International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Hardware Implementation of Biomedical Data Encryption using FPGA

Salma M. Saif[1], Ali E. Taki El_Deen[2], Mohy E. Abo-Elsoud[3]

[1]Electronics and Communications Dept, Mansoura University, Egypt
[2]IEEE senior member, Alexandria University, Egypt
[3]IEEE senior member, Electronics and Communications Dept, Mansoura University, Egypt

Abstract: ***If privacy is outlawed, only outlaws will have privacy. Maintaining privacy in our personal communications is something everyone desires. Without encryption, it would be very easy for sensitive data to be stolen and used malevolently. This paper presents AES encryption algorithms and a comparison between them and other encryption algorithms such as DES, RSA, and Blowfish. It also presents some statistical tests which test the randomness of the used key. A hardware implementation of the AES on the recent Xilinx Spartan-6 FPGA is applied on text and biomedical images. Such device is to speed up the AES algorithm and to reduce logic area. In addition, the comparison between FPGA and Matlab is introduced.***

Keywords: AES, FPGA, Medical image security, Encryption, Decryption.

1. Introduction

The protection of sensitive information against unauthorized access or fraudulent changes has been of prime concern throughout the centuries. Modern communication techniques, using computers connected through networks, make all data even more vulnerable for these threats. Also, new issues have come up that were not relevant before, e.g. how to add a (digital) signature to an electronic document in such a way that the signer can not deny later on that the document was signed by him/her [1].

The security of communications and commerce in a digital age relies on the modern incarnation of the ancient art of codes and ciphers. Underlying the birth of modern cryptography is a great deal of fascinating mathematics, some of which has been developed for cryptographic applications, but much of which is taken from the classical mathematical canon [2].

Cryptography is the study of methods for sending messages in secret (namely, in enciphered or disguised form) so that only the intended recipient can remove the disguise and read the message (or decipher it) [3].

Cryptography plays a crucial role in many aspects of today's world, from internet banking and ecommerce to email and web-based business processes. Understanding the principles on which it is based is an important topic that requires a knowledge of both computational complexity and a range of topics in pure mathematics [4].

This paper is organized as follow: Section (2) covers different types of multimedia data. Section (3) provides a brief discussion of AES encryption algorithm. Section (4) presents a comparison between AES encryption algorithms and other techniques such as DES, RSA, and Blowfish. Section (5) discusses some statistical tests that are used to test the randomness of the used key. Section (6) gives some notes about FPGA and its advantages over ASIC. Section (7) shows the experimental results. Finally the paper is concluded in Section (8).

2. Multimedia Data

i. Text

Inclusion of textual information in multimedia is the basic step towards development of multimedia software [5]. Text data may be taken form keyboard or loaded from any text file.

ii. Image

Another interesting element in multimedia is graphics does not have a single agreed format. They have different format to suit different requirements. The size of a graphic depends on the resolution it is using. A computer image uses pixel or dots on the screen to form itself [5].

2D images may be taken from a digital camera or loaded from Matlab gallery of any image format.

3. AES Encryption Algorithm

In 1997, the same year in which DES was definitely broken by a brute-force attack; NIST announced an initiative to develop a new encryption standard, called the Advanced Encryption Standard (AES), which would replace DES. The selection process was open and the candidates had to meet a series of requirements among which the most important were support for key lengths of 128, 192, and 256 bits and a block size of 128 bits. The evaluation criteria focused on aspects related to security, cost, and implementation characteristics and, after a selection process that took three years to be completed, in October 2000, NIST announced that the algorithm **Rijndael**, designed by Belgian cryptographers Joan Daemen and Vincent Rijmen, would become AES [6].

AES consists of so-called layers. Each layer manipulates all 128 bits of the data path. The data path is also referred to as the state of the algorithm. There are only three different types of layers. Each round, with the exception of the first, consists of all three layers as shown in Fig. 1: the plaintext is denoted as x, the ciphertext as y and the number of rounds as nr. Moreover, the last round nr does not make use of the MixColumn transformation, which makes the encryption and decryption scheme symmetric [7].

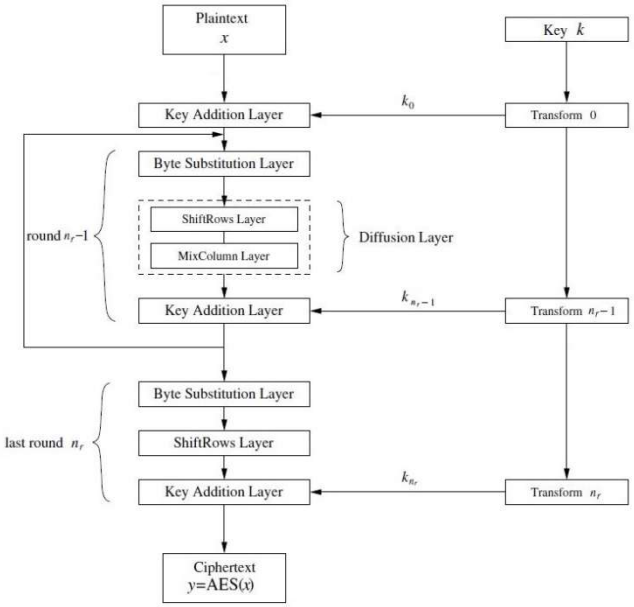


Figure 1: AES block diagram

We now describe the steps in more detail. The 128 input bits are grouped into 16 bytes of 8 bits each, call them $a_{0,0}$, $a_{1,0}$, $a_{2,0}$, $a_{3,0}$, $a_{4,0}$, $a_{0,1}$, $a_{1,1}$, ….., $a_{3,3}$. These are arranged into a 4 x 4 matrix

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$

The Byte Substitution Transformation:
In this step, each of the bytes in the matrix is changed to another byte by a table, called the S-box [8].

The Shiftrow Transformation:
The four rows of the matrix are shifted cyclically to the left by offsets 0, 1, 2, and 3 [8].

The Mixcolumn Transformation:
MixColumn layer is a matrix operation which combines (mixes) blocks of four bytes [7].

The Round Key Addition:
The round key, derived from the key, consists of 128 bits, which are arranged in a 4 x 4 matrix consisting of bytes. This is XORed with the output of the MixColumn step [8].

For decryption, the Byte Substitution layer becomes the Inv Byte Substitution layer, the ShiftRows layer becomes the Inv ShiftRows layer, and the MixColumn layer becomes Inv MixColumn layer. However, as we will see, it turns out that the inverse layer operations are fairly similar to the layer operations used for encryption. In addition, the order of the subkeys is reversed, i.e., we need a reversed key schedule [7].

4. Comparison between AES and Other Algorithms

Table 1 presents a comparison between AES encryption algorithm and DES, RSA, and Blowfish encryption algorithms.

Table 1: Comparison between AES, DES, RSA, and Blowfish

	Key type	*Key size*	*Block size*
AES	Symmetric	128 bits	128 bits, 192 bits, and 256 bits
DES	Symmetric	64 bits (56 bits are actually used)	64 bits
RSA	Asymmetric	Not specified	Not specified
Blowfish	Symmetric	64 bits	From 32 bits to 448 bits

5. Random Number Generation Tests

The NIST Test Suite is a statistical package consisting of 15 tests that were developed to test the randomness of (arbitrarily long) binary sequences produced by either hardware or software based cryptographic random or pseudorandom number generators. These tests focus on a variety of different types of non-randomness that could exist in a sequence [9]. We are going to present four tests; frequency test, serial test, poker test, and run test.

A. Frequency (Monobit) Test:
The purpose of this test is to determine whether the number of 0's and 1's in the sequence are approximately the same, as would be expected for a random sequence. Let n0, n1 denote the number of 0's and 1's in s, respectively. The statistic used is [10]

$$X_1 = \frac{(n_0 - n_1)^2}{n} \qquad (1)$$

B. Serial (Two bit) Test:
The purpose of this test is to determine whether the number of occurrences of 00, 01, 10, and 11 as subsequences of the sequence are approximately the same, as would be expected for a random sequence. Let n_0, n_1 denote the number of 0's and 1's in the sequence, respectively, and let n_{00}, n_{01}, n_{10}, n_{11} denote the number of occurrences of 00, 01, 10, 11 in the sequence, respectively [10].

$$X_2 = \frac{4}{(n-1)}(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2) - \frac{2}{n}(n_0^2 + n_1^2) + 1 \qquad (2)$$

C. Poker Test:
Let m be a positive integer such that $\left[\frac{n}{m} \geq 5.(2^m)\right]$, and let $k = \left[\frac{n}{m}\right]$. Divide the sequence into k non-overlapping parts each of length m, and let n_i be the number of occurrences of the i_{th} type of sequence of length m, $1 \leq i \leq 2^m$. The poker test determines whether the sequences of length m each appear approximately the same number of times in the sequence, as would be expected for a random sequence. The statistic used is [10]:

$$X_3 = \frac{2^m}{k}\left(\sum_{i=1}^{2^m} n_i^2\right) - k \qquad (3)$$

D. Run Test:
The purpose of the runs test is to determine whether the number of runs (of either zeros or ones) of various lengths in the sequence s is as expected for a random sequence. The expected number of gaps (or blocks) of length i in a random sequence of length n is $e_i = (n - i + 3)/2^{i+2}$. Let k be equal to the largest integer i for which $e_i \geq 5$. Let Bi, Gi be the number of blocks and gaps, respectively, of length i in the sequence for each i, $1 \leq i \leq k$. The statistic used is [10]:

$$X_4 = \sum_{i=1}^{k} \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(G_i - e_i)^2}{e_i} \qquad (4)$$

6. Implementation of the AES

FPGA technology has become mature to provide relevant for cryptographic system. The AES is widely used block cipher with a rich implementation literature for both hardware and software. Most AES implementations for reconfigurable devices are based on configurable logic such as look-up tables (LUTs) and flip-flops. Some researchers noted an AES design that takes advantage of the slice structure and 6-input LUTs of the new FPGAs such as Virtex-5 or Spartan-6 without using embedded RAM or DSP blocks.

A hardware implementation of the AES on the recent Xilinx Spartan-6 XC6SLX45 is chosen. The use of the nodes of FPGA-based image processing allows one to satisfy requirements such as low power consumption, small logic area, and reconfigurability of the hardware structure.

VHSIC (Very High Speed Integrated Circuit) Hardware Description Language (VHDL) usage has risen rapidly since its inception and is used by literally tens of thousands of engineers around the globe to create sophisticated electronic products. One of the best uses of VHDL today is to synthesize Application Specific Integrated Circuits (ASIC) and Field Programmable Gate Array (FPGA) devices [12].

With advanced FPGAs, they are preferred over ASIC. They have advantages over ASIC because of faster hardware solution. Also, they have a wider applicability than ASIC.

FPGA design flow provides the use of Xilinx ISE 14.2 suite, starts with design entity using schematic, HDL, or EDIF (Electronic Design Interchange Format). It performs design implementation and verification until it is correct and complete.

7. Experimental Results

AES encryption technique was applied on text, and grey scale biomedical image data using Matlab R2013a on windows7 which its specifications' are: (i) 64-bit operating system, (ii) processor of 2.10GHz, and (iii) 3 GB RAM for software simulation which shown that 256-bits AES given the best result for select technique. The original, encrypted and reconstructed text was shown. The original, encrypted and reconstructed grey scale biomedical image was shown in Fig.5-7 for 128, 192, and 256-bits key AES. The processing time for text encryption and decryption is shown in Fig.3, 4. The processing time for biomedical image encryption and decryption is shown in Fig.8, 9.

Also a hardware description of AES encryption algorithm using FPGA is performed with the aid of hardware description language VHDL. Synthesize of the VHDL code is done using ModelSim SE 6.3 software for simulation and optimization. Implementation (i.e. Translate, Map and Place and Route) of the code is carried out on Xilinx – Project Navigator, ISE 14.2 suite. Xilinx Spartan-6 XC6SLX45-CSG324C is used for hardware evaluation. The usage of hardware devices results in significance elaboration of the design performance. The simulation results are shown in Fig.10, 11.

Why Spartan-6:
Here is a comparison between FPGA Xilinx families.

Table 2: Differences between Xilinx families

	Spartan-3, Virtex-4	Virtex-5, Virtex-6, Spartan-6
No. of slices per 1 CLB slice	4	2
No. of LUTs per each slice	2	4
No. of LUT inputs	4	6
Max. single-port memory size per LUT	16 x 1	64 x 1
Max. shift register size per LUT	16 bits	32 bits

Between 25 and 50% of all slices can also use their LUTs as distributed 64-bit RAM or as 32-bit shift registers (SRL32) or as two SRL16s. From table 2, we can summarize that there are only 2 slices per CLB slice in Spartan-6 each contains 4 6-inputs LUTs while 4 slices per CLB slice are exist in Spartan-3 families each contains 2 4-inputs LUTs and from table 5 the results show that Spartan-6 requires a much smaller number of LUTs for the design than Spartan-3A which lead to the use of less silicon area in Spartan-6, built upon 45 nm triple-oxide technology while Spartan-3 is built upon 90 nm triple-oxide technology. Spartan-6 consumes less power than Spartan-3A, Virtex-4, and Virtex-5 and is higher in performance.

1-Text encryption:
128-bits AES:
Plaintext:
Mohamed Ibrahim El-masry. , Age: 50, Sex: male

Ciphertext:
úÎ8ü;ý ñõøÎ¦ÉïÙRwòk Ä=ewtZËý'u]LÚº-
¬áså-Þ£m²ÅÉ.!~¨þ(Þ£m²ÅÉ.!~¨þ(Þ£m²ÅÉ.!~¨þ(Þ£m²ÅÉ.!~¨þ(
Þ£m²ÅÉ.!~¨þ(
192-bits AES:
Plaintext:
Mohamed Ibrahim El-masry. , Age:50 , Sex: male

Ciphertext:
å£r n¦Û
ßÊÚ Â]yÉÛEq¼½h§ðPÝ ³U¯¼- . f?Ý T A ^è õã(ôömû¼H ° Ù x õã(ôömû¼H ° Ù x õã(ôömû¼H°Ù x õã(ôömû¼H°Ù x õã(ôömû¼H°Ù x

256-bits AES:
Plaintext:
Mohamed Ibrahim El-masry. , Age:50 , Sex: male

Ciphertext:
oËñùT2"ãéaE¿Ý»l³æþàþÀðºÄ%;äö}JÈÐ&Ý-Ø?çÃrg÷&Ý-Ø ?çÃrg÷&Ý-Ø?çÃr
g÷&Ý-Ø?çÃrg÷&Ý-Ø?çÃr g÷

Statistical of The Key:

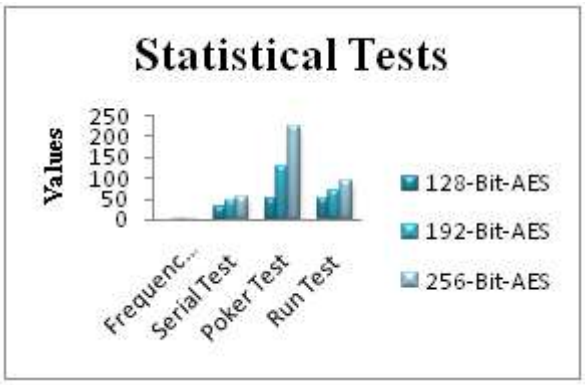


Figure 2: Tests values

Processing Time of Text Data:

According to the usage of FPGA, the processing time in seconds required to encrypt and decrypt text data using different encryption algorithms is given in figures 3, 4.

Text Data Encryption Time:

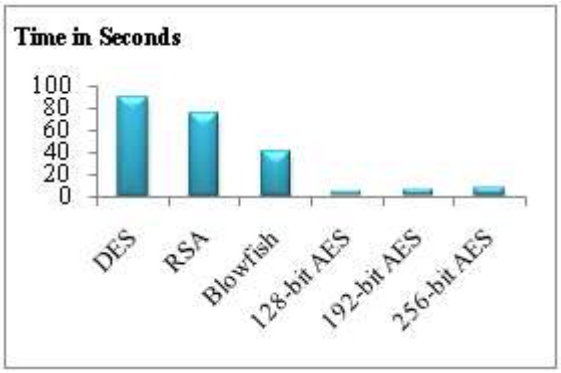


Figure 3: The encryption time for text in seconds

Text Data Decryption Time:

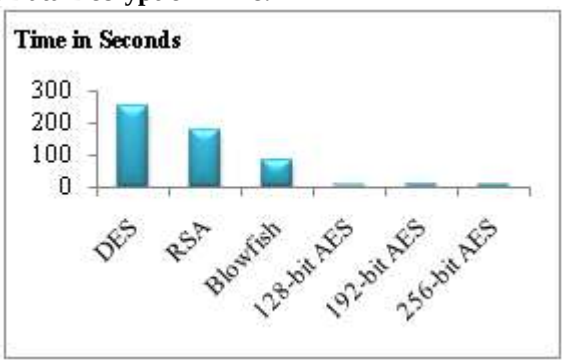


Figure 4: The decryption time for text in seconds

Table 3: Encryption and decryption time for text data

	Encryption Technique	*Using Matlab*	*Using FPGA*
Encryption time in seconds	DES	400	90
	RSA	300	75
	Blowfish	100	40
	128- bits AES	5	3
	192- bits AES	7	5
	256- bits AES	9	8
Decryption time in seconds	DES	390	250
	RSA	350	180
	Blowfish	120	85
	128- bits AES	8	5
	192- bits AES	9.5	7
	256- bits AES	13	10

2-Image:
The results of applying AES encryption algorithms with 128, 192, and 256 bits key on x-rays are shown in figures 5, 6, and 7.

128-Bits AES:

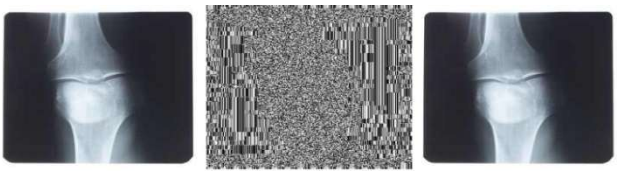


Figure 5: Original, encrypted, and decrypted image respectively

192-Bits AES:

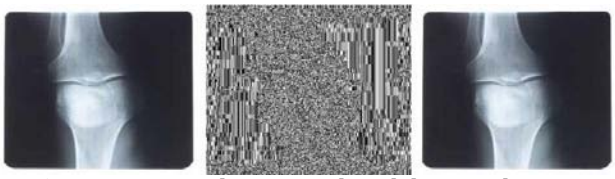


Figure 6: Original, encrypted, and decrypted image respectively

256-Bits AES:

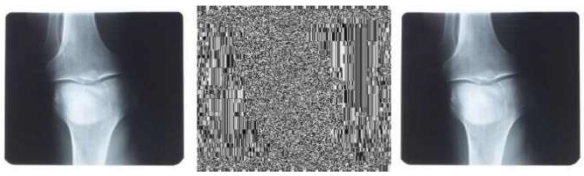


Figure 7: Original, encrypted, and decrypted image respectively

Processing Time of Image:

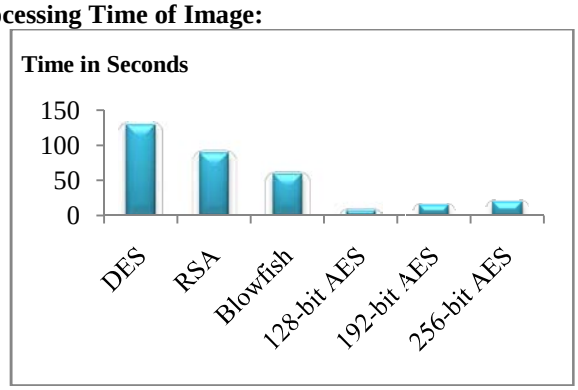


Figure 8: The encryption time for image in seconds

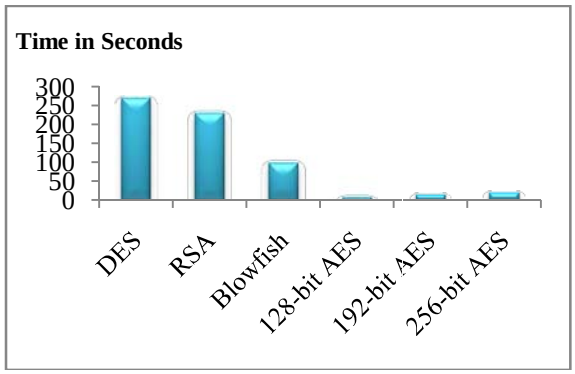


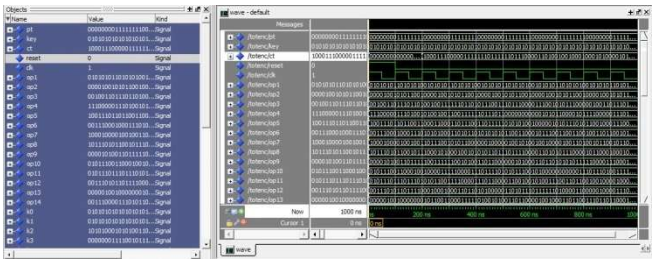
Figure 9: The decryption time for image in seconds

Table 4: Encryption and decryption time for image

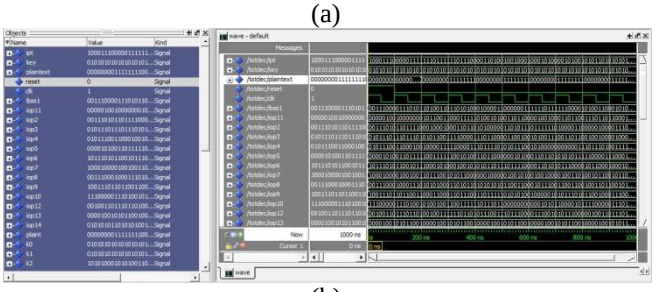
	Encryption Technique	*Using Matlab*	*Using FPGA*
Encryption time in seconds	DES	700	130
	RSA	600	90
	Blowfish	550	60
	128- bits AES	350	8
	192- bits AES	400	15
	256- bits AES	500	20
Decryption time in seconds	DES	780	270
	RSA	700	230
	Blowfish	640	100
	128- bits AES	430	10
	192- bits AES	510	15
	256- bits AES	600	20

It is obvious that the processing time in hardware (FPGA) is faster that software (MATLAB). The results show the superiority of 128-bits AES over the listed algorithms in terms of the encryption time for both text and image. When talking about data security, the 256-bits AES algorithm is the most secure algorithm compared to the listed algorithms.

Simulation Results:

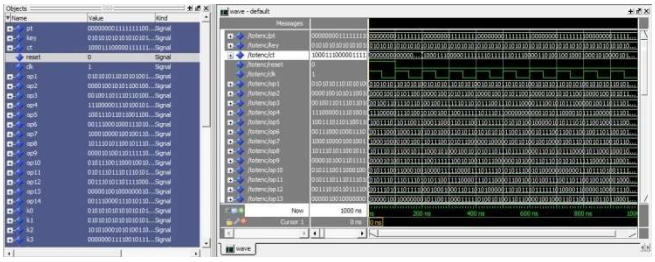


(a)

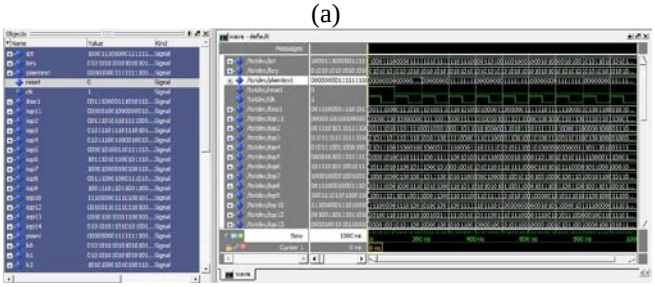


(b)

Figure 10: Pre-Synthesis for text using AES (a) Encryption, and (b) Decryption



(a)



(b)

Figure 11: Pre-Synthesis for Image using AES (a) Encryption, and (b) Decryption

Table 5: Resource used in the FPGA Implementation of AES using Spartan-3A, Virtex-4, Virtex-5, and Spartan-6

		Spartan-3A,	Virtex-4	Virtex-5	Spartan-6
Process		90nm	90nm	65nm	45nm
Static Power Consumption (mw)		27-336	128-1278	267-3028	11-94
IOS	Used	11	11	11	11
	Avail.	372	240	640	218
Global Buffers	Used	1	1	1	1
	Avail.	24	32	32	32
LUTs	Used	59494	36078	8338	8180
	Avail.	11776	12288	69120	27288
CLB Slices	Used	29747	18039	2085	2398
	Avail.	5888	6144	17280	6822
Block RAMs	Used	0	0	0	0
	Avail.	20	36	148	116

8. Conclusion

Encryption is complementary line of defense in protecting multimedia content. In this paper, AES encryption algorithm has been discussed with some statistical tests that are applied on the key to test its randomness. Software and hardware implementations with the aid of MATLAB, Mentor Graphics Tools, and Xilinx – Project Navigator, ISE 14.2 suite for the encryption of text and biomedical images using 256-bits key AES algorithm have been presented. It is clear that the results show that the hardware is much faster than software and increase system security. Moreover, they save logic area and reduce the static power consumption.

References

[1] Henk C.A. van Tilborg, "Fundamentals of Cryptology: A Professional Reference and Interactive Tutorial", ISBN: 0-7923-8675-2, 2000.
[2] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, "An Introduction to Mathematical Cryptography", ISBN: 978-0-387-77993-2, 2008.
[3] Richard A. Mollin, "An Introduction to Cryptography", Second Edition, ISBN: 1584886188 / 9781584886181, 2005.
[4] John Talbot, Dominic Welsh, "Complexity and Cryptography: An Introduction", ISBN: 978-0-511-14070-9, 2006.
[5] Ali E. Taki El_Deen, Mohy E. Abo-Elsoud, Salma M. Saif, "Text and Biomedical Images Disguising using Advanced Encryption Standard", International Journal of Engineering Research and Technology (IJERT), Vol. 2, Issue 12, December 2013.
[6] José Luis Gómez Pardo, "Introduction to Cryptography with Maple", ISBN: 978-3-642-32165-8, 2013.
[7] Christof Paar, Jan Pelzl, "Understanding Cryptography", ISBN: 9783642041006, 2010.
[8] Wade Trappe, Lawrence C. Washington, "Introduction to Cryptography with Coding Theory", Second Edition, ISBN: 0-13-198199-4, 2006.
[9] Andrew Rukhin, Juan Soto, James Nechvatal, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", National Institute of Standards and Technology, 2008.
[10] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, "Hand Book of Applied Cryptography", 4th Edition, ISBN: 9780849385230, 2010.
[11] Xilinx, "Introduction to FPGA Design with Vivado High-Level Synthesis", UG998 (v1.0) July 2, 2013.
[12] Douglas L. Perry, "VHDL: Programming by Example", 4th Edition, ISBN: 0-07-140070-2, 2002.
[13] Clive "Max" Maxfield, "The Design Warrior's Guide to FPGAs", 4th Edition, ISBN: 0-7506-7604-3, 2004.
[14] http://www.xilinx.com/fpga/asic.htm

Author Profile

Salma M. Seif received the B.S.c degree (honors) in Electronics and Communication Engineering (ECE) Department from Mansoura University, Mansoura, Egypt, in 2011. She is currently working towards the MS.c degree in ECE from Mansoura University,Mansoura, Egypt. Her research interests include Cryptography, Watermarking, and Biomedical image applications using FPGA technology.

Ali E. Taki El_Deen received the PhD degree in Electronics and Communications Engineering in "Encryption and Data Security in Digital Communication Systems". He has a lot of publications in various international journals and conferences. His current research interests are in multimedia processing, wireless communication systems, and Field Programmable Gate Array (FPGA) applications.

Mohy Eldin Aboelsoud received the M.Sc. degree in Electronics and Communication Engineering Department from Cairo University, Cairo, Egypt, in 1979, and the Ph.D degree from L'Institut National Polytechnique de Toulouze, Toulouze, France, in 1983. He has worked for Technical Research Center (TRC) from 1970 to 1980. During 1984 – 1990, he was a chairman with Electronics Department at TRC. From 1991 to 1996, he was an associated professor at Mansoura University. He is currently full professor with ECE Department of Mansoura University since 1996. His research interests, analog/digital VLSI circuit, FPGA design, switched-resistor networks, electronic system for neural networks and digital signal processing applications.