Advanced Encryption Standard with Galois Counter Mode using Field Programmable Gate Array.

IOP Conf. Series: Journal of Physics: Conf. Series **1019** (2018) 012008 doi:10.1088/1742-6596/1019/1/012008

Advanced Encryption Standard with Galois Counter Mode using Field Programmable Gate Array.

Nabihah Ahmad[1*], Lim Mei Wei[1], M. Hairol Jabbar[1]

[1] Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia,
86400 Parit Raja, Batu Pahat, Johor, Malaysia.

*Email: nabihah@uthm.edu.my

Abstract. Nowadays, the protection of transferring data is important to prevent the data hack easily. Advanced Encryption Standard with Galois Counter Mode (AES-GCM) plays an important role to provide high assurance of authenticity and data confidentiality in electronics, computers and other communication applications. This paper presents the implementation of AES-GCM by using Field Programmable Gate Array (FPGA) and AES-GCM designs in parallel- pipelined to achieve high performance in term of throughput and latency. The implementation of AES-GCM in FPGA by using 128-bit of input data block, Initialization vector (IV) and Additional Authenticated Data (AAD) to provide a high speed of authenticated encryption/ decryption. The key length of AES-GCM is 256-bit to provide the high security system and the operation of key expand designed in parallel to optimize operation time of AES-GCM. The proposed architecture is designed in Verilog hardware description language (HDL) and implemented using DE1-SoC with Cyclone V device. A parallel-pipelined of AES-GCM is introduced and it is operated at 10 MHz, achieved throughput of 16.84 Gbps, utilized of 11,196 slices. AES-GCM is carried out with the key-length of 256-bit is suitable to perform at high speed of electronic applications in term of security.

1. Introduction

Advanced Encryption Standard with Galois Counter Mode (AES-GCM) is introduced by United States of America National Institute for Standard and Technology (NIST). AES-GCM is suitable to employ in communication or electronic applications [3]. AES-GCM is a block cipher mode of operation that provides high speed of authenticated encryption and data integrity. Todays, the level of privacy protection is insufficient and make the data is been hacked easily. The FPGA is suitable to implement for AES-GCM by ensuring the confidentiality and integrity of the bit-stream[4].

AES-GCM have two main functions are block cipher encryption and multiplication over the field $GF(2^{128})$[5]. The authenticated encryption operation takes Initialization Vector (IV), Additional Authenticated Data (AAD),secret key and plaintext as an input in128-bit and gives a 128-bit ciphertext and authentication tag,T. The AES-GCM algorithm encrypts or decrypts with 128-bit, 192-bit or 256-bit of cipher key. The number of rounds executed transformations of AES depends on the length of cipher key [6][7][8]. Thus, the number of rounds executed is increased, the time taken to generate output is longer and will affect performance of AES-GCM.

Research of previous works shows the performance of AES-GCM in [17] is the better than other works. The method to design the AES-GCM was right-to-left exponentiation of pipelined GF multipliers. A group of parallel generators are created multiple blocks of input. This makes the more

efficient of the architecture which achieved the throughput 327.7 Gbps. The design pipelined S-box of AES is the efficient method to achieve the high performance of AES. The multiple of rounds transformation to execute, AES will consume more time as the performance of AES depends the key size in the design.is required. The performance of previous work [5], the throughput 42.7 Gbps, utilize with 2815 slices. The pipelined of S-box is needed to consider when designing the architecture of AES-GCM.

In this paper, the performance of AES-GCM is analyzed when the implementation of AES-GCM encryption using DE1-SoC with Cyclone V device. The performance of AES-GCM is introduced in term of throughput and latency. Verilog HDL is used to describe the behavior of AES-GCM by using Quartus II. The parallel-pipelined of AES-GCM is introduced to achieve the objectives of this paper.

2. Architecture of AES-GCM

The AES architecture is designed in parallel due to higher performance of AES-GCM encryption and decryption. The parallel architecture of key expansion is designed as greater key size is used. Figure 3 shows the key size is 256-bits will cause the execution of 14 rounds substitution and permutations needed. The key size depends on the desired security level. The standard number of transformation rounds of AES-256 is fourteen rounds. As shown in Table 1, AES encrypts a 128-bit of plaintext or decrypts 128-bit of ciphertext by repeatedly applying the same round transformation a number of times depending on the key size.

Table 1. Key-Block-Round Combinations.

	Key Length (32-bit word)	Block Size (32-bit word)	Number of Round
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

When the key size implemented is larger, the encryption or decryption will take longer time. The main function of AES encryption is according the key schedule. The proposed key size of this project is 256-bit. Thus, the proposed design of AES is in parallel that can optimize the time taken to execute the key expand and encryption of AES. For the AES algorithm, the length of the input block and the output block are 128-bit. The key length designed is 256-bit, which reflects the number of rounds to be performed is 14 rounds. The more rounds to be executed will make the more consuming in time. Therefore, the parallel data paths are designed for improving the performance of AES in term of speed to execute. Figure 1 shown the architecture designed in 256-bit key.

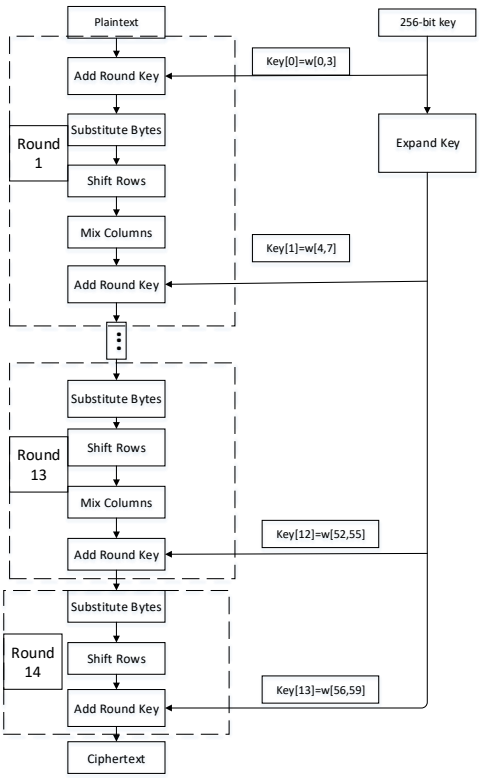

Figure 1. AES encryption with 256-bit of key length.

In Mix Columns, each column of the State is in four term polynomial. By reducing XOR gates in the critical paths, the architecture of Mix Colum is to be more efficient. The standard polynomial equations of the first column of Mix Column as shown Equation 1,

$$s'_{0,c} = \{[02] \cdot s_{0,c}\} \oplus \{[03] \cdot s_{1,c}\} \oplus s_{2,c} \oplus s_{3,c} \quad (1)$$

After the expansion of the equation, the equation shows as Equation 2

$$s'_{0,c} = [02]\, s_{0,c} \oplus [02]\, s_{1,c} \oplus s_{1,c} \oplus s_{2,c} \oplus s_{3,c} \quad (2)$$

The function of multiplication of [02] in hexadecimal is represented by the XTime as shown in Figure 2. After the expansion of the equation, the critical path of the Mix Column transformation is four with the XOR gate. An efficient of Mix Column implementation architecture can be derived by minimizing the number of XOR gates. The equation can be simplified and rewritten as Equation 3

$$s'_{0,c} = [02]\{ s_{0,c} \oplus s_{1,c} \} \oplus \{s_{1,c} \oplus s_{2,c}\} \oplus s_{3,c} \quad (3)$$

As shown in Figure 3, the critical path of the Mix Column is reduced, the three of critical paths XORed to generate the output of Mix Column. Therefore, one XTime is required with the three XOR gates.

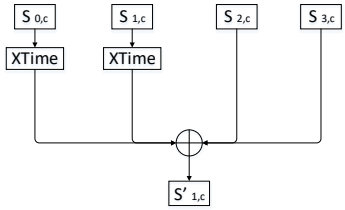

Figure 2. MixColumn transformation.

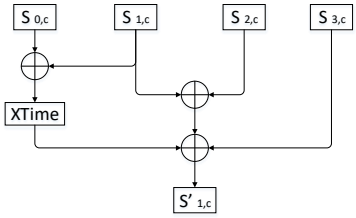

Figure 3. The efficient of Mix Column transformation

AES-GCM has two main components, AES engine and GHASH function as shown in Figure 4. In the hash subkey for GHASH function, 128-bit of "zero"block as an input is encrypted and the hash subkey will store in the register. The 96-bit of IV is appended with $0^{31}\|1$ and generated Nonce in counter 1 at the same time the 32-bit incrementing function is applied to form the next counter block. The Nonce acts as an input of AES encryption will generate the intermediate hash value Yi. When it received additional authentication data , AAD and it is appended with ciphertext, GHASH function is generated the authentication tag. The same function of authenticated encryption and authenticated decryption. The output of the authenticated decryption is authentication tag generated need to match the encryption tag. If not matched, the decryption function is failed.

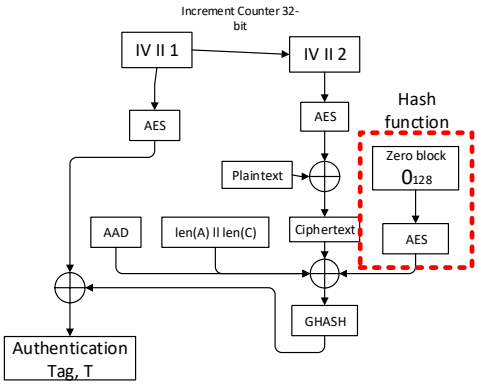

Figure 4. AES-GCM encryption/decryption.

Verilog HDL is used as the hardware description language as the flexibility to exchange among environments. The code is pure Verilog HDL that easy to implement on Cyclone V DE1-SoC board.

2.1. FPGA Implementation of AES-GCM Architecture

The AES-GCM design is programmed in Cyclone V 5CSEMAS5 FPGA on DE1-SoC board. Pins are selected in the pin plan. After programming AES-GCM in DE1-SoC board, the real functionality is done. As shown in Figure 5, DE1-SoC will encrypt and decrypt the data and it will interface with the memory addresses to store or access the data. The output data will display on computer screen. This project is suitable for the electronic applications which requires higher security, such as the Personal Cloud Storage device, Virtual Private Network (VPN) and others.

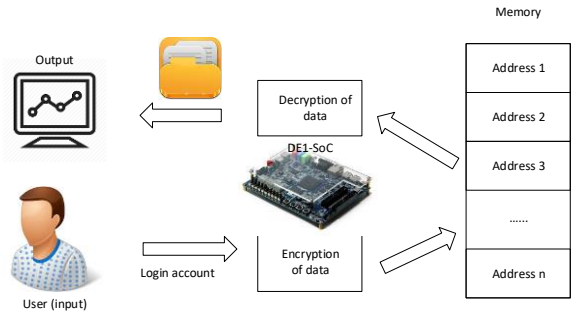

Figure 5. Overall architecture of AES-GCM.

3. Results and discussion

The proposed design is implement with Verilog HDL and it is simulated and synthesized using Quartus II. And ported to Cyclone V DE1-SoC device. The pipelined architecture of GCM operation is proposed.

The performance of AES-GCM is analysed in terms of throughput and latency. Latency is defined as the total time needed to complete the AES-GCM encryption and decryption process of the 128-bit of an input data block. The accurate value is calculated from the time a block of data enters encryption or decryption, until data leaves the process. The latency is calculated as shown in Equation 4:

Latency = Cycles per Encrypted Block(C) x Time period (T) (4)

where C is the clock per byte to encrypt a block of data, T is the reciprocal of the frequency clock, 1/ f_clk.
Latency depends on the key size and the operation mode of AES-GCM. In the parallel-pipelined of AES-GCM designed, asynchronous reset is applied in the circuit. After the reset is 0, the next stage is executed. In the proposed design, there are 10 stages to be executed. The result of latency obtained is 7.6 ns.

The throughput indicated the speed of the AES-GCM encryption and decryption process. It also defined as 128-bit of input can be encrypted and decrypted in a unit of time. Thus, the throughput of AES-GCM is calculated in bit per second (bps) as shown in Equation 5:

Throughput = 128 bits/ latency (5)

$$= 128\text{-bit}/ (7.6\times 10^{-9})$$
$$= 16.84 \text{ Gbps}$$

Table 2 shows the performance of proposed work, the throughput and the utilization of the proposed work compared with the previous works.The throughput of proposed design achieved 16.84 Gbps, utilize 11,196 slices. The throughput of proposed design compares with previous works, the proposed work is 94.93% slower than the targeted previous work. The architecture of previous work is a parallel GHASH function while the proposed work is pipelined. In the GHASH function, seven clock cycles are taken for one stage. Therefore the time consumed is more on the GHASH function. The utilization of the proposed work is lesser than the previous work, 93.17% and the architecture of the proposed work will consume less power than the previous work. The pipelined design of GHASH in AES-GCM is consumed less area of FPGA compared with the previous work.

Table 2. Performance of AES-GCM proposed.

Design	Device	Throughput (Gbps)	Frequency (MHz)	Slices
Proposed work	Altera Cyclone V	16.84	10	11,196
[17]	Xilinx Virtex Ultrascale	327.7	320	163,850
[12]	Xilinx Virtex 5	102.4	242	12,161,
[3]	Xilinx Virtex 5	82	641	2,472,
[5]	Synopsys DesignVision	0.118	233	146
[15]	Synopsys DesignVision	42.7	333	2,815
[20]	Xilinx Virtex 4	10.49	82	13,368
[16]	Altera Stratix	5.61	43.86	12,827

4. Conclusion

The proposed design was successfully completed with the implementation of authenticated encryption and decryption of AES-GCM. The proposed design is 16.84 Gbps, utilized with 11,196 slices. Compared with the previous work, the proposed design is slower than 94.93%, utilize is less than previous work 93.17%. The key length of previous work is 128-bit, but the key length proposed is 256-bit. This implementation can be to introduce in confidential corporate documents, government documents, and personal cloud storage device or person information protection which emphasis on the data security. The proposed design of AES-GCM is aimed for the high speed of the encryption and decryption.

Acknowledgement
This work was financial supported by FRGS Grant Vot No. 1538.

References

[1] A. M. Deshpande, M. S. Deshpande and D. N. Kayatanavar 2009 FPGA implementation of AES encryption and decryption *2009 International Conference on Control, Automation, Communication and Energy Conservation* pp. 1-6.
[2] M. Kosug, M. Yasuda and A. Satoh 2015 FPGA implementation of authenticated encryption algorithm Minalpher *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)* pp. 572-576
[3] M. Pitchaiah, P. Daniel, and Praveen 2012 Implementation of Advanced Encryption Standard Algorithm *Int. J. Sci. Eng. Res*. vol 3 no. 3 pp. 1–6,
[4] V. Arun, K. Vanisree, and D. L. Reddy 2014 Implementation of AES-GCM encryption algorithm for high performance and low power architecture Using FPGA *International Journal of Research and Applications* vol. 1 no. 3 pp. 120–131
[5] D. A. McGrew and J. Viega 2004 The Security and Performance of the Galois/Counter Mode (GCM) of *Operation Proceedings of the 5th International Conference on Cryptology in India*Secur pp. 343–55

[6] K. M. Abdellatif, R. Chotin-Avot, and H. Mehrez 2013 Efficient AES-GCM for VPNs using FPGAs *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)* pp. 1411–1414
[7] A. Jose 2016 A High Throughput Low Power AES-GCM for FPGAs *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* vol. 5, no. 4, pp. 183–187
[8] L. Henzen and W. Fichtner 2010 FPGA parallel-pipelined AES-GCM core for 100G ethernet applications *FPGA Parallel-Pipelined AES-GCM Core 100G Ethernet Appl. ESSCIRC 2010 - 36th Eur. Solid State Circuits Conf.* vol. 1 no. 2 pp. 202–205
[9] T. Chen, W. Huo, and Z. Liu 2010 Design and efficient FPGA implementation of Ghash core for AES-GCM *Des. Effic. FPGA Implement. Ghash Core AES-GCM* pp. 1–4
[10] H. Qin, T. Sasao, and Y. Iguchi 2006 A design of AES encryption circuit with 128-bit keys using look-up table ring on FPGA *IEICE Trans. Inf. Syst.* vol. E89–D no. 3 pp. 1139–1147
[11] M. Abdellatif, R. Chotin-Avot, and H. Mehrez 2014 Authenticated encryption on FPGAs from the static part to the reconfigurable part *Microprocess. Microsyst.* vol. 38, no. 6, pp. 526–538
[12] Navabi, Zainalabedin 1999 Verilog digital system design *McGraw-Hill*
[13] Maxfield, Clive 2011 FPGAs: instant access *Newnes*
[14] Nemati, Hamid R. 2010 Applied Cryptography for Cyber Security and Defense: Information Encryption and Cyphering *IGI Global*
[15] B. Sunar, E. Savas, and Ç. K. Koç 2003 Constructing Composite Field Representations for Efficient Conversion *Constr. Compos. F. Represent. Effic. Convers*. vol. 52 no. 11 pp. 1391–1398
[16] M. Mozaffari-Kermani and A. Reyhani-Masoleh 2010 Efficient and high-performance parallel hardware architectures for the AES-GCM *IEEE Trans. Comput.* vol. 61 no. 8 pp. 1165–1178
[17] B. Buhrow, K. Fritz, B. Gilbert, and E. Daniel 2015 A Highly Parallel AES-GCM Core for Authenticated Encryption of 400 Gb / s Network Protocols *2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig)* pp. 1–6
[18] S. M. Wadi and N. Zainal 2013 Rapid Encryption Method based on AES Algorithm for Grey Scale HD Image Encryption *Procedia Technology* vol. 11 no. 12 pp. 51–56
[19] J. Daemen, V. Rijmen, and K. U. Leuven 1999 AES Proposal : The Rijndael Block Cipher pp. 45
[20] N. Fips 2001 197: Announcing the advanced encryption standard (AES) Adv. ENCRYPTION Stand *(AES) Technology Lab. Natl. Inst. Stand.* vol. 2009 pp. 8–12