Advanced Encryption Standard Algorithm: Issues and Implementation Aspects

Ahmed Fathy[1], Ibrahim F. Tarrad[2],
Hesham F.A. Hamed[3], and Ali Ismail Awad[1,4]

[1] Faculty of Engineering, Al Azhar University, Qena, Egypt
[2] Faculty of Engineering, Al Azhar University, Cairo, Egypt
[3] Faculty of Engineering, Minia University, Minia, Egypt
[4] Member of Scientific Research Group in Egypt (SRGE)
{ahmedf.abdelfatah,tarradif}@gmail.com, hfah66@yahoo.com, aawad@ieee.org

Abstract. Data encryption has become a crucial need for almost all data transaction application due to the large diversity of the remote information exchange. A huge value of sensitive data is transferred daily via different channels such as e-commerce, electronic banking and even over simple email applications. Advanced Encryption Standard (AES) algorithm has become the optimum choice for various security services in numerous applications. Therefore, many researches get focused on that algorithm in order to improve its efficiency and performance. This paper presents a survey about the cutting edge research conducted for the AES algorithm issues and aspects in terms of developments, implementations and evaluations. The contribution of this paper is targeted toward building a base for future development and implementation of the AES algorithm. It also opens door for implementing the AES algorithm using some machine learning techniques.

Keywords: Cryptography, Advanced Encryption Standard, FPGA, ASIC, Machine Learning.

1 Introduction

Cryptography is the process of transferring data into scrambled format, but at the same time, it allows the intended recipient to restore the original data by using a secret key. Encryption and decryption are the two major functions in any cryptography system. Encryption is transferring data into unintelligible format by secret key to guarantee the user privacy. Decryption is the opposite function used to recovery the original encrypted data by using secret key. Data encryption is an important process in almost all data transaction applications [1].

There are two classes of data encryption algorithms; symmetric key and asymmetric key [2]. In symmetric or private key algorithm, the communication is achieved by using only one key. In contrary, asymmetric key algorithm uses more than one key for data encryption and data restore. One key is a public

A. Ell Hassanien et al. (Eds.): AMLTA 2012, CCIS 322, pp. 516–523, 2012.

key, and it is used for data encryption, where the other key is a private key, and it is used for data decryption. The symmetric algorithms are much faster than asymmetric key which need bigger key and complex computation [1], [3].

Advanced Encryption Standard (AES) [4] algorithm is one of the symmetric key block ciphers with block size varies from 64 to 256 bits as the processors become more sophisticated. However, the AES can accept block size up to 256 bit, its speed still slow compared to the stream-based ciphers in the time of all applications are seeking for faster encryption process such as web servers and Automatic Teller Machines (ATMs). On the other hand, some AES applications are keep struggling for low implementation area such as smart card and cellular phone related hardware. Therefore, the encryption speed and implementation area are the two important factors of the real time deployment of AES algorithm. The problem of AES spreading out is the compromising between the encryption/decryption speed and the implementation area.

Field Programmable Gate Array (FPGA)[5] is an Integrated Circuit(IC) that can be repeatedly reconfigured as requested by the operated applications, and it can produce different behaviour by simple configuration changes [6]. According to the previous property and its low cost, FPGA is considered as a good environment for simulating the hardware implementation of the AES encryption algorithm [7]. This paper emphasizes the deployment of AES algorithm on the FPGA environment and focusing on the cutting edge approaches for enhancing the encryption speed and reducing the required implementation area. Due to their friability and inelegance, machine learning techniques are also good candidates for efficient AES development and cryptanalysis processes [8].

The contribution of this research lies on reporting the state-of-the art implementations of the AES algorithm over the FPGA modules, and get focusing on the raised implementation issues and aspects. This contribution is significant for any future developments and implementations of the AES encryption algorithm. It is considered as a ground truth information for evaluating any new AES developments by comparing the output results of the proposed approaches with the results reported in this research.

The rest of this paper is organized as follows. Section 2 presents the theoretical background of the AES algorithm with respect to algorithm's structure, data encryption and data decryption methodologies. Section 3 reports the cutting edge research about the AES implementation over the FPGA environment with speed and area considerations. Furthermore, the new machine learning implementation of AES are also reported in section 3. Conclusions and future work are reported in Section 4.

2 Advanced Encryption Standard Algorithm

Data Encryption Standard (DES) [9] was considered as a model for the symmetric key encryption which has a key length of 56 bits. but this key length is become small and can easily be hacked [10]. The National Institute of Standards

Table 1. Different AES keys and their attributes

	Key length (N_k) words	Block size (B_b) words	Number of round (N_r)
AES128	4.0	4.0	10.0
AES192	6.0	4.0	12.0
AES256	8.0	4.0	14.0

and Technology NIST [11] released a contest to choose a new symmetric cryptograph algorithm that would be called Advanced Encryption Standard to replace the DES. A five algorithms have been chosen as Mars, RC6, Rijndael, Serpent and Twofish. After two years of detailed evaluation NIST announced Rijndael as a proposed AES [12], [13]. AES has length of bits 128 which can encrypt and decrypt with the three different key lengthes as 128 bit, 192 bit and 256 bit which known as AES128, AES192 and AES256 [14]. The key length, block size and the number of rounds for each AES mode (128, 192, and 256) are reported in Table 1.

The encryption operation is performed on a two dimensional array of bytes (Each block is organized as a 4 × 4 matrix of bytes) called `State` which consists of 4 rows of bytes and each row has N_b bytes. By considering AES 128 algorithm in which its initial round state is XOR with a selected key. The regular round consist from four main operations which called `SubBytes`, `ShiftRows`, `MixColumns`, and `AddRoundKey`. In the last round, only three operations are found while `MixColumns` operation is eliminated [15].

`SubBytes` are nonlinear transformation that uses 16 identical 256-byte substitution tables (S-box) for independently mapping each byte of `State` into another byte. S-box entries are generated by computing multiplicative inverses in Galois Field GF(2^8) and applying an affine transformation. `SubBytes` can be implemented either by computing the substitution or using look up table. `ShiftRows` cyclically shifts the bytes in the second, third and fourth rows by one, two and three, respectively. This function requires no resource hardware and it can be executed on an FPGA as plain wiring [12]. `MixColumns` is a linear transformation and it is conducted on the `State` array column by column. The key scheduler of the AES generates a total of N_b $(N_r + 1)$ words in order to complete the encryption and decryption processes [16], [17], [18].

The decryption process is the inverse operation of encryption which inverse the round transformations to obtain the original plain data. The round transformations of the decryption process have four functions: `AddRoundKey`, `InvMixColumn`, `InvShiftRows` and `InvSubBytes`, respectively. `AddRoundKey` is an `XOR` function. `InvShiftRows` have the same function as `ShiftRows` but only in the inverse direction. The first row is not changed, but the second is shifted by one, the third is shifted by two and last is shifted by three. The `InvSubBytes` transformation is done using a permutation table called InvS-box that has 256 numbers (from 0 to 255) [10]. It is worth notice that the AES algorithm can operate in four modes, Cipher Block Chain (CBC), Cipher Feedback (CFB), Output Feedback (OFB) and Electronic Code Block (ECB) [1], [19].

Table 2. AES restrictions and achievements by [20]

Target FPGA Device	Virtex XCV600 BG 560-6
Optimization goal	Speed
Maximum operating frequency	140.390MHz
Encryption/decryption throughput	352 Mbit/sec
Total memory usage	130248 Kbytes

3 FPGA Implementations

The FPGA toolkit consists of a matrix of programmable logic cells, known as configurable logic blocks (CLBs) with a grid of interconnecting lines and switches between them. The I/O cells exist around the perimeter and they work as interface between the interconnect lines and the chip's external pins [5]. The exact functionality of a logic cell varies with the manufacturer of the chip, but it is typically comprised of a small amount of functional logic and/or some register storage capability. Programming (configuring) the FPGA consists of specifying the logic function of each cell and the switches on the interconnecting lines.

3.1 AES Speed Optimization

Ghewari et al. [20] used iterative operations in order to reduce the consumed resources by the AES. The AES algorithm is implemented using VHDL language in Xilinx ISE 9.2 with Device XCV600 of Xilinx Virtex Family. The achieved encryption/decryption throughput is around 352 Mbit/s. The problem with this approach is a high memory consumption. Table 2 shows the restrictions and achievements of the mentioned algorithm.

Thulasimani et al. [21] developed a hardware architecture and implementation for AES128, AES192 and AES256 on the same hardware, and they used the iterative round technique to reduce the resource consumption. The AES has been implemented using the VHDL code running on Xilinx 9.2 with maximum throughput around 666.67 Mbit/s. The advantage of this approach is the low power consumption as the the implementation of three keys on single chip leads to consumed power reduction which is useful for applications such that radio software. The implementation results and achievements of this approach is shown in Table 3.

Table 3. AES restrictions and achievements by [21]

Target FPGA Device	XC2V6000BF957-6
Optimization goal	Speed
Maximum operating frequency	62.5MHz
Encryption/decryption throughput	666.67 Mbit/sec
Total memory usage	Not reported

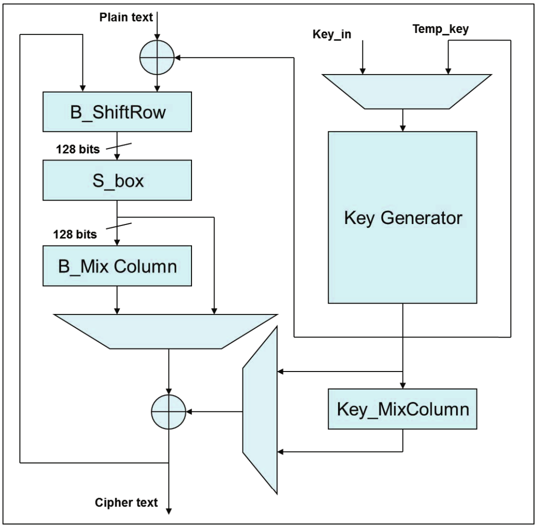

Fig. 1. Flowchart of the AES implementations over the ASIC chip

Yin et al. [22] proposed an implementation of the AES algorism on Application Specific Integrated Circuit (ASIC) which supports three key lengthes as $128, 192$ and 256. The encryption/decryption processes are implemented using the same sequential architecture, but with more efficient control. The implementation flowchart is shown in Fig.1. The achieved throughput with different key lengths is shown in Table 4.

Liberatori et al. [23] used Spartan3 to implement AES algorism which have only 173 single ended I/O pins. Therefore, two buses of 64 bits have been used for input and output data and another 32 bus for encryption key and the remaining 13 pins have been used for managing the control signal. They have used Xilinx6.3i, ModelSim5.8 and ModelSim3.0SE plus for the implementation purpose. The achievements of the proposed approach are reported in Table 5.

The above reported researches are interested only in enhancing the encryption/decription speed regardless of the consumed FPGA area as some of these research did not even consider reporting the value of implementation area. Enhancing the encryption/decription speed does not optimize the overall deployment of AES as some applications are interesting in small implementation area. The following part reports the researches about optimizing the AES implementation area over the FPGA.

Table 4. AES throughputs with different key lengthes [22]

	Key length 128 bit	Key length 192 bit	Key length 256 bit
Data length 128 bit	1.03 Gbps	0.853 Gbps	0.73 Gbps

Table 5. AES restrictions and achievements by [23]

Target FPGA Device	Spartan 3 XC3s200FT256-5
Optimization goal	Speed
Maximum operating frequency	91.049MHz
Encryption/decryption throughput	224 Mbit/sec
Total memory usage	Not reported

3.2 AES Area Optimization

Hamalainen et al. [15] tried to reduce the implementation area (number of gates) by parallelizing the AES operations on FPGA. The high level architecture consists of byte permutation, `MixColumn` multiplier, parallel to serial converter, S-box and key scheduler. The proposed design is implemented on a 0.13 μm CMOS technology, therefore, this approach is appreciated for low cost and low power applications. The achieved results are shown in Table 6.

Another area improvement approach is introduced by Rady et al. [24]. The proposed AES core architecture consists of three units; controller unit, interfaces unit and the main Enc/Dec AES unit with the key expansion and storage. The proposed architecture introduced two ways to improve the AES area. The first way is by iterative the key expansion and ordinary round. The second way is by sharing specific resource in the ordinary round and key expansion. This design introduce improvement in area implementation of the Enc/Dec AES core by VHDL code with Spartan3(Xc3s400) FPGA kit using the iteration architecture and the resource sharing. This design consumes 2699 slices and two BRAM. It is simulated with the timing simulation of ModelSim V 6.0.

Away from the AES hardware implementations, machine learning techniques contribute very well not only for AES development, but also for encryption hacking and cryptanalysis operations. Siddeeq and Ali [25] used the Neural Networks technique for building and simulating the AES based cryptosystem. The initial weights for neural network representing the key used in both encryption and decryption processes. The simulation results prove that the performance of NN-AES is equivalent to the normal one. Albassal and Wahdan [26] used the neural networks for attacking and cryptanalysis of a Feistel type block cipher. The achieved results can be used in the future for filling some holes of encryption algorithm. They have also extended their work to include Genetic Algorithms as a machine learning techniques for cryptanalysis process [27].

Table 6. CMOS implementations of AES for area optimization [15]

	Width (bit)	Area (Kgates)	Freq. (MHz)	Power μW/MHz	Thro. Mbps
Area	8	3.1	152	37	121
Power	8	3.2	130	30	104
Speed	8	3.9	290	62	232

4 Conclusions and Future Work

Data encryption is a basic notion of information security, and it is used by almost all data transaction applications. The Advanced Encryption Standard (AES) is one of the most efficient encryption algorithms, and it has received great research attention due its simplicity and applicability. The AES related researches are focused in the encryption speed and the hardware implementation area as two important factors for algorithm‘s performance. This paper presented some cutting edge researches for enhancing the AES speed and implementation area. According to the reported results, it is clear that these is trade-off between the two factors. As a future work, we are going to continue this research in order to get the maximum encryption speed in limited implementation area. Moreover, some machine learning techniques can be used for simulating the AES encryption and cryptanalysis processes.

References

1. van Tilborg, H.C.A.: Encyclopedia of Cryptography and Security. Springer-Verlag New York, Inc., Secaucus (2005)
2. Nedjah, N., de Macedo Mourelle, L.: A Versatile Pipelined Hardware Implementation for Encryption and Decryption Using Advanced Encryption Standard. In: Daydé, M., Palma, J.M.L.M., Coutinho, Á.L.G.A., Pacitti, E., Lopes, J.C. (eds.) VECPAR 2006. LNCS, vol. 4395, pp. 249–259. Springer, Heidelberg (2007)
3. Paar, C., Pelzl, J.: Understanding Cryptography: A Textbook for Students and Practitioners, 1st edn. Springer Publishing Company, Incorporated (2009)
4. Burr, W.E.: Selecting the advanced encryption standard. IEEE Security and Privacy 1(2), 43–52 (2003)
5. Kilts, S.: Advanced FPGA Design: Architecture, Implementation, and Optimization. Wiley-IEEE Press (2007)
6. Gomes, O., Moreno, R., Pimenta, T.: A fast cryptography pipelined hardware developed in FPGA with VHDL. In: The 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), pp. 1–6 (October 2011)
7. Mijalli, M.H.A.: Efficient realization of S-Box based reduced residue of prime numbers using Virtex-5 and Virtex-6 FPGAs. American Journal of Applied Sciences 8(8), 754–757 (2011)
8. Dileep, A., Sekhar, C.: Identification of block ciphers using support vector machines. In: International Joint Conference on Neural Networks, IJCNN 2006, pp. 2696–2701 (2006)
9. National Institute of Standards and Technology: FIPS PUB 46-3: Data Encryption Standard (DES) (October 1999), `http://www.itl.nist.gov/fipspubs/fip186-2.pdf`, supersedes FIPS 46-2
10. Hoang, T., Nguyen, V.L.: An efficient FPGA implementation of the advanced encryption standard algorithm. In: IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), pp. 1–4 (March 2012)
11. National Institute of Standards and Technology, `http://www.nist.gov/index.html`

12. Zambreno, J., Nguyen, D., Choudhary, A.: Exploring Area/Delay Tradeoffs in an AES FPGA Implementation. In: Becker, J., Platzner, M., Vernalde, S. (eds.) FPL 2004. LNCS, vol. 3203, pp. 575–585. Springer, Heidelberg (2004)
13. Wali, M.F., Rehan, M.: Effective coding and performance evaluation of the Rijndael Algorithm (AES). In: Student Conference on Engineering Sciences and Technology, SCONEST 2005, pp. 1–7 (August 2005)
14. Tillich, S., Feldhofer, M., Popp, T., Großschädl, J.: Area, delay, and power characteristics of standard-cell implementations of the AES S-Box. Journal of Signal Processing Systems 50(2), 251–261 (2008)
15. Hamalainen, P., Alho, T., Hannikainen, M., Hamalainen, T.: Design and implementation of low-area and low-power AES encryption hardware core. In: The 9th EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools, DSD 2006, pp. 577–583 (2006)
16. Elumalai, R., Reddy, A.R.: Improving diffusion power of AES Rijndael with 8 × 8 MDS Matrix. International Journal of Scientific and Engineering Research 2(3) (March 2011)
17. Rais, M.H., Qasim, S.M.: Efficient fpga realization of S-Box using reduced residue of prime numbers. International Journal of Computer Science and Network Security (IJCSNS) 10(1), 74–96 (2010)
18. Huang, J., Seberry, J., Susilo, W.: A Five-Round Algebraic Property of the Advanced Encryption Standard. In: Wu, T.-C., Lei, C.-L., Rijmen, V., Lee, D.-T. (eds.) ISC 2008. LNCS, vol. 5222, pp. 316–330. Springer, Heidelberg (2008)
19. Yenuguvanilanka, J., Elkeelany, O.: Performance evaluation of hardware models of advanced encryption standard (AES) algorithm. The IEEE Southeastcon, 222–225 (April 2008)
20. Ghewari, P.B., Jaymala, M., Patil, K., Chougule, A.B.: Efficient hardware design and implementation of AES cryptosystem. International Journal of Engineering Science and Technology 2(3), 213–219 (2010)
21. Thulasimani, L., Madheswaran, M.: A single chip design and implementation of aes-128/192/256 encryption algorithms. International Journal of Engineering Science and Technology 2(5), 1052–1059 (2010)
22. Yin, H., Debiao, H., Yong, K., Xiande, F.: High-speed ASIC implementation of AES supporting 128/192/256 bits. In: International Conference on Test and Measurement, ICTM 2009, vol. 1, pp. 95–98 (December 2009)
23. Liberatori, M., Otero, F., Bonadero, J., Castineira, J.: AES-128 cipher. high speed, low cost FPGA implementation. In: The 3rd Southern Conference on Programmable Logic, SPL 2007, pp. 195–198 (Febraury 2007)
24. Rady, A., El Sehely, E., El Hennawy, A.: Design and implementation of area optimized AES algorithm on reconfigurable FPGA. In: Internatonal Conference on Microelectronics, ICM 2007, pp. 35–38 (December 2007)
25. Siddeeq, Y.A., Ali, H.M.: AES cryptosystem development using neural networks. International Journal of Computer and Electrical Engineering (IJCEE) 3(2), 309–314 (2011)
26. Albassal, A., Wahdan, A.M.: Neural network based cryptanalysis of a feistel type block cipher. In: International Conference on Electrical, Electronic and Computer Engineering, ICEEC 2004, pp. 231–237 (September 2004)
27. Albassal, A., Wahdan, A.M.: Genetic algorithm cryptanalysis of a feistel type block cipher. In: International Conference on Electrical, Electronic and Computer Engineering, ICEEC 2004, pp. 217–221 (September 2004)