International Journal of Engineering & Technology, 7 (1.5) (2018) 230-233

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Efficient and high speed key-independent AES-based authenticated encryption architecture using FPGAs

A. Murali*, K Hari Kishore

Department of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur,
Andhra Pradesh, India 522502
**Corresponding author E-mail: amurali3@gmail.com*

Abstract

Data manipulations are made with the use of communication and networking systems. But at the same time, data integrity is also a needed and important property that must be maintained in every data communicating systems. For this, the security levels are provided with cryptographic primitives like hash functions and block ciphers which are deployed into the systems. For efficient architectures, FPGA-based systems like AES-GCM and AEGIS-128 plays in the best part of the re-configurability, which supports the security services of such communication and networking systems. We possibly focus on the performance of the systems with the high security of the FPGA bit streams. GF (2^{128}) multiplier is implemented for authentication tasks for high-speed targets. And also, the implementations were evaluated by using vertex 4.5 FPGA's

Keywords*: FPGAs, re-configurability, AES-GCM, AEGIS-128.*

1. Introduction

For confidentiality and the integrity, AES-GCM is propagated in various security levels. AES and the GHASH core components are implemented and these define the performance levels determining the inherent computation feedbacks. This is possible for only the key-synthesized scheme.

The design of high speed systems of AES-GCM regarding GHASH functions are the present challenges to overcome. For this KOA-based GHASH with FPGAs is one of the pipelined mechanisms. To perform the two processes (encryption and authentication) the components BRAMs, composite field, LUT-based sub-bytes are implemented.

Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR) define a portfolio for AE algorithms that have advantages for AES-GCM systems with efficient and secured environments even in specific states. AEGIS is a stream cipher with the large state with continuous updating states [4]. AEGIS achieves high performance with strong security alerts [4]. AEGIS-128 processes a 16-byte message block just with 5 AES-round functions.

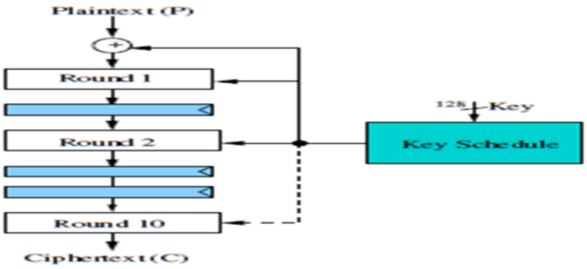


Fig. 1: Pipelined design.

In previous workouts [1, 2, 5], the different architectures of FPGA-based AES-GCM were developed. But this made clear that these contributions depict a common challenge which is related to the hardware performance of the system. Considering the high performance and the integrity, we assume an efficient pipelined KOA-based GHASH (shown in figure 2) to obtain a key-independent AES-GCM as shown in figure 1. To reduce the complexity of this approach, KOA has been chosen which the multiplication process in the GHASH is. Whereas in the other challenges like throughput, we have introduced a method to obtain a feedback-free multiplier.

From the previous workouts, [5] the reduction factor is not considered even the use of GHASH design, there is no difference in the throughput calculation result. [2] has proposed the 4-parallel AES-GCM using the pipelined KOA. It achieved the authentication in 11 clock cycles and 18 frames of 128-bits.

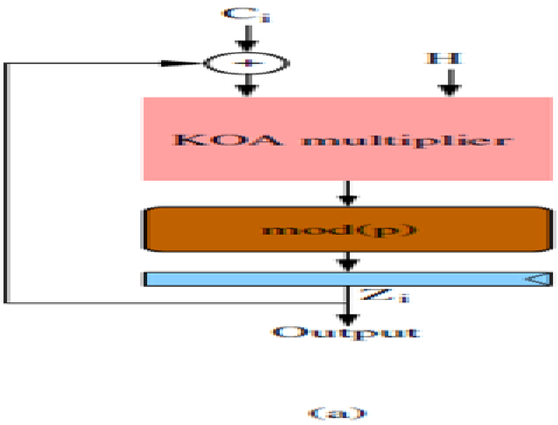


Fig. 2: (a) KOA-based GHASH.

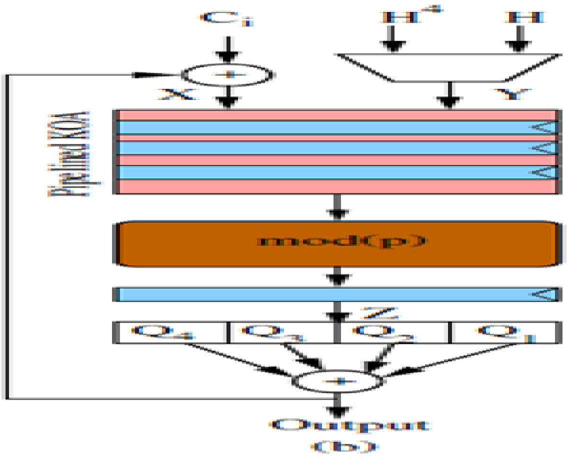
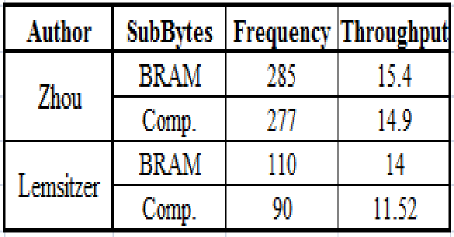


Fig. 2: (b) Pipelined KOA-based GHASH

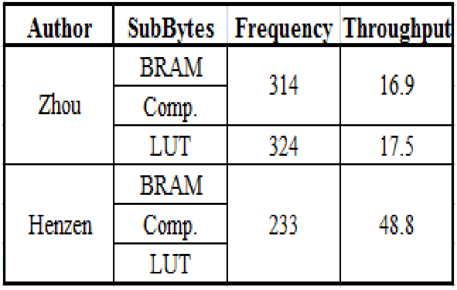
The following table shows the hardware comparisons of the previous AES-GCM systems with vertex-4, 5 on FPGAs. The maximum throughput from implementing a single AES-GCM core on Vertex-5 reads as 17.9 Gbps [5]. But as a result, the maximum throughput calculation from the parallel cores reads as 48.8 Gbps [2].

Table 1: Hardware comparison of previous AES-GC systems (Vertex-4)



Author	SubBytes	Frequency	Throughput
Zhou	BRAM	285	15.4
	Comp.	277	14.9
Lemsitzer	BRAM	110	14
	Comp.	90	11.52

Table 2: Hardware comparison of previousAES-GC systems (Vertex-5)



Author	SubBytes	Frequency	Throughput
Zhou	BRAM	314	16.9
	Comp.		
	LUT	324	17.5
Henzen	BRAM	233	48.8
	Comp.		
	LUT		

2. Efficient Koa-Based Ghash

In order to improve the performance of AES-GCM, pipelined KOA-based GHASH is introduced for the key-independent circuit. And also KOA is selected to reduce the consumed area of the design by multiplication process in the GHASH. Throughput can be changing in increasing levels as the feedback-free multiplier method is used as stated before. The figure 3 describes our proposed architecture of pipelined KOA-Based GHASH.

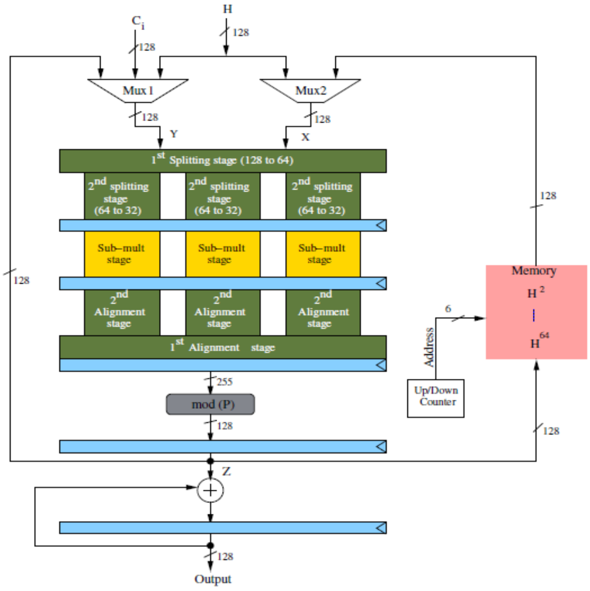


Fig. 3: Proposed Pipelined KOA-Based GHASH

Here, we use 2-step KOA. The two 128-bit inputs (C_i and H) split two times to use the 32-bit multipliers. Each of the sub-stage has three 32-bit multipliers. Hence, the complexity of the multiplier is reduced and the pipelining approach is implemented for reducing the data path of the multiplier. Finally, this architecture combines the following sections:

- 2-step pipelined "KOA for GF multiplication.
- Memory (for storing values)
- 6-bit Up-Down counters for addressing memory.
- For switching, Mux1, 2 are utilized between H, C_i and memory output.

The MAC generation is described as follows:

2.1 At Initialization stage

This includes the storing the H values in the memory. From starting point, H is passed to ports (X, Y) and the counter counts UP. Now H_2 appears on Z port after 4 clock cycles. Since we have used 4-stage pipelined KOA. After that, the memory stores the value H_2 and it is passed to Y port. Similarly, H to X port to generate H_3. This process continuously repeats till all the values are filled from H_2 to H_{64}.

2.2 Output generation

After the first stage of initialization, the output generation is automatically take part in the process. The counter starts counting Down by considering first input $C_{1.}$ This input is passed to port Y and the memory passes H_{64} to port X. After first clock cycle the process repeats. This takes end when passing the input C_{64} to port Y and H to port X. The output is then calculated by XORing Z values. It takes minimum of 64 clock cycles with 5 more additional clock cycles. Hence the throughput of the proposed architecture is shown as :

$$Throughput(Mbps)=F_{max(MHz)} \text{ X } 128 \text{ X } (64/69) \quad (1)$$

We recommend the use of CLBs for memory implementation because of 6-input LUTs or instead BRAMs can also be used as an alternative.

This approach has two main considerable advantages to the previous works.

1. It reduces the reduction factor.
2. In the case of changing the key, 252 clock cycles are needed to initialize the memory. Hence, reconfiguration is not necessary.

3. Hardware Implementation of Aegis-128 Architecture

AEGIS depends on AES round(A, B) as A is defined to be 16-byte state and B is 16-byte round Key. It uses some useful functions as Shift Rows, Sub Bytes, mix Columns and Add Round Key. These are shown in figure 4.

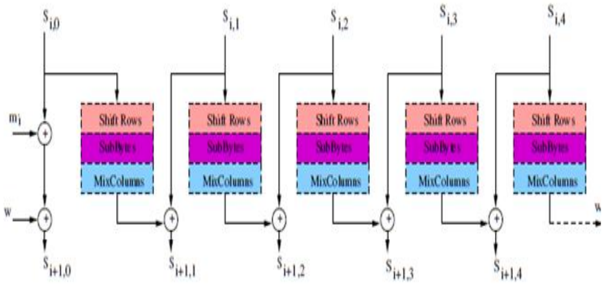


Fig. 4: The state update function of AEGIS-128

AEGIS encrypts and authenticates with 128-bit key and an initialization vector which has the message length as less than 2^{64}-bits. And the length of MAC must be less than or equal to 128-bits.

There are three stages in AEGIS-128:

1. Initialization: It loads the key and it takes 10 steps to run the cipher with that key.
2. Encryption: After the first stage of initialization, 16-byte plaintext block is used to update the state at each step of the encryption process. And this block is encrypted to cyphertext. If this size of last message block is less than 128 bits, then it is padded with 0 bits and is used to update the state. Since only a full block can be used for the updation state.
3. Finalization(MAC generation): After encryption, the authentication MAC is generated using another 7 steps. The message that is used at this stage is said to be a part of that state at the end of the encryption, length of the message and the associated data.

Coming to the next step, Decryption and the verification process. Here, only the exact values are considered for the decryption and verification process. Where as Decryption is more similar to the encryption process.

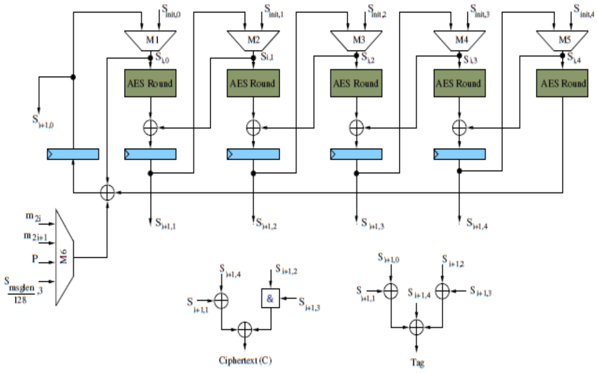


Fig. 5: Proposed High-speed AEGIS-128 Architecture

The figure 5 shows the proposed architecture which includes only 5 AES-rounds are used to obtain the state values in one clock cycle. But in hardware implementation, it needs 80 S-Boxes and 20 Mix columns. Here the sub Byte stage is implemented with LUT and composite field methods. Multiplexers from M_1 to M_5 are utilized to switch between the initialization mode to encryption and finalization modes.

Our goal is to produce an efficient and high-speed architecture of AEGIS-128. The three processes Authentication, Encryption and the Decryption are performed by five values of S. For one AES round, the values of S are updated for every 5 clock cycles.

In our proposed architecture as shown in the figure 5, the 5 AES rounds are used in order to generate the values from $S_{i+1;0}$ to $S_{i+1;4}$ in one clock cycle.

1. Initialization stage of AEGIS-128:

As discussed above, it consists of loading key into the state in 10 steps (i=-10 to -1). This state performs as follows:

The multiplexers (M_1, M_5) passes the values $S_{init;0}$, $S_{init;1}$, $S_{init;2}$, $S_{init;3}$ and $S_{init;4}$ to the five AES rounds in the first step. In further 9 steps; the values of the above variables of S are fed to the multiplexers (M_1, M_5). Finally, M_6 passes the values m2i = K_{128} or m_{2i+1}=K_{128} initialization i as odd according to the step number (even or odd). This takes 10 clock cycles.

2. Encryption of AEGIS-128:

In this stage, when M_6 passes the plaintext P, to be mixed with $S_{i;0}$. Encrypting 128-bit input takes only one clock cycle since it takes 5 AES rounds.

3. Finalization of AEGIS-128:

After processing all the input message encryption, the finalization starts in order to achieve the authentication task. At this stage, it takes only 6 clock cycles.

Any process of AEGIS-128 depends on the generation of the values from $S_{i+1;4}$ that takes only one clock cycle. And the throughput is calculated as:

$$Throughput(Mbps)=F_{max(MHz)} \text{ X } 128 \qquad (2)$$

Our approach can be tuned to handle the decryption and authentication by processing C_i for performing the decryption and there is no change in terms of the authentication stage".

4. Hardware comparison on vertex-5

We believe that, the previous design is that closest one to our proposed which can be reliable with its specification types, high performance, technology and the methodology. The key variations are used in this work and no need of reconfiguration for this architecture. Our design shows a slight change in the consumed area compared to the other architecture. And even the comparatively, the throughput goal is much better than the previous work.

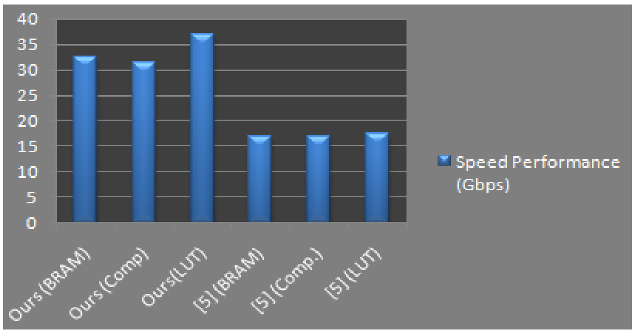


Fig. 6: Performance of the AEGIS-128 on Vertex-5

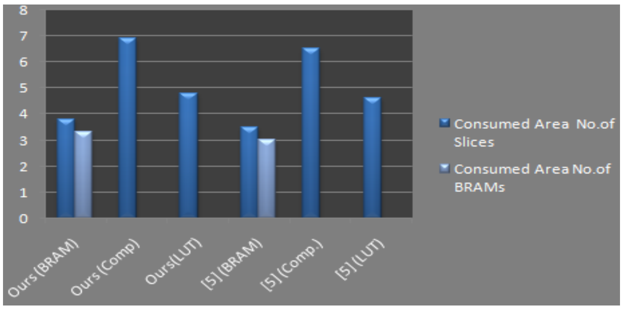


Fig. 7: Complexity of the Hardware in AEGIS-128 on Vertex-5

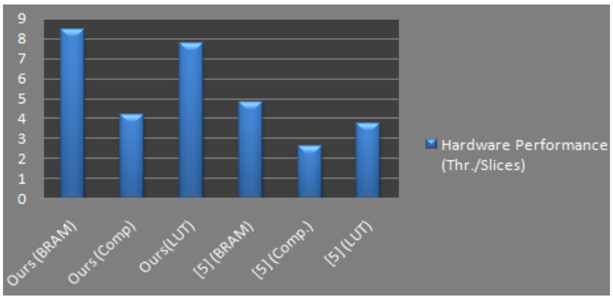


Fig. 8: Hardware performance of the AEGIS-128 on Vertex-5

5. Conclusion

In this article, we finally achieved improvement in the performance by proposing an efficient technique called pipelined KOA-based GHASH. With the GHASH function, the reduction factor of the throughput is decreased with the feedback-free design. To increase the flexibility, the 3 components BRAM, composite field and LUT-based are implemented with the multiplier. AEGIS-128 is compared with the previous results in which it performs only with five AES rounds for better performance. The throughput result is highly increased with the virtex-5 FPGAs.

References

[1] G. Zhou, H. Michalik, and L. Hinsenkamp., “Efficient and High-Throughput Implementations of AES-GCM on FPGAs”. International Conference on Field- Programmable Technology (FPT), 2007, 185-192

[2] L. Henzen and W. Fichtner., “FPGA Parallel-Pipelined AES-GCM Core for 100G Ethernet Applications”. Proceedings of the ESSCIRC, 2010, 202-205.

[3] S. Lemsitzer, J. Wolkerstorfer, N. Felber, and M. Braendli., “Multi-Gigabit GCM-AES Architecture Optimized for FPGAs. Cryptographic Hardware and Embedded Systems-CHES”, 2007, 227-238.

[4] Hongjun Wu and Bart Preneel. Aegis., “A fast authenticated encryption algorithm”. 2013. http://eprint.iacr.org/.

[5] G. Zhou, H. Michalik, and L. Hinsenkamp.., “Improving Throughput of AES-GCM with Pipelined Karatsuba Multipliers on FPGAs. Reconfigurable Computing: Architectures, Tools and Applications”, 2009, 193-203.

[6] C.W. Tseng., “Lock your designs with the virtex-4 security solution”. XCell Journal, XILINX, Spring, 2005.

[7] Xilinx. Virtex-5 FPGA Data Sheet:DC and Switching Characteristics. URL http://www.xilinx.com/support/documentation/data$_$sheets/ds202.pdf.

[8] Akashi Satoh, Sumio Morioka, Kohji Takano, and Seiji Munetoh., “A Compact Rijndael Hardware Architecture with S-box Optimization”. In Advances in CryptologyASIACRYPT 2001, 239-254. Springer, 2001.

[9] Dr. Seetaiah Kilaru, Hari Kishore K, Sravani T, Anvesh Chowdary L, Balaji T “Review and Analysis of Promising Technologies with Respect to fifth Generation Networks”, 2014 First International Conference on Networks & Soft Computing, ISSN:978-1-4799-3486-7/14,pp.270-273,August2014.

[10] Meka Bharadwaj, Hari Kishore "Enhanced Launch-Off-Capture Testing Using BIST Designs” Journal of Engineering and Applied Sciences, ISSN No: 1816-949X, Vol No.12, Issue No.3, page: 636-643, April 2017.

[11] N Bala Dastagiri, Kakarla Hari Kishore "Reduction of Kickback Noise in Latched Comparators for Cardiac IMDs” Indian Journal of Science and Technology, ISSN No: 0974-6846, Vol No.9, Issue No.43, Page: 1-6, November 2016.

[12] A Murali, K Hari Kishore, D Venkat Reddy "Integrating FPGAs with Trigger Circuitry Core System Insertions for Observability in Debugging Process” Journal of Engineering and Applied Sciences, ISSN No: 1816-949X, Vol No.11, Issue No.12, page: 2643-2650, December 2016.

[13] Mahesh Mudavath, K Hari Kishore "Design of CMOS RF Front-End of Low Noise Amplifier for LTE System Applications Integrating FPGAs” Asian Journal of Information Technology, ISSN No: 1682-3915, Vol No.15, Issue No.20, page: 4040-4047, December 2016.

[14] P Bala Gopal, K Hari Kishore, B.Praveen Kittu “An FPGA Implementation of On Chip UART Testing with BIST Techniques”, International Journal of Applied Engineering Research, ISSN 0973-4562, Volume 10, Number 14 , pp. 34047-34051, August 2015

[15] S Nazeer Hussain, K Hari Kishore "Computational Optimization of Placement and Routing using Genetic Algorithm” Indian Journal of Science and Technology, ISSN No: 0974-6846, Vol No.9, Issue No.47, page: 1-4, December 2016.

[16] N Bala Gopal, K Hari Kishore "Analysis of Low Power Low Kickback Noise in Dynamic Comparators in Pacemakers” Indian Journal of Science and Technology, ISSN No: 0974-6846, Vol No.9, Issue No.44, page: 1-4, November 2016.

[17] T. Padmapriya and V. Saminadan, “Improving Throughput for Downlink Multi user MIMO-LTE Advanced Networks using SINR approximation and Hierarchical CSI feedback”, International Journal of Mobile Design Network and Innovation- Inderscience Publisher, ISSN : 1744-2850 vol. 6, no.1, pp. 14-23, May 2015.

[18] Rajesh, M., and J. M. Gnanasekar. &quot;An optimized congestion control and error management system for OCCEM.&quot; International Journal of Advanced Research in IT and Engineering 4.4 (2015): 1-10.

[19] S.V.Manikanthan and K.srividhya "An Android based secure access control using ARM and cloud computing", Published in: Electronics and Communication Systems (ICECS), 2015 2nd International Conference on 26-27 Feb. 2015,Publisher: IEEE,DOI: 10.1109/ECS.2015.7124833.

[20] M. Rajesh, Manikanthan, “GET-UP-AND-GO EFFICIENT MEMETIC ALGORITHM BASED AMALGAM ROUTING PROTOCOL”, International Journal of Pure and Applied Mathematics, ISSN NO:1314-3395, Vol-116, No. 21, Oct 2017.