ISSN:0975-9646

Image Encryption using AES Algorithm based on FPGA

Anup.Gujar

M.Tech
Dept Of Electronics And Communication
Ashoka College of Eng&Tech. JNTU HYDERABAD

ABSTRACT
With the fast progression of data exchange in electronic way, information security is becoming more important in data storage and transmission. Because of widely using images in industrial process, it is important to protect the confidential image data from unauthorized access.This paper presents the design of a 128 bit encoder using AES Rijndael Algorithm for image encryption. The AES algorithm defined by the National Institute of Standard and Technology(NIST) of United States has been widely accepted. Optimized and Synthesizable VHDL code is developed for the implementation of 128- bit data encryption and process. Xilinx ISE9.2i software is used for synthesis.Timing simulation is performed to verify the functionality of the designed circuit.

General Terms
Cryptography, Encryption and Decryption Algorithms ,Secret Key.

Keywords

Advanced Encryption Standard, Rijndael, S-box.

1. INTRODUCTION

Transmission of sensitive data over the communication channel have emphasized the need for fast and secure digital communication networks to achieve the requirements for secrecy, integrity and non reproduction of exchanged information. Cryptography provides a method for securing and authenticating the transmission of information over insecure channels.It enables us to store sensitive information or transmit it across insecure networks so that unauthorized persons cannot read it.

The urgency for secure exchange of digital data resulted in large quantities of different encryption algorithms which are evaluated on the basis of throughput ,speed of operation and area requirements.There are mainly two types of cryptographic algorithms: symmetric and asymmetric algorithms. Symmetric systems such as Data Encryption Standard(DES),3DES, and Advanced Encryption Standard(AES) uses an identical key for the sender and receiver; both to encrypt the message text and decrypt the cipher text. Asymmetric systems such as Rivest-Shamir-Adelman(RSA) & Elliptic Curve Cryptosystem(ECC) uses different keys for encryption and decryption. Symmetric cryptosystems is more suitable to encrypt large amount of data with high speed.

To replace the old Data Encryption Standard ,in September 12 of 1997,the National Institute of Standard and Technology(NIST) required proposals to what was called Advanced Encryption Standard (AES).Many algorithms were presented originally with researches from 12 different nations. On October 2nd 2000,NIST has announced the Rijndael Algorithm is the best in security, performance, efficiency, implement ability & flexibility. The Rijndael algorithm was developed by Joan Daemen of Proton World International and Vincent Rijmen of Katholieke University at Leuven.

AES encryption is an efficient scheme for both hardware and software implementation. As compare to software implementation, hardware implementation provides greater physical security and higher speed. Hardware implementation is useful in wireless security like military communication and mobile telephony where there is a greater emphasis on the speed of communication.Most of the work has been presented on hardware implementation of AES using FPGA.

2. CONVENTIONAL CRYPTOSYSTEM

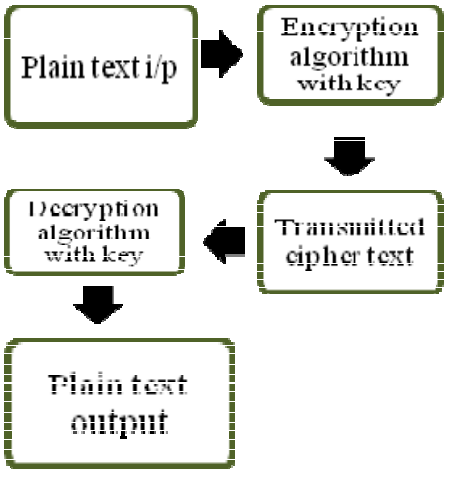

Figure:1 General block diagram

A symmetric cryptosystem is shown in figure:1 and has five ingredients:

i. Plain text: this is the original message or data that fed into the algorithm as input.
ii. Encryption algorithm: the algorithm performs various substitutions and transformations on the plaintext.
iii. Secret key: this is also an input to the algorithm and its value is independent of the plaintext. The algorithm will produce a different output depending on the specific key.
iv. Cipher text: this is the scrambled message produced as output. It depends on the plaintext and the secret key.
v. Decryption algorithm: this is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

2.1 The AES Algorithm

The AES Algorithm is a symmetric-key cipher, in which both the sender and the receiver use a single key for encryption and decryption. The data block length is fixed to be 128 bits, while the length can be 128,192,or 256 bits. In addition, the AES algorithm is an iterative algorithm. Each iteration can be called a round, and the total number of rounds is 10,12, or 14, when key length is 128,192, or 256, respectively. The 128 bit data block is divided into 16 bytes. These bytes are mapped to a 4x4 array called the State, and all the internal operations of the AES algorithm are performed on the State.

Table 1: AES parameters

Algorithm	Key length (Nk words)	Block Size (Nb words)	Number of rounds (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

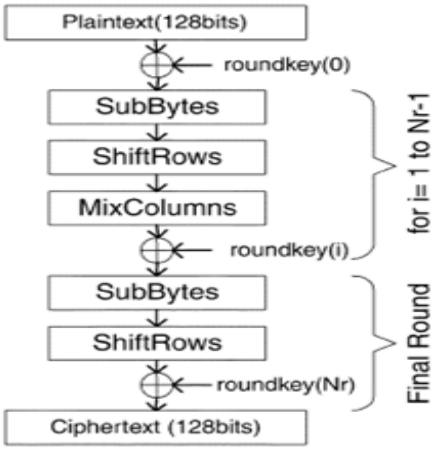

Figure:2 Detailed Block diagram of encryption part

In the encryption of the AES algorithm(Figure:2), each round except the final round consists of four transformations:

i. SubBytes: Operates in each byte of the State independently. Each byte is substituted by corresponding byte in the S-box.
ii. ShiftRow: Cyclically shifts the rows of the State over different offsets.
iii. MixColumn: In this operation the column of the State are considered as polynomials over GF(2^8) and are multiplied with a fixed polynomial. The MixColumn component doesnot operate in the last round of the algorithm.
iv. AddRoundKey: Involves bit-wise XOR operation.

3. DESIGN OF 128BIT ENCODER

3.1 Methodology

The encryption process is iterative in nature. Each iterations are known as rounds .For each round 128 bit input data and 128 bit key is required .That is,need 4 words of key in one round.So the input key must be expanded to the required number of words, which depends upon the number of rounds. The output of each round serves as input of next stage.In AES system,same secret key is used for both encryption and decryption.So it provides simplicity in design.

3.2 Design steps

3.2.1 State array

The input to the encryption algorithm is a single 128-bit block
.This block is copied into the State array,which is a square matrix
of bytes. State array is modified at each stage of encryption.Similarly,the 128 bit key is depicted as a square matrix of bytes.The ordering of bytes within a matrix is by column.

3.2.2 Key expansion

Key expansion is an important for both encryption and decryption.

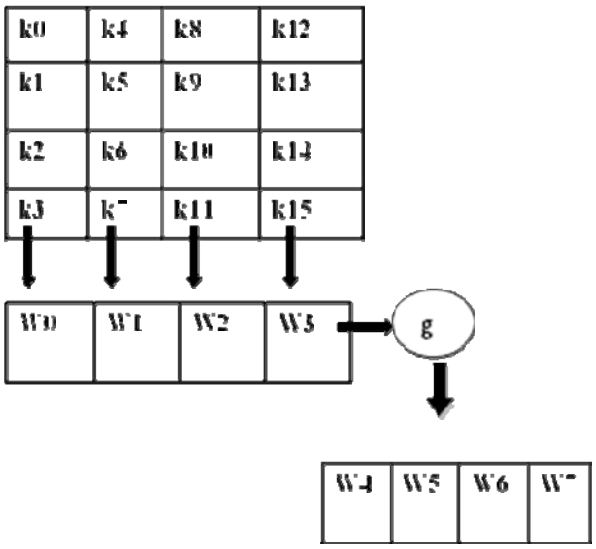

Figure:3 KeyExpansion algorithm

The AES key expansion algorithm takes as input a 4-word (16 bytes) key and produces a linear array of 44 words (176 bytes).This is sufficient to provide a 4-word round key for the initial Add Round Key stage and each of the 10 rounds of the cipher.

The following pseudocode describes the expansion:

Table 2: Pseudocode for KeyExpansion

```
KeyExpansion(byte key[16],word[44])
{
  word temp
for (i=0; i<4; i++)
   w[i]=(key[4*i],key[4*i+1],key[4*i+2],key[4*i+3]);for
(i=4; i<44; i++)
 {
  temp =w[i-1];
 if(i mod 4 = 0)
     temp=SubWord(RotWord(temp)) xor Rcon[i/4];
w[i] = w[i-4] xor temp
 }
}
```

The key is copied into the first 4 words of the expanded key.The reminder of the expanded key is filled in 4 words at a time.Each added word w[i] depends on the immediately preceding word, w[i-1] and the word four positions back, w[i-4]. In three out of four cases, a simple XOR is used. For a word..

1. RotWord performs a one-byte circular left shift on a word.This means that an input word [b0,b1,b2,b3] is transformed into [b1,b2,b3,b0].
2. SubWord performs a byte substitution on each byte of its input word,using the S-box.
3. The result of step 1 and step2 is XORed with a round constant Rcon[j].

The round constant is a word in which the three rightmost bytes are always 0.Thus the effect of XOR of a word with Rcon is to only perform an XOR on the left byte of the word.The round constant is for each round and is defined as Rcon[j] = (RC[j],0,0,0), with RC[1]=1; RC[j]=2*RC[j-1] and with multiplication over the field GF(2^8).

The values of RC[j] in hexadecimal are :

Table 3: Round constant values

j	RC[j]
1	01
2	02
3	04
4	08
5	10
6	20
7	40
8	80
9	1B
10	36
11	6C
12	d8
13	A6
14	4d

3.2.3 AddRound Key

The 128 bits of State array are bitwise XORed with the 128 bits of the round key(4 words of the expanded key).The operation is viewed as a columnwise operation between the 4 bytes of the State array column and one word of the round key (Figure 4).

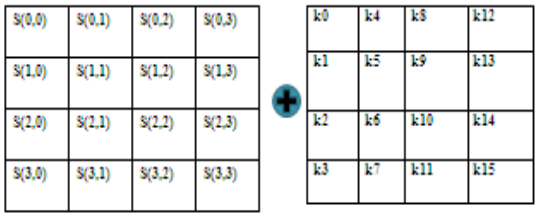

Figure 4: XOR operation between State and key word

3.2.4 Substitute Bytes

AES defines a 16 x 16 matrix of byte values, called an S-box, that contains a permutation of all possible 256 8-bit values. Each byte of State array is mapped into a new byte in the following way: The leftmost 4 bits of the byte are used as a row value and the leftmost 4 bits are used as a column value. These row and column values serve as indexes into the S-box to select a unique
8-bit output value (Figure 5).

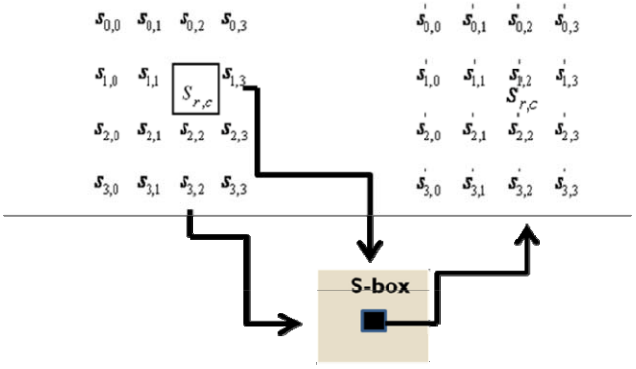

Figure 5: Sub-byte Operation

3.2.5 Shift Rows

The first row of State array is not altered. For the second row , a 1-byte circular left shift is performed. For the third row,a 2- byte circular left shift is performed. For the third row, a 3-byte circular left shift is performed.

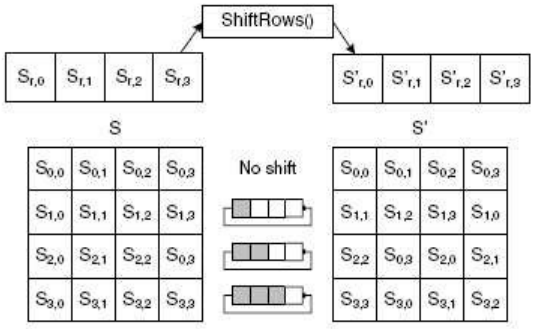

Figure 6: Shifting operation

3.2.6 Mix Column

It operates on each column individually.Each byte of a column is mapped into a new value that is a function of all four bytes in the column.The transformation can be defined by following matrix multiplication on State array.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S(0,0) & S(0,1) & S(0,2) & S(0,3) \\ S(1,0) & S(1,1) & S(1,2) & S(1,3) \\ S(2,0) & S(2,1) & S(2,2) & S(2,3) \\ S(3,0) & S(3,1) & S(3,2) & S(3,3) \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} S'(0,0) & S'(0,1) & S'(0,2) & S'(0,3) \\ S'(1,0) & S'(1,1) & S'(1,2) & S'(1,3) \\ S'(2,0) & S'(2,1) & S'(2,2) & S'(2,3) \\ S'(3,0) & S'(3,1) & S'(3,2) & S'(3,3) \end{bmatrix} \quad -- \quad (1)$$

Each element in the product matrix is the sum of products of elements of one row and one column.In this case,individual additions and multiplications are performed in GF(2^8).The Mix Column transformation on a single column j(0 <= j <= 3) of State array can be expressed as

$$\begin{aligned} S'(0,j) &= (2.s(0,j)) \text{ xor } (3.s(1,j)) \text{ xor } (s(2,j)) \text{ xor } (s(3,j)) \\ S'(1,j) &= (s(0,j)) \text{ xor } (2.s(1,j)) \text{ xor } (3.s(2,j)) \text{ xor } (s(3,j)) \\ S'(2,j) &= (s(0,j)) \text{ xor } (s(1,j)) \text{ xor } (2.s(2,j)) \text{ xor } (3.s(3,j)) \\ S'(3,j) &= (3.s(0,j)) \text{ xor } (s(1,j)) \text{ xor } (s(2,j)) \text{ xor } (2.s(3,j)) \quad -- \quad (2) \end{aligned}$$

3.3 Experiment Analysis

Each round has 4 operations and it is iterative in nature.So the output of first round is fed to the second round as input data and perform the same operations with another set of keys.This process continued until the last round reach.In the last round,there is no mix column operation. The State array obtained after the last round is the required cipher text for transmission (Figure 7).

Input data(128bit):

[11111110000111100001111100001111111010101010101000000000011110001110101000110011010111000110001110000111001100011000101]

Input key(128 bit):

[11000010101000010111100000001111111111100000000000011111111111010101010101010101010000000000011111111111000000011100111001100000]

Output(128bit):

[01101010000001010010001101101110011101111101110001111001111000100000011100111110011000001100001000100101011010010110100110110111]

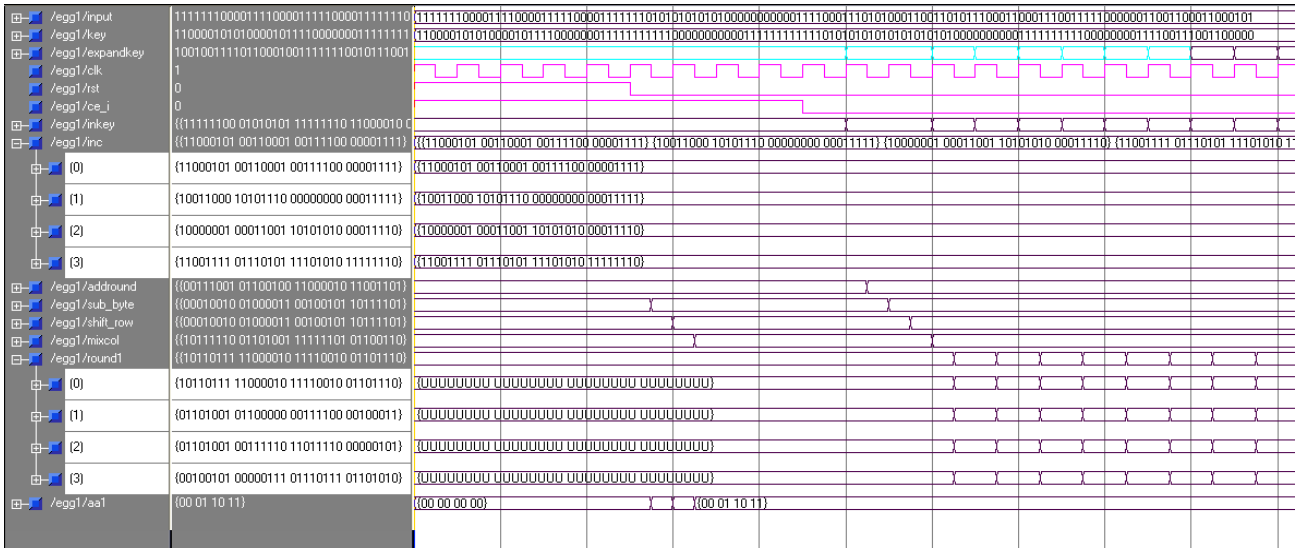

Figure 7: Encrypted data

The 128 bit input data is encoded into another set of 128 bit data using 128 bit secret key.Schematic diagram for both encoder and key expansion unit is obtained.

Expand the 128 bit key into 256 bits. Initially,map the input key into 4x4 matrix as that of the input data.Then performs the key expansion algorithm in columnwise.

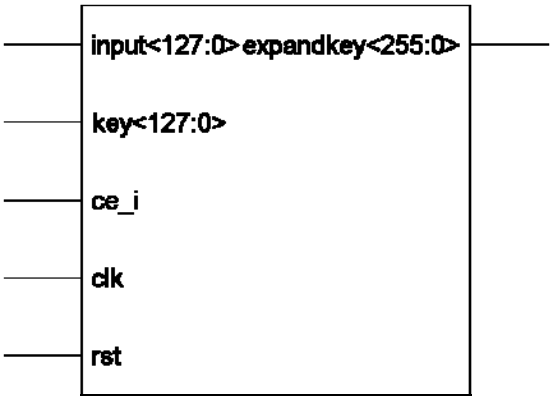

Figure 8: Schematic Diagram for Encoder

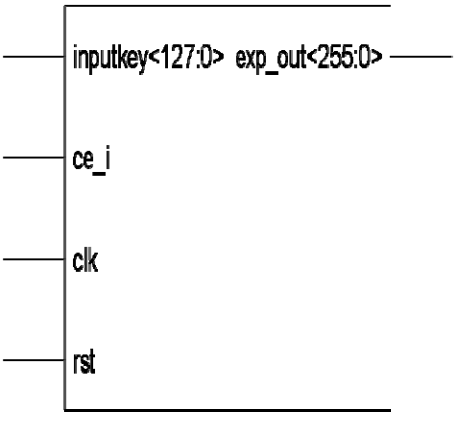

Figure 9: Schematic Diagram for key expansion unit

4. IMAGE ENCRYPTION

An image can be encrypted by combining MATLAB with the encoder. Each pixel in an image is represented by 8 bits,ie 1 byte.Using MATLAB convert the pixel values into bytes.These byte values are then used as input to the encoder. The 128 bit encoder then convert this byte into corresponding encoded byte. The encoded bit values are then converted into decimal values for pixels. Repeat this operation for each pixels (Figure 10).

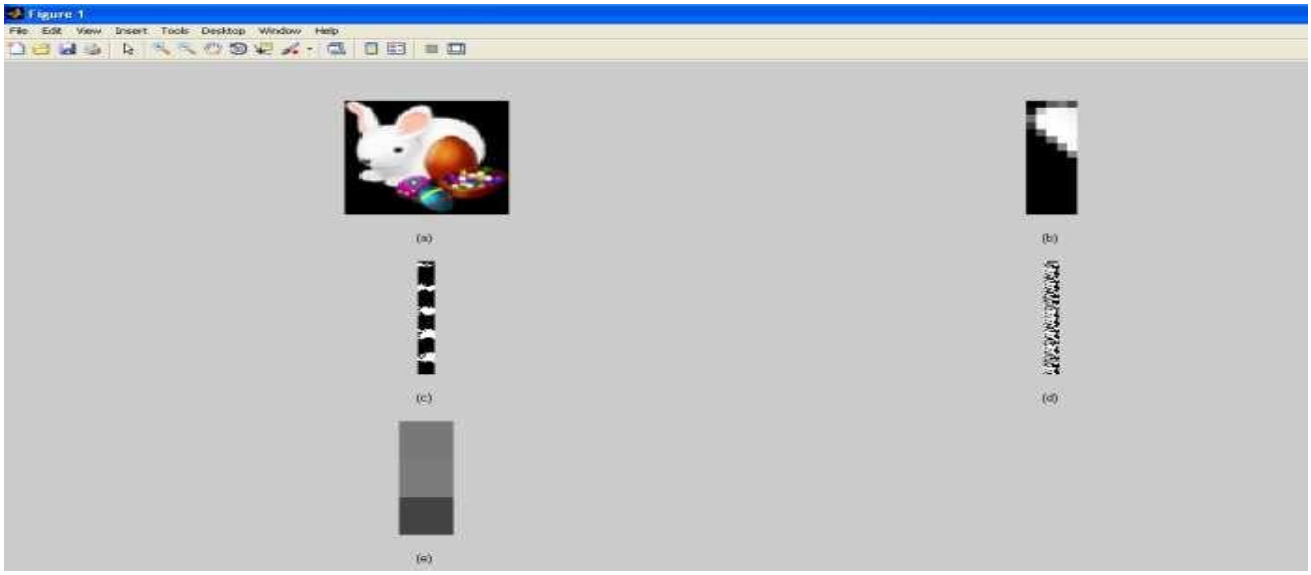

Figure 10:Image encryption

(a) Original image, (b) Desired part of the image
(c) Binary image of (b),(d) Encrypted binary image of (b)
(e) Encrypted image

5. REFERENCES

[1]. Bruce schneier"*Applied Cryptography*" 2nd Edition published by John Wiley&SonsInc.

[2]. William stallings "*Cryptography and Network Security*" 3rd Edition published by Pearson Education Inc and Dorling Kindersley Publishing Inc.

[3]. M. Zeghid, M. Machhout, L. Khriji, A. Baganne, and R. Tourki "*A Modified AES Based Algorithm for ImageEncryption*" World Academy of Science, Engineering and Technology 27, 2007

[4]. Abdelfatah A. Yahya and Ayman M. Abdalla"*A Shuffle Image-Encryption Algorithm*"Department of Computer Science, Al-Zaytoonah University of Jordan, Journal of Computer Science 4 (12): 999-1002, 2008

[5]. Xinmiao Zhang, Student Member,IEEE, and Keshab K. Parthi, Fellow, IEEE "*High-Speed VLSI Architecture for AES Algorithm*" IEEE Transactions on VLSI, Vol.12, No.19, September 2004

[6]. Alireza Hodjat, Student Member, IEEE, and Ingrid Verbauwhede, Senior Member, IEEE "*Area-Throughput Trade-Offs for Fully Pipelined30 to 70 Gbits/s AES Processors*" IEEE Transactions on Computers, Vol.55, no.4, April2006

[7]. Pawel Chodowiec and Kris Gaj "*Very compact FPGA implementation of the AES Algorithm*",in Proc. Of Cryptographic hardware and embedded system workshop,pp.319-333,2003

[8]. F.Rodriguez-Henriquez,N.A Saquib and A. Diaz-Perez"*4.2 Gbits/secSingle Chip FPGA implementation of the AES Algorithm*", ElectronicsLetters, Vol.39, No.15, pp.1115-1116,2003

[9]. N. Sklavos and O. Koufopavlou, Member, IEEE "*Architectures and VLSI Implementations of the AES-Proposal Rijndael*" IEEE Transactions on Computers, Vol. 51, No. 12, December2002.

[10]. J.Bhasker " *A VHDL Primer*".3rd Edition published by Pearson Education Inc and Dorling Kindersley Publishing Inc.

[11]. J. Elbirt, W. Yip, B. Chetwynd, and C. Paar. *An FPGA implementation and performance evaluation of the AES block cipher candidate algorithm finalist.* presented at Proc.3rdAESConf.(AES3).[Online].Available:http://csrc.nist.gov/encryption/aes/round2/conf3/aes3papers.html.

[12]. V. Fischer and M. Drutarovsky, "*Two methods of Rijndael implementation in reconfigurable hardware,*" in Proc. , pp. 77–92, CHES 2001, Paris, France,May 2001.

[13]. H. Kuo and I. Verbauwhede, "*Architectural optimization for a 1.82 Gbits/sec VLSI implementation of the AES Rijndael algorithm,*" in Proc. CHES 2001, pp. 51–64, Paris, France, May 2001.

[14]. M. McLoone and J. V. McCanny, "*Rijndael FPGA implementation utilizing look-up tables,*" in IEEEWorkshop on Signal Processing Systems, pp.349–360, Sept. 2001.