International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 16, No.1 (Aug-2024)

http://dx.doi.org/10.12785/ijcds/160167

AES 32: An FPGA implementation of Lightweight-AES for IoT Devices

Sumit Singh Dhanda[1], **Brahmjit Singh**[1] and **Poonam Jindal**[1]

[1] *Department of Electronics and Communication, National Institute of Technology, Kurukshetra, India*

Received 27 May 2023, Revised 06 May 2024, Accepted 11 May 2024, Published 10 Aug. 2024

Abstract: IoT is marked by the resource-constrained devices. Information security is the main challenge that arise due to wireless transmission of data by ubiquitous sensors. The rapid expansion of IoT setups with resource-constrained devices has spurred research into low-cost information security solutions. This study presents an efficient version of AES for high throughput. The AES's data path is 32-bit compressed. Implementation has been carried out on different FPGA families. Data path compression and use of BRAMs has led to improved throughput with savings in resource consumption. Loop-unrolled AES results in the consumption of 2669 slices which 12 times as big as this design. While 32-bit AES with 128-bit data path consumes 4 times more resources than proposed design which uses 223 slices and 5 BRAMs on Artix-7 FPGA. The proposed design delivers throughput in the range of 2.2 to 3.5 Gbps and achieves efficiency of 1.75 Mbps-7.8 Mbps per slice on different FPGAs. It outperforms different lightweight ciphers and constrained AES implementations in existing literature.

Keywords: Advance Encryption Scheme (AES), Internet of Things (IoT), Field Programmable Gate Arrays (FPGA), Data path, Information security

1. INTRODUCTION

Cisco estimates that the number of connected devices will rise from 50 billion by 2020 to reach 500 billion by 2025 [1]. To ensure the information security in Internet of Things (IoT) small sensors and devices need to be safeguarded against sniffing attacks. Smart grids are also an application of IoT. Enormous data exchange and openness of resource sharing among smart meters in smart grid have also generated challenges of data security [2]. Data privacy and data leakage are also an important concern at cloud level as well [3]. Confidentiality of the information can be enhanced with the help of Block ciphers. These are used for ensuring information security [4], [5] in various standards. Blockchain technology is another important area where cryptographic algorithms serve as the base of security [6]. For this kind of application, the most secure block cipher is thought to be Advanced Encryption Standard (AES). National Institute of Standards and Technology (NIST) [7], released FIPS-197 in which AES was adopted as a standard symmetric cipher. It ensures confidentiality at two levels

i) For high throughput applications such as e-commerce or in case of trunk communication.
ii) For lower data rates it can be used for resource constrained devices.

Software and hardware implementations of AES are utilized for these purposes. Hardware implementation of AES is preferred as compared to software implementation for high throughput applications. These implementations are carried out either on field programmable gate arrays (FPGA) or on application specific integrated chips (ASIC). Major research areas of AES implementation are highlighted in Fig 1. To minimize the delay highly pipelined architectures are implemented. Area reduction is achieved by iterative architecture. Several optimizations in the basic operations such as SubBytes or Mix-Columns, arithmetic operations etc. are also used for the same. Further, resource sharing [8] has also been used to minimize the area and increase the speed of the architecture while maintain the integrity of the cipher. Data-path reduction [9] is one of the resource sharing techniques to achieve the smaller area implementation of AES. Due to the ever-increasing demand of security solutions for resource constrained devices researchers are still working in the direction of developing new architectures of AES. Various attempts are reported in literature towards optimizations are broadly focused in two categories:

i) Pipelined (fully or partially) architecture for implementation of high speed.
ii) Compact and low-power architecture for the low resources or low-cost devices and feedback mode of operations.

E-mail address: dhandasumit@gmail.com, brahmjit.s@gmail.com, poonamjindal81@nitkkr.ac.in

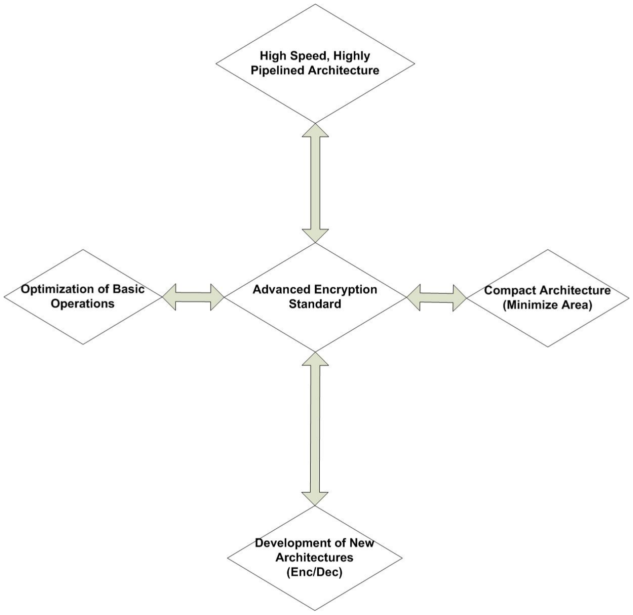
High Speed, Highly Pipelined Architecture
Optimization of Basic Operations
Advanced Encryption Standard
Compact Architecture (Minimize Area)
Development of New Architectures (Enc/Dec)

Figure 1. Research directions in AES

Major contribution of this work is to explore the adaption of AES-128 to low-cost devices in IoT by effective resource utilization. A two-step approach is applied to minimize the latency and resource consumption. First step considers the compression of data path to 32-bits. Use of BRAMs available in FPGA, maximizes the utilization of available resources. Although, this design utilizes 32-bit data path like others [10-12] but optimum utilization resources enable it to stand out among these designs. Efficient use of available block RAMs has enabled it to minimize the resources. Use of BRAMs for the implementations of the S-boxes yielded heavy reduction in the resource consumption. As the standard AES design is utilized for the work and no changes has been made to original structure and operations of AES structure. Hence, the security of the cipher remains same as original AES.

The contribution of this work are as follows:

i) In this paper, a new high performance constrained architecture has been presented for resource constrained devices.
ii) Architecture makes use of BRAMs to minimize the resource consumption on FPGAs. It achieves the optimum utilization of FPGA resources.
iii) It also presents the state of the art in the field of research.
iv) The result is compared with existing designs and lightweight implementations for IoT.

Rest of the paper is organized as follows: Various contemporary implementations are presented alongside older ones in Section 2. Section 3 provides the implementation details. Implementation results of the proposed design are presented, compared and discussed while identifying its applications in Section 4. In the end Section 5, draws conclusions and provides the future work directions.

2. RELATED WORK

In [8], authors used the 32-bit data path, resource sharing between these encryption and decryption units and subfield arithmetic to minimize the hardware requirement. In [10], a low power AES architecture with an optimized S-box have been implemented on an FPGA with 128-, 192- and 256-bit keys. An ASIC implementation for the AES processor has been carried out in [11] which is capable of delivering the throughput of 2.29 Gbps. In [12], authors carried out 32-bit implementation on FPGA with the help of pre-computed key expansion for FPGA. S-box is implemented as LUTs. The design used the dedicated memory blocks that were available on FPGA. Shift rows is performed with addressing logic. It is made possible by arranging the state-bytes in such a manner that were efficiently stored in shift registers. The same method has been used in [13] to reduce storage requirements and implement data paths of various sizes. In [14],authors have improved the FPGA resource consumption using T-box method. In [15], a theoretical design for the AES architecture was presented to optimize the resource consumption. In [9], authors carried out a fully parallel and loop unrolled implementation of AES using composite field arithmetic and LUT based T-Boxes. It was carried out for two different architectures one was 8-bit S-box based while the other was 32-bit data path. The architectures were optimized for high speed and low latency. The theoretical architecture presented by in [15] was utilized in [16] with a core added with decryption functionality and 8-bit data path. Data path contains S-box implementation in combinatorial logic. A study focused on the IoT devices and their design was presented in [17] but they left small devices. The power consumption for AES has been reported 42 mW in this study which is not appropriate for the constrained IoT devices. Hence, the 128-bit architectures mentioned in [17], [18], [19] are not suitable for the implementation in constrained devices due to power requirements. Similarly, [20] utilizes 32-bit data path and has power consumption in micro-watt level but the area requirements make it unsuitable for the small sensors.In [21], an asynchronous design has been presented for 128-bit data-path AES that consumes lesser power but the area requirements are high for the small devices and power consumption is still a concern for resource constrained devices. In [22], on FPGA, a simplified version of the AES algorithm is realised. To obtain the least amount of latency in an ad hoc voice link, the mixcolumns step is deleted.

In [23],a new AES crypto-hardware accelerator was presented for the devices such as Bluetooth controller. It uses power efficient designs for S-box, MixColumns, Shiftrows and their inverses. The area occupied is 3120 GE for the 130 nm CMOS technology. In [24], a new design named nano-AES was presented utilizing 8-bit data path.

It was an ASIC implementation which achieved 35-2.4% improvements over previous works. In [25], have presented 8-bit architecture for the SILC, CLOC, AES-JAMBU, and COLM authenticated ciphers. All of these are designed by modifying AES core. AES-JAMBU used the least resources among all of these. A crypto-engine for AES-GCM was purposed in [26], which generates the throughput of 100 Gbps. It can be utilized in optical transport networks. It is designed using 40 nm library. AES has been adapted to design a chaos-based algorithm for the encryption in [27]. It provides security for images and data. Authors have tested the scheme for different tests and attacks and high resistance has been reported against such attacks. Security issues of AES based designs are highlighted in next few works. True random number generators (TRNGs) have a statistical weakness due to physical randomness. A post-processing method can be used to solve this issue. An S-box based solution have been proposed in [28] In [29],], a correlation scan attack against XOR compaction is proposed. In [30], LC-FARES was presented. It has the capability to identify injected-faults. Sixteen 8-bit registers are used, in a 32-bit architecture, for implementing ShiftRows. A flexible AES design, that can choose from different defense mechanisms, key sizes and mode of operations etc., is presented in [31] using an agile approach. It uses Chisel framework to achieve reduced code size. Authors have designed an advanced crypto-hardware for AES. It supports variable key sizes in multiple modes [32]. The designs are synthesized using 7 nm CMOS technology. In [33], authors have presented a lightweight cipher using Lorentz-chaotic system (LCS). It occupies only 27 slices and uses feistel structure. LCS has been used to generate the random numbers which are used in the key. Numbers of works have been reported on the AES optimizations. There is need for the reduction in resource consumption of AES. Data path compression is one of the popular strategies for the area minimization. But only few works have been reported for lightweight-AES for the IoT applications. It is a big clearly highlights a gap in the literature and motivated us to adopt following methodology:

i) AES-128 is adapted to AES-32 by data path compression. BRAMs are used to further minimize the resource consumption.
ii) Verilog is used for coding the design which is synthesized on PlanAhead software.
iii) Thereafter, it was implemented on different FPGAs.
iv) Based on these FPGA implementation design is compared with existing designs.

Proposed design outperforms existing works in throughput and area.

3. 32-bit Data Path Implementation

This 32-bit iterative architecture was designed for high throughput with minimized resource-consumption. It is shown in Fig 2 below. MixColumns are 32-bit in size just like main data path. Key generation is one of the most important operation in AES. By generating keys on-the-fly one can save the time required which will result in enhanced performance. For the same purpose independent s-box is used in this design. Initial input to the proposed design has a size of 128-bits. For SubBytes operation, it is converted into four blocks of 32-bits each.

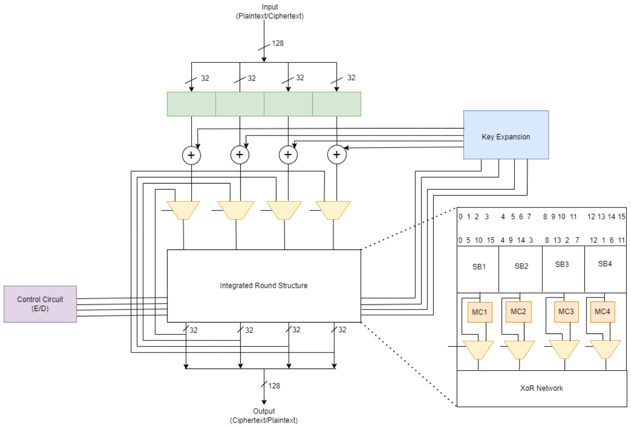

Figure 2. BRAM based 32-bit Iterative architecture

ShiftRows operates on input bytes and arrange them into a fixed sequence. Every time input is fed this sequence is repeated. This fact has led to permute the output-bytes in a fixed pattern which is similar to the standard design of Chadoweic and Gaj [12] (2003) and N. Pramstaller, et al. [13] (2004). The next operation is the MixColumns operation, which has four and is 32-bit wide. The last step is adding a round key, which is accomplished by using a different SubBytes unit. In this case, the computation is done instantly. Either galois field (GF) arithmetic or the S-Box can be stored as a look-up table (LUT) are used in the SubBytes operation. Here, S-box is implemented as a LUT, which uses a few more resources than a design based on GF. However, we have stored the S-box entries using the block RAM (BRAM) that is included in the FPGA. The same data-path and resources are used by both encryption and decryption in this system. Multiplexers have been used to make this possible. The Key Expansion unit uses a separate S-box as well, but it uses less resources now that the FPGA has been modified to use the on-board block RAM. 5 BRAMS have been used in all in the Artix-7 FPGA. It has enabled the design to reduce the resource requirement heavily. Total 256 entries have been made for the byte substitution table. In this process, the ‘case’ statement has been utilized for the byte substitutions. It is an area consuming process but it will help in faster execution of the cipher. Due to 32-bit data path and sharing of S-box, the number of cycles required to implement one round now become 5 (4 cycles for main data and one cycle for key expansion). Multiplexers helps in sharing of resources. The SubBytes architecture is shown in Fig 3. It is a 32-bit wide operation which is divided in four 8-bit wide operations individually. Hence, these are calculated individually and combined in the end. The Key expansion module utilizes

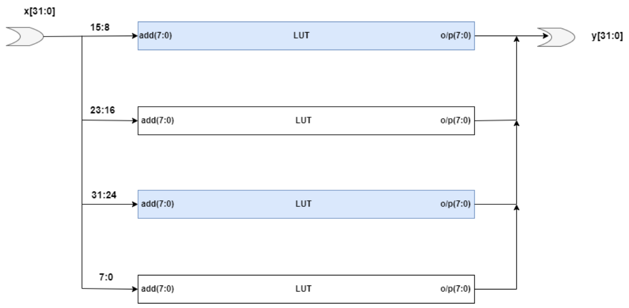

Figure 3. SubBytes structure for AES-32

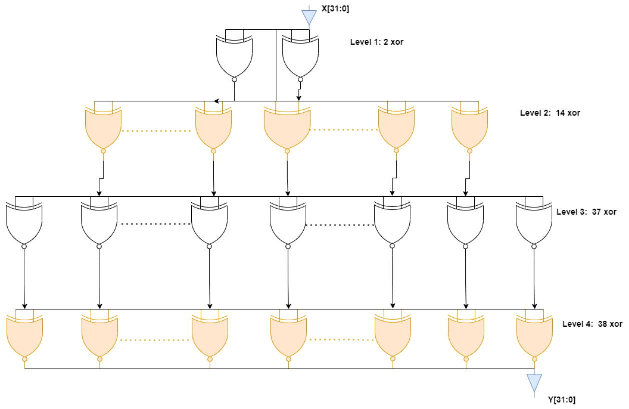

Figure 4. MixColumns structure for AES-32

a separate SubBytes module in this design and hence it is able to calculate the output in minimum cycles.

Although, the cost is paid in terms of BRAMs and additional control circuitry. Inverse SubBytes is similar to SubBytes operations and uses same number of resources.

Four levels of logic constitute the MixColumns operation. Fig 4 shows the different levels of logic used in the MixColumns design. There are total 4-levels of logic used in this design and a total of 91 XOR operations are needed. There are two XOR gates on first level, Fourteen XOR gates are present on level 2. While level 3 consist of thirty-seven gates and finally thirty-eight can be found in level 4. The inverse MixColumns operation is quite similar to the design but there are five levels of logic.

4. Result and Discussions

The initial implementation of the design is carried out with Xilinx Vivado software version 2014 and Artix-7 FPGA. On the other hand, for the comparison with exiting designs the synthesis is carried out using Xilinx PlanAhead software. Different FPGAs have been used for the implementations. While mentioning the FPGAs, we have used 'V' for Virtex, 'K' for Kintex and 'Sp' for Spartan family while numeric values 5, 6, 7 or alphabet 'E' etc. represent the generation of the particular family.

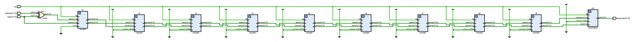

Figure 5. Top level Schematic for AES-128

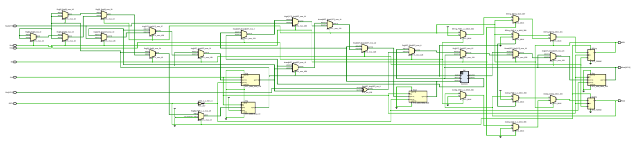

Figure 6. Schematic for AES-32 with BRAMs

This 7-series FPGA have two types of slices; slice-M and slice-L. Here, the advantage of using slice-M is that it can utilize its LUTs to configure distributed RAM (DRAM). It helps in better utilization of resources. Another strategy that we have adopted is to utilize the BRAM for S-box. BRAM on 7-series FPGAs has storage capacity up to 36 Kbits which makes it ideal suited for the S-box storage. It can also be used for other storage as well. Since S-box as LUT has 256 entries and each entry is a byte long, using slice resources or DRAM for the same will be a waste of resources. The top-level schematic of the AES-32 and AES-128 bits is displayed in Figs. 5 and 6, respectively. As previously mentioned, AES-32 is implemented as an iterative architecture, but AES-128 is constructed as a loop unrolled design.

Table I represents the resource consumption of the AES-32 on Artix-7 FPGA and its comparison with AES-128. It shows that total 568 LUTs have been used while the number of slices stands at 223. The design also utilizes 5 BRAMs available on FPGA. These BRAMs are used for the implementation of the S-boxes which are implemented as the LUT. It helps in the better resource utilization. There are two types of slices available on 7-series FPGAs slice-M and slice-L. One of Slice-M's advantages is that they may be modified to create DRAMs, which can then be utilized for storage while the software applies optimization techniques. Of the 223 slices that have been employed in our design, 40 percent are slice-M and 60 percent are slice-L. However, using BRAMs is how the resource consumption is primarily reduced. We have compared our implementation to AES-128, which was implemented on the same Artix-7 FPGA, in order to highlight the savings that our implementation has accomplished. It has loop unrolled architecture. The comparison is shown in the I and depicted in Fig 7 . It shows that AES-128 consumes 2668 slices and a total of 9571 LUTs.

The following are the outcomes for the use of the Artix-7 FPGA's resources: Compared to AES-32, there has been an overall improvement in resource utilization of 91.64 percent. We have also implemented AES-32 using a 128-bit data channel in a similar manner. It makes use of 1231 LUTs and 424 slices. These results show that an area savings of 47.40 percent can be realized simply by compressing the data route to 32 bits. The bar chart in Fig. 8 illustrates the same. The utilization of BRAMs in the

TABLE I. AES-32's comparison with the two implementation

Design	Slices	LUTs	Improvement (%)
AES-128 (loop unrolled architecture)	2669	9571	91.64
AES-32-bit with 128-bit data path	424	1231	47.40
AES-32-bit	223	568	

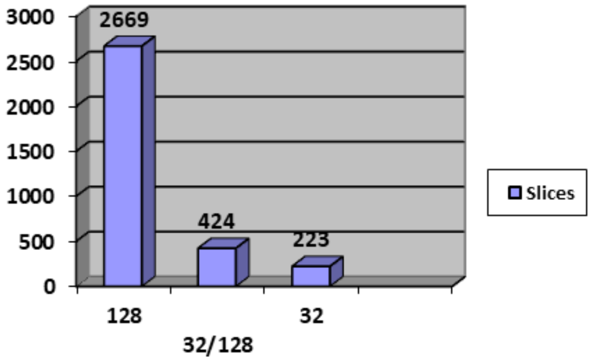

Figure 7. Comparison of three basic implementations of AES among each other

S-box implementation results in a significant reduction in area when compared to the current implementations, even for 32-bit implementations. This reduction helps in making design compact and better suited for the small devices.

The proposed design is compared with the existing ones based on three factors. The number of slices consumed for the implementation. The maximum frequency of operation (Fmax) design has clocked on the FPGA. The throughput delivered by the proposed design and its efficiency which is calculated as throughput per slice (TPS). Mega Hertz (MHz) is the unit for calculating Maximum frequency of operation. It is the maximum value recorded when design is implemented on a specific FPGA. Throughput, T is recorded in mega-bits per second (Mbps) and presented via equation (1)

$$T = \frac{B \times F}{N} \tag{1}$$

where F is the frequency at which the FPGA operates, N is the number of clock cycles used to encrypt or decrypt the entire block of data, and B is the number of bits processed at a time that makes up the block size. Equation is used to determine TPS, efficiency, and throughput per slice.

$$TPS = \frac{T}{R} \tag{2}$$

Here, R represents the total number of resources—that is, slices or LUTs—that the design uses when it is implemented on a certain FPGA. It stands for slices here. Efficiency offers a clearer picture for the precise investigation of how the design uses resources. We have synthesized the design in Xilinx PlanAhead and implemented it on various Xilinx FPGAs in order to compare it with the existing designs. It gave us the information we needed to compare the results in-depth with previous designs. In order to do the design comparison, we have used the approach described by [34], which involves using the notion of ”equivalent slices” and a ”normalization method.”

The idea of ”equivalent slices” is applied in the process of comparing the old and new gadgets. Since a lot of designs rely on lookup tables that are kept in the FPGA's BRAMs. Thus, slices and BRAMs make up the two components of resource use. BRAMs have been converted to slices for the implemented design, this value has been added to the total. After which the ”normalization method” and the idea of ”equivalent slices” were used to conduct the design comparison.

A thorough analysis of the literature indicates that various FPGA types are employed in implementation. All the FPGAs ranging from Virtex series have been used for the same. For a fair comparison, a ”normalized TPS” calculation is therefore given. The following presumptions are used to calculate the implementations of FPGA families previous to Virtex-5:

i) These FPGAs have two LUTs per slice, whereas FPGAs made after Virtex-5 have four LUTs per slice. For these FPGAs, the occupied space is therefore divided by two.
ii) 64 slices have been attributed to a single BRAM of size 18 Kb in these FPGAs, but starting with Virtex-5, single BRAM with 36 Kb size has been attributed 128 slices.
iii) Lastly, normalization factor 1.22 (550/450) has been used for multiplication for the normalization to the operating frequencies attained by these FPGAs in order to equalize the frequency of operation. This is because the highest frequency of operation for these FPGAs is 450 MHz, but the maximum frequency of operation for Virtex-5 and later models is 550 MHz.

The normalized frequency, throughput, and TPS are computed for FPGAs of the older generation—that is, those manufactured prior to Virtex-5—using this normalization criterion. We can observe the contrast between the intended design and the designs found in published works.

Fig 8 presents the resource consumption of all the designs. The design by [14] is the most resource constrained. While [12] are second most constrained implementation. Our design is fourth among these designs in terms of resource consumption. The results are depicted the equivalent slice calculations. Hence, the BRAM occupancy increases the number of slices. But being the part of the FPGA architecture, their use increases the utilization of available resources which would be wasted otherwise. This use of BRAMs helps the design to achieve high operating

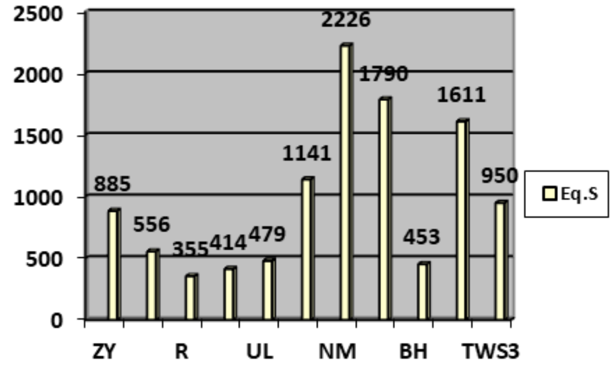

Figure 8. Comparison of Resource Consumption among Designs

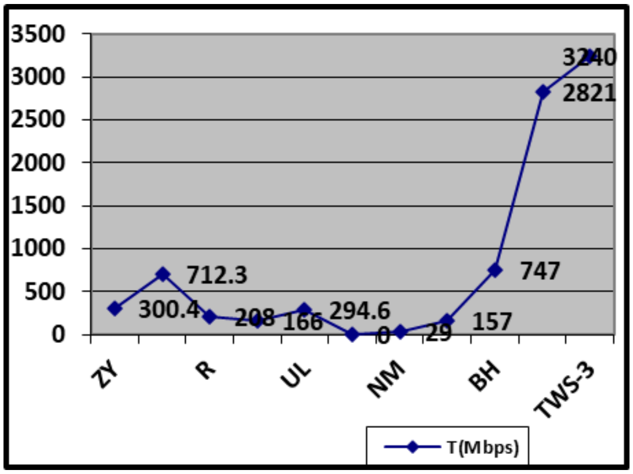

Figure 9. Throughput comparison among designs

frequencies.

Fig 9 depicts the comparison of the designs on the basis of throughput which is measured in megabits per second (Mbps). Both implementations of the proposed design provide the best throughput among presented designs with 2821 and 3240 Mbps. It shows that in terms of performance proposed design provides superior throughput. Fig 10 shows the comparison on the basis of maximum frequency of operation. These all are normalized results and designs by [41] and [36] occupies the first and second place in terms of maximum frequency. The proposed design occupies third place on Spartan-3 FPGA while fourth place is for the Virtex-5 FPGA implementation of this design. Hence, design performs satisfactorily on this parameter.

In addition to the comparison with the current designs The suggested design and the most recent lightweight ciphers suggested for the IoT needs are contrasted in Table IV. Table IV presents a comparison among different lightweight ciphers. These primitives are chosen for comparison on the basis of implementation of slices, frequency

TABLE II. COMPARISON WITH DIFFERENT DESIGNS WITH NORMALIZATION

Work	FPGA	Slice + BRAM	Equivalent Slices	Fmax (MHz)	T (Mbps)
[35] ZY	V-5	885 4992	885 4992	103.3 116	300.4 1350
[36] NB	V-5	556	556	256	712.3
[14] R	Sp-3	163+3	355	71	208
[12] CG	Sp-2	222+3	414	60	166
[37] UL	Sp-3	287+3	479	123.464 (101.2)	294.4
[38] NK	V-4	2281	1141	167.14 (137)	-
[39] NM	V800-4	4452	2226	28.06 (23)	29
[40] LO	V-E	3580	1790	-	157.07
[41] RBH	V-5	69+3	453	257	747
This work	V-5	459+9	1611	220	2821
	Sp-3	619(/2) + 10	950	253.15 (207.5)	3240.32

TABLE III. COMPARISON OF NORMALIZED THROUGHPUT AND TPS AMONG DIFFERENT DESIGNS

Work	T (Mbps)	TPS (Mbps/ Slice)
[35] ZY	300.4 1350	0.339 0.270
[36] NB	712.3	1.28
[14] R	208	0.70
[12] CG	166	0.32
[37] UL	294.4	0.61
[39] NM	29	0.013
[40] LO	157.07	0.08774
[41] RBH	747	1.65
This work V-5	2821	1.75
Sp-3	3240.32	3.414

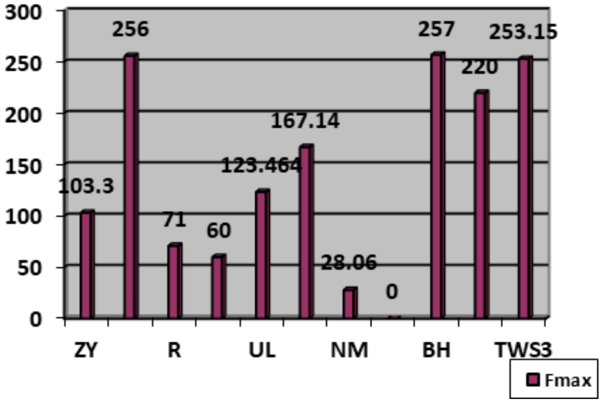

Figure 10. Frequency comparison among designs

TABLE IV. A COMPARISON WITH OTHER LIGHTWEIGHT PRIMITIVES

Design	FPGA	Slices	Fmax	Throughput (Mbps)	TPS (Mbps/slice)
PRESENT [42]	V-6	157	186.3	372.6	2.37
LED [43]	V-7	217	169.09	338.18	1.55
HIGHT [43]	V-7	252	372.3	744.6	2.950
SIMON [44]	V-7	95	219	292	3.07
XTEA [45]	K-7	228	345.9	26.42	0.115
LC-FARESS [30]	V-6	283	166	241.4	0.85
	V-7	277	310	450.9	1.62
	K-7	271	330	480	1.77
[46]	V-6	6577	170	2188.7872	3.33
	V-7	666	213	2734.5792	4.11
	K-7	536	213	2734.5792	5.10
THIS WORK	V-7	489+5	240.269	3075	6.28924
	K-7	460+5	274.907	3518	7.64958

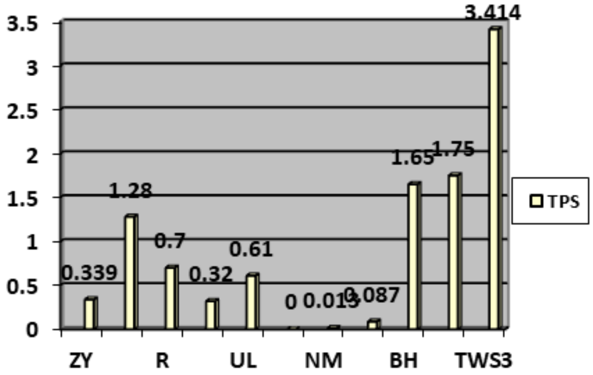

Figure 11. TPS comparison among designs

of operation, throughput and TPS. As can be observed from the table IV, proposed design is not as constrained in implementation as the other ciphers. It consumes maximum resources among all the ciphers [47]. The main reason behind this is the block size processed in these ciphers. Proposed lightweight-AES algorithm processes 128-bit block in comparison to the 64-bit block size of LED, XTEA, SIMON and HIGHT [48]. Small block size means that size of the ciphers and processing time will be small. But iterative architecture of AES-32, data path compression reduces the area required. Use of BRAMs further minimize the resource consumption. AES-32 has been compared with the design in [48] which does not utilize BRAMs and s-box has been implemented in galois field. It clearly highlights that proposed design achieves the maximum utilization of available FPGA resources. Comparison with [48] further highlights that implementation of s-box in BRAMs results in 50 to 90% improvement in the throughput. The number of cycles remained 10 due to a separate s-box for key-schedule. Hence, AES-32 is able to achieve higher throughput with small area. AES-32 can be operated in nearly same frequencies. The throughput of the design underlines the performance of the design. It is best among all the designs in the table. TPS (efficiency) data shows that the design is able to achieve the optimum utilization of FPGA resources. It presents maximum performance per unit resource among all the designs. A thorough comparison reflects that slightly more consumption of resources by the purposed design enables it to deliver best performance among all. It helps in delivering highest throughput and maximum TPS. The TPS results obtained by the AES-32 reflects the optimum utilization of resources of the FPGA along with best per slice performance. The suggested design may now be processed more quickly and with less resource consumption thanks to the use of BRAMS in the implementation. The employment of distinct ”SubBytes” for key-schedule has contributed to delivering high throughput and exhibiting low latency, which is another factor in the optimal performance. Its low resource consumption makes it a good choice for a variety of Internet of Things use cases, including surveillance, smart lighting, smart buildings, and AC control. It is desirable for rapid response applications, including smart grid applications, because of its low latency. Normally, IoT applications are associated with small devices. These devices are either sensors or actuators which transmits information on very small rate. Many IoT applications such as surveillance etc. transmits the data at higher rate. Hence, to ensure the encryption of such high information one has to have high throughput for the encryption scheme like the proposed design.

Table IV presents the comparitive performance evaluation amongst the different lightweight ciphers which is depicted in Fig 11. These primitives are chosen for comparison based on implementation of Slices, frequency of operation, throughput and TPS. As observed from the table IV, proposed design is not as constrained in implementation as the other ciphers. It consumes maximum resources among all the ciphers. Proposed lightweight-AES algorithm processes 128-bit block in comparison to the 64-bit block size of LED, XTEA, SIMON and HIGHT. Small block size means that size of the ciphers and processing time will be small. But iterative architecture of AES-32, data path compression reduces the area required. Use of BRAMs further minimize the resource consumption. The number of cycles remained 10 due to a separate s-box for key-schedule. Hence, AES-32 is able to achieve higher throughput with small area.

AES-32 can be operated in nearly same frequencies. The throughput of the design underlines the performance of the design. It is best among all the designs in the table IV. It provides 4 to 133 time more throughput in comparison to these ciphers. TPS (efficiency) data shows that the design is able to achieve the optimum utilization of FPGA resources. It presents maximum performance per unit resource among all the designs. The improvement in efficiency ranges from 2 to 66 times. A thorough comparison reflects that slightly more consumption of resources by the purposed design enables it to deliver best performance among all. It helps in delivering highest throughput and maximum TPS. The TPS results obtained by the AES-32 reflects the optimum utilization of resources of the FPGA along with best per slice performance. Use of BRAMS in the implementation has enabled the faster processing and low-resource consumption of the proposed design. The employment of distinct ”SubBytes” for key-schedule has contributed to delivering high throughput and exhibiting low latency, which is another factor in the optimal performance. Its low resource consumption makes it a good choice for a variety of Internet of Things use cases, including surveillance, smart lighting, smart buildings, and AC control. It is desirable for rapid response applications, including smart grid applications, because of its low latency. Normally, IoT applications are associated with small devices. These devices are either sensors or actuators which transmits information on very small rate. Many IoT applications such as surveillance etc. transmits the data at higher rate. Hence, to ensure the encryption of such high information one has to have high throughput for the encryption scheme like the proposed design.

5. Conclusion and Future Work

In this work, we have adapted AES-128 to AES-32 employing data path compression strategy. Sharing the resources between encryption and decryption path, the LUTs requirement is minimized. Effective utilization of FPGA resources has led to further reduction in the number of slices and improvement in throughput over existing designs. AES-32 which is nearly 6.9 times smaller in comparison to loop-unrolled AES-128 is more suitable for small IoT devices. Utilizing five on-board block RAMs overall consumption of LUTs is remarkably reduced. As a result, the number of slices needed for AES implementation is reduced to 223 slices. On the Spartan-3 chip, the suggested design achieves a throughput of 3.2 Gbps, although in other Xilinx FPGAs, it stays between 2.2 and 3.5 Gbps. The design’s efficiency also confirms that it makes the best use of the available resources. Each slice has a speed range of 1.75 Mbps to 7.8 Mbps. It is a good choice for the various Internet of Things use cases, including surveillance, smart lighting, smart buildings, and AC control, because to its high throughput and low resource requirements. For applications that require faster response times, such smart grid applications, low latency could be advantageous. BRAMS are good way for resource reduction in FPGA. But its implementation in gates occupies large area. In future, effort will be on further reduction of slice consumption and developing more lightweight-ciphers for IoT applications.

References

[1] S. Alnefaie, S. Alshehri, and A. Cherif, “A survey on access control in iot: models, architectures and research opportunities,” *International Journal of Security and Networks*, vol. 16, no. 1, pp. 60–76, 2021.

[2] E. Y. Dari, A. Bendahmane, and M. Essaaidi, “Verification-based data integrity mechanism in smart grid network,” *International Journal of Security and Networks*, vol. 16, no. 1, pp. 1–11, 2021.

[3] A. Ghorbel, M. Ghorbel, and M. Jmaiel, “A model-based approach for multi-level privacy policies derivation for cloud services,” *International Journal of Security and Networks*, vol. 16, no. 1, pp. 12–27, 2021.

[4] P. Jindal and B. Singh, “Quantitative analysis of the security performance in wireless lans,” *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 3, pp. 246–268, 2017.

[5] ——, “Analyzing the security-performance tradeoff in block ciphers,” in *International conference on computing, communication & automation*. IEEE, 2015, pp. 326–331.

[6] A. Echchabi, M. M. S. Omar, and A. M. Ayedh, “Factors influencing bitcoin investment intention: the case of oman,” *International Journal of Internet Technology and Secured Transactions*, vol. 11, no. 1, pp. 1–15, 2021.

[7] N. Fips, “Advanced encryption standard (aes) fips pub 197,” *Technology Laboratory, National Institute of Standards…*, vol. 2009, pp. 8–12, 2001.

[8] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, “A compact rijndael hardware architecture with s-box optimization,” in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2001, pp. 239–254.

[9] T. Good and M. Benaissa, “Very small fpga application-specific instruction processor for aes,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 7, pp. 1477–1486, 2006.

[10] K. U. Järvinen, M. T. Tommiska, and J. O. Skyttä, “A fully pipelined memoryless 17.8 gbps aes-128 encryptor,” in *Proceedings of the 2003 ACM/SIGDA eleventh international symposium on Field programmable gate arrays*, 2003, pp. 207–215.

[11] I. Verbauwhede, P. Schaumont, and H. Kuo, “Design and performance testing of a 2.29-gb/s rijndael processor,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 3, pp. 569–572, 2003.

[12] P. Chodowiec and K. Gaj, “Very compact fpga implementation of the aes algorithm,” in *International workshop on cryptographic hardware and embedded systems*. Springer, 2003, pp. 319–333.

[13] N. Pramstaller, S. Mangard, S. Dominikus, and J. Wolkerstorfer, “Efficient aes implementations on asics and fpgas,” in *International Conference on Advanced Encryption Standard*. Springer, 2004, pp. 98–112.

[14] G. Rouvroy, F.-X. Standaert, J.-J. Quisquater, and J.-D. Legat, “Compact and efficient encryption/decryption module for fpga implementation of the aes rijndael very well suited for small embedded applications,” in *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.*, vol. 2. IEEE, 2004, pp. 583–587.

[15] T. Järvinen, P. Salmela, P. Hämäläinen, and J. Takala, “Efficient byte permutation realizations for compact aes implementations,” in *2005 13th European Signal Processing Conference*. IEEE, 2005, pp. 1–4.

[16] P. Hamalainen, T. Alho, M. Hannikainen, and T. D. Hamalainen, “Design and implementation of low-area and low-power aes encryption hardware core,” in *9th EUROMICRO conference on digital system design (DSD’06)*. IEEE, 2006, pp. 577–583.

[17] S. Agwa, E. Yahya, and Y. Ismail, “Power efficient aes core for iot constrained devices implemented in 130nm cmos,” in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2017, pp. 1–4.

[18] Y. Wang and Y. Ha, “Fpga-based 40.9-gbits/s masked aes with area optimization for storage area network,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 1, pp. 36–40, 2013.

[19] M. M. Wong, M. D. Wong, A. K. Nandi, and I. Hijazin, “Construction of optimum composite field architecture for compact high-throughput aes s-boxes,” *IEEE transactions on very large scale integration (VLSI) systems*, vol. 20, no. 6, pp. 1151–1155, 2011.

[20] D.-H. Bui, D. Puschini, S. Bacles-Min, E. Beigné, and X.-T. Tran, “Aes datapath optimization strategies for low-power low-energy multisecurity-level internet-of-things applications,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 12, pp. 3281–3290, 2017.

[21] N. El-meligy, M. Amin, E. Yahya, and Y. Ismail, “130nm low power asynchronous aes core,” in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2017, pp. 1–4.

[22] K. Kumar, K. Ramkumar, and A. Kaur, “A lightweight aes algorithm implementation for encrypting voice messages using field programmable gate arrays,” *Journal of King Saud University-Computer and Information Sciences*, 2020.

[23] N. Ahmad and S. R. Hasan, “A new asic implementation of an advanced encryption standard (aes) crypto-hardware accelerator,” *Microelectronics Journal*, vol. 117, p. 105255, 2021.

[24] K. Shahbazi and S.-B. Ko, “Area-efficient nano-aes implementation for internet-of-things devices,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 1, pp. 136–148, 2020.

[25] M. Jahanbani, N. Bagheri, and Z. Norozi, “Lightweight implementation of silc, cloc, aes-jambu and colm authenticated ciphers,” *Microprocessors and Microsystems*, vol. 72, p. 102925, 2020.

[26] E. Mobilon and D. S. Arantes, “100 gbit/s aes-gcm cryptography engine for optical transport network systems: architecture, design and 40 nm silicon prototyping,” *Microelectronics Journal*, vol. 116, p. 105229, 2021.

[27] M. Maazouz, A. Toubal, B. Bengherbia, O. Houhou, and N. Batel, “Fpga implementation of a chaos-based image encryption algorithm,” *Journal of King Saud University-Computer and Information Sciences*, 2022.

[28] A. M. Garipcan and E. Erdem, *Design, FPGA implementation and statistical analysis of a high-speed and low-area TRNG based on an AES s-box post-processing technique*. ISA transactions, Elsevier, 2021.

[29] Y. Sao, S. S. Ali, D. Ray, S. Singh, and S. Biswas, “Co-relation scan attack analysis (cosaa) on aes: A comprehensive approach,” *Microelectronics Reliability*, vol. 123, p. 114216, 2021.

[30] S. Sheikhpour, S.-B. Ko, and A. Mahani, “A low cost fault-attack resilient aes for iot applications,” *Microelectronics Reliability*, vol. 123, p. 114202, 2021.

[31] X. Guo, M. El-Hadedy, S. Mosanu, X. Wei, K. Skadron, and M. R. Stan, “Agile-aes: Implementation of configurable aes primitive with agile design approach,” *Integration*, vol. 85, pp. 87–96, 2022.

[32] P. Nannipieri, S. Di Matteo, L. Baldanzi, L. Crocetti, L. Zulberti, S. Saponara, and L. Fanucci, “Vlsi design of advanced-features aes cryptoprocessor in the framework of the european processor initiative,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 2, pp. 177–186, 2021.

[33] A. Shailaja and K. G. Ningappa, “A low area vlsi implementation of extended tiny encryption algorithm using lorenz chaotic system,” *International Journal of Information and Computer Security*, vol. 14, no. 1, pp. 3–19, 2021.

[34] D.-S. Kundi, A. Aziz, and N. Ikram, “A high performance st-box based unified aes encryption/decryption architecture on fpga,” *Microprocessors and Microsystems*, vol. 41, pp. 37–46, 2016.

[35] Z. Yuan, Y. Wang, J. Li, R. Li, and W. Zhao, “Fpga based optimization for masked aes implementation,” in *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2011, pp. 1–4.

[36] N. Benhadjyoussef, M. Karmani, M. Machhout, and B. Hamdi, “A hybrid countermeasure-based fault-resistant aes implementation,” *Journal of Circuits, Systems and Computers*, vol. 29, no. 03, p. 2050044, 2020.

[37] U. Legat, A. Biasizzo, and F. Novak, “A compact aes core with on-line error-detection for fpga applications with modest hardware resources,” *Microprocessors and microsystems*, vol. 35, no. 4, pp. 405–416, 2011.

[38] N. Kamoun, L. Bossuet, and A. Ghazel, “Sram-fpga implementation of masked s-box based dpa countermeasure for aes,” in *2008 3rd International Design and Test Workshop*. IEEE, 2008, pp. 74–77.

[39] N. Mentens, L. Batina, B. Preneel, and I. Verbauwhede, “E09: An fpga implementation of rijndael: Trade-offs for side-channel security,” *IFAC Proceedings Volumes*, vol. 37, no. 20, pp. 493–498, 2004.

[40] L. Ordu and B. Ors, “Power analysis resistant hardware implementations of aes,” in *2007 14th IEEE International Conference on Electronics, Circuits and Systems*. IEEE, 2007, pp. 1408–1411.

[41] R. Bani-Hani, K. Mhaidat, and S. Harb, “Very compact and efficient 32-bit aes core design using fpgas for small-footprint low-power embedded applications,” *Journal of Circuits, Systems and Computers*, vol. 25, no. 07, p. 1650080, 2016.

[42] W. Zhao, Y. Wang, and R. Li, “A unified architecture for dpa-resistant present,” in *2012 International Conference on Innovations in Information Technology (IIT)*. IEEE, 2012, pp. 244–248.

[43] S. Subramanian, M. Mozaffari-Kermani, R. Azarderakhsh, and M. Nojoumian, “Reliable hardware architectures for cryptographic block ciphers led and hight,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 10, pp. 1750–1758, 2017.

[44] P. Ahir, M. Mozaffari-Kermani, and R. Azarderakhsh, “Lightweight architectures for reliable and fault detection simon and speck cryptographic algorithms on fpga,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 4, pp. 1–17, 2017.

[45] K. Tian, *Fault-Resilient Lightweight Cryptographic Block Ciphers for Secure Embedded Systems*. Rochester Institute of Technology, 2014.

[46] S. S. Dhanda, B. Singh, P. Jindal, and D. Panwar, “A highly efficient fpga implementation of aes for high throughput iot applications,”

Journal of Discrete Mathematical Sciences and Cryptography, vol. 25, no. 7, pp. 2029–2038, 2022.

[47] H. Zodpe and A. Sapkal, "An efficient aes implementation using fpga with enhanced security features," *Journal of King Saud University-Engineering Sciences*, vol. 32, no. 2, pp. 115–122, 2020.

[48] S. S. Dhanda, B. Singh, and P. Jindal, "Lightweight cryptography: a solution to secure iot," *Wireless Personal Communications*, vol. 112, no. 3, pp. 1947–1980, 2020.

Sumit Singh Dhanda received B.Tech and M.Tech degrees in Electronics and Communication Engineering from Kurukshetra University, Kurukshetra in 2005 and 2011 respectively. He has teaching experience of 10 years and currently pursuing his Doctoral Degree with ECE Department at National Institute of Technology, Kurukshetra, India. He has published 10 research papers in International/National conferences. His research interests include security algorithms for Internet of Things and wireless and mobile communication.

Brahmjit Singh has completed Bachelor of Engineering in Electronics Engineering from Malaviya National Institute of Technology, Jaipur, Master of Engineering with specialization in Microwave and Radar from Indian Institute of Technology, Roorkee and Ph.D. degree from GGS Indraprastha University, Delhi. He is with the Department of Electronics and Communication Engineering, National Institute of Technology, Kurukshetra working as Professor having 24 years of teaching and research experience. He is currently serving as Dean PD and Regional Coordinator, Regional Academic Centre for Space at NIT Kurukshetra. He has held several administrative and academic positions in NIT Kurukshetra which include Chairman ECE Department, Professor in-Charge Centre of Computing and Networking, and Member Planning and Development Board. He was also incharge of Siemens Centre of Excellence at NIT Kurukshetra. He has published 100 research papers in International/National Journals and conferences, organized several conferences and short term courses. His current research interests include 6G, Cognitive Radio, and Security Algorithms for Wireless Networks and Mobility Management in wireless networks. He has been awarded The Best Research Paper Award on behalf of 'The Institution of Engineers (India)'. He is the member of IEEE, Life member of IETE, and Life Member of ISTE.

Poonam Jindal received B.E degree in Electronics and Communication Engineering from Punjab Technical University, Punjab in 2003, M.E degree in Electronics and Communication Engineering from Thapar University, Patiala in 2005 (India). She is working as Assistant Professor with Electronics and Communication Engineering Department, National Institute of Technology, Kurukshetra, India and completed her Doctoral Degree at National Institute of Technology, Kurukshetra, India. She has published 50 research papers in International/National journals and conferences. Her research interests include security algorithms for wireless networks and mobile communication. She is a member of IEEE..