Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Design and implementation of various datapath architectures for the ANU lightweight cipher on an FPGA

Vijay DAHIPHALE[†‡1], Gaurav BANSOD[†1], Ankur ZAMBARE[1], Narayan PISHAROTY[2]

[1]*Pune Institute of Computer Technology, Pune 411043, India*

[2]*Iziel Healthcare Pvt. Ltd., Pune 411028, India*

[†]E-mail: vijaydahiphale96@gmail.com; gaurav249@gmail.com

Received Oct. 29, 2018; Revision accepted May 9, 2019; Crosschecked Mar. 6, 2020

Abstract: Since the dawn of the Internet of Things (IoT), data and system security has been the major concern for developers. Because most IoT devices operate on 8-bit controllers with limited storage and computation power, encryption and decryption need to be implemented at the transmitting and receiving ends, respectively, using lightweight ciphers. We present novel architectures for hardware implementation for the ANU cipher and present results associated with each architecture. The ANU cipher is implemented at 4-, 8-, 16-, and 32-bit datapath sizes on four different field-programmable gate array (FPGA) platforms under the same implementation condition, and the results are compared on every performance metric. Unlike previous ANU architectures, the new architectures have parallel substitution boxes (S-boxes) for high throughput and hardware optimization. With these different datapath designs, ANU cipher proves to be the obvious choice for implementing security in extremely resource-constrained systems.

Key words: Lightweight cryptography; Internet of Things (IoT); Embedded security; Encryption; FPGA; Datapath design
https://doi.org/10.1631/FITEE.1800681 **CLC number:** TN918

1 Introduction

The 21st century has witnessed the majority of advancements in technology. Decreasing the size of the electronic components has proven to follow the same path. We started with the "Internet of People" and now we are moving towards the "Internet of Things" (IoT). Today, many electronic IoT based devices are capable of accessing a network. All of these devices are connected to the Internet, and can send and receive data on a network. Security plays a major role in data transfer (Dahiphale et al., 2019b). Data Encryption Standard (DES) (National Institute of Standards and Technology, 1999) and the Advanced Encryption Standard (AES) (National Institute of Standards and Technology, 2001) allow us to implement a secure environment at both the transmitting and receiving ends. Most of the embedded systems are based on low-end 8-bit microcontrollers, which have limited computational and processing power. In addition, AES and DES ciphers require a great deal of memory, gate equivalent (GE), and power consumption for their implementation (Poschmann, 2009). Therefore, these ciphers cannot be efficiently implemented on resource-constrained devices (Poschmann, 2009). This issue has led to the emergence of a new field called "lightweight cryptography," which handles security issues involved in resource-constrained environments.

Some devices in an embedded system consist of software programmable processors, and security can be implemented in the system by introducing a lightweight cipher in its software stack. However, devices such as radio frequency identification (RFID) tags and smart cards do not have any software

‡ Corresponding author
ORCID: Vijay DAHIPHALE, https://orcid.org/0000-0002-7113-3666

programmable processors. Hence, providing security through hardware implementation of lightweight ciphers plays a very important role for these devices (Dahiphale et al., 2019a).

The International Organization for Standardization (ISO) certified cipher "PRESENT" (Bogdanov et al., 2007) requires about 1560 GE for its application-specific integrated circuit (ASIC) level hardware implementation, while the ANU (Bansod et al., 2016a) cipher needs only 1015 GE, making it much more lightweight than PRESENT. Compared with PRESENT, the memory size of ANU is smaller and the power consumption is lower. ANU needs only 22 mW dynamic power, while the PRESENT cipher consumes 39 mW power (Bansod et al., 2016b). GE required by ANU for hardware implementation is 30% less than that required by PRESENT, and the power consumption of ANU is around 42% less than that of PRESENT.

ANU is a balanced Feistel based network. ANU supports 64-bit plaintext and 128/80-bit key length, and it has a total of 25 rounds. In this study, we implement the ANU architecture on a field-programmable gate array (FPGA) to analyze the results on actual hardware. Each application in IoT features different requirements for datapath depending on the data bus size of the controller (Xu et al., 2014). To meet the demands of developers, we implement the ANU architecture using 4-, 8-, 16-, and 32-bit datapaths (Lara-Nino et al., 2017; Okabe, 2017). All these datapath architectures are compared with various metrics, such as power consumption, throughput, GE, lookup table (LUT), efficiency, clock cycle, and latency. The ANU cipher needs 16 4-bit substitution boxes (S-boxes) for 64-bit data, two 32-bit permutation layers, two 32-bit circular shift blocks, and three 2-input 32-bit exclusive ORs (EXORs).

In this study, we describe the new ANU architectures and present several FPGA implementations for ANU architectures and present the results associated with each architecture. The architectures are based on serial and parallel datapaths. Using these results, a user can choose the most suitable architecture for the application. Previous ANU architectures (Bansod et al., 2016a) are based on a single datapath for an S-box, whereas the new architectures introduce parallel S-boxes for faster operations. Optimization of datapaths has led to a significant reduction in the number of GE required for hardware synthesis. Results include the calculations of the number of S-boxes required for each cycle. All the metric comparisons are in a tabular form for all architectures.

2 Lightweight block ciphers

Lightweight block ciphers play an important role in providing security for a resource-constrained environment (Poschmann, 2009). ANU is a lightweight block cipher with a Feistel-type structure. ANU is a cipher with substitution and permutation layers to provide good nonlinearity in plaintext and ciphertext. Twenty-five rounds of ANU provide good security against basic and advanced attacks (Bansod et al., 2016a). The round-based structure of the ANU block cipher is shown in Fig. 1.

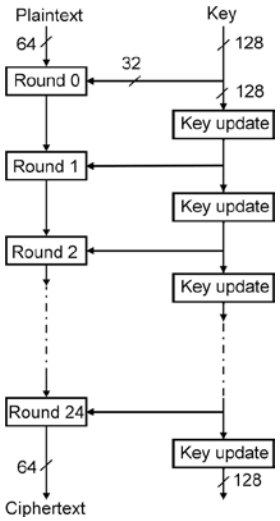

Fig. 1 Round-based structure of ANU

Initially, the key is EXORed in round zero, and after each round, the key gets updated. The updated key is then EXORed in the respective round (Fig. 1). At the end of the 25 rounds, ANU cipher produces the ciphertext and key for decryption of ciphertext at the receiving end.

3 ANU block diagram

A single round of the ANU cipher design is shown in Fig. 2. P_MSB_i consists of 32-bit MSB and

P_LSB_i consists of 32-bit LSB for the i^{th} round (MSB means the most significant bit and LSB means the least significant bit). The function F consists of a circular shift and an S-box. To provide more nonlinearity in plaintext and ciphertext, ANU uses two S-boxes in function F. BP is bit permutation, which is used to shuffle the bits and produce complexity in the cipher. RK_i consists of 32-bit LSB of the round key, which is EXORed with 32-bit LSB of data for the i^{th} round (Dahiphale et al., 2019b).

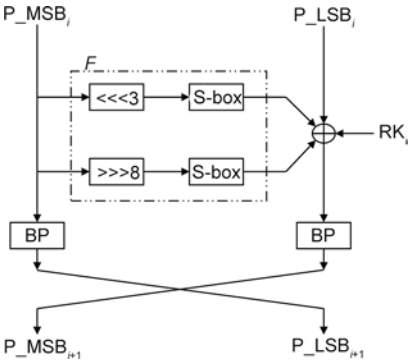

Fig. 2 Block diagram of ANU

3.1 Substitution box (S-box)

S-box is an essential component in the security analysis of the cipher (Bogdanov et al., 2007). ANU uses a 4-bit S-box, i.e., 4-bit input and 4-bit output (Fig. 3). This 4-bit S-box is used repetitively for substitution of 32-bit input data.

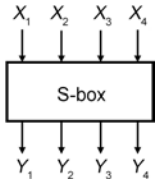

Fig. 3 4-bit S-box used in ANU

S-box is the only nonlinear block in the cipher. Therefore, values in the S-box are chosen in such a way that all S-box properties are fulfilled. The S-box used in ANU is shown in Table 1.

Table 1 S-box used in ANU

X	$S(X)$	X	$S(X)$	X	$S(X)$	X	$S(X)$
0	2	4	1	8	4	C	F
1	9	5	C	9	3	D	6
2	7	6	A	A	8	E	5
3	E	7	0	B	D	F	B

3.2 Permutation layer

ANU has a 32-bit permutation layer (P-layer). The permutation layer permutes the bits. P-layers produce good avalanche effect, and thus increase the randomness in respective bit positions (Dahiphale et al., 2019b). The 32-bit P-layer of ANU is shown in Table 2, and the diagram view is shown in Fig. 4.

Table 2 32-bit P-layer used in ANU

i	BP(i)	i	BP(i)	i	BP(i)	i	BP(i)
00	20	08	22	16	11	24	09
01	16	09	18	17	15	25	13
02	28	10	30	18	03	26	01
03	24	11	26	19	07	27	05
04	17	12	19	20	14	28	12
05	21	13	23	21	10	29	08
06	25	14	27	22	06	30	04
07	29	15	31	23	02	31	00

3.3 Circular shifts

Normal shift operations may cause loss of some bits. Therefore, circular shifts are used in the ANU cipher design. It uses two circular shifts: left circular shift by three bits (<<<3) and right circular shift by eight bits (>>>8).

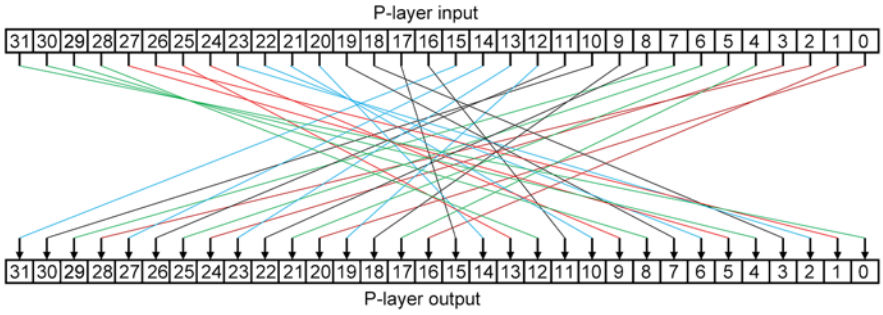

Fig. 4 P-layer structure of ANU

The diagram view of these shift operations is shown in Fig. 5.

3.4 EXOR and swapping

The 32-bit LSB from the 128-bit key is EXORed with the 32-bit LSB of plaintext for the specific round. The EXOR operation is depicted in Fig. 2. At the end of each round, 32-bit MSB and 32-bit LSB are swapped to create good avalanche effect (Dahiphale et al., 2019b).

3.5 Encryption algorithm

Algorithm 1 summaries the encryption process of the ANU cipher.

Algorithm 1 Encryption

Input: plaintext (64-bit), key (80- or 128-bit)
Output: ciphertext (64-bit, i.e., MSB and LSB after 25 rounds)

1 key=K_{127}, K_{126}, …, K_0
2 **for** i=0 to 24 **do** // i specifies the round number
3 RK_i=K_{31}, K_{30}, …, K_0
// Extracting 32-bit key from the 128-bit key
4 temp1=sbox (msb<<<3) // S-box of <<<3 32-bit MSB
5 temp2=sbox (msb>>>8) // S-box of >>>8 32-bit MSB
6 lsb=temp1⊕temp2⊕lsb⊕RK_i // EXOR of temp1, temp2, // 32-bit LSB, and 32-bit LSB of key
7 msb=player(msb), lsb=player(lsb)
// Bit permutation of 32-bit MSB and 32-bit LSB
8 swap(msb, lsb)
// Swapping of 32-bit MSB and 32-bit LSB
9 keyschedule(key) // Key scheduling algorithm
10 **end for**

3.6 Key scheduling algorithm

Key scheduling for ANU is robust, and is motivated by the PRESENT cipher (Bogdanov et al., 2007). After calling the function of "keyschedule" in the encryption algorithm, key scheduling performs Algorithms 2 and 3 (Dahiphale et al., 2019b).

Algorithm 2 128-bit key scheduling

Input: key
Output: encrypted 128-bit key of the i^{th} round

1 key=(key<<<13) // Left circular shift by 13
2 key[K_3, K_2, K_1, K_0]=S[K_3, K_2, K_1, K_0], key[K_7, K_6, K_5, K_4]=S[K_7, K_6, K_5, K_4]
// S-box substitution of 8-bit LSB of key
3 key[K_{63}, K_{62}, K_{61}, K_{60}, K_{59}]=[K_{63}, K_{62}, K_{61}, K_{60}, K_{59}]⊕i
// 5-bit EXOR with round counter (i.e., i)

Algorithm 3 80-bit key scheduling

Input: key
Output: encrypted 80-bit key of the i^{th} round

1 key=K_{79}, K_{78}, ..., K_0
2 key=(key<<<13) // Left circular shift by 13
3 key[K_3, K_2, K_1, K_0]=S[K_3, K_2, K_1, K_0]
// S-box substitution of 4-bit LSB of key
4 key[K_{63}, K_{62}, K_{61}, K_{60}, K_{59}]=[K_{63}, K_{62}, K_{61}, K_{60}, K_{59}]⊕i
// 5-bit EXOR with round counter (i.e., i)

4 FPGA implementation of cipher

The FPGA device is a configurable IC which can be configured by users in different ways. An FPGA

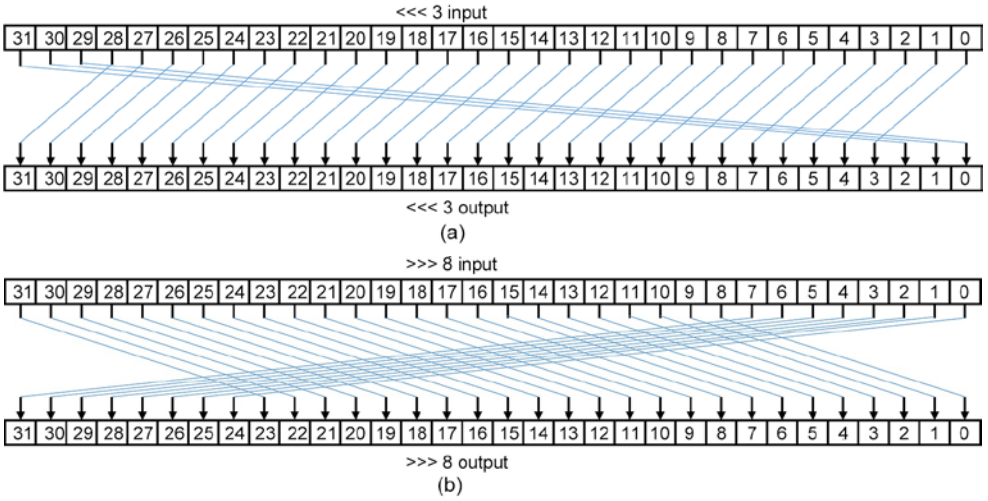

Fig. 5 Diagram view of the left circular shift operation by three bits (a) and the right circular shift operation by eight bits (b)

consists of components such as LUTs, flip flops, FPGA slices, and logic elements (Xilinx, 2018a, 2018b; Dahiphale et al., 2019a). The performance of the FPGA device is dependent on all these components (Lara-Nino et al., 2017; Okabe, 2017). Generally, complex circuits can be implemented using high-end FPGA devices. Circuit performance is calculated by different parameters, such as latency, power consumption, throughput, efficiency, the area required, the maximum frequency that can be applied to the circuit, and propagation delays (Dahiphale et al., 2019a).

FPGA implementation of the lightweight cipher is application-dependent. For example, for memory-constrained applications, a specific architecture that yields fewer GEs will be selected. Likewise, the architectures for high throughput and low latency will be different (Okabe, 2017). It is a challenging task to achieve the utmost efficiency for every parameter. A solution can be derived by proposing trade-offs in some important design metrics (Xu et al., 2014; Dahiphale et al., 2019a).

By considering different applications, we implemented ANU in four different architectures with different datapath sizes. We implemented ANU in a serialized and round-based architecture (Table 3).

Table 3 Architecture description

Architecture	Type	Data size (bit)	Key size (bit)	Datapath size (bit)
D1	Serialized	64	128	4
D2	Serialized	64	128	8
D3	Serialized	64	128	16
D4	Round-based	64	128	32

Table 3 shows that serialized implementations are with 4-, 8-, and 16-bit datapath architectures, and that round-based implementation is with a 32-bit datapath architecture.

Fig. 6 shows the top-level layout for all the ANU architectures. All the architectures have the same number of inputs and outputs although they differ in the datapath size.

4.1 4-bit datapath (D1)

D1 is the serialized architecture with a 4-bit datapath (Fig. 7). This architecture consists of two 32-bit registers, i.e., LSB and MSB, which are used as shift registers. The architecture uses two S-boxes simultaneously to speed up the operation.

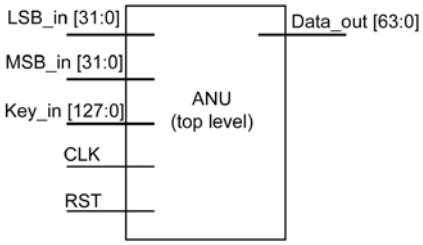

Fig. 6 A top level model of ANU

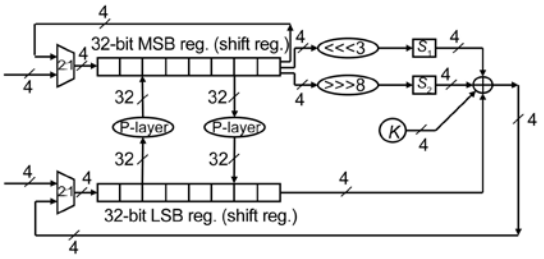

Fig. 7 4-bit datapath architecture of ANU

The 4-state finite state machine (FSM) is required to implement this architecture. Initially, the data is loaded into both the shift registers in state 0. After successfully loading the data, the encryption process starts from state 1. In the first state, the 32-bit MSB shift register in the 4-bit datapath architecture corresponding to shifting states (<<<3 and >>>8) will be selected for processing. After processing, substitution and EXOR operations are performed on these 32-bit MSB shift registers in the 4-bit datapath architecture in a single cycle. This operation on the 32-bit MSB shift register requires a total of eight clock cycles. After performing the substitution and EXOR operations on each nibble, data from both registers is circularly shifted by 4 bits, and the MSB and LSB registers act as a shift register in state 1. In the next stage, bit permutation of MSB and LSB takes place with key scheduling (Dahiphale et al., 2019b). Thus, to complete one round, the 4-bit datapath architecture requires 8+1=9 clock cycles. Fig. 8 shows the operations for each state of the D1 architecture.

Because ANU is a 25-round cipher, to encrypt the block of 64-bit data, a total of 9×25=225 clock cycles are required.

The architecture of key scheduling is shown in Fig. 9. Two S-boxes are used for key scheduling to

reduce the overhead in the implementation of the datapath (Dahiphale et al., 2019b).

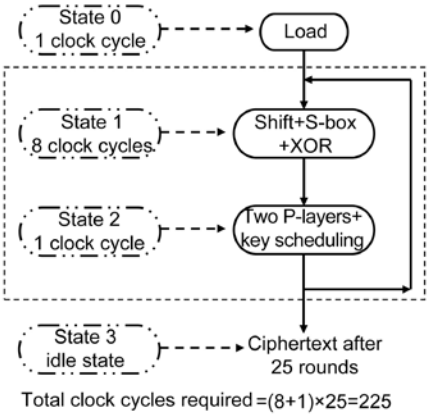

Fig. 8 Clock cycles required per state for D1

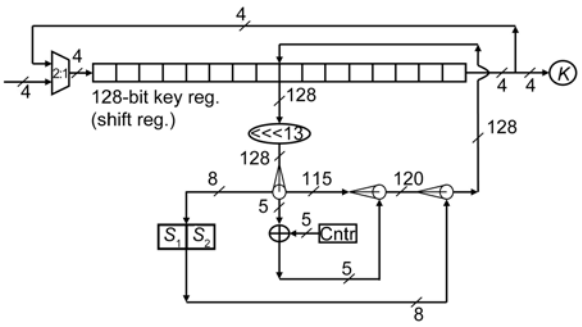

Fig. 9 Key scheduling architecture of ANU

The architecture requires the fewest GEs compared with other architectures for its hardware implementation. GE required for the ANU lightweight cipher is calculated using the standard ASIC library IBM 8RF (0.130 ms). Standard library values for different gates are shown in Table 4 (Bansod et al., 2016a; Dahiphale et al., 2019a).

Table 4 GE required for each logic circuit

Gate	GE	Gate	GE
EXOR	2	AND	1.25
MUX	2.25	OR	1.25
DFF	4.25		

GE calculations of this architecture are shown in Tables 5 and 6. The control logic for this architecture implementation requires five flip flops for the round counter, two flip flops for the state machine, and three flip flops for state 1 control. Hence, the control logic requires a total of 10 flip flops. Shift operation and permutation layers do not require any GE because these layers are implemented using wires, and practically, they may consume a few gates (Dahiphale et al., 2019a).

Table 5 GE calculation for D1

Data layer	GE required
Two 32-bit registers	2×32×4.25=272
Three 4-bit EXORs	3×4×2=24
Two S-boxes	24+24=48
Two 4-bit MUXs	2×4×2.25=18
Control logic	10×4.25=42.5
P-layer	0
Shift	0
Total	404.5

Table 6 GE calculation for key scheduling

Key schedule	GE required
128-bit register	128×4.25=544
5-bit EXOR	5×2=10
Two S-boxes	24+24=48
4-bit MUX	4×2.25=9
Total	611

As shown in Tables 5 and 6, the total number of GEs=data−layer+(key scheduling)=(404.5+611) GE= 1015.5 GE.

Hence, implementation of this architecture requires around 1015 GE.

4.2 8-bit datapath (D2)

D2 is the serialized architecture of ANU with an 8-bit datapath size. The architecture shown in Fig. 10 is similar to that of D1. The only changes are the datapath size and the number of S-boxes. A single round of this architecture requires a total of five clock cycles. Therefore, to encrypt the block of 64-bit data, it requires a total of 5×25=125 clock cycles (Dahiphale et al., 2019b). The architecture for key scheduling is similar to that of D1 except for a change in the datapath size (8 bits). For the data layer and key layer, a total of 490.25 GE and 620 GE are required for the implementation, respectively. Combining the data layer and key layer, the 8-bit datapath architecture requires a total of around 1110 GE.

4.3 16-bit datapath (D3)

The serialized architecture of ANU with a 16-bit datapath size is shown in Fig. 11. It uses a total of

eight S-boxes for the data layer and two separate S-boxes for key scheduling. The architecture is similar to those of D1 and D2. The only changes are the datapath size and the number of S-boxes. This architecture requires fewer clock cycles than D1 and D2. Hence, this architecture produces higher throughput. D3 architecture operates in a similar manner to D1 and D2. D3 requires only two clock cycles in state 1 to perform substitution and EXOR operation of 32-bit MSB shift register. State 2 performs the same function as those in D1 and D2 in a single clock cycle. Hence, 25-round ANU requires a total of (2+1)×25=75 clock cycles to encrypt a block of 64-bit data. Fig. 12 shows the operations for ANU in the 16-bit datapath architecture.

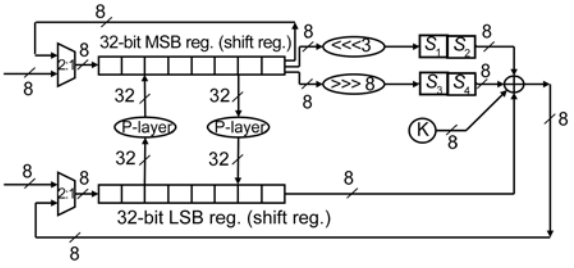

Fig. 10 8-bit datapath architecture of ANU

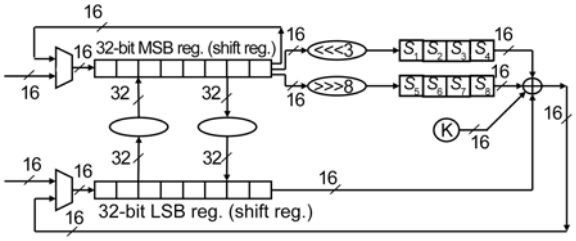

Fig. 11 16-bit datapath architecture of ANU

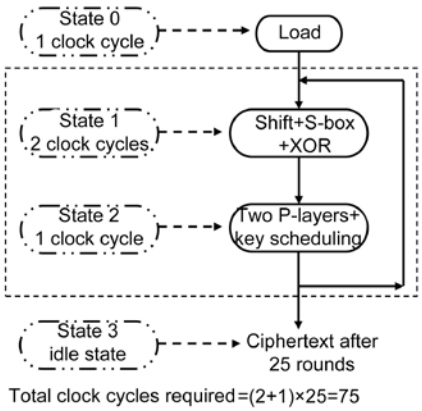

Fig. 12 Clock cycles required per state for D3

Implementation of this architecture requires 666 GE for the data layer and 638 GE for key scheduling. Hence, D3 requires a total of 1304 GE for hardware synthesis. Note that, in this case, only one flip flop is used to control state 1. Thus, the control logic requires a total of eight flip flops.

4.4 32-bit datapath (D4)

Fig. 13 shows the round-based architecture of ANU. The datapath size is 32-bit in this case. The architecture shown in Fig. 13 is a little different compared with the first three architectures (Figs. 7, 10, and 11), and it uses 16 S-boxes for data-layer implementation. It also uses two separate S-boxes for key scheduling. This architecture achieves a higher throughput compared with the first three architectures. One round of ANU is performed in just one cycle. This produces a latency of only 25 clock cycles to encrypt the block of 64-bit data. As a result, this architecture produces the highest throughput. This architecture has much less FSM overhead because one complete round is performed in only one clock cycle. Fig. 14 shows the operations of ANU in the round-based architecture.

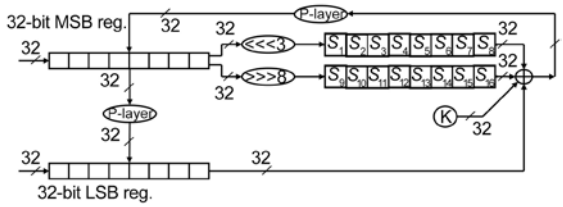

Fig. 13 32-bit datapath architecture of ANU

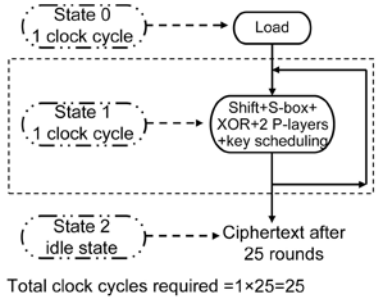

Fig. 14 Clock cycles required per state for D4

Round-based implementation requires more GE compared with the first three implementations. It requires 877.75 GE for the data layer and 674 GE for key scheduling. Therefore, D4 requires a total of 1551.75 GE for hardware implementation. In this

case, only seven flip flops (five round counters and two state machines) are required for control logic because one round is performed in one cycle. Table 7 shows the GE comparison between the proposed architectures.

Table 7 GE comparison between the proposed architectures

Architecture	Data size (bit)	Key size (bit)	GE
D1	64	128	1015
D2	64	128	1110
D3	64	128	1304
D4	64	128	1551

4.5 Performance metrics

The critical and important metrics for hardware implementation of ANU cipher in FPGAs are speed, area, power consumption, and energy dissipation. All these metrics are platform-dependent; thus, choosing the right platform for a specific datapath is very critical. The data block size for all the architectures is the same and the frequency is around 13.56 MHz.

4.5.1 Platform

ANU datapath architectures are implemented in a Xilinx FPGA using ISE design suite 14.7. The Verilog is the HDL. ANU is implemented on LUT-4 and LUT-6 based devices to implicitly know its performance in different FPGA families. Spartan-3 (xc3s700an-5fgg484) and Virtex-4 (xc4vlx25-12ff668) devices are used for LUT-4 based implementation with speed grades of −5 and −12, respectively. Spartan-6 (xc6slx45t-3fgg484) and Virtex-5 (xc5vlx50t-3ff1136) devices are used for LUT-6 based implementation with a speed grade of −3. The frequency of 13.56 MHz is the ISO standardized frequency generally used for a smart card related operations, and therefore all the results are calculated by setting the frequency of 13.56 MHz (Dahiphale et al., 2019a).

4.5.2 Footprint area

The area metric includes flip flops, LUTs, and FPGA slices to implement the cipher on an FPGA. High-performance FPGA with LUT-6 based devices gives the most compact solution for hardware implementation compared with LUT-4 based FPGAs. The values of all these parameters are mentioned in the design summary, and should be as low as possible, so that the design can be implemented efficiently in an area-constrained environment.

4.5.3 Speed

Speed parameters include latency, maximum frequency, and throughput. Latency is the number of clock cycles required to encrypt the block of 64-bit data. Latency depends only on the architecture; hence, the architecture should be designed in such a way that it has little latency. The maximum frequency for the circuit also depends upon the architecture. It is the inverse of the longest path delay. We calculated three different throughput rates for the proposed architectures, which include the maximum throughput, throughput at 13.56 MHz, and throughput per slice at the frequency of 13.56 MHz: maximum throughput (Thr)=maximum_frequency·data_block_size/latency; throughput at 13.56 MHz (Thr^*)=(13.56 MHz)·data_block_size/latency; throughput per slice=Thr^*/(number of slices); maximum throughput per slice=Thr/(number of slices).

4.5.4 Power consumption

Power consumption is an important metric for battery operated devices. There are two types of power, i.e., static power and dynamic power. Static power is the power consumed by circuit when it is stable (not operating); i.e., it is not switching. Dynamic power is the power consumed by a circuit in an operating state. The total power dissipation, i.e., the sum of these two, should be as low as possible to have good battery backup for the system. The power consumption depends on both the block size and the frequency.

4.5.5 Energy consumption

Energy consumption is the total energy required to process the block of 64-bit data. Energy depends on the total power dissipation, the latency of the architecture, and the operating frequency of the circuit. Energy consumption should be as low as possible: energy=total_power·latency/(13.56 MHz); energy per bit=energy/data_block_size.

5 Results and evaluation

All the four architectures are evaluated by different performance metrics. Table 8 shows the

comparison among D1–D4 based on the floor area and throughput. Table 9 shows the comparison of the proposed architectures based on latency, energy, and power. The most suitable architecture can be selected considering the required performance metrics. Comparisons of all the datapath architectures with other existing lightweight cipher are depicted in Tables 10 and 11. All the performance metrics are dependent only on the platform used for hardware implementation (Dahiphale et al., 2019a).

Figs. 15–23 compare different metrics of all the architectures of the ANU cipher implemented on different platforms. The platforms used for FPGA implementation are Spartan-6, Spartan-3, Virtex-5, and Virtex-4. Each graph shows a comparison between each of the 4-, 8-, 16-, and 32-bit architectures based on different performance metrics, which include area, energy, latency, power consumption, and throughput. The frequency used while calculating all the performance metrics is 13.56 MHz, which is the most appropriate for RF applications in IoT.

5.1 Area

The throughput per slice is a design metric to illustrate the efficiency of the architectures when it is desired to study the trade-off between the area reduction and performance of the architecture. Moreover, the static power for all the architectures is constant and the dynamic power is variable. The dynamic power is dependent on the switching frequency, leading to variable power consumption with a change in the frequency. The power analysis demonstrates how selecting the appropriate FPGA board can deliver a change with the significance of an order of magnitude.

Spartan-6 facilitates the most efficient implementation of the proposed architectures. On Spartan-6, 208 flip flops, 233 LUTs, and 62 FPGA slices are required for D4. The area requirement is less for D4 compared with other three architectures. This makes D4 the most suitable architecture for implementation in an area-constrained environment. Because the area requirements of D1–D3 are comparable with that of D4, any of these architectures can be efficiently implemented depending on the system requirements.

5.2 Power consumption

Spartan-3 and Spartan-6 can be used for implementation of the proposed architectures to meet the minimum power requirements of the circuit. On Spartan-6, D4, D3, D2, and D1 need the total power

Table 8 Comparisons of the proposed architectures over area and throughput

Device	Architecture	Data size (bit)	Key size (bit)	Flip flop	LUT	Slice	Latency	F_{max} (MHz)	Thr_{max} (Mb/s)	Thr^* (Mb/s)	Thr^*/slice (kb/s)
Spartan-3 xc3s700an-5fgg484	D1	64	128	208	441	258	225	238.100	67.72	3.85	14.922
	D2	64	128	210	460	265	125	243.253	124.54	6.94	26.188
	D3	64	128	210	496	276	75	241.456	206.04	11.57	41.920
	D4	64	128	199	513	272	25	262.123	671.03	34.71	127.610
Spartan-6 xc6slx45t-3fgg484	D1	64	128	220	350	95	225	266.766	75.88	3.85	40.526
	D2	64	128	216	357	93	125	266.766	136.58	6.94	74.623
	D3	64	128	211	372	100	75	257.346	219.60	11.57	115.700
	D4	64	128	208	233	62	25	351.276	899.26	34.71	559.838
Virtex-4 xc4vlx25-12ff668	D1	64	128	204	647	404	225	325.404	92.55	3.85	9.529
	D2	64	128	205	668	419	125	329.937	168.92	6.94	16.563
	D3	64	128	236	712	385	75	460.607	393.05	11.57	30.051
	D4	64	128	204	530	281	25	396.833	1015.89	34.71	123.523
Virtex-5 xc5vlx50t-3ff1136	D1	64	128	203	416	165	225	428.743	121.95	3.85	23.333
	D2	64	128	204	417	178	125	496.475	254.19	6.94	38.988
	D3	64	128	204	426	183	75	496.475	423.65	11.57	63.224
	D4	64	128	199	270	103	25	559.566	1432.48	34.71	336.990

* represents the parameter calculated at a fixed frequency of 13.56 MHz. F_{max}: maximum frequency

Table 9 Comparison of the proposed architectures over power and energy consumption

Device	Architecture	Data size (bit)	Key size (bit)	Latency	Static power (mW)	Dynamic power (mW)	Total power (mW)	Energy* (μJ)	Energy*/data_ size (nJ/bit)
Spartan-3 (xc3s700an-5fgg484)	D1	64	128	225	36.38	13.43	49.81	0.826	12.906
	D2	64	128	125	36.41	19.64	56.05	0.516	8.062
	D3	64	128	75	36.40	16.68	53.08	0.293	4.578
	D4	64	128	25	36.35	6.59	42.94	0.079	1.234
Spartan-6 (xc6slx45t-3fgg484)	D1	64	128	225	36.19	10.09	46.29	0.768	12.000
	D2	64	128	125	36.20	10.83	47.03	0.433	6.765
	D3	64	128	75	36.20	10.34	46.54	0.257	4.015
	D4	64	128	25	36.25	13.77	50.02	0.092	1.437
Virtex-4 (xc4vlx25-12ff668)	D1	64	128	225	233.05	17.16	250.21	4.151	64.859
	D2	64	128	125	233.07	18.36	251.44	2.317	36.203
	D3	64	128	75	233.06	17.55	250.60	1.386	21.656
	D4	64	128	25	233.08	18.62	251.69	0.464	7.250
Virtex-5 (xc5vlx50t-3-ff1136)	D1	64	128	225	560.26	24.79	585.04	9.707	151.671
	D2	64	128	125	560.24	22.79	583.03	5.374	83.968
	D3	64	128	75	560.22	21.28	581.50	3.216	50.250
	D4	64	128	25	560.16	15.19	575.35	1.060	16.562

* represents the parameter calculated at a fixed frequency of 13.56 MHz

Table 10 Comparison between the round-based ANU architecture (D4) and the round-based implementation of standard cipher

Device	Architecture/ cipher	Data size (bit)	Key size (bit)	Flip flop	LUT	Slice	Latency	F_{max} (MHz)	Thr_{max} (Mb/s)	Thr_{max}/slice (kb/s)
Spartan-3 (xc3s700an-5fgg484)	D4	64	128	199	513	272	25	262.123	671.03	2467.02
Spartan-6 (xc6slx45-3fgg484)	D4	64	128	208	235	61	25	351.276	899.26	14 741.96
Spartan-6 (xc6slx45t-3fgg484)	D4	64	128	208	233	62	25	351.276	899.26	14 504.19
Spartan-6 (xc6slx75-3fgg484)	D4	64	128	208	234	61	25	351.276	899.26	14 741.96
Spartan-6 (xc6slx75t-3fgg484)	D4	64	128	208	233	61	25	351.276	899.26	14 741.96
Spartan-3 (xc3s50-5) (Anandakumar et al., 2014)	LED (χ)	64	128	76	391	199	48	78.790	104.80	530.00
	LED $(\chi)^2$	64	128	76	444	227	48	87.630	116.54	510.00
	LED $(\chi)^4$	64	128	76	456	233	48	98.700	131.20	560.00
Spartan-3 (xc3s200-5ft256) (Hanley and O'Neill, 2012)	PRESENT (C1)	64	128	200	381	191	55	179.950	209.40	1096.33
Spartan-3 (xc3s400-5) (Poschmann, 2009)	PRESENT	64	128			202	32	254.000	508.00	2510.00
Spartan-3 (Bulens et al., 2008)	AES	128	128			1800		150.000	1700.00	900.00

To be continued

Table 10

Device	Architecture/ cipher	Data size (bit)	Key size (bit)	Flip flop	LUT	Slice	Latency	F_{max} (MHz)	Thr_{max} (Mb/s)	Thr_{max}/slice (kb/s)
Virtex-II (Standaert et al., 2007)	ICEBERG	64	128			631		254.000	1016.00	1610.00
Virtex-II (xc2v4000) (Mace et al., 2007)	SEA	126	126			424		145.000	156.00	370.00

Table 11 Comparison of the serialized ANU architectures (D1–D3) with the serialized implementation of standard cipher

Device	Architecture/ cipher	Datapath size (bit)	Data size (bit)	Key size (bit)	Flip flop	LUT	Slice	Latency	F_{max} (MHz)	Thr_{max} (Mb/s)	Thr_{max}/slice (kb/s)
Spartan-3 (xc3s700an-5fgg484)	D1	4	64	128	208	441	258	225	238.100	67.72	262.48
	D2	8	64	128	210	460	265	125	243.253	124.54	469.96
	D3	16	64	128	210	496	276	75	241.456	206.04	746.52
Spartan-6 (xc6slx45t-3fgg484)	D1	4	64	128	220	350	95	225	266.766	75.88	798.73
	D2	8	64	128	216	357	93	125	266.766	136.58	1468.60
	D3	16	64	128	211	372	100	75	257.346	219.60	2196.00
Spartan-3 (xc3s50-5) (Anandakumar et al., 2014)	LED (χ)	4	64	128	216	302	167	1680	137.340	5.20	30.00
		16	64	128	72	233	122	960	118.250	7.88	60.00
	LED $(\chi)^2$	8	64	128	219	388	203	912	142.010	9.97	50.00
		8	64	128	48	167	86	1152	120.750	6.71	80.00
	LED $(\chi)^4$	16	64	128	227	414	219	528	128.730	15.6	70.00
		16	64	128	70	248	127	384	117.870	19.65	150.00
Spartan-3 (xc3s200-5ft256) (Lara-Nino et al., 2017)	PRESENT (C4)	16	64	80	153	215	124	133	213.810	102.89	829.75
Spartan-3 (xc3s200-5ft256) (Lara-Nino et al., 2017)	PRESENT (C5)	16	64	128	201	264	151	136	194.630	91.59	606.55
Spartan-3 (xc3s50-5) (Yalla and Kaps, 2009)	PRESENT		64	128			117	256	114.800	28.46	240.00
	HIGHT		64	128			91	160	163.700	65.48	720.00
Spartan-3 (xc3s50-5) (Kaps, 2008)	xTEA		64	128			254	112	62.600	35.78	140.00
Spartan-3 (Exc3s500) (Guo et al., 2008)	PRESENT		64	80			271				
Spartan-3 (Exc3s500) (Aysu et al., 2014)	SIMON		128	128			36		136.000	3.60	100.00
Spartan-3 (xc3s50-5) (Chu and Benaissa, 2012)	AES		128	128			184	160	45.600	36.50	200.00
Spartan-3 (xc3s50-5) (Kaps and Sunar, 2006)	AES		128	128			393	534		16.86	40.00

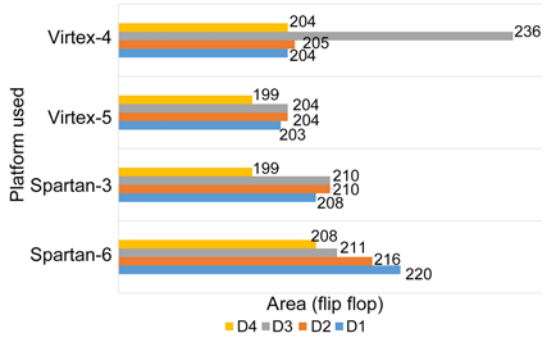

Fig. 15 Comparison of D1–D4 over the area used in terms of flip flops

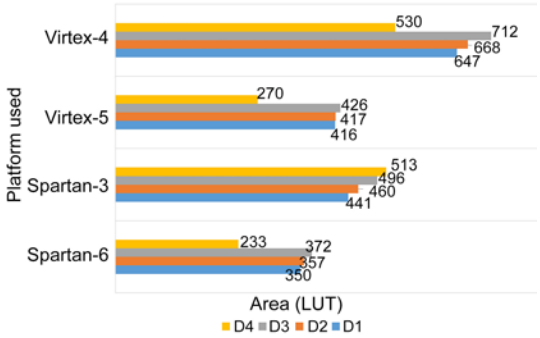

Fig. 16 Comparison of D1–D4 over the area used in terms of LUTs

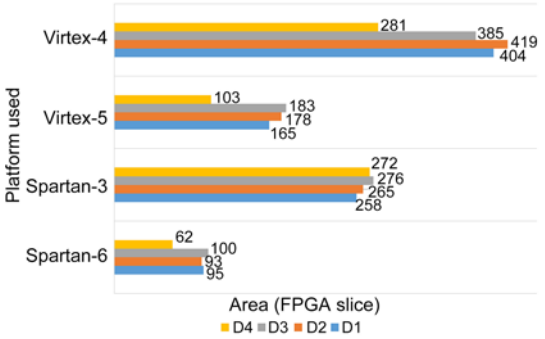

Fig. 17 Comparison of D1–D4 over the area used in terms of FPGA slices

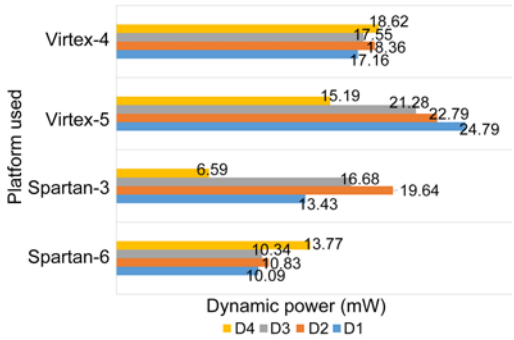

Fig. 18 Comparison of D1–D4 over the dynamic power consumption

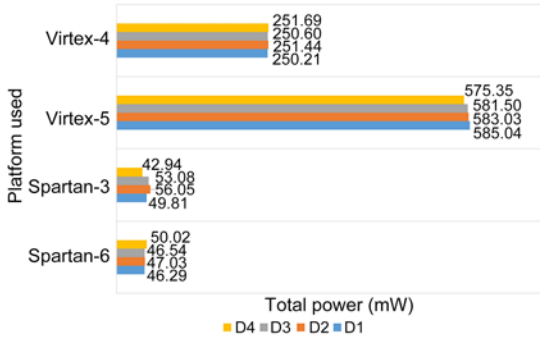

Fig. 19 Comparison of D1–D4 over the total power consumption

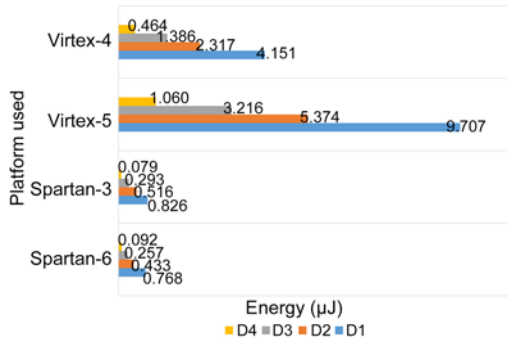

Fig. 20 Comparison of D1–D4 over the energy consumption

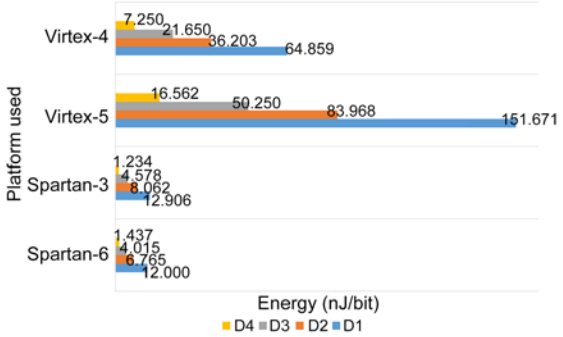

Fig. 21 Comparison of D1–D4 over the energy consumption per bit

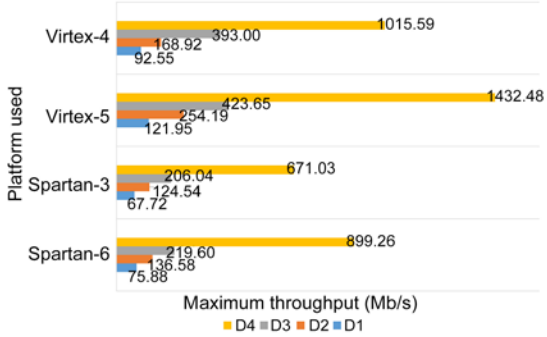

Fig. 22 Comparison of D1–D4 over the throughput at the maximum operating frequency

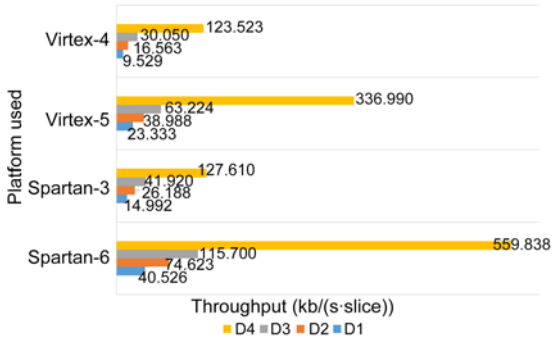

Fig. 23 Comparison of D1–D4 over the throughput per slice

of 50.02, 46.54, 47.03, and 46.29 mW, respectively. Among all the architectures, the dynamic power is variable, but the static power and the total power consumption do not vary massively. This makes the selection of architecture more resource- and frequency-dependent.

5.3 Energy

Spartan-6 provides higher energy efficiency. D4 needs only 0.092 μJ energy on Spartan-6, which is 16.45% more than Spartan-3, 80.17% less than Virtex-4, and 91.32 % less than Virtex-5.

5.4 Throughput

On each platform, the total energy requirement of D4 is less than those of three other architectures. In terms of energy per bit, D4 maintains the best results.

The throughput of all architectures is the same on all FPGA platforms at the frequency of 13.56 MHz, because the latency is independent of the platform. The changes of throughput are the maximum throughput and the throughput per slice. Because D4 is the round-based architecture, it has less latency and higher throughput compared with D1–D3 on all platforms.

From Tables 8 and 9 and Figs. 15–23, it can be concluded that LUT-6 based platforms give a more compact implementation of ANU compared with LUT-4 based platforms. Spartan-6 and Virtex-5 are LUT-6 based FPGAs, whereas Spartan-3 and Virtex-4 are LUT-4 based FPGAs.

As stated above, Spartan-6 is the most suitable FPGA platform for hardware implementation of the proposed architectures in a resource-constrained environment. Virtex-5 is preferable for higher throughput requirements. When viewing Figs. 15–23 and Tables 8 and 9 for comparison, users can choose the most suitable architecture for their applications. Every FPGA board gives the best results for the 32-bit architecture over three other architectures.

6 Conclusions

In this study, we compared different ANU cipher architectures. The architectures were defined by different datapath sizes. D1–D4 each needed 64-bit data and a 128-bit key for encryption. Depending on the datapath size, the performances varied significantly. The 4-bit datapath architecture was developed for slower applications where the time bound is flexible; similarly, the 32-bit datapath architecture was developed for faster applications where power consumption is not a major concern.

All the architectures were implemented on four different FPGA platforms. Among all these platforms, the Spartan platform allowed us to efficently implement the hardware for a resource-constrained environment, like IoT with lower throughput requirement. The Virtex FPGA board enabled the implementation of a faster architecture with higher power dissipation. Tables 8 and 11 and Figs. 15–23 can help designers and developers choose the most suitable architecture for their application.

In all the proposed architectures, the 32-bit architecture, i.e., D4, can be considered the best. It facilitated the maximum throughput, maximum throughput per slice, lower hardware requirement, lower energy consumption, lower power consumption, and lower latency than other proposed architectures.

Contributors

Vijay DAHIPHALE designed the research. Vijay DAHIPHALE and Gaurav BANSOD implemented the architectures on different FPGA platforms. Vijay DAHIPHALE, Gaurav BANSOD, and Ankur ZAMBARE drafted the manuscript. Narayan PISHAROTY helped organize the manuscript. Vijay DAHIPHALE and Gaurav BANSOD revised and finalized the paper.

Compliance with ethics guidelines

Vijay DAHIPHALE, Gaurav BANSOD, Ankur ZAMBARE, and Narayan PISHAROTY declare that they have no conflict of interest.

References

Anandakumar NN, Peyrin T, Poschmann A, 2014. A very compact FPGA implementation of LED and PHOTON. Int Conf on Cryptology in India, p.304-321. https://doi.org/10.1007/978-3-319-13039-2_18

Aysu A, Gulcan E, Schaumont P, 2014. SIMON says, break the area records for symmetric key block ciphers on FPGAs. http://eprint.iacr.org/2014/237

Bansod G, Patil A, Sutar S, et al., 2016a. ANU: an ultra lightweight cipher design for security in IoT. *Secur Commun Netw*, 9(18):5238-5251. https://doi.org/10.1002/sec.1692

Bansod G, Patil A, Sutar S, et al., 2016b. An ultra lightweight encryption design for security in pervasive computing. 2nd Int Conf on Big Data Security on Cloud, p.79-84. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2016.29

Bogdanov A, Knudsen LR, Leander G, et al., 2007. PRESENT: an ultra-lightweight block cipher. Int Workshop on Cryptographic Hardware and Embedded Systems, p.450-466. https://doi.org/10.1007/978-3-540-74735-2_31

Bulens P, Standaert FX, Quisquater JJ, et al., 2008. Implementation of the AES-128 on Virtex-5 FPGAs. Int Conf on Cryptology in Africa, p.16-26. https://doi.org/10.1007/978-3-540-68164-9_2

Chu JF, Benaissa M, 2012. Low area memory-free FPGA implementation of the AES algorithm. 22nd Int Conf on Field Programmable Logic and Applications, p.623-626. https://doi.org/10.1109/FPL.2012.6339250

Dahiphale V, Raut H, Bansod G, 2019a. Design and implementation of novel datapath designs of lightweight cipher rectangle for resource constrained environment. *Multimed Tools and Appl*, 78:23659-23688. https://doi.org/10.1007/s11042-019-7587-3

Dahiphale V, Bansod G, Zambare A, 2019b. Lightweight datapath implementation of ANU cipher for resource-constrained environments. Intelligent Computing, p.834-846. https://doi.org/10.1007/978-3-030-22868-2_57

Guo X, Chen Z, Schaumont P, 2008. Energy and performance evaluation of an FPGA-based SoC platform with AES and PRESENT coprocessors. Int Workshop on Embedded Computer Systems, p.106-115. https://doi.org/10.1007/978-3-540-70550-5_12

Hanley N, O'Neill M, 2012. Hardware comparison of the ISO/IEC 29192-2 block ciphers. IEEE Computer Society Annual Symp on VLSI, p.57-62. https://doi.org/10.1109/ISVLSI.2012.25

Kaps JP, 2008. Chai-Tea, Cryptographic Hardware Implementations of xTEA. Int Conf on Cryptology in India, p.363-375. https://doi.org/10.1007/978-3-540-89754-5_28

Kaps JP, Sunar B, 2006. Energy comparison of AES and SHA-1 for ubiquitous computing. Int Conf on Embedded and Ubiquitous Computing, p.372-381. https://doi.org/10.1007/11807964_38

Lara-Nino CA, Diaz-Perez A, Morales-Sandoval M, 2017. Lightweight hardware architectures for the PRESENT cipher in FPGA. *IEEE Trans Circ Syst*, 64(9):2544-2555. https://doi.org/10.1109/TCSI.2017.2686783

Mace F, Standaert FX, Quisquater JJ, 2007. FPGA implementation(s) of a scalable encryption algorithm. *IEEE Trans Very Large Scale Integr Syst*, 16(2):212-216. https://doi.org/10.1109/TVLSI.2007.904139

National Institute of Standards and Technology, 1999. Data Encryption Standard (DES). Federal Information Processing Standards. http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

National Institute of Standards and Technology, 2001. Advanced Encryption Standard (AES). Federal Information Processing Standards. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

Okabe T, 2017. FPGA implementation and evaluation of lightweight block cipher-BORON. *Int J Eng Dev Res*, 3(4):2321-9939.

Poschmann A, 2009. Lightweight Cryptography—Cryptographic Engineering for a Pervasive World. PhD Thesis, Europaeischer University, Germany.

Standaert FX, Piret G, Rouvroy G, et al., 2007. FPGA implementations of the ICEBERG block cipher. *Integration*, 40(1):20-27. https://doi.org/10.1016/j.vlsi.2005.12.008

Xilinx, 2018a. Spartan-3A FPGA Family: Data Sheet. http://www.xilinx.com/support/documentation/data_sheets/ds529.pdf [Accessed on Dec. 18, 2018].

Xilinx, 2018b. Spartan-6 FPGA Configuration. http://www.xilinx.com/support/documentation/user_guides/ug380.pdf [Accessed on Mar. 22, 2018].

Xu T, Wendt JB, Potkonjak M, 2014. Security of IoT systems: design challenges and opportunities. IEEE/ACM Int Conf on Computer-Aided Design, p.417-423. https://doi.org/10.1109/ICCAD.2014.7001385

Yalla P, Kaps JP, 2009. Lightweight cryptography for FPGAs. Int Conf on Reconfigurable Computing and FPGAs, p.225-230. https://doi.org/10.1109/ReConFig.2009.54