Paper: ASAT-13-CE-23

13th International Conference on
AEROSPACE SCIENCES & AVIATION TECHNOLOGY,
ASAT- 13, May 26 – 28, 2009, E-Mail: asat@mtc.edu.eg
Military Technical College, Kobry Elkobbah, Cairo, Egypt
Tel : +(202) 24025292 – 24036138, Fax: +(202) 22621908

FPGA Implementation of Reconfigurable Parameters AES Algorithm

A. E. Rohiem[*], F. M. Ahmed[**] and A. M. Mustafa[*]

Abstract: In this paper, a novel method of using customized (AES) variable parameters is introduced. This method depends on a continuous parameters reconfiguration and a customization of each internal block. The customization depends on varying the four transformations (polynomial and affine transformations for S-Box (SB), ShiftRows (SR) transformation, and MixColumn (MC) transformation). Internal AES blocks (SB, SR, and MC) are varied each round. Further more, these blocks are randomly interconnected during each session. The ciphered output was tested using avalanche, strict avalanche, and other NIST tests. This method overcomes (ECB) mode problems which appear when there is high redundancy in the plain data and also increasing strength against brute force attacks. The proposed AES is implemented on Field programmable Gate Arrays (FPGAs).

Keywords: Advanced Encryption Standard (AES); Electronic CodeBook (ECB); Field Programmable Gate Array (FPGA); National Institute of Standards and Technology (NIST).

1. Introduction

According to advanced progress in internet and wireless communication, users have an increasing demand of secure devices for data transmission over insecure channels. So, information system should be equipped with encryption and robustness techniques. The NIST [1] selected the Rijndael algorithm, which was developed by Joan Daemen and Vincent Rijmen, to replace the data encryption standard (DES) algorithm [2] as the new advanced encryption standard (AES) algorithm [3]. The AES has been used in many applications from internet routers, Virtual Private Networks (VPNs), mobile phone applications and electronic financial transactions

The AES algorithm can be efficiently implemented by hardware and software. Software implementations cost the smallest resources, but they offer only limited physical security. Because of the growing requirements for high-speed, high-volume secure communications combined with physical security, hardware implementation of cryptography takes place. The proposed AES is implemented based on FPGA technology. This technology, referred to as reconfigurable hardware, offers many advantages for future vendors and users of cryptographic equipment. However, many research papers based on implementation of the AES using FPGAs [4-8] have been presented. The most attacks to the AES systems focus on

[*] Egyptian Armed Forces
[**] Egyptian Armed Forces, Email: fkader2003@yahoo.com

the key, which may be leakage from the internal message or betrayer. This leads to the demand of the variations of an AES system to prevent the key loss from causing an immediate risk to the system. Recently, Barkan and Biham [9] and Jing [10] have proposed the idea of varying the parameters of the AES algorithm including the field irreducible polynomial, the affine transformation in the SubBytes, the offsets in the SRs, and the polynomial in the MCs.

The advantage of such variations in AES systems is that they increase the resistance regarding side channel attacks. As a result, it is necessary to design a cryptographic system with more variations, which is also called reconfigurable parameters system. In this paper, novel method of continuous reconfigurable parameters AES system architecture with Altera Stratix II FPGA [11] is introduced.

This paper is organized as follows; after the introduction, the AES algorithm is introduced. In section 3, the reconfigurable parameters of AES algorithms are explained. A Novel Method against ECB Mode is introduced in section 4. The implementation of the proposed system with FPGA is presented in section 5. Finally, conclusion and references are in section 6 and section 7.

2. AES Algorithm

The AES is a symmetric block cipher algorithm, in which the key length can be independently specified to be 128,192, or 256 bits [3]. The AES specification uses the same three key size alternatives but limits the block length to 128 bits. The number of AES rounds depends on the key length. In this paper, we use the key length of 128 bits (AES-128) as a model for general explanation. Figure 1 shows the overall structure of the AES, and the content of each round in both encryption and decryption. According to Federal Information Processing Standards (FIPS) PUB197, The data block is considered as a square matrix of bytes copied into a state array, which is modified at each stage of encryption or decryption. After the final stage, the state array is copied into an output matrix.

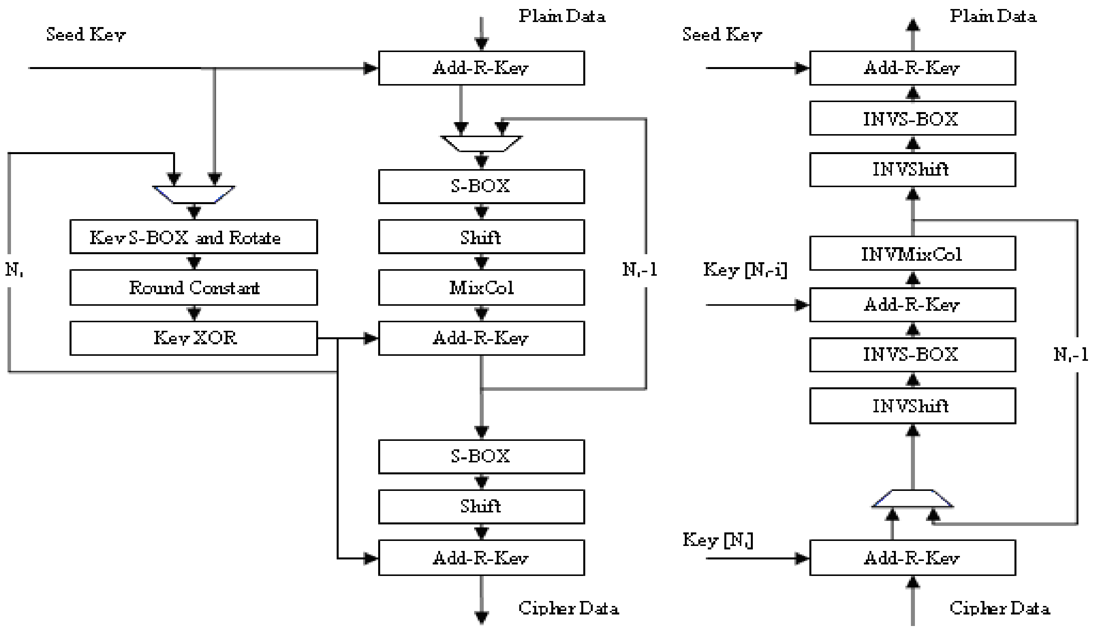


Fig. 1 AES Structure

Similarly, the 128-bit key is considered as a square matrix of bytes. This key is then expanded into an array of key schedule words; each word is four bytes and the total key schedule is 44 words for the 128-bit key.

The input data block passes through a round function which is iterated 10 times. Meanwhile, the key schedule expands. Each round consists of four different functions: SubBytes, SRs, MCs and AddRoundKey (ARK) which are applied to the state array.

The SubBytes is a byte substitution, it uses a non-linear table (SB) for each state byte. This step is to provide adequate resistance against differential and linear cryptanalysis attacks. The SB consists of a multiplicative inverse in GF(2^8) with the irreducible polynomial given, in equation (1):

$$f(\chi) = \chi^8 + \chi^4 + \chi^3 + \chi + 1 \tag{1}$$

For all elements except zero, this is mapped to itself. Then applying an affine transformation, which includes multiplication by a matrix and then making XOR with the hex value 63.

The SHs is a cyclic shift of each row by different byte offsets. Row 0 is not changed. Row 1 is left rotated by one time. Row 2 is left rotated twice, and row 3 three times.

The MC transformation is a linear combination of all the four bytes in the same column of the State over GF(2^8), multiplied with a fixed polynomial $a(\chi)$ modulo χ^4+1 as the data is multiplied with the polynomial, given in equation (2):

$$a(\chi) = \{03\}\chi^3 + \{01\}\chi^2 + \{01\}\chi + \{02\} \tag{2}$$

The ARK transformation is an XOR operation of the data block with the round key. Each round is identical except that the initial round is XORed with the seed key and the last round MC is excluded.

To decrypt the data (ciphertext), the procedure is simply the inverse of its encryption process. In other words, the standard round consists of Inverse SRs (ISR), Inverse Sub-Bytes (ISB), ARK, and Inverse MCs (IMC) transformations. The initial round adds an ARK, and the final round excludes the IMC. Furthermore, it is not necessary to change the key schedule. In the ISRs transformation, each row í is right rotated by í byte(s) instead of left SR. The ISBs transformation is obtained by applying the inverse of the affine transformation followed by taking the multiplicative inverse in GF (2^8). Also, the inverse SB is applied to each byte of the state array. The IMC transformation multiplies a polynomial formed by the same column of the State over GF (2^8) modulo x4 + 1 with a fixed polynomial, given in equation (3):

$$a^{-1}(\chi) = \{0e\}\chi^3 + \{0b\}\chi^2 + \{0d\}\chi + \{09\} \tag{3}$$

The ARK transformation shares the same name with its own inverse because it only involves the XOR operation.

3. Reconfigurable Parameters AES Algorithm.

It is known that the cryptanalysts has many ways to attack cryptographic systems, and there are human factors causes even more problems. The main reason is that these attacks focus on the keys. Therefore, the most serious problem in AES system is the key lost which may be caused by weak protocol or internal break-in (like betrayer).As a result, there should be a new requirement for a cryptographic system preventing from causing an immediate risk with key loss. To meet this requirement, extra parameter(s) should be added to the system other than the key without increasing the system complexity. The AES system actually has four parameters in each round. These parameters are the field irreducible polynomial, the affine transformation in the SB, the offsets in the SRs, and the polynomials, $a(\chi)$ in the MC. The main key and these parameters can be negotiated by different channels between the encryption and decryption sides. This will increase the difficulty of accessing the information from attackers. In this case, even if the main key is lost, the attacker can't get the right parameter(s), because of the variations [10] of the parameters.

The behaviors of all multipliers and multiplicative inverse operations in AES system are affected by the finite field generator polynomial. In the traditional AES, AES algorithm uses the irreducible polynomial given in equation (1) to produce a field in GF (2^8). This irreducible polynomial is one of 30 irreducible polynomials with degree 8 over GF (2^8). Polynomial transformation variations can be achieved by using each of these 30 polynomials to be the field polynomial.

The traditional AES affine transformation is given in equation (4), where $b[i]$ is the bit number ί in the data byte after multiplicative inverse, $\overline{b}[i]$ is the bit number ί in the data byte after affine, $c_{[i]}$ is the bit number ί in the constant with the hex value 63

$$\overline{b}[i] = b[i] + b[(i+4) \bmod 8] + b[(i+5) \bmod 8] + b[(i+6) \bmod 8] + b[(i+7) \bmod 8] + c_{[i]} \tag{4}$$

The inverse affine transformation calculated by XORing the constant with the data byte and then gets its inverse affine as given in equations (5), (6) respectively:

$$\overline{b}_{[i]} = \overline{b}_{[i]} + c_{[i]} \tag{5}$$

$$b_{[i]} = \overline{b}_{[(i+2) \bmod 8]} + \overline{b}_{[(i+5) \bmod 8]} + \overline{b}_{[(i+7) \bmod 8]} \tag{6}$$

The number of random invertible 8x8 matrixes can be calculated as follows:

$$N_n = \prod_{i=0}^{n-1} (2^n - 2^i) \tag{7}$$

The number of variation of the non-zero constant is 255 [8]. Therefore, the count of different affine transformation is 194822323021283328000 in AES system. The affine transformation variation can be achieved by using any two invertible 8×8 matrixes and any non-zero constant.

The SR variation can be achieved by using many SR offsets. Most Known offsets are given in Table 1. The traditional AES use offset 5 in encryption mode [12] and offset 13 in decryption mode.

In MCs transformation, every column of the State is multiplied with a fixed polynomial $a(\chi)$ modulo χ^4+1. $a(\chi)$ is coprime to χ^4+1 then there exists an invertible polynomial $a^{-1}(\chi)$ for the IMC, so we can use another polynomials and their inverses to have many MCs, IMCs to achieve the MC-IMC variations. The next step is to use the principle of variation to suggest a new method to solve the ECB mode problems and also increase the strength of the AES against brute force attack.

Table 1 Shift Offsets

Shift Offset	Inverse Shift offset
1	1
2	-
3	11
4	-
5	13
6	-
7	7
8	-
9	9
10	-
11	3
12	-
13	5
14	-
15	15

4. A Novel Method against ECB Mode

Using ECB mode on long fixed sequences of plain data that doesn't change results in a repeated cipher data sequences, even if parameters variation is used i.e. (different transformations for SB polynomials, SR offsets and MCs polynomials). A picture that has fixed colors for many sequences is shown in Fig. 2(a). This picture is encrypted using an AES algorithm with variable parameter in ECB mode as shown in Fig. 2(b). Even it is so difficult to get data by brute force, but cryptanalysts by some statistical tests, can guess the plain data.

This problem can be solved by using other modes of operation, but these modes not combine the strength against brute force attack as the novel method combine. In this paper the strength of variation against brute force and solving the ECB Problem is combined, and this can be done by using a new method that uses the variable parameters technique by a new way that depends on making random combinations between these parameters at each session. These random combinations are controlled by a random generator which creates, with each session, a random path between different S-boxes, different SRs and different MCs, where every round contains different parameters. This path is changed with every data frame without affecting real time applications. However, Fig.2(c) shows the ciphered picture with the proposed new method.

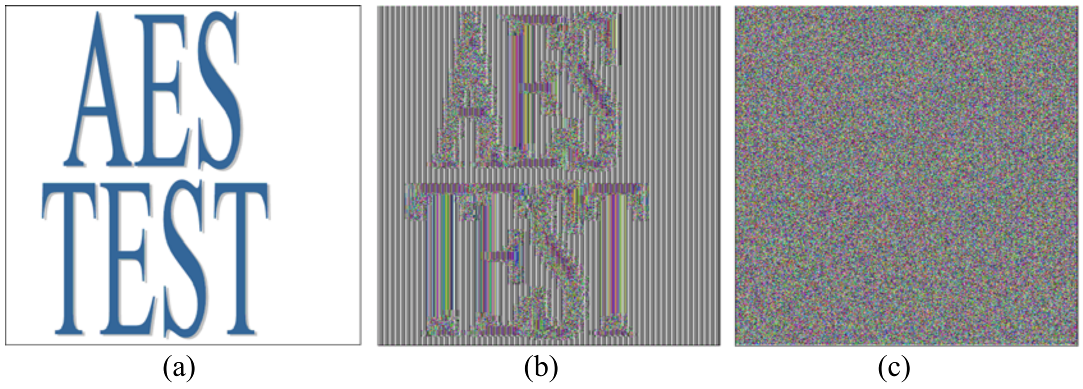


Fig. 2 Encryption using the ECB mode and the proposed mode
(a) Plain Picture
(b) Encryption in ECB Picture
(c) Encryption in ECB mode using proposed technique

Various researching regarding the hardware implementation of the AES based on different criteria such as speed, cost, and reliability. For speed using LUTs for S-boxes is comparatively faster than using composite field. In this paper, the propose method is implemented using the reconfigurable FPGAs. Figure 3 shows the data path and the random generator controller path in the encryption and decryption mode.

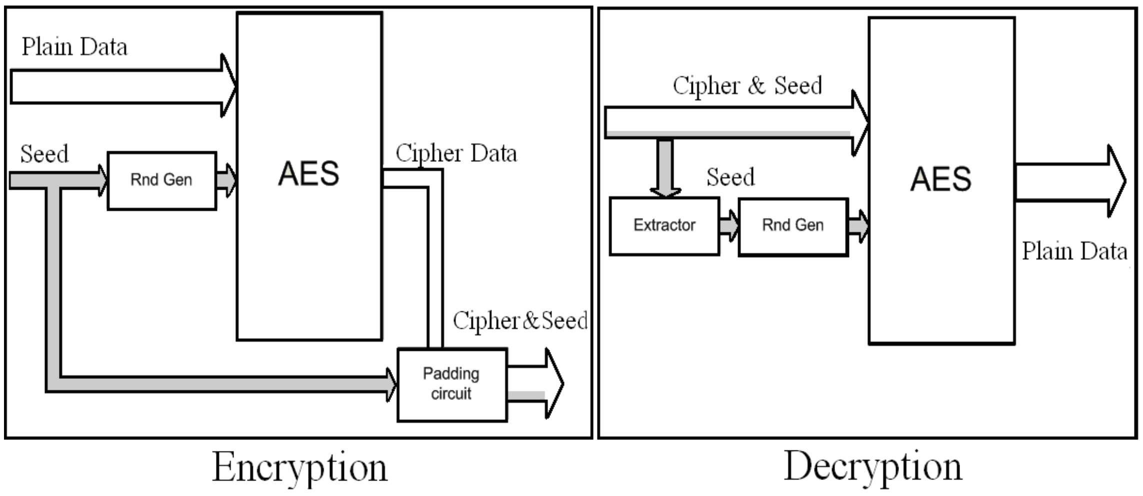


Fig. 3 System Main Blocks

The main controller of the system in the encryption mode injects the seed value into the random sequence generator which will produce the random sequence that will be used for the parameter variation. In the decryption mode, the seed value is extracted from the incoming cipher message and is used to generate the same random sequence which was used by the transmitter to insure a synchronized operation.

5. Implementation of the Proposed Method using FPGA

In the proposed design, different S-Boxes for all rounds are collected in one block, all different SRs in one block, and all different MCs in one block, for encryption mode. In decryption mode, the same procedure shall be done where all ISBs are collect in one block, and ISRs in one block and all IMCs in one block in a reversed order.

When the data is ready, the main controller sets the In_MUX to pass the input data to the SBs blocks, and in the same time enables the random generator to generate a random sequence. This sequence is used to identify the data path between blocks. Input data is fed to all SBs, but the output will be selected according to the random generator selector and so the SH and the MCs, when this output is ready, the main controller sets the In_MUX to pass the feedback and loop on the same path for 9 times, where there is no MCs in the last round.

However, Figure 4 demonstrate the operation of the encryption part, where, an OR gate is used to choose the SR output directly for the last round. All SBs and ISBs are generated for different polynomials and affine transformations by software. Their avalanche, strict avalanche and bit independence [13] are tested to get the best results of them, and then SBs, are chosen. Other polynomials are then chosen and the cipher output is tested by NIST [14] tests.

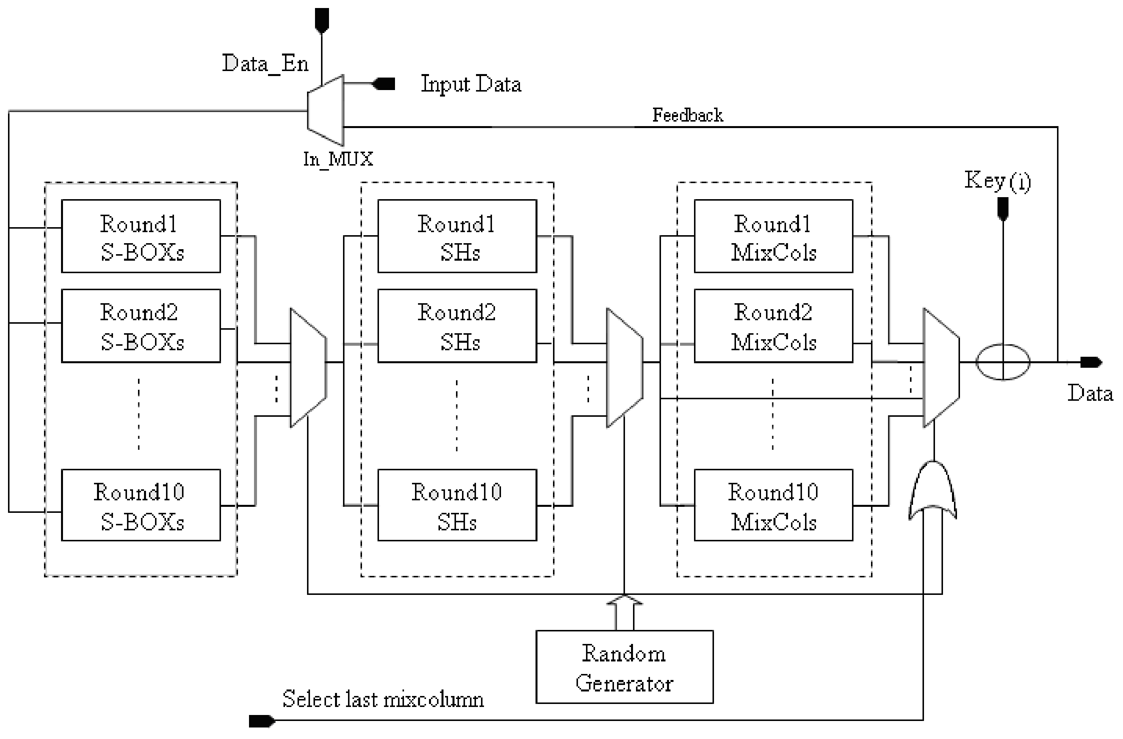


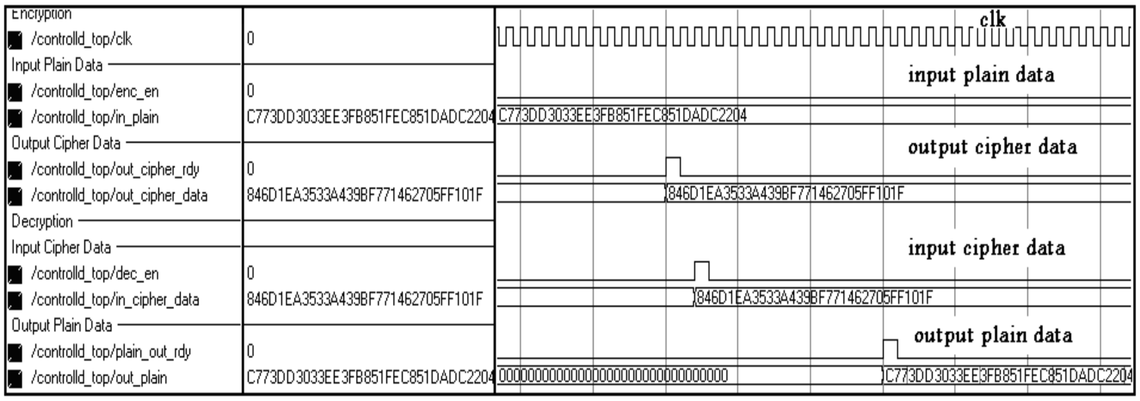
Fig. 4 Encryption Structure

Altera EP2S60F1020C5ES FPGA chip is used for implementation. The device utilization summary is listed in Tabel.2.

Table 2 Device Utilization

Device	usage
Total ALUTs	7,333/48,352 (15%)
Total Registers	1352
Total Memory bits	637,440/2,544,192 (25%)

Figure 5 shows the design simulation wave forms. The outputs take 12 clocks to be ready for both encryption and decryption modes, so the throughput is 3.18 Gbps for clock 298.15 MHz.



```
Encryption
/controlld_top/clk                  0
Input Plain Data
/controlld_top/enc_en               0
/controlld_top/in_plain             C773DD3033EE3FB851FEC851DADC2204   C773DD3033EE3FB851FEC851DADC2204
Output Cipher Data
/controlld_top/out_cipher_rdy       0
/controlld_top/out_cipher_data      846D1EA3533A439BF771462705FF101F   846D1EA3533A439BF771462705FF101F
Decryption
Input Cipher Data
/controlld_top/dec_en               0
/controlld_top/in_cipher_data       846D1EA3533A439BF771462705FF101F   846D1EA3533A439BF771462705FF101F
Output Plain Data
/controlld_top/plain_out_rdy        0
/controlld_top/out_plain            C773DD3033EE3FB851FEC851DADC2204   00000000000000000000000000000000   C773DD3033EE3FB851FEC851DADC2204
```

Fig. 5 Design Simulation Waveform

6. Conclusion

A novel method of using customized (AES) variable parameters is introduced. This method depends on a continuous parameters reconfiguration in a random manner. The customized internal blocks are randomly varied each round while randomly interconnected each session. The proposed AES is implemented using FPGA. The ciphered output was tested using avalanche, strict avalanche, and other NIST tests. This method overcomes (ECB) mode problems which appear when there is high redundancy in the plain data and also increasing strength against brute force attacks.

7. References

[1] J. Daemen, V. Rijmen, AES proposal: Rijndael Document version 2, 1999.

[2] National Institute of Standards and Technology (NIST), Data Encryption Standard (DES), Federal Information Processing Standards Publications (FIPS PUBS) 46-3 (1999).

[3] National Institute of Standards and Technology (NIST), Advanced Encryption Standard (AES), Federal Information Processing Standards Publications (FIPS PUBS) 197 (2001).

[4] Alireza Hodjat, Ingrid Verbauwhede, Area-Throughput Trade-Offs for Fully Pipelined 30 to 70 Gbits/s AES Processors, IEEE TRANSACTIONS ON COMPUTERS, VOL. 55, NO. 4, APRIL 2006

[5] Shen-Fu Hsiao, Ming-Chih Chen, and Chia-Shin Tu, Memory-Free Low-Cost Designs of Advanced Encryption Standard Using Common Subexpression Elimination for Subfunctions in Transformations, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 53, NO. 3, MARCH 2006

[6] T. Chitu, M. Glesner, An FPGA implementation of the AES-Rijndael in OCB/ECB modes of operation, Microelectronics Journal 36 (2) (2005) 139–146.

[7] N. Pramstaller, S. Mangard, S. Dominikus, J. Wolkerstorfer, Efficient AES implementations on ASICs and FPGAs, Lecture Notes in Computer Science 3373 (2005) 98 112.

[8] Ming-Haw Jing, Zih-Heng Chen, Jian-Hong Chen, Yan-Haw Chen, Reconfigurable system for high-speed and diversified AES using FPGA, Microprocessors and Microsystems 31 (2007) 94–102

[9] E. Barkan, E. Biham, in how many ways can you write Rijndael? Lecture Notes in Computer Science 2501 (2002) 160–175.

[10] M.H. Jing, C.H. Hsu, T.K. Truong, Y.H. Chen, Y.T. Chang, The diversity study of AES on FPGA application, in IEEE International Conference on Field-Programmable Technology (FPT) (2002) 390–393.

[11] Stratix II Device Handbook, Volume 1.
Available from: <http://www.altera.com/literature/hb/stx2/stratix2_handbook.pdf>.

[12] Eltayeb Salih Abuelyman, and Mohamed Ahmed El-Affendi, An Optimized Real Time Generation of S-Box Inverses Using Arithmetic Modulo Powers of Two, IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.12, December 2007

[13] I_s_l VERG_IL_I, Melek D. Y¨UCEL, Avalanche and Bit Independence Properties for theEnsembles of Randomly Chosen n _ n S-Boxes, Turk J Elec Engin, VOL.9, NO.2 2001, c TU¨BI_TAK

[14] Song-Ju Kim, Ken Umeno, and Akio Hasegawa, Corrections of the NIST Statistical Test Suite for Randomness, Communications Research Laboratory, Incorporated Administrative Agency 4-2-1, Nukui-kitamachi, Koganei-shi, Tokyo 184-8795, Japan.