Performance Analysis of Software Reference Model of AES Algorithm for Secure Image Encoding

Renjith V Ravi, Dr.Mahalakshmi R

Abstract— with data security being one of the primary research topics in current day industry advancements over security, AES algorithm is one of the most popular techniques for data encryption. In this paper, we develop software reference model for AES algorithm and validate its performance in terms of encryption and decryption capability using various sets of data vectors. For secure image encoding required for unmanned vehicles, Discrete Wavelet Transform (DWT) is adopted for image decomposition, quantization and selection of appropriate sub bands is required prior to encryption to reduce computation time. In this paper, a modified algorithm for secure image encoding is proposed, modeled and is analyzed for its performance. Various images are considered as test cases for encoding, with parallel operation of AES algorithm the total time in encoding the image is reduced to 2 seconds (estimated based on software reference model). PSNR for various images obtained demonstrate the performances of the proposed model for image encoding. Image size of 1024x1024 is considered for encryption and decryption. The results obtained demonstrate the performances of AES algorithm.

Key words: AES algorithm, encryption, decryption, software model, image encoding

1. INTRODUCTION:

Adaptations of autonomous vehicles are growing especially in applications such as manufacturing, hazardous materials handling, surveillance, remote sensing, and defense sector and homeland security. The basic task in any such application is the perception of the environment through one or more sensors predominantly by using image sensors. Many different types of unmanned vehicles (UVs) are being developed for use in Unmanned Aerial Vehicle (UAV), Unmanned Ground Vehicle (UGV), and Autonomous Underwater Vehicle (AUV) environments. Unmanned robotics is actively being developed for both civilian and military use to perform dull, dirty, and dangerous activities. These unmanned vehicles (UVs) are remote-operated and the vehicles are controlled by a human operator via a communications link. Control actions are determined by the operator based upon either direct visual observation or remote viewing through a camera. Since digital video or images taken from UVs transmission system usually includes a compression module that aims to reduce the transmitted bit rate. During transmission over a public communication link encryption of compressed data is very important and hence the cryptography techniques have to be carefully designed. In this work, encryption of the image is carried out using particular key while getting the original image the same key is used. Data transfer through public network is always unsecured, thus security is one of the major challenges that need to be addressed to ensure the reliable data transfer. The security problem therefore becomes an important issue in today's wired or wireless Internet applications. One of the most useful methods to protect data is employing a cryptographic system, as the design of cipher algorithms is based on an advanced mathematical theorem. It usually mixes different types of cryptosystems in a secure protocol to provide a safe channel for data transmission. Generally speaking, asymmetric-key cryptosystems, and RSA here stands for Rivest, Shamir and Adleman who first publicly described it in 1978. Symmetric-key cryptosystems, such as Data Encryption Standard (DES) or Advanced Encryption Standard (AES), are used to encrypt bulk data in the transmission phase. Due to limited computing resources in portable applications, the system usually offloads the security process to dedicated special hardware. Recently, there have been many works on designing cost-effective encryption hardware used in portable applications [1]–[10]. Some works [1]–[5] focus on area reduction of AES, while others [6]–[10] propose to reduce hardware cost for both ECC and RSA cryptosystems. Image processing is finding importance in various applications as ever since last 10 years. Multimedia applications are dominated by image processing. It is mandatory to secure or protect multimedia content from unauthorized access. Protecting the image or video content in a multimedia data is of primary importance as image conveys more information than any other source of data. Images are large in size and require large storage unit, hence encryption of image content is also time consuming [14]. Traditional encryption algorithm make encryption very time consuming for large size of image data, hence encryption algorithms should be customized for larger image sizes and need to be faster. [13,14,15] reports that symmetric key algorithm have computational time less than asymmetric key algorithms. Symmetric key algorithms such as AES, DES have been successfully used

- *Renjith V Ravi is currently pursuing Ph.D program in electronics and communication engineering at Karpagam Academy of Higher Education, India, PH-+919895031051. E-mail: renjith_v_ravi@yahoo.com*
- *Dr.R. Mahalakshmi is currently working as Professor and HOD of Electrical and Electronics at Sree Krishna College of Technology Coimbatore, India*

for encryption of data, hardware algorithms for AES have made them very fast and hence consume very lees time, however for images with large data size AES is still time consuming. Image processing is finding importance in various applications as ever since last 10 years. Multimedia applications are dominated by image processing. It is mandatory to secure or protect multimedia content from unauthorized access. Protecting the image or video content in a multimedia data is of primary importance as image conveys more information than any other source of data. Images are large in size and require large storage unit, hence encryption of image content is also time consuming [12]. Traditional encryption algorithm make encryption very time consuming for large size of image data, hence encryption algorithms should be customized for larger image sizes and need to be faster. [11, 12, 13] reports that symmetric key algorithm have computational time less than asymmetric key algorithms. Symmetric key algorithms such as AES, DES have been successfully used for encryption of data, hardware algorithms for AES have made them very fast and hence consume very lees time, however for images with large data size AES is still time consuming. In [14, 15, 16] fast symmetric architectures for hardware implementation of AES algorithm is reported. These algorithms have not been validated for image encoding. A central consideration for any cryptographic system is its susceptibility to possible attacks against the encryption algorithm such as statistical attack, differential attack, and various brute attacks. Errors in channel also corrupt the encrypted data and hence there is a need for suitable technique that can be used to detect and correct the errors. In this paper, we analyze the performances of AES algorithm for various inputs, keys and noise in channel. Performance analysis carried out helps in identifying a suitable error correcting algorithm for AES. Section II discusses cryptography algorithm in brief. Section III discusses software algorithm for AES, section IV discusses software reference model development for AES algorithm. Section V presents results and discussion and conclusion is presented in Section VI.

2. SECURE IMAGE ENCODING SCHEME

Secured image encoding is one of the novel approaches that have been adopted in UVs for data transmission to the base station. The input image captured by the UVs is transformed using Discrete Wavelet Transform (DWT) to obtain various sub bands, the sub bands are quantized and the quantized sub bands are encrypted. The encoding technique such as Huffman encodes the encrypted data and compresses the data captured, and is transmitted to the base station. Figure 1 shows the block diagram of secured image coding [17]. DWT has been widely used in many different fields of audio and video signal processing. DWT is being increasingly used as effective solutions to the problem of image compression.

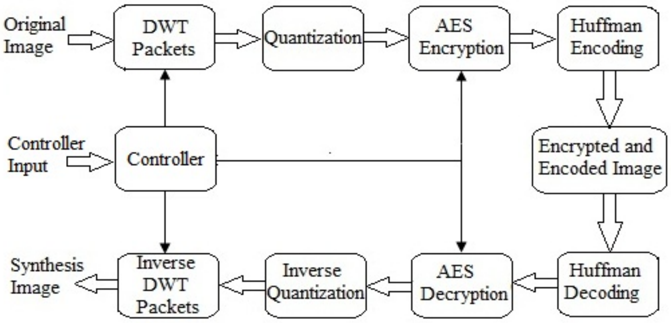
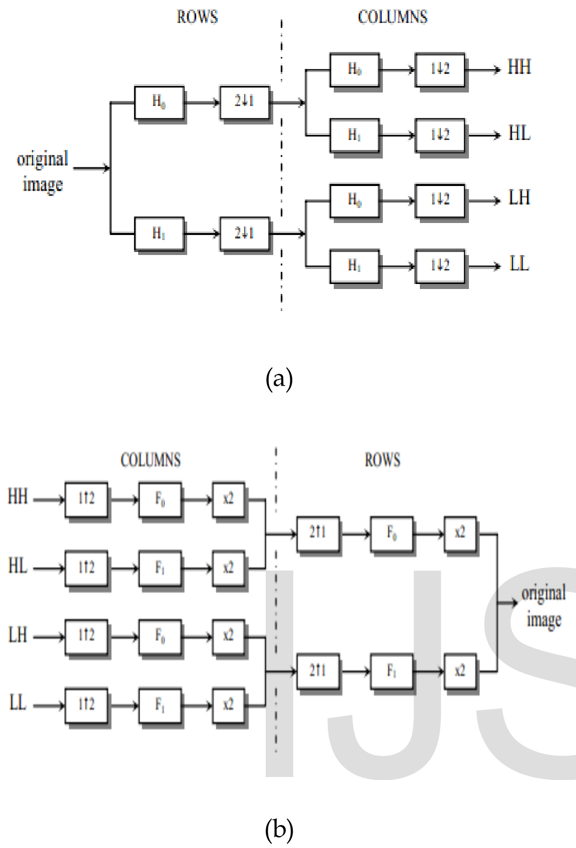


Fig 1 Secure Image Codec [17]

Quantizer is the process of approximating the continuous set of values in the image data with a finite set of values. The design of the quantizer has a significant impact on the amount of compression obtained and loss incurred in a compression scheme. AES is a block cipher with variable key length (128-bit, 192-bit, and 256-bit respectively) and block size of 128-bit. AES need very low memory to make it very well suited for restricted-space environments, in which it also demonstrates excellent performance. Huffman coding is a form of encoding that creates the most efficient set of prefix codes for a given text. The principle is to use a lower number of bits to encode the data that occurs more frequently. The controller is a module needed for optimizing applications security requirements based on a variable system resources. Users can define their security requirements for a particular security service by specifying a security range. G. Liu, T. Ikenaga, S. Goto and T. Baba in their paper have proposed a new video security scheme, which includes two encryption methods. The prominent feature of this method is a shuffling of AC events generated after DCT transformation and quantization stages [18]. DCT introduces blocking artifacts and hence, DWT is adopted. M. Zeghid, M. Machhout, L. Khriji, A. Baganne and R. Tourki [19], in their paper, they analyzed the Advanced Encryption Standard (AES), and they add a key stream generator (A5/1, W7) to AES to ensure improving the encryption performance; mainly for images characterized by reduced entropy. Hardware implementation of DWT with encryption imposes major challenges and have been discussed in [20][21][22],[23]. In this paper we analyze the performance of secure image coding using software reference model.

2.1 DWT for Image Compression

The two-dimensional DWT is becoming one of the standard tools for image fusion in image and signal processing field. The DWT process is carried out by successive low pass and high pass filtering of the digital image or images. This process is called the Mallat algorithm or Mallat-tree decomposition. Figure 2 shows an implementation structure of the 2-D DWT-IDWT. The first level of transformation is performed along the rows and the second level of transformation occurs along the column. The four sub band components (LL, LH, HL and HH) capture the low frequency components (DC component), high frequency compoenents (edges along vertical, horizontal and diagonal axis). On the reverse process, the original image is reconstructed based on inverse transformation process using IDWT.



(a)

(b)

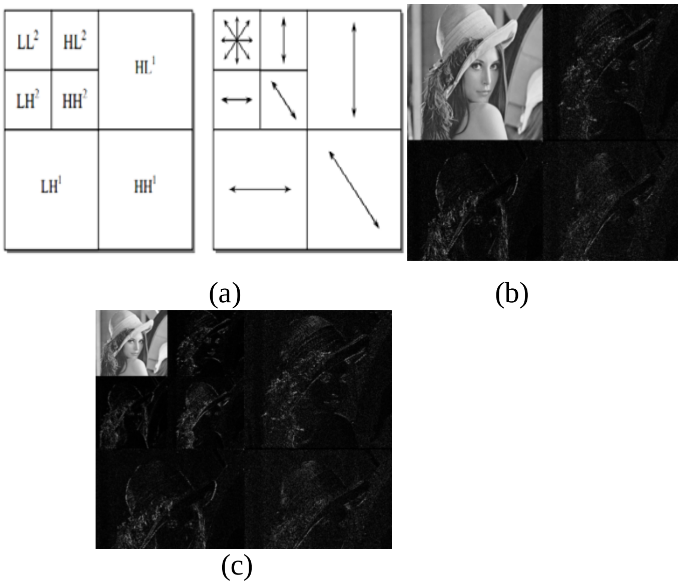
Fig 2 Analysis filter bank structure of (a) DWT and (b) IDWT [18]

The filter coefficients of high pass and low pass sub bands need to satisfy the property are shown in Eq. 1 and Eq. 2 respectively for perfect reconstruction.

$$\frac{1}{2}\left[H_0\left(z^{1/2}\right)X\left(z^{1/2}\right)+H_0\left(-z^{1/2}\right)X\left(-z^{1/2}\right)\right] \quad \text{...........Eq. (1)}$$

$$\frac{1}{2}\left[H_1\left(z^{1/2}\right)X\left(z^{1/2}\right)+H_1\left(-z^{1/2}\right)X\left(-z^{1/2}\right)\right] \quad \text{...........Eq. (2)}$$

Figure 3 shows the pyrimidal decompostion of input image using DWT. In the first level, input image is decomposed into four sub bands (LL,LH,HL, HH), the LL sub band compoenent is furhter decomposed into four more sub band component in the second level. Figure 3(a) shows the results of first level decomposition, Figure 3(b) shows the second level decomposition stage. The information is actually present in the LL sub band the other three sub bands provide information on edges of a given object in the original image.



(a)

(b)

(c)

Fig 3 (a), (b) and (c) Image decomposition step of DWT

Image compression is achieved by quantizing the higher sub bands that are not very significant and transmitting the LL sub band without quantization. An input image of size N x N after two level decomposition will give rise to 7 sub bands (three higher level sub bans of size N/2 x N/2 at the first level, three N/4 x N/4 at the second level and one low frequency sub band of N/4 x N/4). Image compression is achieved by choosing only the significant sub bands that provide information and quantizing all other sub bands.

2.2 AES Encryption and Decryption

Advanced Encryption Standard (AES) is a symmetric block cipher that processes data blocks of 128 bits using the cipher key of length 128, 192, or 256 bits. The AES algorithm [2] organizes the data block in a four-row and row-major ordered matrix. The original AES encryption/decryption procedure is shown in Figure 1. In both encryption and decryption, the AES algorithm uses a round function, which consists of four different byte-oriented transformations:

1) Sub Bytes substitutes each State of the data block with a substitution table (S-box) value of that byte.
2) Shift Rows shifts the each row of the State array by different offsets cyclically, and the offset depends on row-index.
3) Mix Columns transforms each column of the matrix by multiplying it with a constant *GF* polynomial.
4) Add Round Key adds a Round Key to the State by a simple bitwise XOR operation.

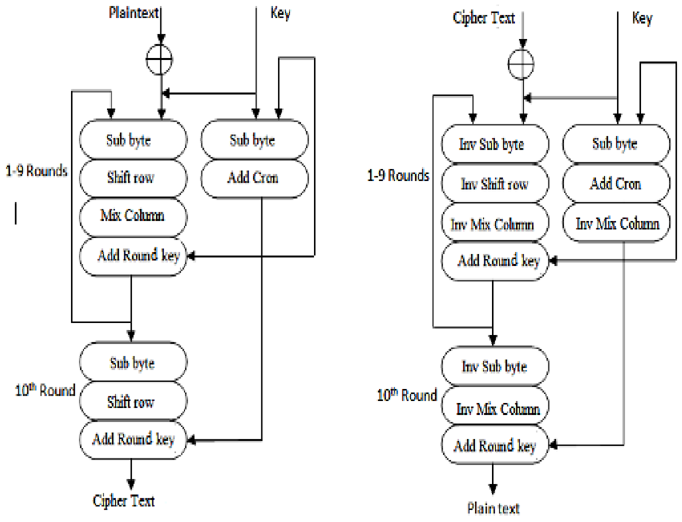


Fig 4 AES Encryption and Decryption

In AES algorithm, each 128-bit information is arranged as a 4 x 4 state, operated by four primitive transformations. During the encryption/decryption process, the four primitive transformations are executed iteratively in N_r rounds, where the value of N_r will be 10, 12, or 14, depending on which key size is selected. In the encryption procedure, the incoming data will first be bitwise XORed with an initial key, and then, four transformations are executed in the following order: Sub-Bytes, ShiftRows, MixColumns, and AddRoundKey. Notice that the MixColumns transformation is not performed in the last round. The execution sequence is reversed in the decryption process, where their inverse transformations are InvSubBytes, InvShiftRows, InvMixColumns, and AddRoundKey, respectively. Since each round needs a round key, an initial key is used to generate all round keys before encryption/decryption. In the AES algorithm, the SubBytes transformation is a nonlinear byte substitution composed of two operations: 1) 1) Modular inversion over GF(2^8), modulo an irreducible polynomial $p(x) = x^8 + x^4 + x^3 + x + 1$ and 2) affine transformation defined as y=Mx +v, where M is an 8 x 8 b matrix, v is an 8-b constant, and x/y denotes 8-b input/output. In the MixColumns transformation, the 128-b data arranged as a 4 x 4 state are operated column by column. The four elements of each column form a four-term polynomial that is multiplied by a constant polynomial $C(x) = \{03\} x^3 + \{01\}x^2 + \{01\}x + \{02\}$ modulo $x^4 + 1$. The ShiftRows transformation is a simple operation in which each row of the state is cyclically shifted right by different offsets. The AddRoundKey transformation is a bitwise XOR operation of each round key and current state.

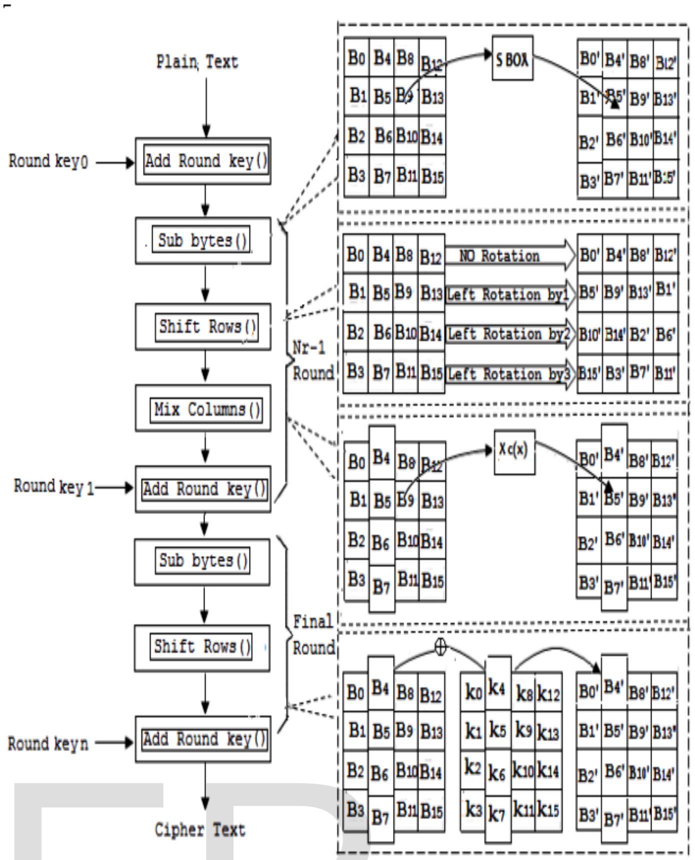


Fig 5 AES algorithm

S-box generation is carried out in two steps, first find Inverse of the element over GF(2^8) modulo the irreducible polynomial given in Eq. 1

$$x^8 + x^4 + x^3 + x + 1 \text{ ----------------------------------- Eq.1}$$

Second, apply affine transformation of the form y = Mx + C to the inverse, where M = 8 x 8 bit matrix, C = 8-bit constant and x/y = 8-bit input/output. Shift rows means cyclic shift of each row to the left by a predefined offset as shown in Figure 2. Mix column operates on each column individually. Each byte is mapped into a new value which is a function of all 4 bytes in that column. In Add Round Key, the 128 bits of state are bitwise XORed with 128 bits of the round key. Each round key generated in the key expansion and scheduling process. 10 rounds of the whole AES process are repeated for a key length of 128 bit. The round keys are generated by a key expansion process. The expanded key is 176 bytes long. Software modeling of AES encryption algorithm using Matlab has been carried out and various transformations used in the algorithm in converting plain text to cipher text were studied. The software modelling has been carried out such that the main program calls the initialization function and the encryption function which in turn calls other sub functions to accomplish the task of encryption. Next section discusses the software model for AES algorithm

3. SOFTWARE REFERENCE MODEL FOR AES ALGORITHM

In this work, we propose a software reference model to analyze the performances of AES algorithm. The algorithm discussed in Figure 4 is modeled in Matlab. The input operands are expressed in Hexadecimal format, the input operand range is set between 0 to 255 and hence 8 bits are used for representation using hexadecimal number representation format. The plain text or initial state of 128 bits of data is arranged as a 4 x 4 matrix of 16 bytes and a round key of length 128 bits is also generated from the initial key. The input data is transformed using each transformations namely sub_bytes, shift_rows, mix_columns and add round key. Finally the cipher text obtained after 10 such repetitions (rounds) is computed. The initialization function returns the expanded key schedule w, the, substitution table s_box and the polynomial matrix poly_mat. It uses function s_box_gen which in turn calls functions find_inverse and affine_transform for generating the substitution box. Every element in the substitution box is obtained by performing an affine transformation on the multiplicative inverse of the element in the binary extension field GF (2^8). To obtain the key schedule 'w', AES_init function calls the functions sub_box, rot_word and rcon_gen. The key expansion function takes the user supplied 16 bytes long key and utilizes the previously created round constant matrix rcon and the substitution table s_box to generate a 176 byte long key schedule w, which will be used during the encryption processes. The substitution table, s_box is used by the expanded key schedule function key_expansion and the encryption function cipher to directly substitute a byte (element of GF(2^8)) by another byte of the same finite field. The function s_box_gen creates the S-box by searching for the inverses of all elements of GF(2^8) by the use of find_inverse and by applying affine transformations to all inverses using aff_trans function. mod_pol denotes the standard AES modular reduction polynomial and is given as 283d = 100011011b = $x^8 + x^4 + x^3 + x + 1$. Key expansion and polynomial generation function steps are crried out to develop the final block diagram of AES algorithm. The main function Cipher takes s_box , key schedule and poly_mat generated by the AES_init function as well as the 16 byte plain text and produces the encrypted text. It re-arranges the plain text into a 4x4 byte state matrix and calls functions Add_round_key, Sub_Bytes, Shift_ Row and Mix_Columns in order to generate cipher text.

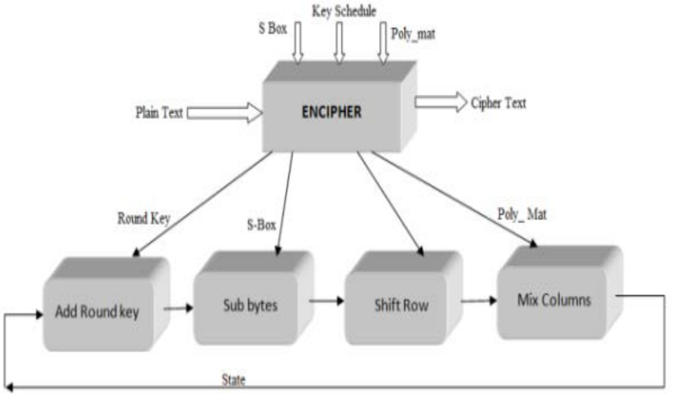
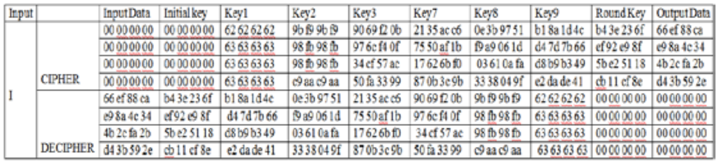


Fig 6 Cipher function for encryption

Figure 6 shows the block diagram of 'Cipher' function for encryption. In Figure 6, add_round_key function performs a bitwise xor of the state matrix and the round key matrix. The function shift_rows cyclically permutes (shifts) the rows of the state matrix to the left. The mix_columns transformation computes the new state matrix S0 by left-multiplying the current state matrix S by the polynomial matrix P.

3.1. Image Encoding using AES

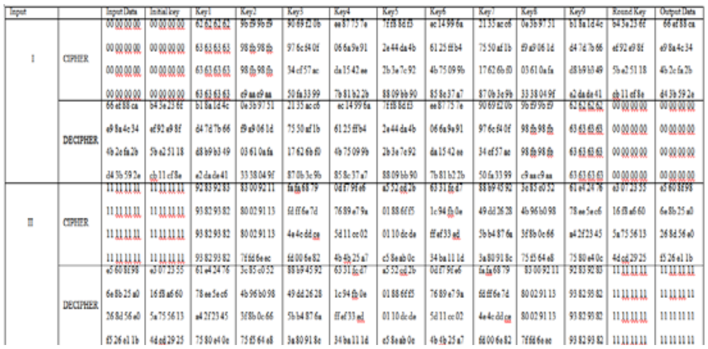
In this work, images that are of size 512 x 512 are encoded using AES algorithm developed in the previous section. Input image of size 512 x 512 which is a gray scale image is represented using 8 bits per pixel. The total number of bits in one frame is 512*512*8 = 2097152 bits. AES algorithm encodes 128 bits, hence the image is sub divided into 16384 frames and each frame is encoded and decoded using the AES algorithm. Figure 7 shows the results of encryption and decryption using 9 sets of keys (Keys 4 to 6 are not shown in figure 10). These are the results with different input plaintexts with using same key for encryption.



Input		InputData	Initial key	Key1	Key2	Key3	Key7	Key8	Key9	RoundKey	OutputData
I	CIPHER	00 00 00 00	00 00 00 00	62 62 62 62	9b f9 9b f9	90 69 f2 0b	21 35 ac c6	0e 3b 97 51	b1 8a 1d 4c	b4 3e 23 6f	66 ef 88 ca
		00 00 00 00	00 00 00 00	63 63 63 63	98 fb 98 fb	97 6c f4 0f	75 50 af 1b	f9 a9 06 1d	d4 7d 7b 66	ef 92 e9 8f	e9 8a 4c 34
		00 00 00 00	00 00 00 00	63 63 63 63	98 fb 98 fb	34 cf 57 ac	17 62 6b f0	03 61 0a fa	d8 b9 b3 49	5b e2 51 18	4b 2c fa 2b
		00 00 00 00	00 00 00 00	63 63 63 63	c9 aa c9 aa	50 fa 33 99	87 0b 3c 9b	33 38 04 9f	e2 da de 41	cb 11 cf 8e	d4 3b 59 2e
	DECIPHER	66 ef 88 ca	b4 3e 23 6f	b1 8a 1d 4c	0e 3b 97 51	21 35 ac c6	90 69 f2 0b	9b f9 9b f9	62 62 62 62	00 00 00 00	00 00 00 00
		e9 8a 4c 34	ef 92 e9 8f	d4 7d 7b 66	f9 a9 06 1d	75 50 af 1b	97 6c f4 0f	98 fb 98 fb	63 63 63 63	00 00 00 00	00 00 00 00
		4b 2c fa 2b	5b e2 51 18	d8 b9 b3 49	03 61 0a fa	17 62 6b f0	34 cf 57 ac	98 fb 98 fb	63 63 63 63	00 00 00 00	00 00 00 00
		d4 3b 59 2e	cb 11 cf 8e	e2 da de 41	33 38 04 9f	87 0b 3c 9b	50 fa 33 99	c9 aa c9 aa	63 63 63 63	00 00 00 00	00 00 00 00

Fig 7 Results of encryption and decryption

In the next set of test analysis, 128 bit of input is encrypted using various sets of keys. In order to validate the performances of AES algorithm, input bits with all '0's and all '1's have been used as test vector to carry out the analysis. Figure 8 shows the results obtained. These are the results with same input plaintexts with using different key for encryption.

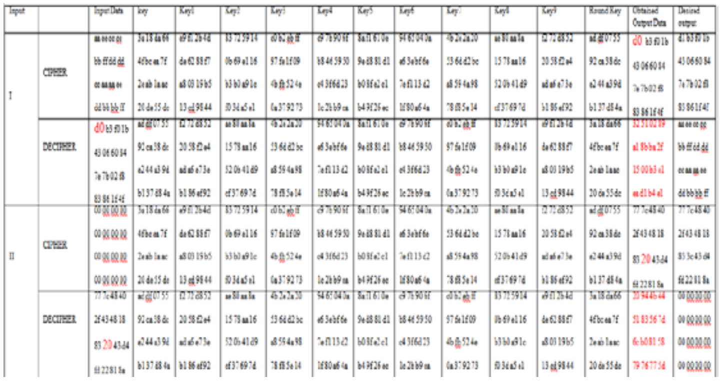


Input		Input Data	Initial key	Key1	Key2	Key3	Key4	Key5	Key6	Key7	Key8	Key9	Round Key	Output Data
I	CIPHER	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	62 62 62 62 63 63 63 63 63 63 63 63 63 63 63 63	9b f9 9b f9 98 fb 98 fb 98 fb 98 fb c9 aa c9 aa	90 69 f2 0b 97 6c f4 0f 34 cf 57 ac 50 fa 33 99	ee 87 75 7e 06 6a 9e 91 da 15 42 ee 7b 81 b2 2b	7f f8 8d f3 2e 44 da 4b 2b 3e 7c 92 88 09 bb 90	ec 14 99 6a 61 25 ff b4 4b 75 09 9b 85 8c 37 a7	21 35 ac c6 75 50 af 1b 17 62 6b f0 87 0b 3c 9b	0e 3b 97 51 f9 a9 06 1d 03 61 0a fa 33 38 04 9f	b1 8a 1d 4c d4 7d 7b 66 d8 b9 b3 49 e2 da de 41	b4 3e 23 6f ef 92 e9 8f 5b e2 51 18 cb 11 cf 8e	66 ef 88 ca e9 8a 4c 34 4b 2c fa 2b d4 3b 59 2e
	DECIPHER	66 ef 88 ca e9 8a 4c 34 4b 2c fa 2b d4 3b 59 2e	b4 3e 23 6f ef 92 e9 8f 5b e2 51 18 cb 11 cf 8e	b1 8a 1d 4c d4 7d 7b 66 d8 b9 b3 49 e2 da de 41	0e 3b 97 51 f9 a9 06 1d 03 61 0a fa 33 38 04 9f	21 35 ac c6 75 50 af 1b 17 62 6b f0 87 0b 3c 9b	ec 14 99 6a 61 25 ff b4 4b 75 09 9b 85 8c 37 a7	7f f8 8d f3 2e 44 da 4b 2b 3e 7c 92 88 09 bb 90	ee 87 75 7e 06 6a 9e 91 da 15 42 ee 7b 81 b2 2b	90 69 f2 0b 97 6c f4 0f 34 cf 57 ac 50 fa 33 99	9b f9 9b f9 98 fb 98 fb 98 fb 98 fb c9 aa c9 aa	62 62 62 62 63 63 63 63 63 63 63 63 63 63 63 63	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
II	CIPHER	11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11	11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11	92 83 92 83 93 82 93 82 93 82 93 82 93 82 93 82	83 00 92 11 80 02 91 13 80 02 91 13 7f fd 6e ec	fa fa 68 79 fd ff 6e 7d 4e 4c dd ce fd 00 6e 82	0d f7 9f e6 76 89 e7 9a 5d 11 cc 02 4b 4b 25 a7	a5 52 cd 2b 01 88 6f f5 01 10 dc de c5 8e ab 0c	63 31 fc d7 1c 94 fb 0e ff ef 33 ed 34 ba 11 1d	88 b9 45 92 49 dd 26 28 5b b4 87 6a 3a 80 91 8c	3c 85 c0 52 4b 96 b0 98 3f 8b 0c 66 75 f5 64 e8	61 e4 24 76 78 ee 5e c6 a4 2f 23 45 75 80 e4 0c	e3 07 23 55 16 f8 a6 60 5a 75 56 13 4d cd 29 25	e5 60 8f 98 6e 8b 25 a0 26 8d 56 e0 f5 26 e1 1b
	DECIPHER	e5 60 8f 98 6e 8b 25 a0 26 8d 56 e0 f5 26 e1 1b	e3 07 23 55 16 f8 a6 60 5a 75 56 13 4d cd 29 25	61 e4 24 76 78 ee 5e c6 a4 2f 23 45 75 80 e4 0c	3c 85 c0 52 4b 96 b0 98 3f 8b 0c 66 75 f5 64 e8	88 b9 45 92 49 dd 26 28 5b b4 87 6a 3a 80 91 8c	63 31 fc d7 1c 94 fb 0e ff ef 33 ed 34 ba 11 1d	a5 52 cd 2b 01 88 6f f5 01 10 dc de c5 8e ab 0c	0d f7 9f e6 76 89 e7 9a 5d 11 cc 02 4b 4b 25 a7	fa fa 68 79 fd ff 6e 7d 4e 4c dd ce fd 00 6e 82	83 00 92 11 80 02 91 13 80 02 91 13 7f fd 6e ec	92 83 92 83 93 82 93 82 93 82 93 82 93 82 93 82	11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11	11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11

Fig 8 Simulation results with same input and different keys

For the results shown in Figure 7 and Figure 8, the last columns show the encrypted output and decrypted output. The decrypted output matches with the input shown in column one. Experimental analyses have also been carried out to demonstrate AES algorithm to decrypt using two different keys for encryption and decryption respectively. The input data sets chosen for analysis are encrypted using one set of keys and decrypted using another set of keys. From the results obtained it is found that the decrypted data does not match with the input operand, thus proving the effectiveness of encryption algorithm.

Further in this work, in order to analyze the performance of AES algorithm, the encrypted data is introduced with noise and is decrypted using the keys. The results shown in Figure 9 demonstrates that for the selected set of test vectors the AES algorithm is not able to reconstruct the correct output when the error is introduced. Hence it is found that the transmitted data over channel when corrupted by noise, cannot be decrypted with the keys used for encryption. These are the results with different input plaintexts with using same key for encryption with error obtained at output side of cipher. The error bits where marked with red colour text.



Input		Input Data	key	Key1	Key2	Key3	Key4	Key5	Key6	Key7	Key8	Key9	Round Key	Obtained Output Data	Desired output
I	CIPHER	aa ee cc gg bb ff dd dd cc aa gg ee dd bb bb ff	3a 18 da 66 4f bc ea 7f 2e ab 1a ac 20 de 55 dc	e9 f1 2b 4d de 62 88 f7 a8 03 19 b5 13 cd 98 44	83 72 59 14 0b 69 e1 16 b3 b0 a9 1c f0 3d a5 e1	c0 b2 eb ff 97 fe 1f 09 4b fb 52 4e 0a 37 92 73	c9 7b 90 6f b8 46 59 50 c4 3f 6d 23 1c 2b b9 ca	8a f1 61 0e 9e d8 81 d1 b0 8f e2 c1 b4 9f 26 ec	94 65 04 0a e6 3e bf 6e 7e f1 13 d2 1f 80 a6 4a	4b 2e 2a 20 53 6d d2 bc a8 59 4a 98 78 f8 5e 14	ae 80 aa 8a 15 78 aa 16 52 0b 41 d9 cf 37 69 7d	f2 72 d8 52 20 58 f2 e4 ad a6 e7 3e b1 86 ef 92	ad df 07 55 92 ca 38 dc e2 44 a3 9d b1 37 d8 4a	d0 b3 f0 1b 43 06 60 84 7e 7b 02 f8 83 86 1f 4f	d1 b3 f0 1b 43 06 60 84 7e 7b 02 f8 83 86 1f 4f
	DECIPHER	d0 b3 f0 1b 43 06 60 84 7e 7b 02 f8 83 86 1f 4f	ad df 07 55 92 ca 38 dc e2 44 a3 9d b1 37 d8 4a	f2 72 d8 52 20 58 f2 e4 ad a6 e7 3e b1 86 ef 92	ae 80 aa 8a 15 78 aa 16 52 0b 41 d9 cf 37 69 7d	4b 2e 2a 20 53 6d d2 bc a8 59 4a 98 78 f8 5e 14	94 65 04 0a e6 3e bf 6e 7e f1 13 d2 1f 80 a6 4a	8a f1 61 0e 9e d8 81 d1 b0 8f e2 c1 b4 9f 26 ec	c9 7b 90 6f b8 46 59 50 c4 3f 6d 23 1c 2b b9 ca	c0 b2 eb ff 97 fe 1f 09 4b fb 52 4e 0a 37 92 73	83 72 59 14 0b 69 e1 16 b3 b0 a9 1c f0 3d a5 e1	e9 f1 2b 4d de 62 88 f7 a8 03 19 b5 13 cd 98 44	3a 18 da 66 4f bc ea 7f 2e ab 1a ac 20 de 55 dc	32 51 02 f9 a1 8b ba 2f 15 00 b3 e1 ea d1 b4 e1	aa ee cc gg bb ff dd dd cc aa gg ee dd bb bb ff
II	CIPHER	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	3a 18 da 66 4f bc ea 7f 2e ab 1a ac 20 de 55 dc	e9 f1 2b 4d de 62 88 f7 a8 03 19 b5 13 cd 98 44	83 72 59 14 0b 69 e1 16 b3 b0 a9 1c f0 3d a5 e1	c0 b2 eb ff 97 fe 1f 09 4b fb 52 4e 0a 37 92 73	c9 7b 90 6f b8 46 59 50 c4 3f 6d 23 1c 2b b9 ca	8a f1 61 0e 9e d8 81 d1 b0 8f e2 c1 b4 9f 26 ec	94 65 04 0a e6 3e bf 6e 7e f1 13 d2 1f 80 a6 4a	4b 2e 2a 20 53 6d d2 bc a8 59 4a 98 78 f8 5e 14	ae 80 aa 8a 15 78 aa 16 52 0b 41 d9 cf 37 69 7d	f2 72 d8 52 20 58 f2 e4 ad a6 e7 3e b1 86 ef 92	ad df 07 55 92 ca 38 dc e2 44 a3 9d b1 37 d8 4a	77 7c 48 40 2f 43 48 18 83 20 43 d4 6f 22 81 8a	77 7c 48 40 2f 43 48 18 83 3c 43 d4 6f 22 81 8a
	DECIPHER	77 7c 48 40 2f 43 48 18 83 20 43 d4 6f 22 81 8a	ad df 07 55 92 ca 38 dc e2 44 a3 9d b1 37 d8 4a	f2 72 d8 52 20 58 f2 e4 ad a6 e7 3e b1 86 ef 92	ae 80 aa 8a 15 78 aa 16 52 0b 41 d9 cf 37 69 7d	4b 2e 2a 20 53 6d d2 bc a8 59 4a 98 78 f8 5e 14	94 65 04 0a e6 3e bf 6e 7e f1 13 d2 1f 80 a6 4a	8a f1 61 0e 9e d8 81 d1 b0 8f e2 c1 b4 9f 26 ec	c9 7b 90 6f b8 46 59 50 c4 3f 6d 23 1c 2b b9 ca	c0 b2 eb ff 97 fe 1f 09 4b fb 52 4e 0a 37 92 73	83 72 59 14 0b 69 e1 16 b3 b0 a9 1c f0 3d a5 e1	e9 f1 2b 4d de 62 88 f7 a8 03 19 b5 13 cd 98 44	3a 18 da 66 4f bc ea 7f 2e ab 1a ac 20 de 55 dc	20 94 4b 44 51 83 56 7d 6c b0 81 58 79 76 77 5d	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Fig 9 AES results with channel error

From the analysis carried out it is found that the AES algorithm consumes time as there are several iterations to be performed, and during each iteration the data manipulation is carried out using a set of keys. When input data source is image which has large set of data in terms of pixels, encryption of information takes more time; hence it is required to choose alternate techniques that can be combined with AES to minimize the processing time. Further error in channel corrupts the data and it becomes difficult to decrypt from the corrupted information. The algorithm developed in this work demonstrates the performances of AES for image encoding. One of the major limitations of the encoding scheme is that the input image which is of size 512 x 512 and consists of 16,384 frames of 128 bits, requires 16 seconds for encoding. The computation time is identified using inbuilt function in Matlab for compiling the program. Without the channel error, the PSNR for various images are computed and shown in Table 1. The PSNR is found to be in the range of 41dB to 52dB for various images. The average time for encoding is 16 seconds.

4. MODIFIED ALGORITHM FOR SECURE IMAGE CODING

The major limitations observed in image encoding are the total computation time in AES algorithm when applied to image data (16,384 frames). In order to reduce the computation time, input image is transformed to sub bands using DWT and each sub band is quantized and encoded using AES. The computation time is reduced and also is more suitable for real time applications. The modified block diagram for secure image encoding is shown in Figure 10. In the modified algorithm the input image is decomposed into 7 sub bands of high and low frequency components. The LL2 sub band is encoded using AES algorithm. The other sub bands are LH2, HL2 and HH1 are chosen and quantized. Thus the chosen sub bands are encrypted using AES algorithm independently. The total number of bits that are encoded after 2D DWT and quantization are shown in Table 1.

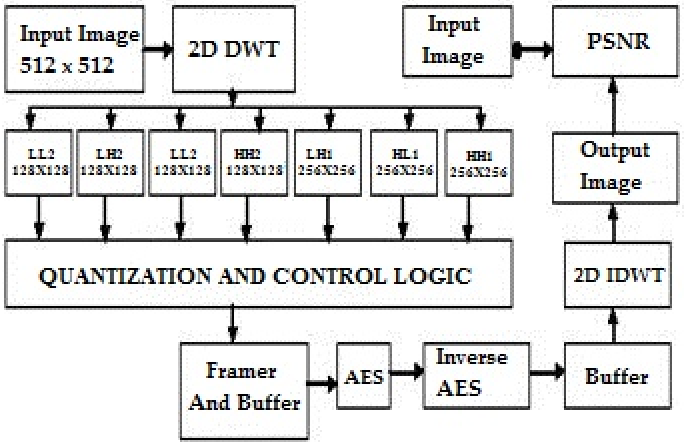


Fig 10 Modified Secure Image Encoding

In the modified encoding scheme LH2, HL2 and HH1 are chosen as they hold the high frequency components of the input image along the vertical, horizontal and diagonal directions. The vertical component and horizontal component information is captured from the second level decomposed sub band, the diagonal component is captured form the first level sub band component. Further, the LL2 component can be decomposed to four more sub bands in the third level and suitable sub band components can be quantized and encoded using AES algorithm independently.

TABLE 1

MODIFIED SECURE IMAGE ENCODING

Input Image Size	Bits per frame	LL2/LH2/ HL2/HH2 size (in No. of bits)	LH1/HL1/ HH1 size (in No. of bits)	No. of bits to be encoded using AES after selection
64 x 64	32768 (256)	2034	9126	15228 (119)
128 x 128	131072 (1024)	9126	36864	64242 (502)
256 x 256	524288(4096)	36864	147456	258048 (2016)
512 x 512	2097152(16384)	147456	589824	1032192(8064)
1024 x 1024	8388608(65536)	589824	2359296	4128768(32256)

The table 1 shown above presents the number bits encoded using AES algorithm for various image sizes. In the image encoding scheme without DWT, the number of frames to be encoded (each of 128 bits), are shown in brackets in column 2. After decomposition using 2D-DWT using two level, the number of frames to be encoded is shown in brackets in column 5. After DWT, the pixels are represented using 9 bits. The total number of frames to be encrypted after DWT is reduced by 50% thus the computation time for AES algorithm is reduced to less than 8 seconds (16 seconds is the time without DWT). As the AES algorithm is encoding the sub bands independently, the total time for encoding is less than 2 seconds in the modified algorithm.

5. RESULTS AND DISCUSSION

The modified algorithm proposed is modeled using Matlab and is verified for its functionality using various test images. Figure 11 shows the test images considered for encoding and decoding. The input image is decomposed into sub bands and the sub bands are encrypted using AES algorithm, the encrypted data is decrypted and inverse DWT is applied to obtain the original image. Peak Signal to Noise Ratio (PSNR) is computed to estimate the performances of the encoding scheme.

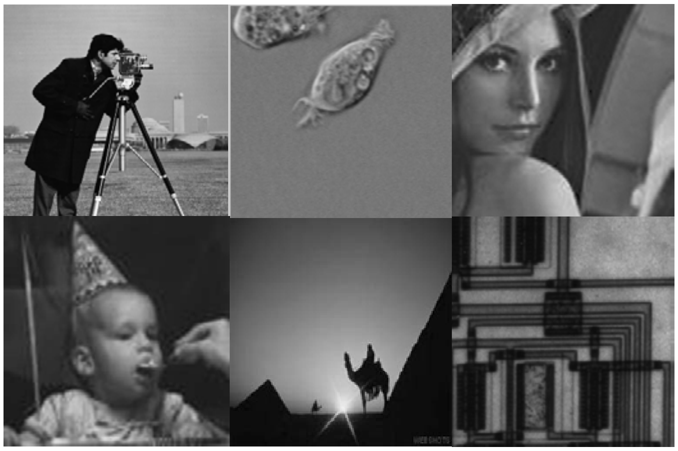


Fig 11 Test images for secure encoding

The software reference model developed in Matlab, performs DWT of each image and the decomposed sub bands are chosen based on the information available in the sub bands. The chosen sub bands are quantized and encrypted. The quantization process in this work compares the sub band coefficients with a threshold, the coefficients below threshold are made zero, and coefficients above the set threshold are retained. In this work, for the LL2 the threshold is set to +/-61, for the LH2 and HL2 the threshold is set to +/-136, and for the HH1 the threshold is set to +/- 212. The thresholds have been identified based on trial and error, so that the information in the decomposed image is not lost. After quantization the information is lost in the decomposed image, however the image size is reduced and thus reduces delay in AES computation. Table 2 shows the PSNR results of AES with DWT and without DWT. Quantization process introduces losses in the image, without quantization the PSNR is almost closer to the PSNR obtained without DWT.

TABLE 2

MODIFIED SECURE IMAGE CODING RESULTS

Test Image	PSNR (without DWT) in dB	PSNR (with DWT and without quantization) in dB	PSNR (with DWT and with quantization) in dB
cell	42	40.89	38.1
circuit	43.4	41.76	37.23
lena	45.3	43.71	40.1
girl	46.9	44.14	41.01
baby	51.65	46.23	42.2
sunset	50.43	47.12	41.19
water lilies	49.45	47.91	42.9
barbera	47.8	44.86	40.34
cameraman	48.7	46.3	39.87

PSNR results obtained without and with quantization are shown in column 3 and 4 respectively and are compared with the results of image coding with AES encoding in column 2. The maximum deviation in terms of PSNR for the chosen images is 4 dB and 9 dB without quantization and with quantization. In order to improve the PSNR and reduce computation time, the input image can be decomposed into multiple hierarchical sub bands and quantization threshold can be set appropriately to obtain the original image without loss. DWT is performed to reduce computation time in AES encoding on the input image directly. Further, performances of various DWT filters can also be estimated on the reconstruction process. In this work, DB4 wavelets have been used for decomposition and reconstruction. From the results obtained we demonstrate that the choice of DWT filter, selection of sub bands, quantization threshold and parallelism of AES algorithm plays a vital role in secure image encoding for Unmanned Vehicles.

6. CONCLUSION

Cryptography is the study of mathematics related information security. Data encryption leads to confidentiality, data integrity, entity authentication, and data origin authentication. Several standards for data encryption over the last few years, Advanced Encryption Standard (**AES**) algorithm is the best available private key cryptographic algorithm today. Encryption of data sources such as image and text requires time as AES is an iterative algorithm, encrypted data need to be secured, and channel noise effect need to be analyzed. In this work, the software reference model for AES is developed and is validated using various data sets and the performances of AES algorithm is estimated in terms of encryption and decryption capability. The results obtained demonstrate the advantages of AES for encryption of image and text data. The intermediate steps in AES can be fastened using suitable algorithms and can be used to encrypt complex input data sources.

REFERENCES

1. Chen-Hsing Wang, Chieh-Lin Chuang, and Cheng-Wen Wu, *An Efficient Multimode Multiplier Supporting AES and Fundamental Operations of Public-Key Cryptosystems*, IEEE Transactions on Very Large Scale Integration Systems (VLSI), Vol.18, No.4, pp.553-563, 2010

2. Satoh A., Morioka S., Takano K., and Munetoh S., *Unified hardware architecture for128-bit block ciphers AES and Camellia,* Proceedings of cryptographic Hardware and Embedded Systems, pp. 304–318, 2003

3. Bruce Schneir, Applied Cryptography, 2nd Edition, John Wiley and Sons Publishers, 1996

4. Herstein I. N., *Abstract Algebra,* Macmillan Publishing Company, 1990

5. William Stallings, Cryptography and Network Security Principles and Practices, 4th edition, Prentice Hall, 2007

6. Tenca A. F. and Koç C. K., *A scalable architecture for modular multiplication based on Montgomery's algorithm*, IEEE Transactions on Computer Science, Vol. 52, No. 9, pp. 1215–1221, 2003

7. Harris D., Krishnamurthy R., Anders M., Mathew S.,, and Hsu S., *An improved unified scalable radix-2 Montgomery multiplier*, Proceedings of 17th IEEE Symposium on Computer Arithmetic, pp. 172–178, 2005

8. A. Satoh and K. Takano, *A scalable dual-field elliptic curve cryptographic Processor*, IEEE Transactions on Computer Science, Vol. 52, No.4, pp.449–460, 2003

9. Wang J., Zeng X.,, and Chen J., *A VLSI implementation of ECC combined with AES,* Proceedings of International Conference on Solid State and Integrated Circuit Technology, pp. 1899–1904, 2006

10. Dominik Engel Thomas stutz,Andreas Uhl,"A survey on JPEF2000 encryption", Multimedia systems[online] SpringerLink Verlag pp.1 -29, ,2008.

11. Shtewi,A.M. "An Efficient Modified Advanced Encryption Standard (MAES) adapted for image cryptosystems" IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.2, pp 226-232 February 2010

12. Shiguo Lian," Quasi-commutative watermarking and encryption for secure media content distribution",[online], Multimedia Tools and Applications Volume 43, Number 1 / May, 2009

13. K. Gaj, P.Chodowiec, "Fast implementation and fair comparison of the final candidates for advanced encryption standard using field programmable gate arrays", CT-RSA 2001, pp.84-99

14. A. Hodjat, I. Verbauwhede, "A 21.54 Gbits/s Fully Pipelined AES Processor on FPGA". Proceedings of the 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'04).

15. K. Janvinen, M. Tominisko, J. Skytta, "A fully pipelined memorlyess 17, 8 Gpbs AES-128 encryptor", in International symposium of Field programmable Gate arrays, 2003, pp.207-215.

16. M. Mclone, J.V. McCanny, "Rijindael FPGA implementations utilizing look-up tables", J.VLSI signal process, syst. 34(3)(2003)261-275.

17. D. Dia, M. Zeghid, M. Atri, B. Bouallegue, M. Machhout and R. Tourki, "DWT-AES Processor for a Reconfigurable Secure Image Coding", International Journal of Computer Science and Engineering,vol.1,no.2,june 2009.

18. G.Liu, T.Ikenaga, S.Goto and T.Baba, "A Selective Video Encryption Scheme for MPEG Compression Standard", in IEICE Transactions on Fundamentals of Electronics, communications and Computer Sciences, 89 (2006), pp. 194-202.

19. M. Zeghid, M. Machhout, L. Khriji, A. Baganne and R. Tourki, "A Modified AES Based Algorithm for Image Encryption", International Journal of Computer Science and Engineering,1(2007), pp.70-75.

20. Sugreev Kaur and Rajesh Mehra,"High Speed and Area Efficient 2D DWT Processor Based Image Compression", Signal & Image Processing : An International Journal(SIPIJ) Vol.1, No.2, December 2010.

21. Ashwini M. Deshpande, Mangesh S. Deshpande and Devendra N. Kayatanavar, "FPGA implementation of AES Encryption and Decryption", International Conference on Control , Automation, Communication and Energy Conservation -2009, 4th-6th June 2009.

22. A. Mansouri, A. Ahaitouf, and F. Abdi, "An Efficient VLSI Architecture and FPGA implementation of High-Speed and Low Power 2-D DWT for (9, 7) wavelet Filter", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.3, March 2009.

23. D.Dia, M.Zeghid, M.Atri, B.Bouallegue, M.Machhout and R.Tourki, "DWT-AES Processor for a Reconfigurable Secure Image Coding", International Journal of Computer Science and Engineering,vol.1,no.2,june 2009.

24. A. E. Rohiem, F. M. Ahmed and A. M. Mustafa "FPGA Implementation of Reconfigurable Parameters AES Algorithm", 13th International Conference on Aerospace Sciences and Aviation Technology, ASAT- 13, May 26 – 28, 2009.