the desired output. The first operation Sub Bytes are converted using the S-Box which contains the fixed set of data and enables the restrictors to change the data to hexadecimal format. The second operation is where the actual encryption logistics are started by sifting the rows of data in a hierarchical process. The third operation performs the matrix multiplication with already existed matrix and creates a complicated data. The fourth operation performs the Ex-or operation with another column and leads the data to a high security scheme. The same process is repeated for n number of times to meet the desired output of required data. The reverse process of four operations is carried for the decryption process. Let us study the operation process in detail.

5.1 ADD ROUND KEY

Each of the sixteen bytes of the state is XORed against every of the sixteen bytes of some of the expanded key for this round. The expanded Key bytes square measure ne'er reused. therefore once the primary sixteen bytes square measure XORed once more the primary sixteen bytes of the expanded key then the expanded key bytes 1-16 are never used again. Subsequent time the Add round Key operate is named bytes 17-32 are XORed against the state.

The first time Add Round Key gets executed.

State	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR
Exp Key	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

The second time Add Round Key gets executed.

State	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR
Exp Key	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

And so on for every round of execution. Throughout decryption this procedure is reversed. Thus the state is 1st XORed against the last sixteen bytes of the expanded key, then the second last sixteen bytes so on.

5.2 SUB BYTES

During encryption every value of the state is replaced with the corresponding SBOX value.
For example HEX 19 would get changed to HEX D4.
During decryption every value within the state is replaced with the corresponding inverse of the SBOX.
For example HEX D4 would get changed to HEX 19.

5.3 SHIFT ROWS

Arranges the state during a matrix so performs a circular shift for every row. This can be not a bit wise shift. The circular shift simply moves every byte one house over. A byte that was within the second position could end up within the third position after the shift. The circular a part of it specifies that the byte within the last position shifted one house can end up within the initial position within the same row. In Detail: The state is organized in a very 4x4 matrix (square).
Each row is then moved over (shifted) one, two or three spaces over to the proper, looking on the row of the state. Initial row is rarely shifted.

5.4 MIX COLUMNS

This is maybe the toughest step to each understand and explain. There are 2 parts to the present step. the primary can justify that elements of the state are increased against that elements of the matrix. The second can justify however this multiplication is enforced over what's referred to as a Galois Field.
The state is organized into a four row table (as represented within the Shift Row function). The multiplication is performed one column at a time (4 bytes). Every value within the column is eventually increased against each value of the matrix (16 total multiplications). The results of those multiplications are XORed along to provide only four result bytes for subsequent state. So four bytes input, sixteen multiplications twelve XORs and four bytes output. The multiplication is performed one matrix row at a time against every worth of a state column.

Matrix multiplication

2 3 1 1
1 2 3 1
1 1 2 3
3 1 1 2

16 byte state
b1 b5 b9 b13
b2 b6 b10 b14
b3 b7 b11 b15
b4 b8 b12 b16

The first result byte is calculated by multiplying four values of the state column against four values of the primary row of the matrix. The result of every multiplication is then XORed to provide one byte.
b1 = (b1 * 2) XOR (b2*3) XOR (b3*1) XOR (b4*1)
The second result byte is calculated by multiplying constant four values of the state column against four values of the second row of the matrix. The result of every multiplication is then XORed to provide one byte.
b2 = (b1 * 1) XOR (b2*2) XOR (b3*3) XOR (b4*1)
The third result byte is calculated by multiplying constant four values of the state column against four values of the third row of the matrix. The result of every multiplication is then XORed to provide one byte.
b3 = (b1 * 1) XOR (b2*1) XOR (b3*2) XOR (b4*3)
The fourth result byte is calculated by multiplying constant four values of the state column against four values of the fourth row of the matrix. The result of every multiplication is then XORed to provide one byte.
b4 = (b1 * 3) XOR (b2*1) XOR (b3*1) XOR (b4*2)
This procedure is recurrent again with consequent column of the state, till there are no more state columns. Put it all together, the primary column can embrace state bytes 1-4 and can be multiplied against the matrix.

V11. KEY EXPANSION ALGORITHM FLOW

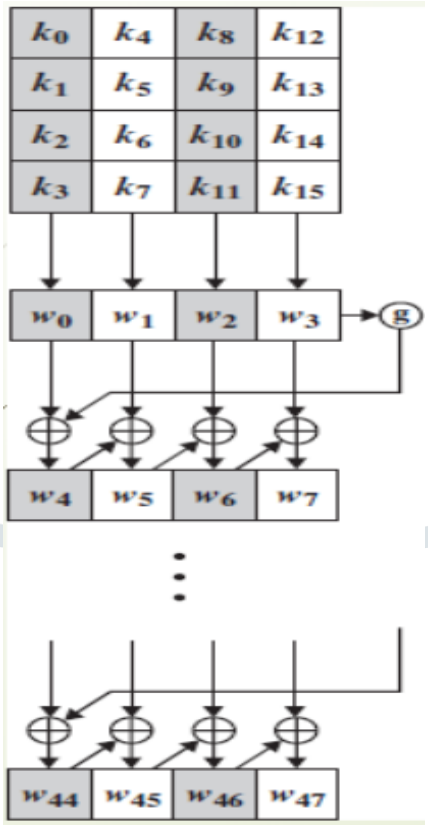

FIG 4 – ALGORITHM FLOW FOR KEY EXPANSION

Prior to encryption or decryption the key should be expanded . The expanded key is utilized in the Add round Key function outlined on top of. anytime the Add round Key operate is termed a distinct a part of the expanded key is XORed against the state. so as for this to figure the expanded Key should be giant enough so it will offer key material for each time the Add round Key operate is dead. The Add round Key operate gets mixed up every round additionally in concert time beyond regulation at the start of the algorithm. so the scale of the expanded key can continually be equal to: sixteen * (number of rounds + 1).

The sixteen within the on top of operate is truly the scale of the block in bytes. This provides key material each|for each} byte within the block throughout every spherical +1

Since the key size is far smaller than the scale of the sub keys, the secret's truly “stretcheout” to produce enough key area for the algorithm. The key expansion routine executes a most of four consecutive functions.

VIII. FLOWCHARTS

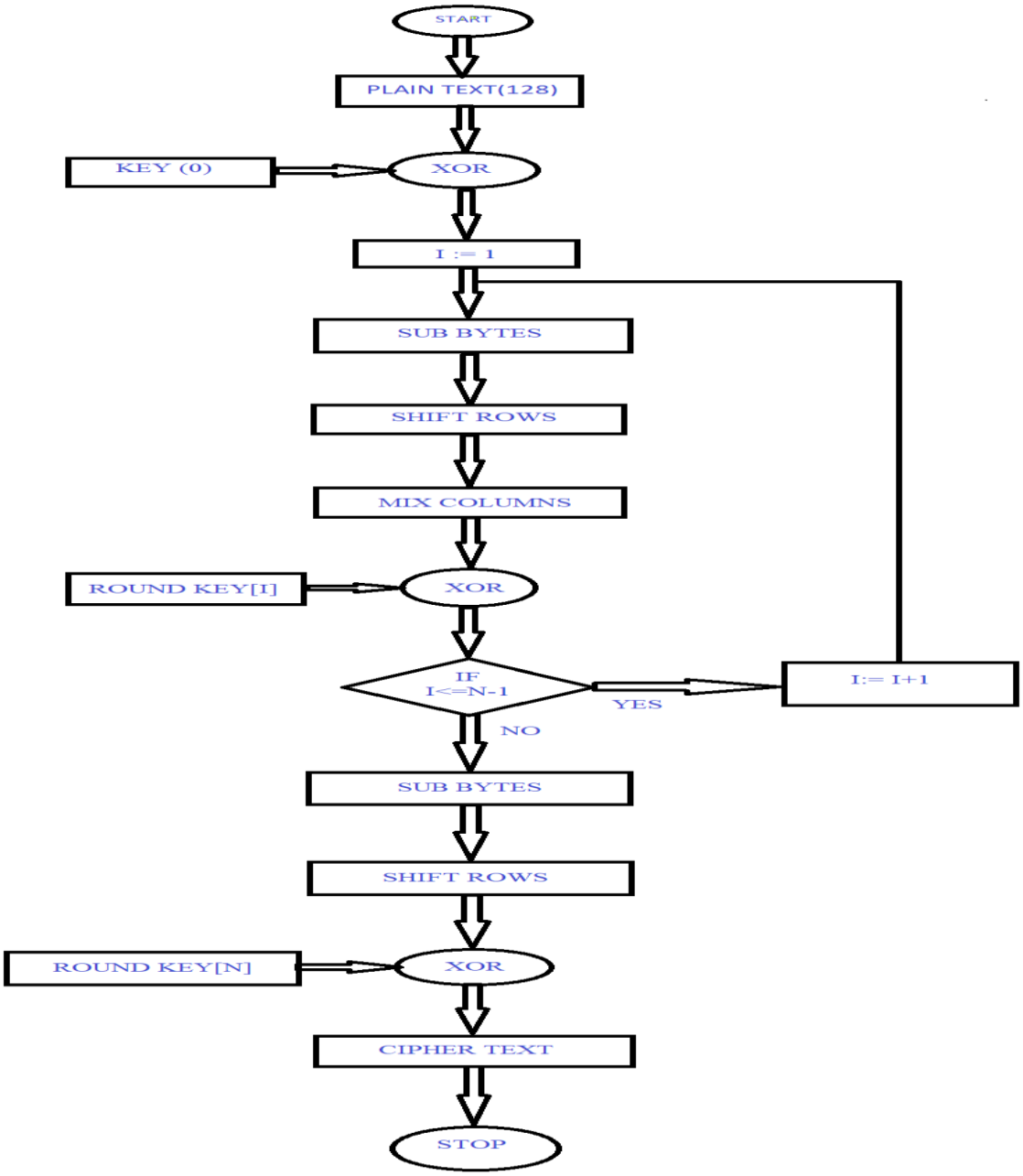

FIG 5 – ENCRYPTION FLOWCHART

Above mentioned figure a flowchart for the encryption process of N rounds we generally use 10 rounds for 128 bits of data. Both the key and plaintext are generated by the user at the time of start and stopped when the user arrives to desired output.

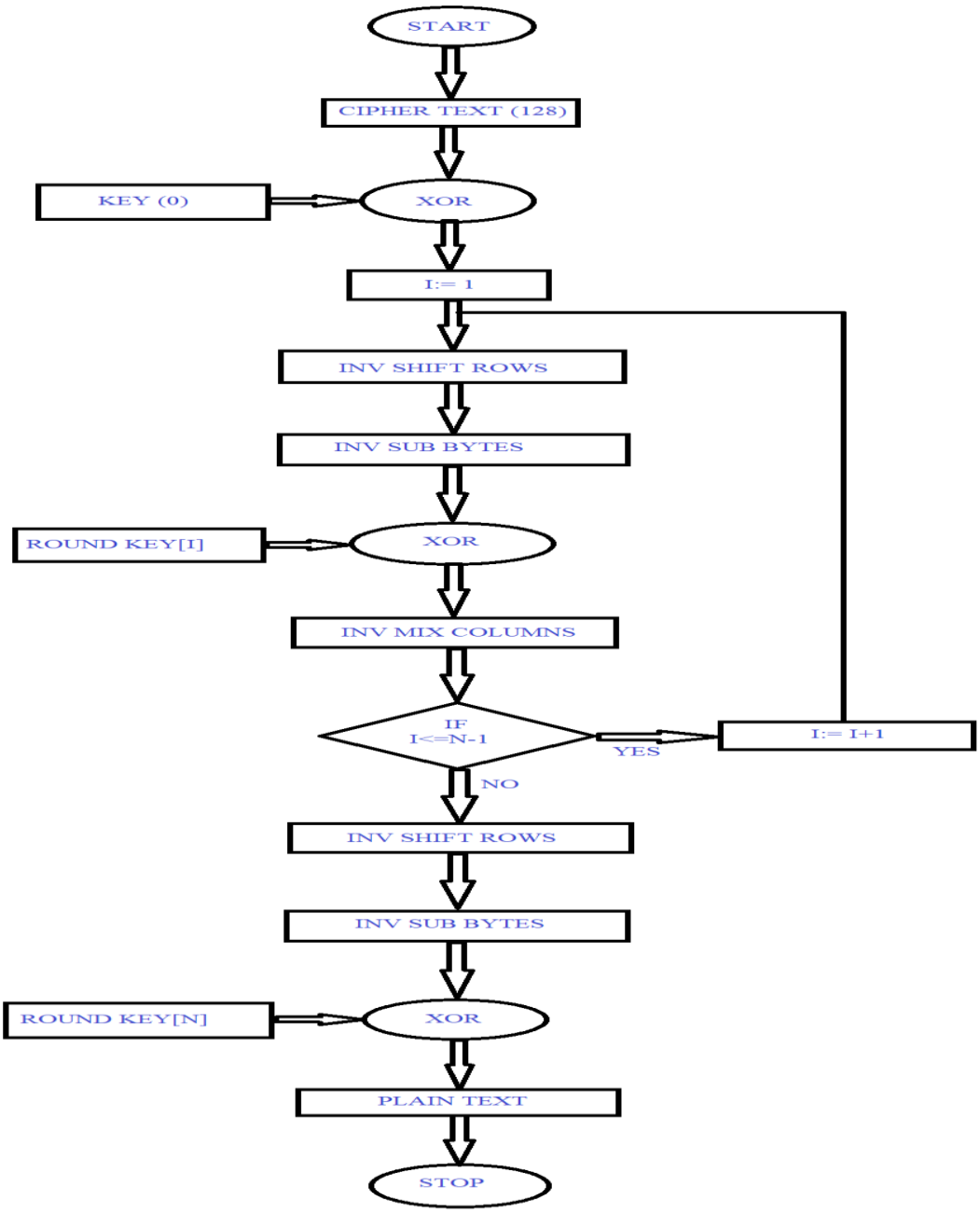

FIG 6 – DECRYPTION FLOWCHART

Below mentioned figure a flowchart for the decryption process of N rounds we generally use 10 rounds for 128 bits of data. Cipher text and key are taken by the user at the time of start and stopped when the user arrives to the actual plaintext.

The flowchart says us about typical operation that it performs and it also mentions that the minimal error in assuming the wrong key can change the informative data to coin-less information. Every function here used is called as inverse and hence perform the backward operation of what the encryption performs which indirectly gives a unique identity to the process of decryption.

IX. RESULTS AND DISCUSSIONS

The encryption and decryption algorithms take their respective inputs in order to achieve the desired outputs.

The encryption algorithm takes the plain text and security key from the user and generates cipher text. We are using Verilog programming language and intuition coding process, where the main code calls all the codes to run the simulation. By this process we can reduce the time factor and it also minimizes to write such a long codes for the 10 rounds of 128 bit encryption process. Every round includes the four major operations as we already know such as sub-bytes, shift-rows, mix-columns and add round key.

The decryption algorithm takes the cipher text and security key from the user to generate the plain text back as the transformation of data is done. As it creates a best security option this is often used in many fields. Decryption can also be performed under the same process that we carried out for the process of encryption.

Let us look over some of the results that we obtained from our project.

Let us first understand the manual result, to do that we will go through an example.

DATA INPUT – PLAIN TEXT – 3243f6a8885a308d313198a2e0370734

KEY – SECURITY KEY – 2b7e151628aed2a6abf7158809cf4f3c

DATA OUTPUT – CIPHER TEXT – 3925841d02dc09fbdc118597196a0b32

The above mentioned are three types that we will get while performing the different algorithms,

For encryption process we will give the plain text and key as input and achieve the cipher text as output.

For decryption process we will give the cipher text and key as input and achieve the plain text as output.

Let us have a look over the simulation result.

Encryption simulation result: -

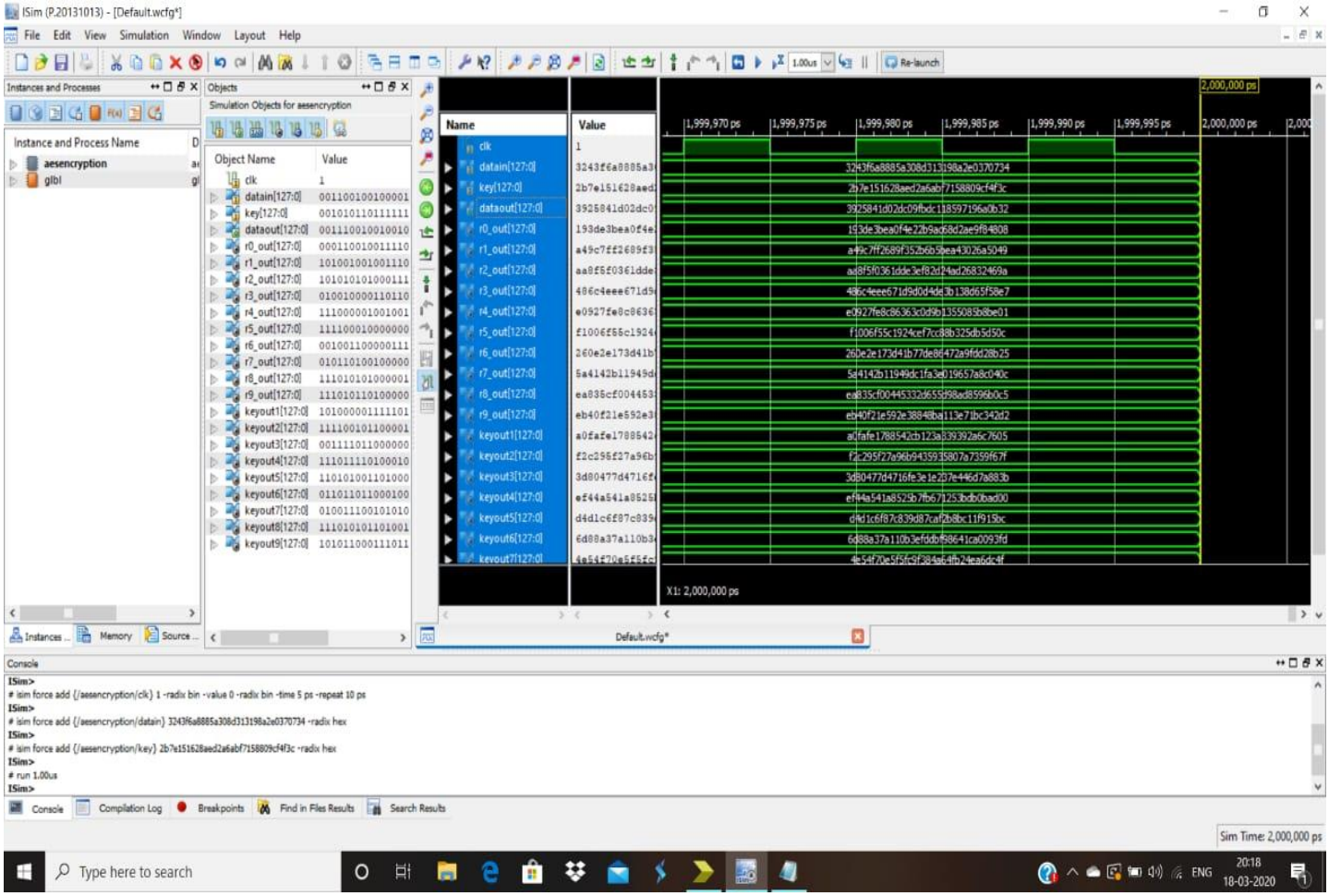

FIG 7 – ENCRYPTION SIMULATION RESULT

Decryption simulation result:-

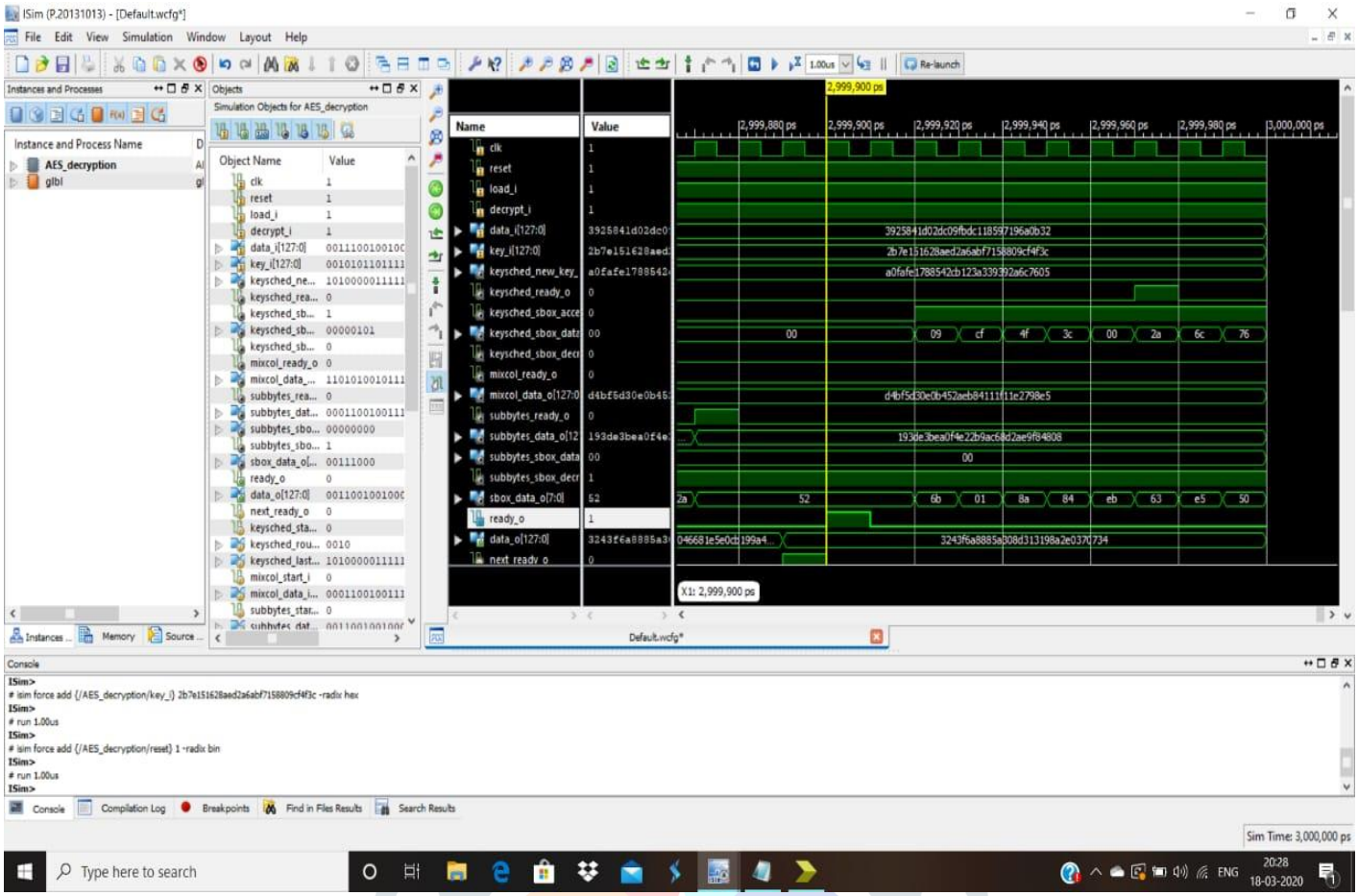

FIG 8 – DECRYPTION SIMULATION RESULT

- Clock High Time: 20 ns.
- Clock Low Time: 20 ns.
- Input Setup Time: 10 ns.
- Output Valid Delay: 10 ns.
- Offset: 0 ns.
- Initial Length of Test Bench: 1500 ns.

The above mentioned are the some of the measures from the project simulation test.

X. XILINX ISE 14.7i

Xilinx ISE (Integrated Synthesis Environment) is also a software system tool created by Xilinx for synthesis and analysis of alpha-lipoprotein styles, sanctionative the developer to synthesize ("compile") their styles, perform temporal arrangement analysis, examine RTL diagrams, simulate a design's reaction to totally different stimuli, and tack together the target device with the computer programmer. The Xilinx ISE is primarily used for circuit synthesis and elegance, whereas ISIM or the ModelSim logic machine is used for system-level testing. The parts that are shipped along with the Xilinx ISE embody are the Embedded Development Kit (EDK), a software system Development Kit (SDK) and Chip Scope professional. the first interface of the ISE is that the Project Navigator, which includes the look hierarchy (Sources), a computer code computer file editor (Workplace), Associate in Nursing output console (Transcript), and a processes tree (Processes). Style the planning that look hierarchy consists of design files (modules), whose dependencies are understood by the ISE and displayed as a tree structure. For single-chip styles there may even be one main module, with different modules enclosed by the foremost module, nearly just like the main() software system in C++ programs. style constraints ar ordered enter modules, that embody pin configuration and mapping. The hierarchical process describes the operations that the ISE can perform on the presently active module in simulation software. The hierarchy includes compilation functions, their

dependency functions, and different utilities. The window additionally denotes problems or errors that arise with every perform. The Transcript window provides standing of presently running operations, and informs engineers on style problems. Such problems may even be filtered to signifies Warnings, Errors, or both.

XI. CONCLUSION

A recent survey says that variety of connected devices through web can rise to 39 billion by 2025. because of this growth within the web usage, it's essential to defend our increasing digital frontier. Securing info on all web primarily based applications like ATM machines, Smartcards, E-commerce, then on is of utmost vital. Security thinks about with the flexibility of a system to forestall unauthorized access to info or services. historically, security problems are related to giant databases. throughout the previous few years, security problems have additionally become vital in embedded period of time systems. Especially, the expansion of web has additionally led to internet primarily based management of embedded systems and devices. Thus many devices poses severe security concerns. one among in every of the first reason that intruders is also prospering is attributable to that information acquired from a system is in a type that is straightforward to browse and comprehend. once we think about the various electronic messages that traverse through the web on a daily basis, it's not tough to see however a well placed network sniffer would possibly capture a wealth of knowledge that users wouldn't wish to have disclosed to unintended readers. Intruders might reveal the data to others, modify it to misrepresent a private or organization, or use it to launch an attack. One resolution to the matter is that through the utilization of cryptography, to forestall intruders from having the ability to use the data that they capture.

This research paper, we have proposed a encryption and decryption algorithm using a AES algorithm. It provides the functionality of calling the programming codes as for their need of performing the required operation which is executed in Xilinx in a new project file. Thus it increases the speed of encryption and decryption process. Which further results in low power consumption, less time compilation and high security for sure as there is less time to encrypt and decrypt the data. It also uses a power consumption of 1.55W and we also reduced the delay the 4.221 ns in the output. All these factors lead to a security increase the encryption and decryption process.

XII. REFERENCES

[1] Xiwei Zhang, Meng Li, Jing Hu, "optimization and implementation of AES algorithm based on xilinx" 2018 IEEE 4th International Conference on Computer and Communications (IEEE 2018), September 4-6, 2018, Beijing, China.

[2] Shuang Chen, Wei Hu, Zhenhao Li, "High performance Data Encryption with AES Implementation on Xilinx", 2019 IEEE 5th International Conference on big Data Security on cloud, (IEEE 2019), Wuhan, Hubei, China.

[3] A. Satoh, S. Morioka, K. Takano, S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization," Proc. ASIACRYPT 2001, pp. 239-254, 2001.

[4] J. Wolkerstorfer, E. Oswald, M. Lamberger, "An ASIC implementation of the AES S-Boxes," Topics in CryptologyCT-AES 2002 Proc. AES Conf. 2002, Feb. 2002.

[5] M. Qiu, Z. Jia, C. Xue, Z. Shao, E. H.-M. Sha, "Voltage assignment with guaranteed probability satisfying timing constraint for real-time multiproceesor DSP," J. VLSI Signal Process. Syst., vol. 46, no. 1, pp. 55-73, Jan. 2007.

[6] K. Gai, M. Qiu, "Blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers," IEEE Trans. Ind. Informat., 2018.

[7] Y. Li, W. Dai, M. Qiu, Z. Ming, "Privacy protection for preventing data over-collection in smart city," IEEE Trans. Comput., vol. 65, pp. 1339- 1350, May 2016.

[8] Chen M C, Hsiao S F, "Low Cost Design of an Advanced Encryption Standard (AES) Processor Using a New Common-SubexpressionElimination Algorithm," IEICE Trans Fundamentals, vol. 92, pp. 3221-3228, December 2009.

[9] Vanitha M, Sakthivel R and Subha S, "highly secured high throughput VLSI architecture for AES algorithm," IEEE International Conference on Devices, Circuits and Systems, Coimbatore, 2012, pp. 403-407.

[10] Zhang X, Parhi K K, "High-speed VLSI architectures for the AES algorithm," IEEE Transaction on Very Large Scale Integration (VLSI) Systems, vol. 12, pp. 957-967, September 2004.

[11] Hammad I, Elsankary K, Elmasry E, "High-Speed AES Encryptor With Efficient Merging Techniques," IEEE Embedded Systems Letters, vol. 2, pp. 67-71, September 2010.

[12] Daemen J, Rijmen V, The design of Rijndael: AES, the advanced Encryption Standard, Berlin, 2002, pp. 56-59.
[13] Oukili S, Bri S, "High speed efficient advanced encryption standard implementation," IEEE International Symposium on Networks, Computers and Communications, Marrakech, 2017, pp. 1-4.
[14] S. Sau , C. Pal and A Chakrabarti "Design and Implementation of Real Time Secured RS232 Link for Multiple FPGA Communication, Proc. Of International Conference on Communication, Computing & Security",2011, ISBN - 978- 1-4503-0464- 1.
[15] xilkernel_v3.00.pdf on *www.xilinx .com.*
[16] R. L. Rivest et al. 1978. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM". Vol. 2 1. pp. 120- 126.
[17] "Cryptography & Network Security ByBehrouzAForouzan".
[18] "Montgomery Algorithm for Modular Multiplication Professor Dr. D. J. Guan" ,August 25, 2003.
[19] "A. Tenca, C. Koc. 1999. A Scalable Architecture for Montgomery Multiplication. Cryptographic Hardware and Embedded Systems", Lecture Notes in Computer Science, No. 17 17, pp. 94- 108.
[20] *http://csrc.nist.gov/publications/fips/fipsI97Ifips-197.pdf.*
[21]B. Schneier. 1996. "Applied Cryptography, Protocols, Algorithms, and Source Code in C", John Wiley and Sons Inc. 2nd Edition. New York, U.S.A.