Optimized Hardware Implementation of Enhanced TRIPLE-DES Using Cluster LUT and Pipelining on SPARTEN FPGA

Amany Sarhan
Faculty of Engineering, Tanta University, Egypt

Marwa Fayez
Faculty of Engineering
Mansoura University, Egypt

Noha Hussen
Faculty of Engineering
Mansoura University, Egypt

ABSTRACT
Due to the rapid use of Internet technology, a need for security mechanisms has appeared to protect the information. Cryptography is one of the most effective techniques used to protect and secure information. Triple Data encryption standard is a cryptography system that provides security in the commercial enterprise. A lot of research has been made over DES and Triple DES algorithms to enhance their performance. In this paper, an Enhanced Triple-DES Algorithm based on Cluster LUT (Look Up Table) and Pipelining (ETDCP) is proposed, as a modification of the Triple DES. ETDCP algorithm uses Cluster LUT in hardware implementation and uses the large embedded memories available in the SPARTAN-E FPGA as hardware designed to obtain the minimum utilized resource. Using cluster LUT diminishes the consumption power by reducing the number of registers and slice/area, which decreases the number of logic utilizations used for Spartan Xilinx FPGA. In addition, ETDCP uses pipelining techniques which will increase the processing rate. The experimental results are based on simulated and synthesized (Xilinx Spartan–E) using ModelSim 6.5 and VHDL code. The results show high throughput/area FPGA implementation. The simulation result also proves that the proposed FPGA implementation of ETDCP algorithm has better speed performance compared to previous implementations of cryptographic algorithms.

General Terms
Security, Cryptography Algorithms., and FPGA implementation

Keywords
Cryptography, DES, FPGA, Spartan-E, Custer LUT, Pipelining

1. INTRODUCTION
In a recent study [1], statistic gives information on the total number of the worldwide Internet users in the last 5 years, as of the most recent reported period, the number of internet users was 3.17 billion, up from 2.94 billion. With this growth in the internet world, data privacy and authentication is one of the most important objectives of a communication system to help the users in getting access to information.

Encryption is the central tool for achieving the data security. Encryption algorithms are typically implemented in one or two ways or both: (1) custom software (2) hardware implementation [1]. Encryption algorithms are implemented easily in software, but it is difficult to be implemented in hardware. On the other hand, hardware solutions can build a software algorithm on hardware devices, which can accelerate its execution. For this reason, FPGA (Field Programmable Array) devices are a highly promising alternative for implementing encryption techniques to achieve superior performance; performance is the most important factor for FPGA designs [2].

The FPGA can be reprogrammed at runtime to support different algorithms. Several modern algorithms were implemented on FPGA's devices such as DES and Tripe-DES [3]. With the evolution of the encryption, it was noted that the process of implementing the algorithm in FPGA only is not enough to improve the efficiency; this became the spotlight on FPGA pipeline architecture and the components necessary to another trend for scientific research in this area. Gate arrays are the main components of the FPGA which are the seeds of programmable logic blocks and programmable routings such as area, resources, data storage elements and the Lookup Table (LUT). The choice of logic blocks is made out of clusters, LUT and flip-flops used in the implementation strongly influence an FPGA's speed and density, which plays a major role in the implementation of secured processing.

The motivation of the work presented in this paper is to introduce an enhancement of the Triple-DES Algorithm (ETDCP) and to implement it as the embedded system on SPARTAN-E FPGA based on Cluster LUT and the Pipelined architecture to parallelize the execution of the stages in order to speed up the whole process. This will enable the application of such implementations as a stand-alone encryption facility to be plugged into any communication system, especially for limited power devices.

The paper is designed as follows. In section 2, the overview and the background of cryptography algorithms and an overview of recent work that tried to enhance the implementation of DES and TDES algorithms are described. Section 3 demonstrates the proposed Enhanced Triple-DES Algorithm based on the Cluster LUT and the Pipelining (ETDCP). The ETDCP pipelined implementation is illustrated in section 4. In section 5, the Cluster LUT's effect on the proposed ETDCP is shown. Section 6 conveys a display of the experimental stimulation assessment and the results gotten subsequently on executing the proposed ETDCP algorithm on SPARTAN-E FPGA using Modelsim6.5. Finally, in section 7 a summary about the contribution of the paper and the proposed topics in the future are given.

2. BACKGROUND AND PREVIOUS RESEARCH
2.1 Related work
Among the diverse encryption algorithms, the most famous algorithm that has been implemented on FPGA in the field of symmetric cipher is the Data Encryption Standard (DES)

algorithm which was produced in the 1970s at IBM. However, DES utilizes one short keyspace (size 265 bits). As a result of the evolution of technology, the key can be predicted in 22 hours with the combined help of 100,000 personal computers joined in a network. A lot of research and development have been made over DES and Triple DES algorithms to enhance their performance. DES is popular because it is fast. The design of pipelined 16 DES rounds; Paterson [4] was able to give a key subordinate information for encryption applied in an FPGA which achieved a bit stream of about 12Gbps.

F-X. Standaert and G. Rouvroy [5] presented an implementation of DES using Boolean masking (XORed with random Boolean values). The masking makes the attack more difficult but has not been able to give any perfect security. DES implementation is also downloadable as were announced by Xilinx Company.

A pipelined implementation of DES was introduced in [6]. The platform in [7] used is Virtex-6 FPGA. The implemented design`s test result shows the possibility of creating data in 16 clock cycles when utilizing the non-pipelined approach. When the pipelined approach is utilized, the first phase only needs essential 17 clock signals, and one clock signal is sufficient for every data cycle. In this design, the throughput was expanded from 4.8 to 18.82 Gbps. However, they mix clock frequency from 1201.923 MHz to 294.031 MHz. The throughput of this pipelined design is more than that of Xilinx which is 15.1 Gbps.

On Spartan –II devices, DES and Triple-DES are already implemented as in [8-9]. The work introduced in [14] presents a speedup factor acquired for the TDES implementation algorithm which, when utilized, requires a reconfigurable functional unit of ADAPTO with a RISC microprocessor (the Altera NIOS- II soft processor).

The components of ADAPTO are described in VHDL (VHSIC Hardware Description Language), it has been implemented on an Altera-Stratix II FPGA and unified with the Nios soft processor by the Custom Logic feature. The measurement of the speedup factor related to the introduction of the reconfigurable hardware accelerator is the major objective. The advanced Encryption Standard (AES) is another famous encryption technique that had attracted the attention to be implemented in FPGA [10]. It was optimized and synthesized as a VHDL code to be implemented by both the decryption and the encryption process with 128-bit data. Xilinx ISE 10.1 software is utilized for simulation. Each program has been tested by trial vectors provided by NIST and output results achieved the least amount of delay.

FPGA implementation of Xilinx Synthesis Tool on Vertex II pro kit Show synthesis results, also it provides computation time for generating the cipher text by AES with 4 S-box and 2 dual ports RAM is 6.922 ns. A triple hill cipher algorithm was introduced and implemented by FPGA [11]. The main goal of this research was to make the algorithm more robust by using three stages of a modified hill cipher; every stage is considered a block cipher with a block length of 128 bits and key length of 256 bits, the plain text to be encrypted is processed by this block cipher in three stages to increase the security. The Arbitrary number generator creates a key which is used for the encryption process. FPGA implementation of the proposed algorithm was accomplished on Space-Grade Virtex-4QV XQR4VSX55-10CF1140 using the Xilinx ISE Design Suite 13.2 as a synthesis tool.

Blowfish algorithm is implemented on FPGA using VHDL programming language [12,13]. The blowfish algorithm is analyzed by ascertaining certain measurements in order to achieve better performance such as encryption time, throughput, avalanche effect, and security. To achieve these results, the research was based on multiple testing scenarios for system reliability. The analysis showed that blowfish algorithm provided performance when implemented in FPGA and shows an alternative algorithm to propose as network security on the internet application.

System on chip (SoC) was used to achieve the implementation of pipelined architecture of a high-speed network security processor (NSP) utilized SSL/TLS protocol. For securing the hardware platform running the application such as e-commerce, virtual private network (VPN) and in other fields that require data confidentiality. In this implementation, Secure Digital (SD) card stored all encryption, hashing, and key exchange algorithms in terms of bit files, not as most current work where they are implemented in hardware [14]. The SOC provides the implemented security algorithms. It also provides the Ethernet communication interface. The cipher is chosen using a preferential Efficient System Index (ESI) based algorithm comprising of throughput, power, and resource for the user. Then, these files are downloaded to the Field Programmable Gate Array (FPGA). The limited reconfiguration feature of the ISE14.4 suite with ZYNQ 7z020-clg484 FPGA platform is introduced by this design. They proved that power throughput and resource of the implemented crypto algorithms are better than the existing works [15].

A pipelined architecture for the implementation of axis parallel binary Decision Tree Classification (DTC) is presented in [16]. This implementation improves the execution time of the algorithm consuming minimal area. They used parallel nodes that are able to individually process data from a streaming source. Each engine processes the data in a pipelined technique and is used by resources more efficiently and increase the throughput. Their implementation shows that the result is 3.5 times faster than the current hardware implementation.

An implementation of the Triple-DES algorithm in a sequential fashion using “loops” in softwares was proposed in [17]. Rouvery et el. [17] implemented the sequential DES and Triple-DES where the key and the mode (encryption/decryption) can be changed on a cycle-by-cycle basis with no dead cycles. The design depended on changing the DES mathematical round in order to reduce the number of resources used in VERTIX-II implementation is based on slices, therefore, reduce the LUTs to obtain the minimum number of blocks to regroup all the logical operations that take 4-bit input and give 1-bit output. Their triple-DES design is based on pipelining and uses 604 slices and produces a throughput of 917 Mbps. The disadvantage is a low speed since the same hardware is used iteratively until the computation is completed. Another disadvantage of the design is that all permutation and expansion operations do not require additional LUT, but only wire critical path. This means, if the critical path a LUT is limited, it will encounter some critical problems with the routing part of the control signal.

CAST core [18] presented the design of Triple-DES using the pipeline technique with Low Gate version. It was implemented to minimize gate count or FPGA resources. The disadvantage of this design is that it does not use any memories such as SRAM or LUT. This is contrary to the goal of increasing the performance because the propagation delay of this high signal, will approximately correspond to one or more LUT delay.

Therefore, at least one LUT must be used to limit the control signals and confused routing.

Chodowiec et al. [19] proposed a triple-DES design with a bit stream of 91 Mbps with less resource, utilizing Virtex resource. For commercial Xilinx IP cores, this paper is also targeted to implementing a full IPSec cryptographic transformation in reconfigurable hardware. The experimental test shows the procedure in which the total encryption and decryption throughput of Triple DES in excess of 1 Gbit/s can be achieved using a single FPGA device such as Virtex. Only up to 80% of the resources of this single FPGA device is required by all cryptographic modules, although it was considered a large value compared to the proposal of using pipeline configuration. The throughput in excess of 3 Gbit/s can be accomplished by using two remaining FPGA devices present on the SLAAC-1V accelerator board.

2.2 Basic DES Algorithm

The overall scheme for DES includes two encryption and decryption schemas as illustrated in Figure 1. The plaintext and key are the two inputs for the encryption function. In this initial step, the plaintext and key are 64 bits. DES algorithm begins with an initial permutation (IP) that rearranges the bits to produce the permuted input, encrypts in 16 "rounds" which involves both permutation and substitution functions, followed by dividing the output of IP step into two parts each weigh 32 bits. In each round, the right-side 32 bits of the block are transformed with the function labeled "f" and the key, then XOR'ed with the left-side 32 bits [2]. Let the "64 bits" of the input block to an iteration consist of a 32-bit block L, followed by a 32-bit block R. Let K be a block of 48 bits chosen from the 64-bit key. Then, the output L'R' of iteration with input LR is defined by:

L' = R'

R' = L $\oplus$ f(R, K)

The key for each round is a subset of the original 64-bit key with bits permuted. At each iteration, a different block K of key bits is chosen from the 64-bit key designated by key. Let Kn be a function which takes an integer n in the range from 1 to 16 and a 64-bit block key as input. This yields 48-bit block output Kn, which is a permuted selection of bits from key Kn = KS (n, KEY) permuted.

2.3 Triple Data Encryption Algorithm (TDEA)

The main aim behind the development of TDES is to solve the security problems presented in DES without having to do a complete redesign of a whole new cryptosystem. TDES is today variable and widely used cryptosystem with usage in a number of Internet protocols [8]. TDES modify the key number of DES by applying the algorithm three times in succession with two different keys as shown in Figure 1.

A TDES algorithm utilizes three keys for the cryptographic engine [3]. Let EK(I) and DK (I) represent the DES encryption and decryption of I using DES key K respectively. Each TDEA of the encryption/decryption process is a compound operation of DES. The following processes are used:

1. TDEA encryption process: it performs the transformation of a 64-bit block "I "into a 64-bit block "O" as follows:
   O = EK1 (DK2 (EK1 (I))).

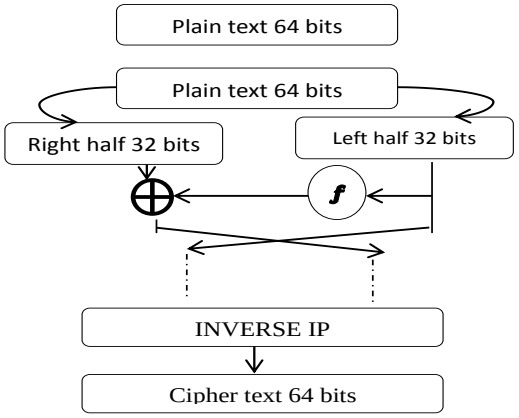


Figure 1: DES Block Diagram

2. TDEA decryption process: it performs the transformation of a 64-bit block I into a 64-bit block O as follows:

   O = Dk2 (Ek1 (Dk2 (I))).

Figure 2 describes the block diagram of Triple-DES.

3. THE PROPOSED ENHANCED TRIPLE-DES ALGORITHM BASED ON CLUSTER LUT AND PIPELINING (ETDCP)

In this paper, an enhancement of the Triple DES algorithm using cluster LUT and pipelining techniques (ETDCP) is proposed, as the modification of the Triple DES. ETDCP offers more features that will enhance the performance of the cryptography for Internet security in a different number of ways:

- Using the Cluster LUT in hardware implementation decreases the number of logic utilizations used on Spartan FPGA, and this will enhance the overall performance.

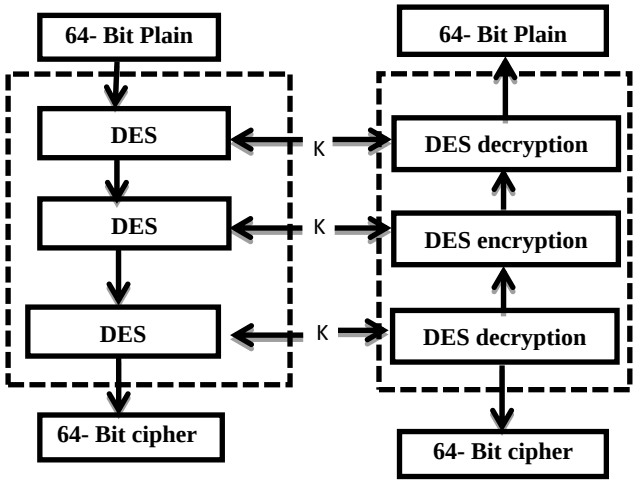


Figure 2: Triple DES Block Diagram

- Using the pipelining technique decreases the number of wired connections, and as a result of this the speed and utilization will increase.
- Using three different keys in the ETDCP increases the confidentiality of the data and helps to save the time of encrypting/decrypting data.

- Using specialized F5MUX multiplexer as a component in Spartan FPGA proved to enhance the performance and density of the FPGA.

ETDCP is an improved mode of DES process in terms of software and hardware implementation. It produces an extra level of security as it receives three different 64-bit keys for three stages of DES, for a whole key length of 192 bits. In the encryption, the entire 192-bits (24 characters) key is entered instead of entering each of the three keys one by one. The ETDCP divides the key into three different sub-keys and performs padding of each key if necessary to be 64 bits long.

The encryption and the decryption of the proposed ETDCP algorithm consist of three iterations of DES, as will be shown in the next subsections. Figure 3 represents the RTL architecture followed by the ETDCP digital implementation.

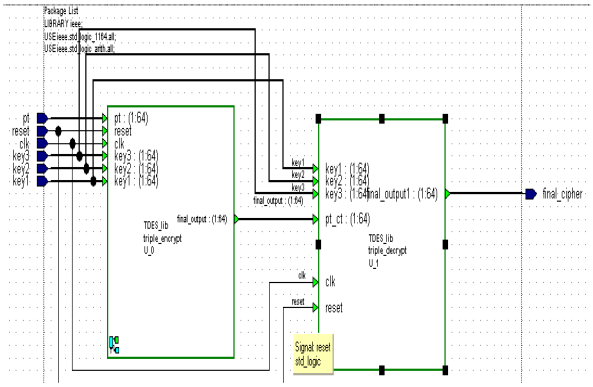


Figure 3: ETDES RTL Block Diagram

3.1 ETDCP Encryption phase

The strategy for encryption is precisely the same as for the normal DES described in section 2.2, but is repeated three times. The 64-bit plain text incorporates and passes through three stages of DES. The first phase is the DES encryption with the first key (K1) for 16 rounds. The second stage is DES Decryption which is a reverse operation of the encryption phase. The data is decrypted with the second key (K2) for 16 rounds in a reverse of the operation. Finally, the data is encrypted by DES Encryption for another 16 rounds using the third key (K3). Thus, in ETDCP the data passes 48 rounds; 24 rounds for encryption and 12 rounds for decryption using three different keys. The steps of the ETDCP encryption phase are shown in figure 4.

3.2 ETDCP Decryption phase

The ETDCP Decryption simply reverses the encryption process. A 64-bit cipher data input passes through 48 rounds; 32 rounds for the decryption and 16 rounds for the encryption with three different keys. It begins with the DES Decryption using the third key (K3), followed by the DES Encryption using the second key (K2), finally the DES Encryption using the first key (K1). The steps of the ETDCP decryption phase are shown in figure 5.

ETDES Encryption Algorithm

The Input:
I: 64 bits of plain text.
KEY: K1, K2, K3
Each key generates 16 sub keys: k1, k2... k16: 16 round keys

Output:
C: 64 bits of cipher text

Algorithm:
Let EK (I) and DK (I) signify the DES encryption and decryption of I using DES key (K), respectively.
Each TDES Encryption process is a compound process of DES encryption.
The transformation of 64-bit block I into a 64-bit block (OE: OPERATION ENCRYPTION) is defined mathematically as follows:
OE(I) = EK1 (DK2 (EK3 (I)))

Figure 4: The steps of ETDCP encryption algorithm

ETDES Decryption Algorithm

Input:
C: 64 bits cipher text.
KEY: K3, K1, K2
Each key generates 16 sub keys: k16, k15, ..., k1: 16 round keys

Output:
64 bit plain text

Algorithm
Let EK (I) and DK (I) represent the DES encryption and decryption of I using DES key (K) respectively.
The transformation of a 64-bit block I into a 64-bit block OD (OPERATION DECRYPTION) is defined as follows:
OD(C) = DK3 (EK2 (DK1 (C)))

Figure 5: The Steps of ETDCP Decryption Algorithm

The ISE foundation will be used to synthesize the ETDCP design and the implementation is via VHDL. The simulation was done by the ISE simulator and the Modalism 9. XE simulator. The Core generation and on-chip verification are performed by Chip scope. Figures 6 and 7 represent the RTL architecture of the ETDCP encryption and the decryption phases.

3.3. Pipelined ETDCP Implementation

Pipelining is an advanced approach to utilizing the possibility to reconfigure the FPGAs components. The ETDCP results that will be shown later proved that pipeline increases the design speed by processing the multiple blocks of data simultaneously. It is achieved by inserting a bank of registers between the combinatorial logic of the implemented circuit. The components of the circuit located between the consecutive registers create pipeline stages. In this study's block ciphers, each round can be considered as a pipeline stage [1].

Using pipelining in the proposed ETDCP is considered an important factor that increases the throughput, computations/ area. The reason for this increase is that pipelining result in running the whole design at a higher frequency which enables it to perform the encryption/ decryption for more data stream. In the proposed ETDCP algorithm, each round of the DES is partitioned and thus pipelined as three stages. First, the data path is separated from the key generation process which will reduce the number of logic levels between subsequent pipeline stages.

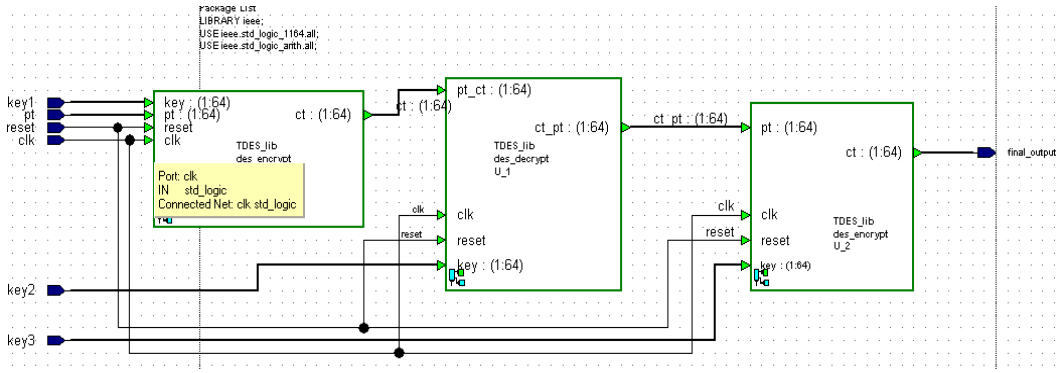


Figure 6: ETDCP RTL Encryption Schematic

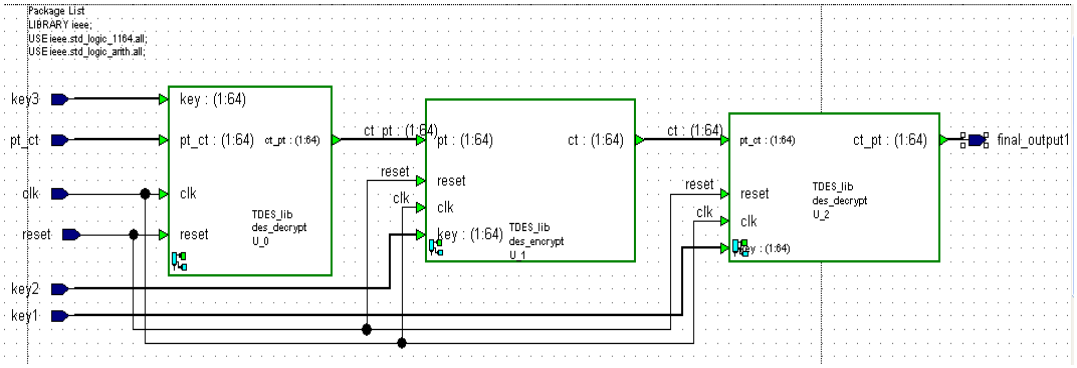


Figure 7: ETDCP Decryption RTL Schematic

When the input accesses the systems, it is encrypted using the encryption key 16 rounds in every stage, a total of 48 rounds. There is a proportional relation between the number of rounds and the number of wires connecting the combinational logic blocks in the Spartan FPGA. This means that when the number of rounds increases, the delay time will increase, which will negatively affect the performance. As a result, the goal is to reduce the number of iterations at every stage of the DES. To decrease the number of wiring between the combinational logic blocks. The pipeline registers are inserted at the end of each DES stage, which results in dividing the path into 48 stages, thus decreasing the area map of the Spartan FPGA. The main advantage of this design is the ability of simultaneously using two or more data-key pairs at the same time during one clock cycle.

Figure 8 gives an overview of the proposed system pipeline architecture. The architecture contains the following lines: the control signals IE (Input Enables), the OE (Output Enable), the Chip Select (CS), the CSCLK (Chip Select Clock) and one indicator (Done).

The pipeline design supports a parallel execution of the three DES stages of the ETDES algorithm. A new data can be entered into the systems as soon as the first stage is done and the data are transferred to the second stage. This results in the whole reduction of the clock cycles and hence a more impact on the performance. Following the various sequences of the procedures used in the proposed architecture process, the hardware pipelines were utilized to overcome the delay caused by the application of three different keys:

a. At the arrival of the clock signal sent by the control unit, all the DES stages are ready to receive input data from the bus. The 64-bit plain text and three different keys are entered into the system where Key1 goes to the DES encryption phase, Key 2 goes to the DES decryption phase and Key 3 goes to the DES decryption phase.

b. The first DES phase, which plays the encryption`s role, is loaded with the first key (Key 1) and X1 (plain text) which is passed through the initial permutation step.

c. The 64 input bits are then split into the left-hand and right-hand parts, each of 32 bits.

d. The output of this phase is the input of the F5MUX which swaps the output of the initial permutation to be entered into the combinational logic Ø.

e. The Spartan combinational logic Ø performs the XOR operation with Key 1 on the two parts of the text producing an output to be stored in the Register 1.

f. The output in the Register 1 will be the input of the Spartan combinational logic ß. The Spartan combinational logic ß combines 64 bits from 32 right and 32 left and performs 8 logic blocks implemented with the S-box. The output is stored in the Register 2.

g. The steps a to f are repeated 16 times.

h. Finally, a final permutation is performed for the bits to produce the final encrypted output. At this point, the control unit gives a clock pulse to pass the output of this stage to the DES decryption stage, and then enters a new plain text to repeat these steps.

The Pseudocode that describes this operation is given in figure 9

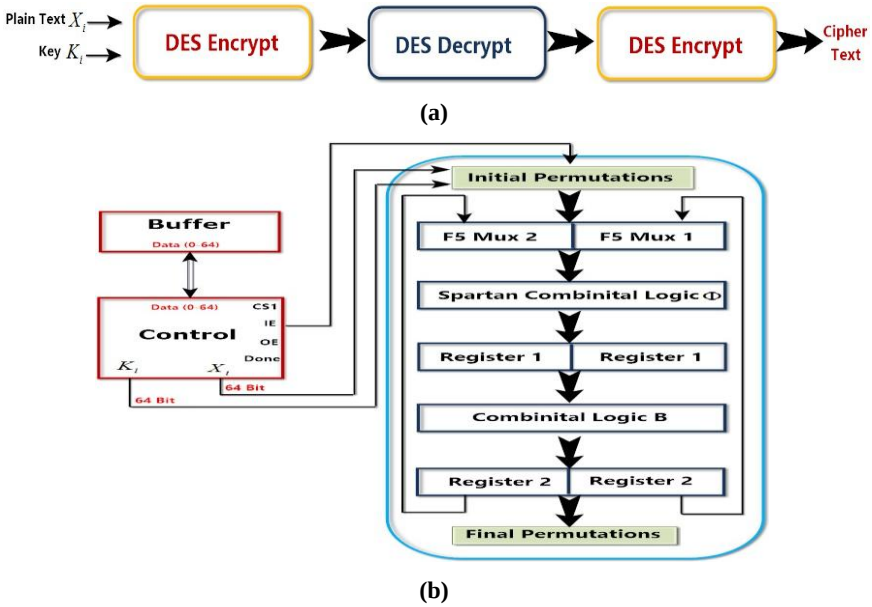


Figure 8: a) Total Steps of ETDCP Architecture, b) Details of One Pipeline Stage

3.4. Using Cluster LUT in ETDCP Implementation

SPARTAN-E FPGA contains both the programmable logic blocks and the programmable routing [16]. As the number of routing between these programmable logic blocks increases, the implementation time increases and as a result, the performance will decrease. Using Cluster LUT in the proposed ETDCP implementation will result in using fewer blocks that will save the mapped area by reducing the connections between the logic blocks. On the other hand, using a large number of inputs will increase the LUT complexity, so, the Cluster LUT was utilized to interconnect a group of several LUTs with local routing to improve the FPGA speed.

The question here is, what is the expected benefit from the use of the cluster LUTs in the ETDCP algorithm? The answer simply is, using the Cluster LUT also prevents the larger logic block unneeded creation of the 48 rounds of the ETDCP algorithm. The Clusters considerably decreases the placement size and routing problem on the Spartan series, in order to include non-linearity into the encryption or decryption scheme. Therefore, to implement the ETDCP sequential circuits, the hold of the design logic blocks should be in mind and the flip-flops, the multiplexer and the distributed memories which utilizes the configuration registers within the Look-Up Tables (LUTs) as general purpose memory cells. One method to improve the performance in the proposed design is to construct larger MUXs using cascade multiple LUTs. This method adds two full levels of logic delays plus an additional routing delay between the cluster LUTs.

In addition, using F5MUX in the ETDCP hardware design enhances the performance and density of FPGA [20]. F5MUX is a function expander; it means that it decreases the number of wiring LUT and the components because if the two LUTs contain independent functions of the same four inputs, the MUX select line becomes the fifth input. In figure 10, F5MUX takes input from two LUTs; one with 3-input LUT and the other is 4-input LUT and it outputs 5-input function. This is an important advantage over the other Spartan FPGA series architectures. Figure 10 shows part of the ETDCP hardware schematic technology using cluster LUT and F5MUX.

The Pseudocode for ETDCP Pipeline encryption

Input: Set of keys {}
Set of{}
Output: Cypher text 64 bits {(),().....()}

```
1.  Xx: For each clock in
2.  {
3.      ( () set
4.      For Each in
5.      {
6.      ()
7.          () in Register1
8.      }
9.          () )
10.         () in Register2
11.         ( () set
12. }
// Decrypt phase input: Cypher text 64 bits{(),().....()}
13. For each clock in
14. {
15. ( () set
16. () )
17.  () in Register2
18.     For Each in
19.     {
20.         ()
21.         () in Register1
22.     }
23.     ( () set
24. }
25. Go to xx
```

Figure 9: Pseudo Code for ETDCP Pipeline Encryption

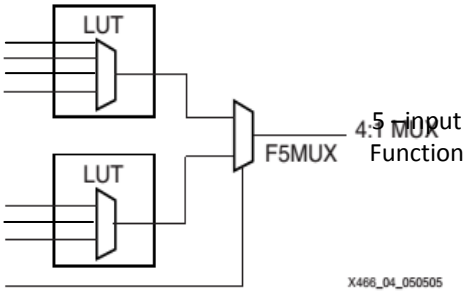


Figure 10: The F5MUX with its output 5-input Function

4. EXPERIMENT EVALUATION

To reach the objective of the evaluation, this section has been divided into three main parts: (1) the evaluation flow, which shows the steps done to evaluate the ETDCP algorithm implementations, (2) the simulation results done for implementing the ETDCP algorithm on (Xilinx Spartan – E) using ModelSim 6.5 and VHDL code and (3) the comparison to the recent cryptographic algorithms.

4.1 ETDCP architecture evaluation flow

The integration of more than a block by using the cluster LUTs will allow the effective finding of a high degree of efficiency. The number of routes and the connection between blocks also need to be reduced. To do so, the best routing for the FPGA architecture, which is as valuable as the logic architecture, must be chosen. Figure 11 illustrates the ETDCP flow on the Spartan FPGA used in these experiments.

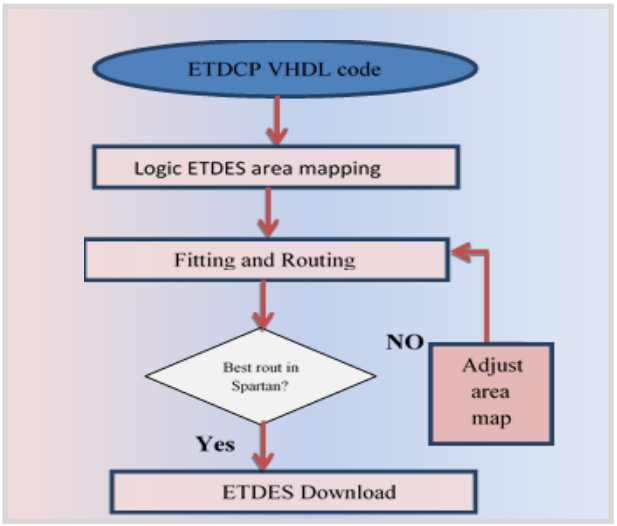
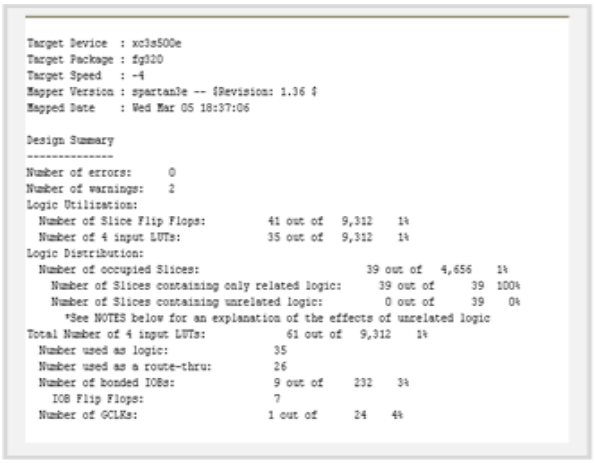


Figure 11: ETDES Architecture Evaluation Flow

The ETDCP VHDL code is translated into logic area mapping technology and independent logic of each encryption and decryption circuits. The optimization for such step is to simplify the logic and remove redundant components. The route between these components is then chosen. A position for each logic block is decided. This minimizes the length of the wires expected to interconnect the hardware. Figure 13 shows the circuit after choosing and optimizing the wire routing.

Figure 12 illustrates the number of resources used in this study's design.



```
Target Device  : xc3s500e
Target Package : fg320
Target Speed   : -4
Mapper Version : spartan3e -- $Revision: 1.36 $
Mapped Date    : Wed Mar 05 18:37:06

Design Summary
--------------
Number of errors:      0
Number of warnings:    2
Logic Utilization:
  Number of Slice Flip Flops:          41 out of   9,312    1%
  Number of 4 input LUTs:              35 out of   9,312    1%
Logic Distribution:
  Number of occupied Slices:                          39 out of   4,656    1%
    Number of Slices containing only related logic:   39 out of      39  100%
    Number of Slices containing unrelated logic:       0 out of      39    0%
      *See NOTES below for an explanation of the effects of unrelated logic
Total Number of 4 input LUTs:          61 out of   9,312    1%
  Number used as logic:                35
  Number used as a route-thru:         26
  Number of bonded IOBs:               9 out of     232    3%
    IOB Flip Flops:                    7
  Number of GCLKs:                     1 out of      24    4%
```

Figure 12: ETDES Area Map

4.2 The Simulation Results of ETDES Algorithm

With the help of MODEL-SIM simulator, the test bench was compiled and stimulated and the test vectors were applied to the proposed design through the test bench. Figure 14 shows the waveform generated for in the encryption of one of the test cases used to validate the algorithm. In this figure, there is 64 bits input (labeled triple_encrypt/PT) and 64 bit output (labeled triple_encrypt/CT) Three different keys of about 64 bits (labeled triple-encrypt/key1, triple-encrypt/keys2, triple-encrypt/key3) were used. This output will be input for the next stage, which is the ETDES decryption as in figure 15.

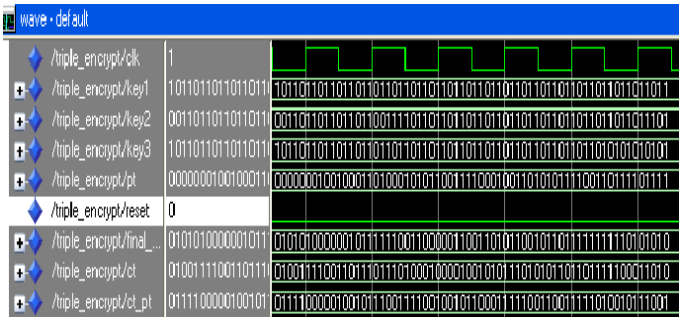


Figure 14: ETDES Encryption Waveform

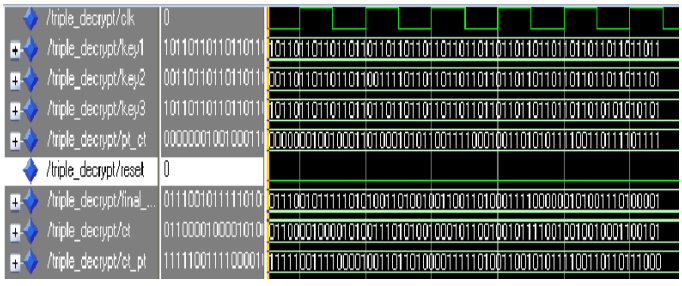


Figure 15: ETDES Decryption Waveform

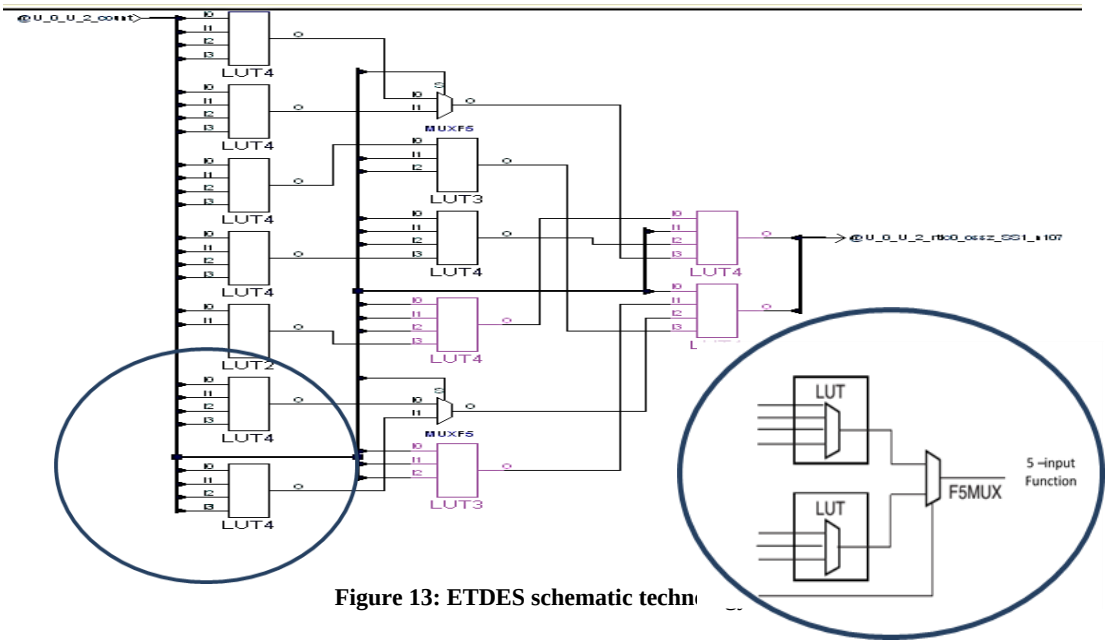


Figure 13: ETDES schematic techn

Figure 16 shows the ETDES encrypt/decrypt waveform, which has 64 bits input (labeled triple_ encrypt_decrypt/PT) and 64 bit output (labeled triple_encrypt_decrypt/final _ cipher). The 64 bit three keys have been given through three ports (labeled triple_encrypt/ key1, triple_encrypt/key2, and triple_encrypt/key3) and the clock has been given through clock input ports. The figure shows that the final output, produced after going through the stages of the encryption and the decryption reviewed previously in section (6), resulted in a match to the original data input and this is the ultimate proof of the success of the ETDES algorithm**.**

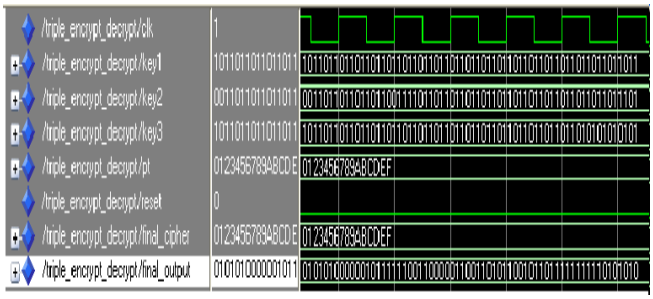


Figure 16: ETDES Encryption/Decryption Waveform

4.3 Comparison results

In this section, a comparison between DES, TDES and ETDCP is presented in Table 1 based on: the cryptography, type, the initial permutation, the number of keys, the key size, the key length, and the number of rounds, delay time, security level, attack time and the speed.

The comparison shows that the proposed architecture of the ETDCP algorithm is more secure than the conventional DES and the TDES. These results have been achieved through increasing the number of keys, key length and the number of rounds. Table 2 shows the difference between Triple-DES, and ETDCP using Cluster LUT utilization for the FPGA different Spartan series. Table 3 shows the ETDCP synthesis report and timing constraints.

Table 1. Comparison between DES, TDES and ETDCP algorithms

Descrption	DES	TDES	ETDCP
Cryptography type	Asymmetric	Asymmetric	Asymmetric
Initial Permutation	PT 64 bit Cipher 64 bit	PT 64 bit Cipher 64 bit	PT 64 bit Cipher 64 bit
Number of keys	One key	Two keys: K1, K2	Three differnet keys: K1,K2,K3
Key Size	64 bits	64 bits	64 bits
Key Length	64 bits	128 bits	192 bits
No of rounds	16 rounds	48 rounds	48 rounds
Delay time	980 ns	1600 ns	67355 ns
Security Level	Insecure	secure	More secure
Attack Time	20 hours	Not yet	Not yet
Speed	Fast	Slow	Slow

Table 3. ETDCP Syntheses on Spartan 3E Report

Speed Grade for ETDCP utilization using Spartan 3E(3S500EFG320)	
Maximum period	**5.628 ns**
Maximum frequency	**1770683 MHZ**
Maximum input arrival time before clock	**7.54**
Maximum output required time before clock	**4.283 ns**
Maximum combinational path delay	**No path delay**
TIMING CONSTRAINS	
Clock period	**5062 ns**
Total number of paths	**67355/6080 ns**

Finally, to evaluate the performance of the ETDCP algorithm, it was compared to ROUVROY [18], CAST [19] and CAJ et al. [20] as shown in Table 4. Three experiments were performed; each experiment has its own goal and settings. To evaluate the experiment`s results, three metrics were used in them: (1) The number of slices used, (2) The Throughput

(number of computations per second (Mbps)), and (3) The Throughput/area (number of computations divided into a number of slices used in FPGA) as shown in figure 17.

Table 4: Comparison with sequential Triple-DES on Vertex-II and our pipelined ETDCP on Spartan implementation

Sequential Triple-DES on Vertex II				Pipelined ETDCP on SPARTAN
	CAST	CAJ et al.	ROUVROY	ETDCP
Number of Slices used	790	614	604	304
Throughput(Mbps)	686	91	917	1033
Throughput/area(Mbps/slice)	0.85	0.15	1.51	3.398

From table 4, it was observed that the proposed implementation decreases the number of slices used due to the pipelining approach and cluster LUTs used. It also shows that the throughput is increased by 50% over the CAST implementation, and by more than 12% over ROUVROY and is almost 10 times the value of CAJ et al. This is due to the advantage of using the pipelining approach which permits new data to be entered while others are being processed. Finally, the throughput/area is increased due to a decrease in the number of slices used and the increase obtained in throughput. As a whole, the comparison results prove that the ETDCP implementation reduces the amount of resources and is able to achieve the goal of increasing the system performance.

5. CONCLUSION

An Enhanced Triple-DES algorithm using the Cluster LUT and pipelining techniques (ETDCP) is proposed in this paper, as a modification of the Triple DES, to enhance the performance of the cryptography for internet security. The ETDCP uses the Cluster LUT in hardware implementation to decrease the number of logic utilizations used in the Spartan FPGA. It also uses three different keys to increase the confidentiality of the data. The ETDCP algorithm implementations are based on using pipelining techniques that decrease the number of wired connections. As a result, this will increase the speed and utilization.

In addition, the F5MUX multiplexer is used in the ETDCP as a component in the Spartan FPGA that will enhance the density of the FPGA. Finally, a preliminary experiment is then conducted, indicating that ETDCP has better results compared to more recent systems. The comparison results showed that the proposed architecture of the ETDCP algorithm is more secure than the conventional DES and the TDES. In addition, the results proved that the ETDCP implementation using the cluster LUTs reduces the amount of resources and is able to achieve the goal of increasing the system`s performance.

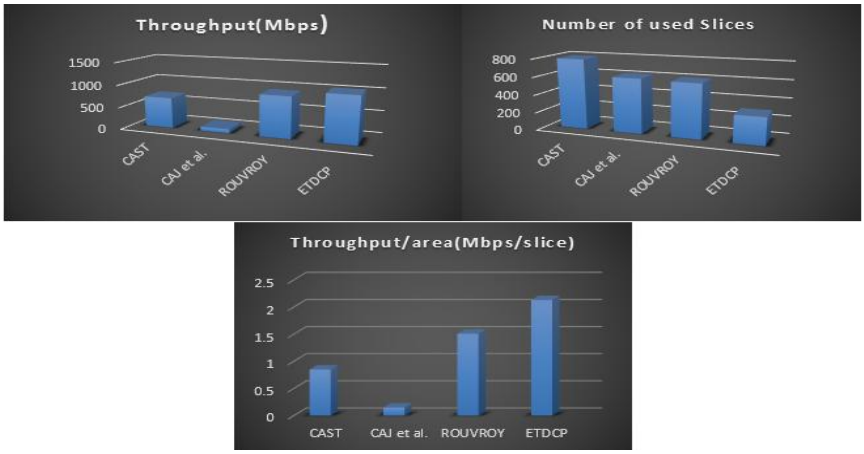


Figure 17: Comparison results evaluating number of slices used, Throughput (Mbps), and Throughput/area by CAST, CAJ et al. and ROUVROY compared to ETDES

Table 2. Comparison between DES, TDES and ETDCP Utilization for FPGA Spartan 3E

	ETDCP utilization for Spartan 3E (3S500EFG320)		TDES utilization for device XC400.package FG320 speed		TDES utilization Spartan 3n	
Factor	USED	UTILIZATION	USED	UTILIZATION	USED	UTILIZATION
Number of slices	39 out of 9312	1%	1583	20%	1622	14%
Number of slices flip flop	41 out 9312	1%	1254	8%	1230	11%
Number of 4-input LUTS	2367 out of 9312	25%	2494	16%	2593	11%
Number of cluster LUTS	958	20%	NON	___	NON	NON
Number of Bounded Iob's	9 out of 232	3%	302	77%	302	37%
Number of GCLK	1 out of 24	4%	1	12%	1	4%

6. REFERENCES

[1] Herman, H.: Pipeline Reconfigurable FPGAs. Journal of VLSI Signal Processing Systems, vol. 24, pp. 129–146, (2000).

[2] Stallings, W.: Cryptography and Network Security: Principles and Practice. 4th Edition, Prentice-Hall, (2006).

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

[4] Patterson, C.: High performance DES encryption in Virtex FPGAs using Jbits. In Proc. of IEEE on Field-Programmable Custom Computing Machines Symposium, Napa Valley, CA, USA, pp. 113 – 121, 17-19 Apr. (2000).

[5] Standaert, F., Rouvroy, G., and Quisquater, J.J.: FPGA Implementations of the DES and TRIPLE-DES Masked Against Power Analysis Attacks. In Proc. of the International Conference on Field Programmable Logic and Applications, FPL '06, Madrid, pp. 1 – 4, 28-30 Aug. (2006).

[6] Schmit, H., Cadambi, S., and Moe, M.: Pipeline Reconfigurable FPGAs. Journal of VLSI Signal Processing Systems, 24, pp. 129–146, (2000).

[7] Pasham, V. and Trimberger, S.: High-Speed DES and Triple DES Encryptor/Decryptor. Xilinx Application Notes, Aug (2001). Available from http://www.xilinx.com/ xapp/xapp270.pdf.

[8] Stinson, D.: Cryptography: Theory and Practice. 3rd Edition, Chapman and Hall/CRC, (2005).

[9] Trimberger, S., Pang, R., and Singh, A.: A 12 Gbps DES Encryptor/Decryptor Core in an FPGA. Lecture Notes on Cryptographic Hardware and Embedded Systems, Springer, Vol. 1965, pp. 156-163, Jan. (2002).

[10] Kaur, A., Bhardwaj, P., and Kumar, N.: FPGA Implementation of Efficient Hardware for the Advanced Encryption Standard. International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol. 2, Issue 3, Feb. (2013).

[11] Khalaf, A., Abd El-karim, M.S., and Hamed, H. F. A.: Triple Hill Cipher Algorithm for Increasing the Security Level of Encrypted Binary Data and its Implementation Using FPGA. In Proc. of the 17th IEEE International Conference on Advanced Communication Technology, (ICACT2015), Seoul, Korea, pp. 454 – 459, July 1-3, (2015).

[12] Wang, K.: An Encrypt and Decrypt Algorithm Implementation on FPGA's. In Proc. of the 5th IEEE International Conference on Semantics, Knowledge and Grid, (SKG 2009), Zhuhai, pp. 298-301, 12-14 Oct. (2009).

[13] Prasetyo, K. N., Purwanto, Y. , and Darlis, D.: An Implementation of Data Encryption For Internet of Things Using Blowfish Algorithm on FPGA. In Proc. of the IEEE 2nd International Conference on Information and Communication Technology (ICoICT), Bandung, pp. 75 – 79, 28-30 May (2014).

[14] Sridevi, S., Himabindu, B., and Alekya, B.: Design of High Performance Pipelined Data Encryption Standard (DES) Using Xilinx Virtex-6 FPGA Technology. The International Journal of Science and Technoledge, 2.2, pp. 53-58, Feb. (2014).

[15] Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., and Re, M.: TDES Cryptography Algorithm Acceleration using a Reconfigurable Functional Unit. In Proc. of the IEEE 21st International Conference on Electronics, Circuits and Systems (ICECS), Marseille, pp. 419 – 422, 7-10 Dec. (2014).

[16] Paul, R., Chakrabarti, A., and Ghosh, R.: Multi Core SSL/TLS Security Processor Architecture and its FPGA Prototype Design with Automated Preferential Algorithm. Microprocessors and Microsystems Journal, Vol. 40, pp. 124–136, Feb. (2016).

[17] Saqib, F., Dutta, A., Ortiz, P., and Pattichis, M.: Pipelined Decision Tree Classification Accelerator Implementation in FPGA (DT-CAIF). IEEE Transactions on Computers, Vol. 64, No. 1, Jan. (2015).

[18] Rouvroy, G., Xavier, F. X., Quisquater, J. J., and Legat, J. J.: Design Strategies and Modified Descriptions to Optimize Cipher FPGA Implementations: Fast and Compact Results for DES and Triple-DES. In Field Programmable Logic and Applications – FPL, LNCS 278, Springer-Verlag Berlin, pp. 181-193, (2003).

[19] CAST. Inc. DES Encryption Core. Available from http://www.cast-inc.com.

[20] Chodowiec, P., Gaj, K., Bellows, P., and Schott., B., Experimental Testing of the Gigabit IPSec-Compliant Implementations of RIJNDAEL and Triple DES Using SLAAC-1V FPGA Accelerator Board. In Proc. of ISC 2001: Information Security Workshop, LNCS 2200, pp.220-234, Springer-Verlag.