Development of Randomized Hybrid Crypto System

Nuthan A.C
Assistant Professor,
Department of ECE, GMIT,
Bharathinagar, Karnataka,
India.

Naveen Kumar M.S
Assistant Professor,
Department of ECE, GMIT, Bharathinagar,
Karnataka, India.

Ravikant. G. Biradar
Assistant Professor,
Department of TE, PESIT,
Bangalore, Karnataka, India.

Dr. Shivanand S Gornale
Assistant Professor and Head
Post Graduate Department of Computer Sc., Govt.
College (Autonomous), Mandya Karnataka, India.

ABSTRACT*-Before the modern era, cryptography was concerned solely with message confidentiality (i.e., encryption) conversion of messages from a comprehensible form into an incomprehensible one and back again at the other end, rendering it unreadable by interceptors without secret knowledge (namely the key needed for decryption of that message). This paper presents a highly secured randomized encryption/decryption technique. Implementation presented in this paper chooses a randomly switching model which combines the two symmetric cryptosystems,Data Encryption Standard (DES) and Advanced Encryption Standard (AES)to work together as a single unit. This module renders it impossible for the intruder information on the algorithms deployed. This model being implemented on FPGA provides excellent area results and a higher level of security is achieved.*

KEYWORDS**: *Hybrid crypto-system, Data Encryption Standard (DES), Advanced Encryption Standard (AES), Random Number Generator (RNG), FPGA.***

I INTRODUCTION

Cryptography is the art and science of protecting information from undesirable individuals by converting it into a form non-recognizable by its attackers while being stored and transmitted [1]. Data cryptography mainly is the scrambling of the content ofthe data, such as text, image, audio, video and so forth to make the data unreadable, invisible orunintelligible during transmission or storage called Encryption. The main goal of cryptography is keeping data secured from unauthorized attackers. The reverse of data encryption is data Decryption.

In modern days, cryptography is no longer limited to secure sensitive military information but recognized as one of the major components of the security policy of any organization and considered an industry standard for providing information security, trust, controlling access to resources, and electronic financial transactions. In recent decades, the field hasexpanded beyond confidentiality concerns to includetechniques for message integrity checking, sender or receiver identity authentication, digital signatures, interactive proofs and secured computation, among others. The modern study of symmetric-key ciphers relates mainly to the study of block ciphers and stream ciphers and to their applications. Block ciphers take as input a block of plaintext and a key, and output a block of cipher text of the same size. The crypto systems which are in use arestatic i.e. only one of the standard algorithms is used at a time. If the same encryption methods are used throughout the day and for days together continuously, the crypto analyzers may crack the code or find out the encryption method and may intrude or spy or hack the confidential data. This necessitates a robust encryption technique. This paper highlights one of the novel approaches of an implementation of a highly secured cryptic algorithm in FPGA which makes impossible for data theft. Many researches have been made with the multiple

algorithms. Instead ofdedicating the hardware i.e. FPGA for single algorithm, the paper [2] presents a hardware implementation of three standard cryptography algorithms on a universal architecture. But in this design, multiple algorithms are implemented on a universal architecture and there is switching among the algorithms at random times.

Though multiple crypto algorithms can be implemented, for simplicity only two algorithms are deployed:

- DES-Data Encryption Standard
- AES-Advance Encryption Standard

II ALGORITHM

Data encryption Algorithm

DES was adopted in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46) [3]. DES is a Feistel-type Substitution-Permutation Network (SPN) cipher. It is a secret-key archetypal block cipher with a block size of 64 bits. DES encrypts a block of 64-bit plaintext into 64-bit cipher text using 64-bit secret key [Left most bit of a block is bit one]. DES uses a 56-bit key which can be broken using brute-force methods, and is now considered obsolete.

Block diagram of the DES encryption is shown in the Figure 1. Figure 2 shows the generation 16 sub keys. A 16 cycle Feistel system is used, with an overall 56-bit key permuted into 16 48-bit subkeys, one for each cycle. With a key length of 56 bits, there are 2^{56} possible keys, which is approximately 7.2 x 10^{16}. Thus a brute-force attack appears impractical [1]. Sometimes DES is said to use a 64-bit key, but 8 of the 64 bits are used only for parity checking, so the effective key size is 56 bits [3][4]. To decrypt, the identical algorithm is used, but the order of sub keys is reversed.

The L and R blocks are 32 bits each, yielding an overall block size of 64 bits. The hash function "f", specified by the standard using the so-called "S-boxes", takes a 32-bit data block and one of the 48-bit subkeys as input and produces 32 bits of output. DES has avalanche effect, a small change in the plaintext or the key should produce a significant change in the cipher extea. A change in one bit of the plaintext or one bit of the key should produce a change in many bits of the ciphertext. Since the time DES was adopted (1977), it has been widely speculated that some kind of backdrop was designed into the cryptic S-boxes, allowing those "in the know" to effectively

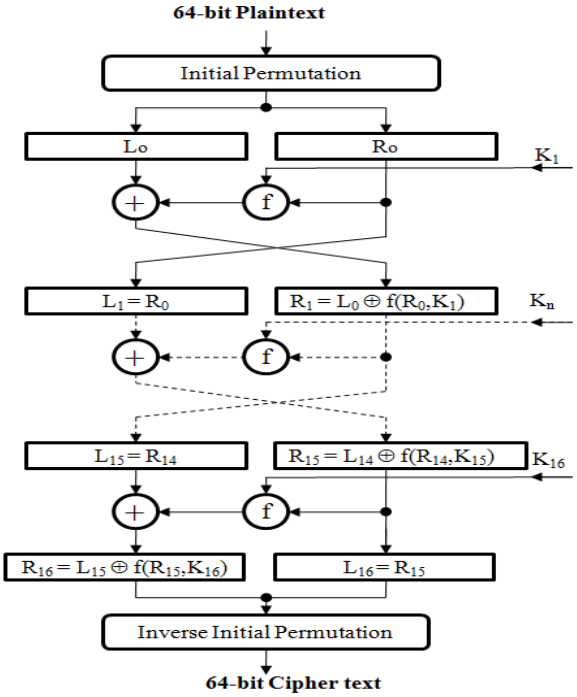

Figure 1 : DES encryption[5]

Combined with the natural parallelism of Feistel ciphers and DES's relatively small key size, have rendered the algorithm obsolete. In 1998, the Electronic Frontier Foundation built a DES Cracker (full specifications available online) for less than $250,000 that can decode DES messages in less than a week [3] [6] [7].

Advanced Encryption

Though triple DES (3DES) is considered to be the best choice (when security is a major concern), its major drawbacks [1] made 3DES impractical for one use.Hence as a replacement NIST in 1997 issued a call for proposals for a new Advanced Encryption Standard (AES) with requirements [1] and evaluation criteria [8]. NIST selected Rijndael (Developed by Dr. Vincent Rijmenand / Dr. Joan Daemen] as the proposed AES algorithm and published a final standard FIPS PUB 197 in November of 2001 [9]. Block diagram of the AES encryption is shown in the Figure 3. Figure 4 shows the generation 16 sub keys. The algorithm processes a data block of size 128 bits using a cipher key of length 128, 192 and 256 bits.

Each data block is a 4×4 matrix called the state on which the basic operations of AES algorithm are performed,after commencing the first round of key on which the basic operations of AES algorithm are performed.

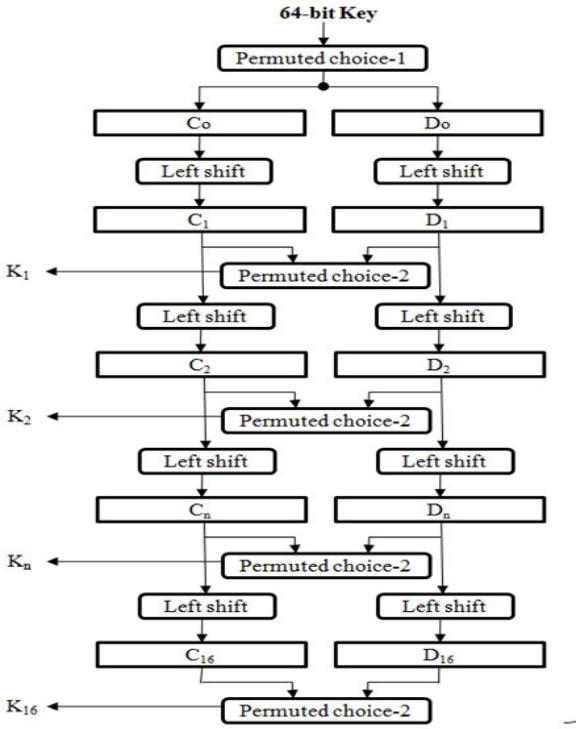

Figure 2 : Generation of subkeys for DES

In addition, around function consisting of four distinctive byteoriented transformations is performed.They are:

- Substitute bytes: Byte by byte substitution of the block using S - boxes (table)
- Shift rows: Circular shift or a simple permutation operation is performed.
- Mix columns: Each column of the block multiplied by a constant matrix, the result obtained is substituted back to the block which makes use of arithmetic over GF (Galois field).
- Add round key: Simple bitwise XOR operation between the current block and expanded key.

These transformations are applied to the data block (i.e. State). The decryption process of AES is the converse of each transmutations explained above except for the Mix column iteration. The structural resemblance for both encryption and decryption makes hardware implementations easier [10].

Random Number Generator

Multiple-bit leap-forward LFSR method utilizes only one LFSR and shifts at several

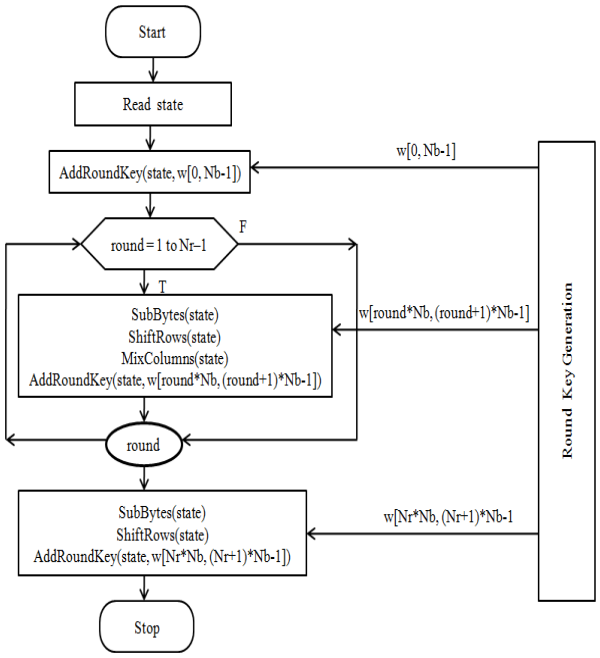

Figure 3: AES encryption

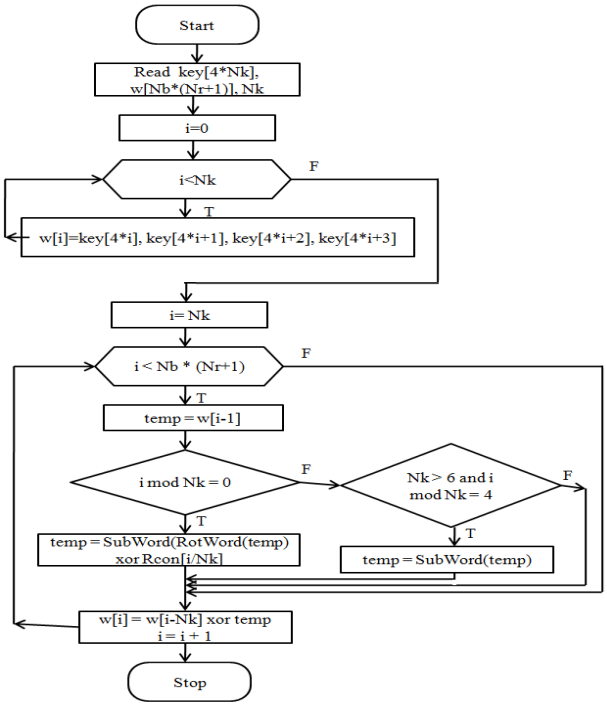

Figure 4 : Generation of subkeys for AES

bits. This method is based on the observation that an LFSR is a linear system and the register state can be written in vector format:

$$q(i+1) = A \bullet q(i)$$

Here, **q**(i +1) and **q**(i) → content of the shift register at $(i+1)^{th}$ and i^{th}steps, **A**→the transition matrix. After the LFSR advances k steps, the equation becomes

$$q(i+1) = A^K \bullet q(i)$$

Let the 4-bit LFSR with

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \therefore A^4 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Therefore, $Q_{next} = A^4 (Q_{present}$

$$q_{0\_next} = q_0 \oplus q_3$$
$$q_{1\ next} = q_0 \oplus q_1 \oplus q_3$$
$$q_{2\_next} = q_0 \oplus q_1 \oplus q_2 \oplus q_3$$
$$q_{3\_next} = q_0 \oplus q_1 \oplus q_2$$

This is realized as shown in Figure 5.

For simplicity instead of 4-bit Leap-Forward LFSR , a 2-bit Leap-Forward LFSR is considered. Therefore the equation becomes

$q_{0\_next}=q_1$
$q_{1\_next}=q_0$

Hence if the seed is "01" then the output sequence is 10 01 10 01

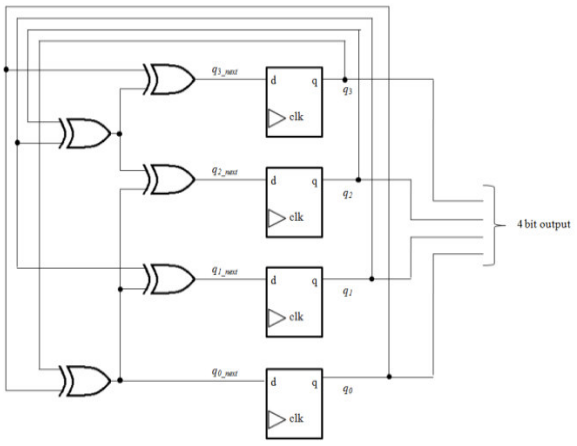

Figure 5: 4-bit Leap-forward LFSR

III PROPOSED SYSTEM AND EXECUTION

The communication between Desktop and FPGA is done using HyperTerminal application supported by Windows OS and UART cable. The Universal Asynchronous Receiver & Transmitter (UART) transmits 11 bits in a frame out of which 8 bits are data bits. First bit is the start bit and the last bit of the frame is stop bit.

Eight characters (64 bits) given after every reset or start-up acts like a password

- One copy of the password is Key for DES encoder/decoder
- Two copies of password which are concatenated (16-character-128bits) is Key for AES encoder/decoder
- First two bits of password act like seed for random number generator
- Each bit is a select line between AES and DES ie if bit is high then AES is active else DES is active

Consider

- Password:
  - NNNNNNNN
  - 4E4E4E4E4E4E4E4E
  - 0100111001001110010011100100111001001110 0100111001001110010011100100111001001110
- Data:
  - nnnn.....nn
  - 6E6E6E6E..... 6E6E
  - 0110111001101110011011100110111001101110 0110111001101110

Therefore

- DES key:
  - NNNNNNNN
  - 4E4E4E4E4E4E4E4E
  - 0100111001001110010011100100111001001110 0100111001001110010011100100111001001110
- AES key:
  - NNNNNNNNNNNNNNNN
  - 4E4E4E4E4E4E4E4E4E4E4E4E4E4E4E4E E4E
  - 0100111001001110010011100100111001001110 0100111001001110010011100100111001001110

01001110010011100100111001001110
01001110010011100100111001001110

- Seed to RNG: 01 (First 2-bits of Password-NNNNNNNN)

For simplicity let us consider first character of the password (NNNNNNNN) ie N (01001110) then the switching between the AES and DES is done as shown in Table 1.Similarly this is continued with the next bit of the password infinitely.

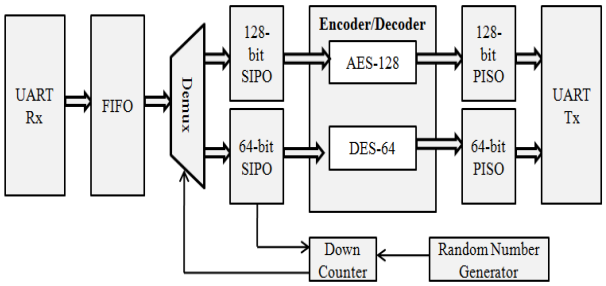

Figure 6: Proposed Module of Encoder/Decoder

Taking "01" as seed the RNG gives "10"(2) as the output which is fed to down-counter. And since the first bit of password is '0' the DES line is selected. The 8 bit data appear at the de-multiplexer, the 8 bits of data are serially shifted to a 64 bits (8 characters) SIPO shift register, and the counter is decremented once. ie.Count value 2 is decremented to 1. When the SIPO contains 64 bits of data to be encoded (or decoded), the load (lda) signal goes high and low and the 64 bits of data is loaded to the AES encoder (or decoder) as shown in Figure 6.

DES encoder(or decoder) has a mod17 counter, DES takes 16 clock pulses to complete the 16 rounds of encoding(or decoding) operations, and at the 17th clock pulse the DONE signal goes high and DES encoder(or decoder) loads encrypted(or decrypted) data into 64 bit PISO shift register. The encoded (or decoded) data is shifted out serially to the UART through the multiplexer. Since count value is non-zero, again the DES line is selected and above is repeated for next 8 characters. But this time when the counter is decremented then the count value becomes '0' hence

- It signals the RNG to give the next random number which is "01"(1)
- Checks the next bit ie the second bit of password and since it is '1' AES line is selected

and the counter is decremented once. ie

Count value 1 is decremented to 0. When the SIPO contains 128 bits of data (16 characters) to be encoded (or decoded), the load (lda) signal goes high and low and the 128 bits of data is loaded to the AES encoder (or decoder) as shown in Figure 6.

AES encoder (or decoder) has a mod11 counter; AES takes 10 clock pulses to complete the 10 rounds of encoding(or decoding) operations, at the 11th clock pulse the DONE signal goes high and AES encoder loads encrypted data into 128 bit PISO shift register. The encoded (or decoded) data is shifted out serially to the UART through the multiplexer. But this time when the counter is decremented then the count value becomes '0' hence

- It signals the RNG to give the next random number which is "10"(2)
- Checks the next bit ie third bit of password and since it is '0' AES line is selected

This is repeated infinitely.

For simplicity let us consider first character of the password (NNNNNNNN) ie N (01001110) then the switching between the AES and DES is done as in Table 1. Table 2 shows the example result of AES and DES encoder and also decoder where the input to the decoder is output to the encoder hence the encoder input and decoder output are same.

The simulation is done on the model of DES and AES encoder and decoder clubbed into one is shown in the Figure 7 and Figure 8. Figure 9 shows the results of the universal encoding where switching between AES and DES are done according to the Table 1.

N	Switching between DES and AES	Rounds	
0	Starts with DES	2 times DES	2 times DES then switch to AES
1	DES (AES	1 time AES	1 time AES then switching to DES
0	AES (DES	2 times DES	3 times DES then switch to AES
0	Remains in DES	1 time DES	
1	DES (AES	2 times AES	5 times AES then switching to DES
1	Remains in AES	1 time AES	
1	Remains in AES	2 times AES	
0	AES (DES	1 time DES	1 time DES

Table 1: Sequence of switching between AES and DES

Similarly this is continued with the next bit of the password infinitely

	DES		AES	
	Hexadecimal	ASCII	Hexadecimal	ASCII
Key	4E4E4E4E4E4E4E4E	NNNNNNNN	4E4E4E4E4E4E4E4E4E4E4E4E4E4E4E	NNNNNNNNNNNNNNNN
Encoder input	6E6E6E6E6E6E6E6E	nnnnnnnn	6E6E6E6E6E6E6E6E6E6E6E6E6E6E6E	nnnnnnnnnnnnnnnn
Encoder output Decoder input	6BBAE23B9849C018	kºâ;¯IÀ	99D0F980D380A06A20E860EA80E0EEC6	Đù Ó j è`ê àî
Decoder output	6E6E6E6E6E6E6E6E	nnnnnnnn	6E6E6E6E6E6E6E6E6E6E6E6E6E6E6E	nnnnnnnnnnnnnnnn

Table 2: Output of AES and DES encoder and decoder

When 16 bytes of data in case of AES or 8 bytes of data in case of DES are fed into Xilinx simulator, researchers can observe that all the 16 bytes or 8 bytes of encrypted or decrypted data for AES or DES respectively, but when the data is dumped on FPGA kit, because of some design details.

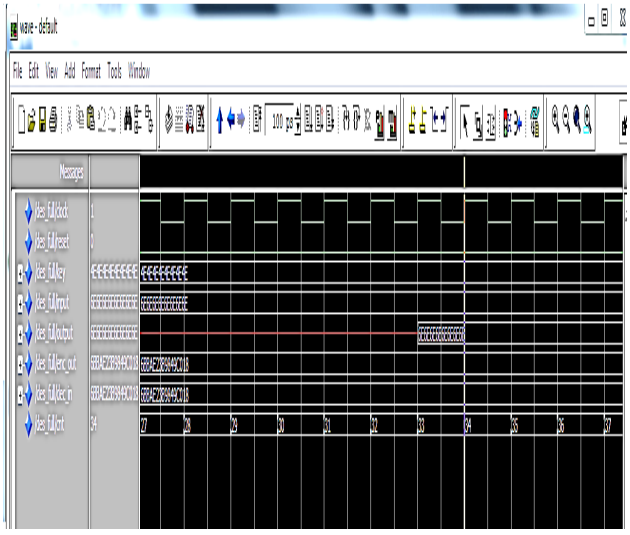

Figure 7: simulation results of DES Encoder and Decoder

Constraints, that is for same clock pulse, from encoder data is loaded to 128 bit register and also the contents of 4bit TEMP register get loaded into a 128 bit register (into LSB). As a result, the researcher is losing one byte of data in case of both AES and DES respectively.

Encoder and decoder have slight inconsistency; the encoded characters correlate with the input data.

Figure 8: simulation results of AES Encoder and Decoder

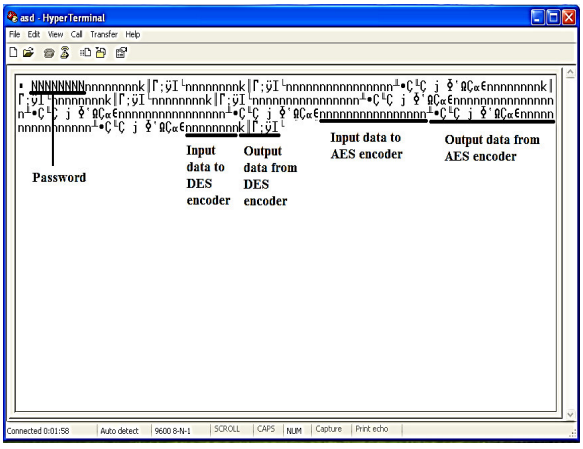

Figure 9 : Xilinx GUI display of encrypted characters

IV CONCLUSIONS

- The concept of universal coding with random switching can be practically implemented
- Implementation of universal coding on the FPGA is successful and multiple data encryption

and decryption standards can be successfully integrated and can be switched randomly among them. So advantages of using a single algorithmare,

- Usage of multiple algorithms helps to increase the security.
- There is a flexibility to remove or add any other cryptographic standards. These changes can be on the fly.
- The cipher key (Password) can be changed with respect to the user or the company requirements.
- While implementing synchronization with the UART, AES encoder and decoder, DES encoder and decoder were the critical aspect of the implementation. This has been handled successfully.
- Since AES and DES are used, the universal coding inherits the statistical and other advantages of AES and DES.
- Switching randomly between the algorithms also enhances security
- Since the password acts like DES key, AES key, seed for RNG and controls switching between AES and DES , universal coding is completely randomized depending upon the password. i.e. the intruder will not know which algorithm is active at what time and for how long. In future this prototype can also be imported to ASIC.

REFERENCES

[1] William Stallings, Cryptography and Network Security Principles and Practices, Fourth Edition, Prentice Hall, 2005.

[2] Yadollah Eslami, Member, IEEE, Ali Sheikholeslami, Senior Member, IEEE, P. Glenn Gulak, Senior Member, IEEE, Shoichi Masui, Member, IEEE, and Kenji Mukaida, “An Area-Efficient Universal Cryptography Processor for Smart Cards”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 14, PP. 43-56, January 2006.

[3] Konheim, A. Cryptography: A Primer. New York: Wiley, 1981.

[4] Behrouz. A. Forouzan, Cryptography and Network Security, Special Indian Edition, Tata Mc-Graw Hill, 2007.

[5] A.A. Zaidan, B. B. Zaidan, AnasMajeed, "High Securing Cover-File of Hidden Data Using Statistical Technique and AES Encryption Algorithm", World Academy of Science Engineering.

[6] M. Abomhara, Omar Zakaria, Othman O. Khalifa , A. A. Zaidan, B. B. Zaidan, “Enhancing Selective Encryption for H.264/AVC Using Advanced Encryption Standard “, International Journal of Computer and Electrical Engineering (IJCEE), ISSN: 1793-8198, Vol.2 , NO.2, PP. 224-229, April2010, Singapore.

[7] Banraplang Jyrwa, Roy Paily, “An Area-Throughput Efficient FPGA implementation of Block Cipher AES algorithm”, IEEE computer society, December 2009.

[8] Advanced Encryption Standard (AES), FIPS 197, November 26, 2001.

[9] Ashwini. MDeshpande. Mangesh S. Deshpande, and Devendra N. Kayatanavar, “FPGA implementation of AES Encryption and Decryption”, International conference on “control, automation, communication and energy conservation” Proc. CACEC’09, 2009, paper 1, p. 1-6.

[10] Daemen, J., and Rijmen, V. "Rijndael: “The Advanced Encryption standard.”Dr. Dobb'sJournal.