International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 6 Issue: 3 100 – 106

Design and Implementation of Triple DES Encryption Scheme

Prabhavathi M,Saranya S,Seby Netto, Sharmily G [1]
IV UGStudents, Dept. of ECE
Sri Shakthi Institute of Engineering and Technology
Coimbatore, Tamilnadu, India

Mrs. Reshma [2]
Assistant Professor, Dept. of ECE
Sri Shakthi Institute of Engineering and Technology
Coimbatore, Tamilnadu, India
Email:reshma@siet.ac.in

S. Raja[3]
Assistant Professor, Dept. of ECE
Sri Shakthi Institute of Engineering and Technology
Coimbatore, Tamilnadu, India
Email:raja.s@siet.ac.in

***Abstract*—**The speed of exhaustive key searches against DES after 1990 began to cause discomfort amongst users of DES. However, users did not want to replace DES as it takes an enormous amount of time and money to change encryption algorithms that are widely adopted and embedded in large security architectures. The DES algorithm was replaced by the Advanced Encryption Standard (AES) by the National Institute of Standards and Technology (NIST). The pragmatic approach was not to abandon the DES completely, but to change the manner in which DES is used. DES is often used in conjunction with Triple DES. It derives from single DES but the technique is used in triplicate and involves three sub keys and key padding when necessary, such as instances where the keys must be increased to 64 bits in length. Known for its compatibility and flexibility, software can easily be converted for Triple DES inclusion. Therefore, it may not be nearly as obsolete as deemed by NIST. This led to the modified schemes of Triple DES (sometimes known as 3DES).3DES is a way to reuse DES implementations, by chaining three instances of DES with different keys. 3DES is believed to still be secure because it requires 2^112 brute-force operations which is not achievable with foreseeable technology. While AES is a totally new encryption that uses the substitution-permutation network, 3DES is just an adaptation to the older DES encryption that relied on the balanced Feistel network. But since it is applied three times, the implementer can choose to have 3 discrete 56 bit keys, or 2identical and 1 discrete, or even three identical keys. This means that 3DES can have encryption key lengths of 168, 112, or 56 bit encryption key lengths respectively. But due to certain vulnerabilities when reapplying the same encryption thrice, it leads to slower performance. In this paper we present a pipelined implementation in VHDL, in Electronic Code Book (EBC) mode, of this commonly used Cryptography scheme with aim to improve performance. We achieve a 48-stage pipeline depth by implementing a TDES key buffer and right rotations in the DES decryption key scheduler. We design and verify our implementation using ModelSim SE 6.3f and Xilinx ISE 8.1i. We gather cost and throughput information from the synthesis and Timing results and compare the performance of our design to common implementations presented in other literatures.

Keywords*-DES, AES, encryption key,VHDL, ModelSim ,Xilinx ISE 8.1i.*

*****

I. INTRODUCTION

Cryptography is an art of composing in mystery symbols and is an antiquated craft; the initially reported utilization of cryptography in composing goes once again to circa-1900 B.C. at the point when an Egyptian copyist utilized non-standard symbolic representations in an engraving. A few masters contend that cryptography showed up spontaneously at some point in the wake of composing was imagined, with requisitions running from strategic messages to war-time fight tactics. It is not at all astonishment, then,that new types of cryptography came not long after the across the board improvement of machine interchanges. In information and telecommunications, cryptography is fundamental when conveying over any non-trusted medium, which incorporates pretty much any system, especially the WWW.Cryptography, then ensures information from theft or change, as well as

ensures information from theft or change, as well as be utilized for client confirmation. There are, when all is said in done, three sorts of cryptographic plans ordinarily used to achieve these objectives: mystery key (or symmetric) cryptography, open-key (or unbalanced) cryptography, and hash works, each of which is depicted beneath. In all instances, the introductory decodedinformation is alluded to as plain-text. It is encoded into figure content, which will thus (ordinarily) be decoded into utilizable plain-text.

Types of Cryptographic Algorithms

There are numerous ways of categorizing cryptographic algorithms. For commitments to this thesis, they will be classified based on the number of keys that are engaged for encryption and decryption, and further demarcated by their application and use. The three kinds of algorithms that is conferred are given below in fig 1

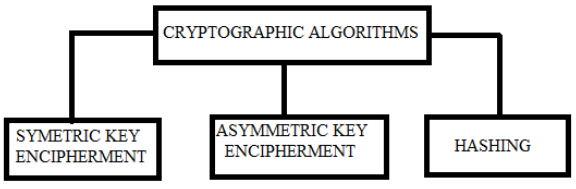

Figure 1: Types of Cryptographic Algorithms

 ISSN: 2321-8169
100 – 106

SYMMETRIC KEY ENCIPHERMENT

Encryption takes as input some plaintext (a stream of data) and a key (a small piece of secret data) and outputs some cipher text. Decryption is the reverse operation: its input consists of the cipher text and a key, and its output is the original plaintext. The idea is that the plaintext cannot be recovered from the cipher text without knowing the correct decryption key. This property is called secrecy. It allows you to send the cipher text through an insecure channel or to store it on an insecure file system.

Authentication takes as input a message and a key and outputs a so-called Message Authentication Code (MAC). A MAC can be seen as a key-dependent checksum. The idea is that nobody can generate a valid MAC for a message without knowing the key. So if message and MAC are sent through an insecure channel then the receiver - given that he knows the key - can verify that the text has not been tampered with (integrity) and that it originatesfrom someone knowing the key (authentication). MACs are also useful if you want to make sure that no intruder (in particular no virus) can alter a file on your system. (If ordinary checksums were used the intruder could change the file and then simply update corresponding checksum.)The security of a cryptographic system depends heavily on the strength of its keys. If an attacker can obtain your keys he can decrypt your messages or fake MACs no matter how good the encryption and authentication algorithms are. In order to support the user LEDA provides secure methods to generate a key from a human-readable passphrase. Of course, the user has the responsibility to choose a good passphrase, i.e. a phrase that cannot be guessed easily by an attacker.

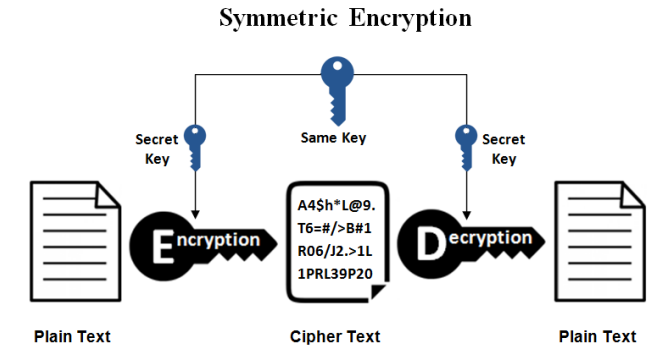

Figure 2: Symmetric Encryption

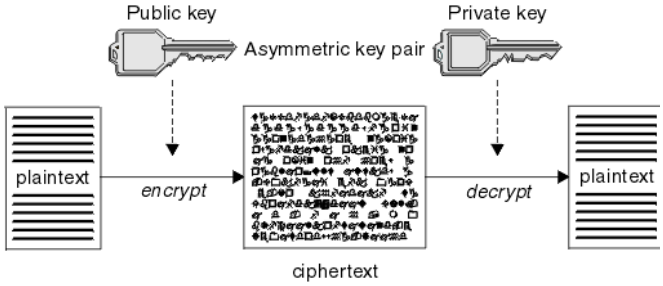

Figure 3: Asymmetric Encryption

Hashing

The hashing algorithm is called the hash function—probably the term is derived from the idea that the resulting hash value can be thought of as a "mixed up" version of the represented value. In addition to faster data retrieval, hashing is also used to encrypt and decrypt digital signatures (used to authenticate message senders and receivers). The digital signatureis transformed with the hash function and then both the hashed value (known as a message-digest) and the signature are sent in separate transmissions to the receiver. Using the same hash function as the sender, the receiver derives amessage-digest from the signature and compares it with the message-digest it also received.

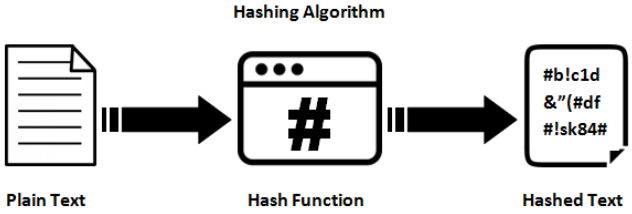

Figure 4: Hashing

II. LITERATRURE SURVEY

An FPGA-based performance comparison of 64-bit block ciphers (Triple-DES, IDEA, CAST128, MISTY1, and KHAZAD) is given in this paper. Two basic architectures are implemented for each cipher. For the non-feedback cipher modes, the pipelined technique between the rounds is used, and the achieved throughput ranges from 3.0 Gbps for IDEA to 6.9 Gbps for Triple-DES. For feedback ciphers modes, the basic iterative architecture is considered and the achieved throughput ranges from 115 Mbps for Triple-DES to 462 Mbps for KHAZAD. The throughput, throughput per slice, latency, and area requirement results are provided for all the cipher implementations.Time performance and area requirements results for 64-bit block ciphers (Triple-DES, IDEA, CAST-128, MISTY1, and KHAZAD) hardware implementations are presented in this paper. Two architectures for each cipher are implemented. For the non-feedback implementations Triple-DES and KHAZAD achieve the best performance and meet better the FPGA characteristics. For the feedback implementations KHAZAD and IDEA appears to have better performance. Triple-DES has the highest latency. .[2] This paper presents a fast and compact FPGA based implementation of the Data Encryption Standard (DES) and Triple Data Encryption Standard (TDES) algorithm, widely used in cryptography for securing the Internet traffic. The main objective of this paper is to provide the reader with a deep insight of the theory and design of a digital cryptographic circuit, which was implemented in a Vertex 5 series (XCVLX5110T) target device with the use of VHDL as the hardware description language. In order to confirm the expected behavior of these algorithms, the proposed design was extensively simulated, synthesized for different FPGA devices both in Spartan and Virtex series from Xilinx viz.

Spartan 3, Spartan 3AN, Virtex 5, Virtex E device families. The novelty and contribution of this work is in three folds: (i) Extensive simulation and synthesis of the proposed design targeted for various FPGA devices, (ii) Complete hardware implementation of encryption and decryption algorithms onto Virtex 5 series device (XCVLX5110T) based FPGA boards and, (iii) Generation of ICON and VIO core for the design and on chip verification and analyzing using Chipscope Pro..[3] The paper presents a suspicious email detection System which detect suspicious activities. In the paper we proposed the use of cryptography strategies for terrorists email detection. Security plays a very important and crucial role in the field of Internet and for email communication. So there is a need of suspicious email detection system which detects all suspicious activities. The need for Suspicious email detection System is increasing due to the rapid usage of Email communication in the Internet world. Triple Data encryption standard (DES) is a private key cryptography system that provides the security in communication system. By using an Enhanced DES algorithm the security has been improved which is very crucial in the communication and field of Internet.[5]This paper presents the design and the implementation of the Triple- Data Encryption Standard (DES) algorithm. A Hardware Triple-DES cryptographic algorithm has been implemented using a Field Programmable Gate Array (FPGA) chip. In order to confirm the expected behavior of the Triple-DES circuitry, the implemented design was extensively simulated and analyzed. The Simulations were run under various clock frequencies. [6] This article discusses the state of the art of cryptographic algorithms as deployed for securing computing networks. While it has been argued that the design of efficient cryptographic algorithms is the “easy” part of securing a large scale network, it seems that very often security problems are identified in algorithms and their implementations. This article discusses the state of the art for a broad range of cryptographic algorithms that are used in networking applications.

III. PROPOSED SYSTEM

The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).DES was published as FIPS 46 in the Federal Register in January 1977. NIST, however, defines DES as the standard for use in unclassified applications. DES has been the most widely used symmetric-key block cipher since its publication. NIST later issued a new standard (FIPS 46-3) that recommends the use of triple DES (repeated DES cipher three times) for future applications.

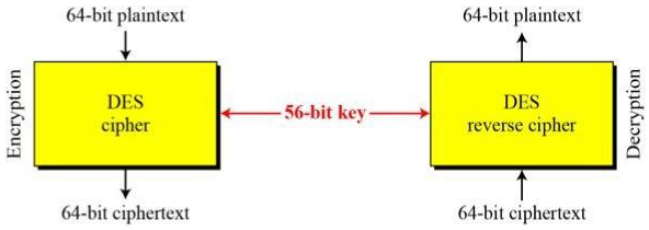

Figure 5: Encryption and Decryption With DES
DES Structure

The encryption process is made of two permutations (P-boxes), which we call initial and final permutations, and sixteen Festalrounds. Each round uses a different 48-bit round key generated from the cipher key according to a predefined algorithm described later in the chapter. Figure 6.2 shows the elements of DES cipher at the encryption site.

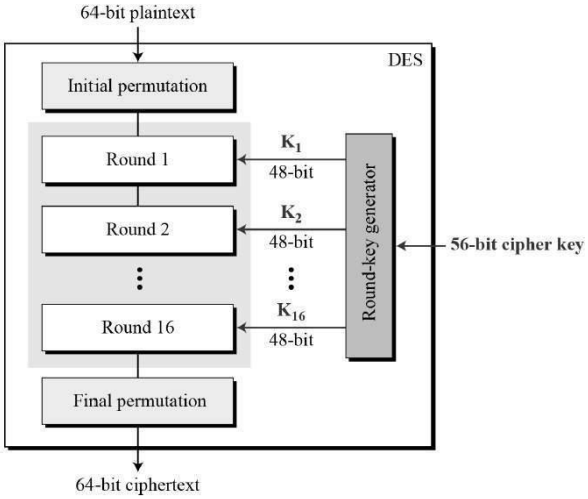

Figure 6: General Structure Of DES

Initial and Final Permutations

Each of these permutations takes a 64-bit input and permutes them according to a predefined rule. We have shown only a few input ports and the corresponding output ports. These permutations are keyless straight permutations that are the inverse of each other. For example, in the initial permutation, the 58th bit in the input becomes the first bit in the output. Similarly, in the final permutation, the first bit in the input becomes the 58th bit in the output. In other words, if the rounds between these two permutations do not exist, the 58th bit entering the initial permutation is the same as the 58th bit leaving the final permutation.

The permutation rules for these P-boxes are shown in Table 6.1. Each side of the table can be thought of as a 64-element array. Note that, as with any permutation table we have discussed so far, the value of each element defines the input port number, and the order (index) of the element defines the output port number.

Rounds

The round takes LI−1 and RI−1 from previous round (or the initial permutation box) and creates LI and RI, which go to the next round (or fi nal permutation box). We can assume that each round has two cipher elements (mixer and swapper). Each of these elements is invertible. The swapper is obviously invertible. It swaps the left half of the text with the right half. The mixer is invertible because of the XOR operation. All

 ISSN: 2321-8169
100 – 106

noninvertible elements are collected inside the function f (RI−1, KI).

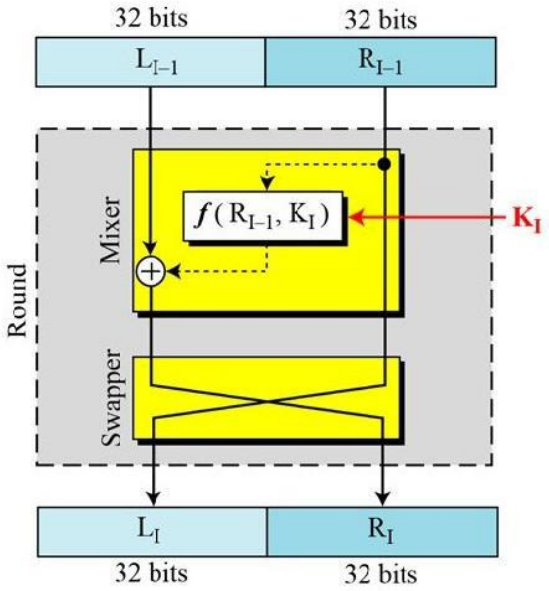

Figure 7: ROUND IN DES

Expansion Permutation

Since RI−1 is a 32-bit input and KI is a 48-bit key, we fi rst need to expand RI−1 to 48 bits. RI−1 is divided into 8 4-bit sections. Each 4-bitsection is then expanded to 6 bits. This expansion permutation follows a predetermined rule. For each section, input bits 1, 2, 3, and 4 are copied to output bits 2, 3, 4, and 5, respectively. Output bit 1 comes from bit 4 of the previous section; output bit 6 comes from bit 1 of the next section.Although the relationship between the input and output can be defined mathematically, DES uses Table to define this D-box. Note that the number of output ports is 48, but the value range is only 1to 32. Some of the inputs go to more than one output. For example, the value of input bit 5 becomesthe value of output bits 6 and 8.

Whitener (XOR)

After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key. Note that both the right section and the key are 48-bits in length. Also note that the round key is used only in this operation.

S-Boxes

The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. The 48-bit data from the second operation is divided into eight 6-bit chunks, and each chunk is fed into a box. The result of each box is a 4-bit chunk; when these are combined the result is a 32-bit text. The substitution in each box follows a pre-determined rule based on a 4-row by 16-column table. The combination of bits 1 and 6 of the input defines one of four rows; the combination of bits 2 through 5 defines one of the sixteen columns.

Cipher and Reverse Cipher

Using mixers and swappers, we can create the cipher and reverse cipher, each having 16 rounds. The cipher is used at the encryption site; the reverse cipher is used at the decryption site. The whole idea is to make the cipher and the reverse cipher algorithms similar. To achieve this goal, one approach is to make the last round (round 16) different from the others; it has only a mixer and no swapper.

Key Generation

The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. However, the cipher key is normally given as a 64-bit key in which 8 extra bits are the parity bits, which are dropped before the actual key-generation process.

Compression P-BOX

The pre-processor before key expansion is a compression transposition step that we call parity bit drop. It drops the parity bits (bits 8, 16, 24, 32… 64) from the 64-bit key and permutes the rest of the bits according to Table 6.12. The remaining 56-bit value is the actual cipher key which is used to generate round keys. The parity drop step (a compression D-box).

Shift Left

After the straight permutation, the key is divided into two 28-bit parts. Each part is shifted left (circular shift) one or two bits. In rounds 1, 2, 9, and 16, shifting is one bit; in the other rounds, it is two bits. The two parts are then combined to form a 56-bit part. Table 3.5 shows the number of shifts for each round.

Compression D-BOX

The compression D-box changes the 58 bits to 48 bits, which are used as a key for a round.

Triple DES Algorithm

The original DES cipher's key size of 56 bits was generally sufficient when that algorithm was designed, but the availability of increasing computational power made brute-force attacks feasible. Triple DES provides a relatively simple method of increasing the key size of DES to protect against such attacks, without the need to design a completely new block cipher algorithm.

Keying Options

The standards define three keying options:

Keying option 1

All three keys are independent. Sometimes known as 3TDEA or triple-length keys. This is the strongest, with 3 × 56 = 168 independent key bits. It is still vulnerable to met-in-the-middle attack, but the attack requires 22 × 56 steps.

Keying option 2

K1 and K2 are independent, and K3 = K1. Sometimes known a S2 TDEA or double-length keys. This provides a shorter key length of 112 bits and a reasonable compromise between DES and Keying option 1.

Keying option 3

All three keys are identical, i.e. K1 = K2 = K3. This is backward compatible with DES, since two operations cancel out. ISO/IEC 18033-3 never allowed this option, and NIST no longer allows it. Each DES key is 8 odd-parity bytes, with 56 bits of key and 8 bits of error-detection. A key bundle requires 24 bytes for option 1, 16 for option 2, or 8 for option 3

Encryption of More Than One Block

As with all block ciphers, encryption and decryption of multiple blocks of data may be performed using a variety of modes of operation, which can generally be defined independently of the block cipher algorithm. However, ANS X9.52 specifies directly, and NIST SP 800-67 specifies that some modes shall only be used with certain constraints on them that do not necessarily apply to general specifications of those modes.

IV. RESULTS AND DISCUSSIONS

Initial Permutation

This permutation is keyless straight permutation that is the inverse of final permutation. For example, in the initial permutation, the 58th bit in the input becomes the first bit in the output.

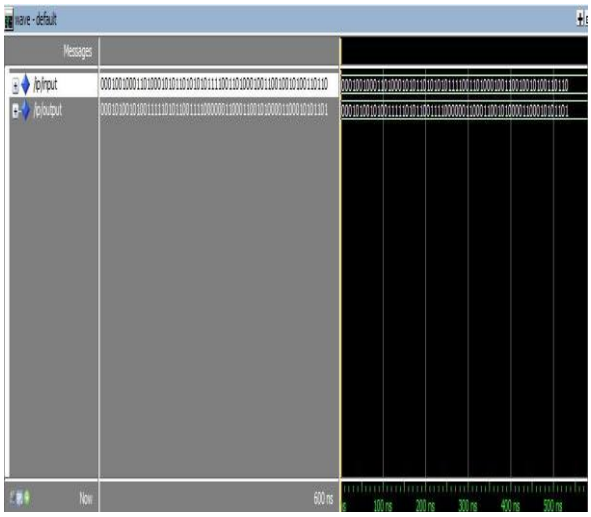

Figure8:Output of Initial Permutation

S BOX

In cryptography, an S-box (substitution-box) is a basic component of symmetric key algorithm which performs substitution. In block ciphers, they are typically used to obscure the relationship between the key and the cipher text — Shannon's property of confusion.

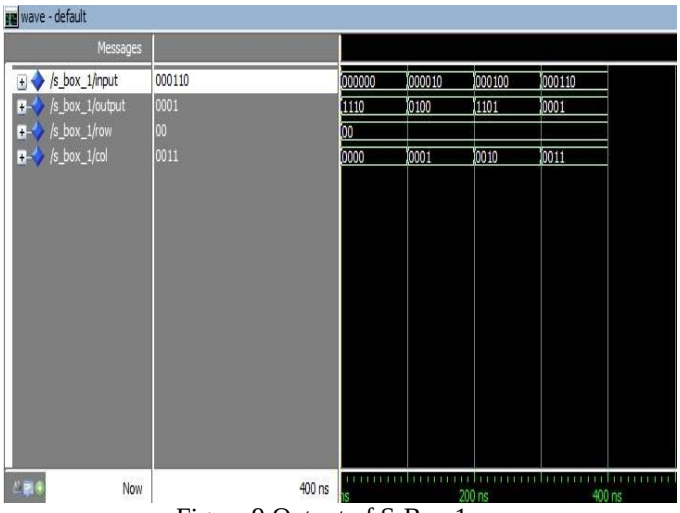

Figure 9:Output of S-Box 1

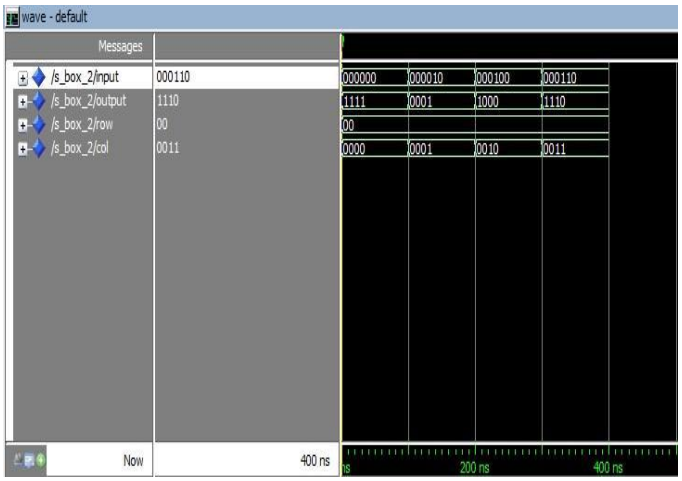

Figure 10: Output of S-Box 2

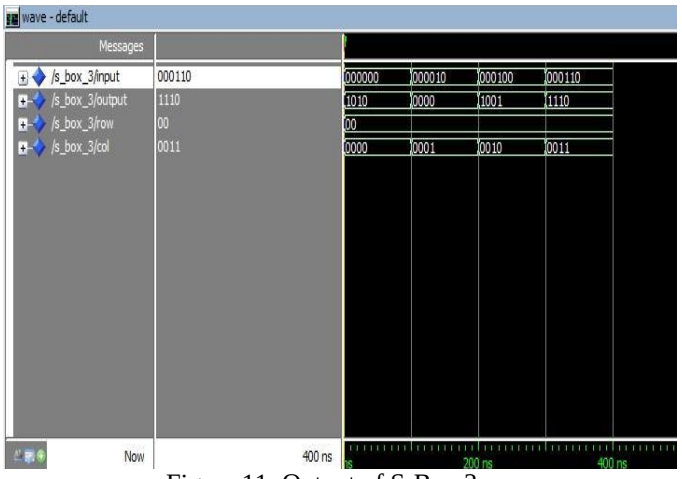

Figure 11: Output of S-Box 3

 ISSN: 2321-8169
100 – 106

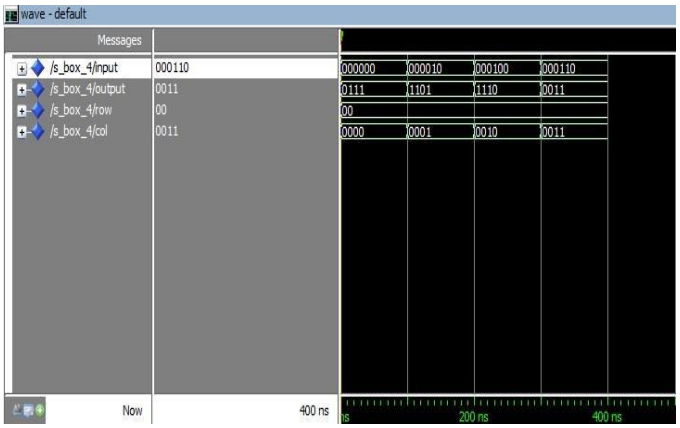

Figure 12: Output of S-Box 4

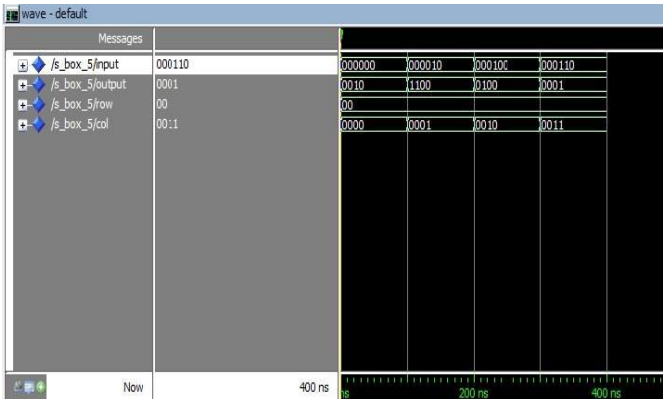

Figure 13: Output of S-Box 5

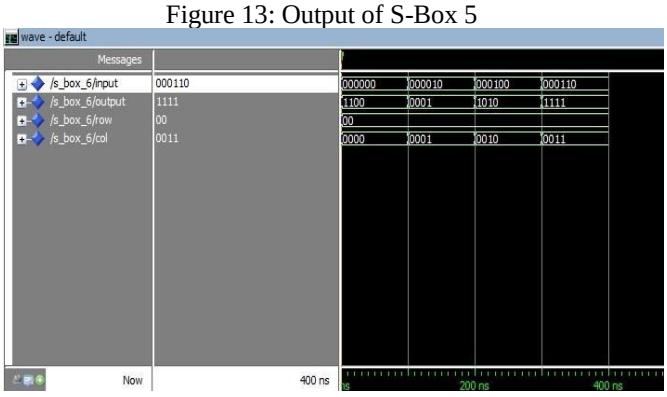

Figure 14: Output of S-Box 6

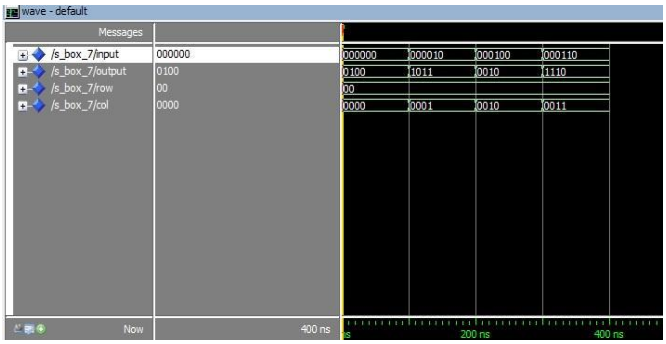

Figure 15: Output of S-Box 7

Key Scheduling

The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. However, the cipher key is normally given as a 64-bit key in which 8 extra bits are the parity bits, which are dropped before the actual key-generation process.

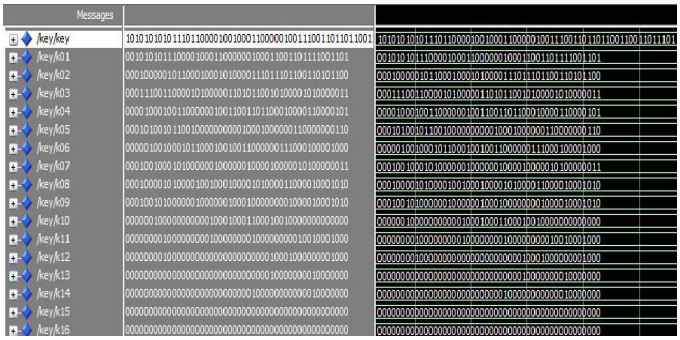

Figure 16: Output of Key Generation

Expansion Permutation

The right 32bits of the plain text gets expanded into 48 bits after passing through the expansion permutation block.

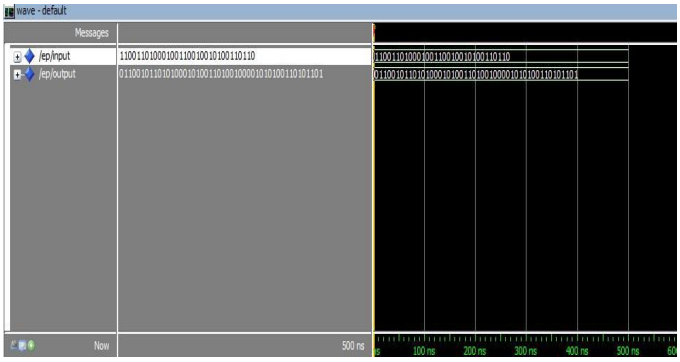

Figure17: Output of Expansion Permutation

Final Permutation

In the final permutation, the first bit in the input becomes the 58th bit in the output.

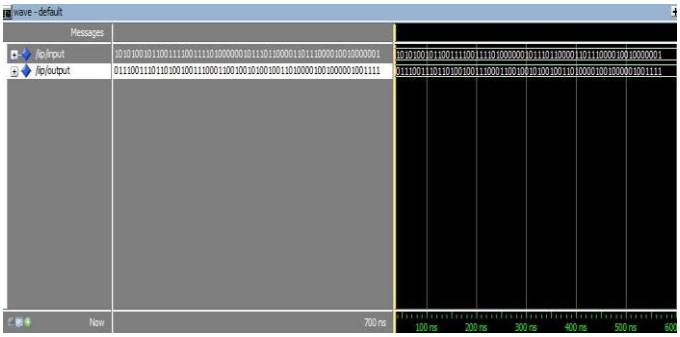

Figure 18: Output of Final Permutation

V. CONCLUSION

The information security can easily be achieved by cryptography algorithm techniques a large number of encryption algorithm have been developed for securing confidential data from the cyberpunks. The aim of current Cryptography is to prevent data from hackers. The strength of

the system is dependent on the length of the key. But to achieve this a large computational time is required, giving a large delay which can be harmful to us. The use of FPGAs can help us to improve this limitation because FPGAs can give enhanced speed. This is due to fact that the hardware implementation of most encryption algorithms can be done on FPGA.The proposed scheme for TDES algorithm has been optimized on the time required to generate the keys or decode data. The algorithm and coding has been implemented on ModelSim software with the help of VHDL language. The synthesis has been done on Xilinx FPGA (Xilinx 9.1e) and the faster clock frequency has been observed in comparison with classical TDES.

The work has been extended in order to increase the security for more severe attacks. The complexity and severity of attacks need a lot of theoretical calculations. There has been seen the scope to further optimize the utilization of resources. The implementation has been further improved so as to get the more efficient usage of the resources and increase in the maximum clockfrequency. Since there are a lot of tradeoffs in practically all the encryption algorithms the major area of research in the future would be to use more than one algorithm combined for a single encryption In the proposed system few gaps have been covered but still a lot of work can be done for the increase in security of the data along with the optimization of resources.

VI. ACKNOWLEDGMENT

The author wish to thank HoD, principal and management of Sri Shakthi Institute of Engineering and Technology, Coimbatore for providing an excellent environment to complete this project in an efficient manner.

VII. REFERENCES

[1] William C. Barker, "Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher", Revised 19 May 2008, NIST Special Publication 800-67, Version 1.1.

[2] He Dr. V. Kamakoti, G. Ananth and U.S. Karthikeyan, "Cryptographic Algorithm Using a Multi-Board FPGA Architecture", Nios II Embedded Processor Design Contest—Outstanding Designs 2005.

[3] Fábio Dacêncio Pereira, Edward David Moreno Ordonez, Rodolfo Barros Chiaramonte , "VLIW Cryptoprocessor: Architecture and Performance in FPGAs", IJCSNS International Journal of Computer Science andNetwork Security, VOL.6 No.8A, August 2006.

[4] Vikram Pasham and Steve Trimberger, “High-Speed DES and Triple DES Encryptor/Decryptor”, Xilinx Application Note: Virtex-E Family and Virtex-II Series, XAPP270 (v1.0) August 03, 2001

[5] Amit Dhir , "Data Encryption using DES/Triple-DES Functionality in Spartan-II FPGAs", White Paper: Spartan- II FPGAs, WP115 (v1.0) March 9, 2000

[6] D. Stinson. “Cryptography: Theory and Practice”, 2nd Edition, Chapman and Hall/CRC, 2002

[7] Toby Schaffer, Member, Alan Glaser, Member, and Paul D. Franzon, "Chip-Package Co-Implementation of a Triple DES Processor", IEEE Transactions on Advanced Packaging, Vol. 27, No. 1, February 2004.

[8] Paris Kitsos, Nicolas Sklavos, Michalis D. Galanis and Odysseas Koufopavlou, “An FPGA-Based Performance Comparison Of The 64-Bit Block Ciphers”, Fifth International Symposium on Intelligent Automation and Control Seville, Spain June 28th-July 1st, 2004.

[9] Andrew S. Tanenbaum, “Computer Networks”, 2003.

[10] http://www.tropsoft.com/