Research Article

Design Space Exploration for High-Speed Implementation of the MISTY1 Block Cipher

Raza Hasan,¹ Yasir Khizar ,² Salman Mahmood,³ and Muhammad Kashif Sheikh ³

¹Department of Computing, Middle East College, Knowledge Oasis Muscat, Seeb, Oman

²College of Information Science, Nanjing University of Aeronautics & Astronautics, Nanjing, China

³Department of Information Technology, Malaysian University of Science & Technology, Petaling Jaya, Malaysia

Correspondence should be addressed to Yasir Khizar; khizar_yasir@hotmail.com

Received 13 April 2021; Revised 22 May 2021; Accepted 1 June 2021; Published 16 June 2021

Academic Editor: Zain Anwar Ali

Copyright © 2021 Raza Hasan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes $2 \times$ unrolled high-speed architectures of the MISTY1 block cipher for wireless applications including sensor networks and image encryption. Design space exploration is carried out for 8-round MISTY1 utilizing dual-edge trigger (DET) and single-edge trigger (SET) pipelines to analyze the tradeoff w.r.t. speed/area. The design is primarily based on the optimized implementation of lookup tables (LUTs) for MISTY1 and its core transformation functions. The LUTs are designed by logically formulating S9/S7 s-boxes and FI and {FO + 32-bit XOR} functions with the fine placement of pipelines. Highly efficient and high-speed MISTY1 architectures are thus obtained and implemented on the field-programmable gate array (FPGA), Virtex-7, XC7VX690T. The high-speed/very high-speed MISTY1 architectures acquire throughput values of 25.2/43 Gbps covering an area of 1331/1509 CLB slices, respectively. The proposed MISTY1 architecture outperforms all previous MISTY1 implementations indicating high speed with low area achieving high efficiency value. The proposed architecture had higher efficiency values than the existing AES and Camellia architectures. This signifies the optimizations made for proposed high-speed MISTY1 architectures.

1. Introduction

With the advances in high-speed wireless applications, the quest to provide secure transfer of data has been of major concern [1, 2]. The efforts are underway to provide a real-time encryption solution for high data transmissions with minimum overhead in terms of power [3–5]. This study primarily focuses on high-speed implementations of a 64 bit MISTY1 block cipher for a wide range of applications, i.e., wireless networks, Ethernet devices, image encryption, and radio network controllers (RNCs) [6].

A 64 bit block cipher MISTY1 is an ISO standardized algorithm designed by Mitsubishi Corporation Electric Limited. It is used to handle a 64 bit block of data or less, e.g., 8 byte personal identification numbers (PINs), and is based on a provable 2^{-56} probability against linear/differential cryptanalysis [7–10]. The differential/integral attacks on MISTY1 require large data as well as computational complexities making it practically infeasible for breaking the

MISTY1 block cipher. The hardware architecture of MISTY1 and its major subfunctions FO and FI constitute a repetitive loop structure [11]. Therefore, the MISTY1 algorithm is suitable for the implementations of resource-constrained and high-speed applications.

To meet the requirement of the Internet of Things, cryptographic algorithms are frequently optimized for area reduction and high throughput implementation or to achieve a good tradeoff between throughput and speed [12–25]. For low-area design, reutilization/logic optimization methodologies have been widely adopted thereby implementing s-boxes using combinational logic [12–20]. A single-round MISTY1 architecture designed for compact implementation is proposed in [20] consisting of only odd-round functions, i.e., $2 \times$ FL functions, $1 \times$ FO function, and $1 \times$ 32 bit XOR. Later, more compact MISTY1 architectures were proposed comprising only one S9/S7 s-box in the FI function [12]. The compact MISTY1 architectures constitute an area of 3041 and 2331 NAND gates, respectively [12].

Finally, $2 \times$ area-efficient MISTY1 design schemes are proposed in [17] based on the combined substitution unit and threshold throughput requirements. The architectures consist of a very low area of 1853/1546 NAND gates and are the most compact implementations to date. In addition, we analyzed the throughput values of the aforementioned studies and found that the compact MISTY1 architectures attained low throughput values, i.e., ≤ 500 Mbps, and are therefore unsuitable for high-speed applications [12–14, 17, 20].

Contrary to low-area cryptographic hardware architectures, high-speed encryption algorithms utilize LUTs/RAMs or optimized combinational logic for s-boxes using pipelined schemes [20–25]. In the recent era, the focus of the studies has also shifted on the efficient implementations measured in the form of throughput-to-area ratio. Owing to high-speed and efficient implementation requirements, the architecture presented in [20] utilizes FPGA RAM blocks for the implementation of S7/S9 s-boxes. However, the straightforward implementation of LUTs for S9/S7 s-boxes (given in MISTY1 specifications) and longer path delay where $4 \times$ XOR operations are executed in a single clock cycle followed by RAM resulted in a large circuit area and reduced throughput values. The architecture presented in [21] utilizes the double-edge trigger methodology for MISTY1 high-speed pipeline implementation but has a longer path delay. Moreover, no architectural modifications/structural optimizations are made for high-speed MISTY1 implementation. On the contrary, although the MISTY1 architecture proposed in [22] achieves high speed, it costs a large area implementing a large number of pipelines. In this study, an effort has been made for high-speed and efficient MISTY1 implementation. In the last couple of years, multiple studies have been published regarding different block ciphers. In [26], researchers proposed a block cipher based on the chaotic generator and implemented it on Xilinx FPGA to prove its effectiveness. Similarly, in [27], Muthalagu and Jain took an existing block cipher algorithm and enhanced its performance to reduce the encryption time.

The unique contributions of the proposed MISTY1 $n=8$ -round pipelined architectures are as follows:

Optimized implementation of MISTY1 S9/S7 s-boxes and transformation functions, i.e., FL, FI, FO, and 32-bit XOR, by logic formulation of 4, 5, and 6 bit input LUTs for area reduction

Designing of MISTY1 and its transformation functions to attribute for the distribution of parallel processing in order to obtain a highly efficient pipelined architecture

High-speed exploration of 8-round MISTY1 architectures by employing SET and DET techniques

This paper is organized into five sections with the introduction, i.e., Section 1, followed by optimizations/designing of LUTs for the implementation of MISTY1 transformation functions described in Section 2. Section 3 proposes $2 \times$ high-speed MISTY1 architectures based on SET and DET pipeline schemes. FPGA implementation results/analysis are described in Section 4. Lastly, a brief conclusion is given in Section 5.

2. Optimized Implementation of MISTY1 Transformation Functions

2.1. FI Function. The optimizations made in the design/implementation of the proposed FI function and its constituent S9 and S7 substitution functions are elaborated in Figures 1(a)–1(e). Figures 1(a) and 1(b) depict the FI function and the equivalent FI with modified S9/S7 paths, respectively. The modifications in Figure 1(b) indicate simultaneous execution of leftmost 9 bits and rightmost 7 bits where the subscripts ‘L’ and ‘R’ represent the leftmost and rightmost bits, respectively. T stands for the TRUNCATE function, and the plus sign showing the summer function is actually the XOR gate. The XOR gate with KI_R is adding on the LSB side to reduce the path delay. The LSB bits are dependent on MSB bits, and the addition of KI_R eliminates the dependency of MSB on LSB bits. We have optimized the LUTs of LSB bits by combining S7 and XOR gate. The hardware cost is reduced by the optimization of LUTs for both MSB and LSB sides. In the next step shown as Figure 1(c), the dotted lines of Figure 1(b) are replaced by LUTs $\{(S9-1 \sim S9-3), (S9-5 \sim S9-7), \text{ and } (S7-1 \sim S7-3)\}$ concatenated by XOR gates. The upper-left LUTs (S9-1 $\sim$ S9-3) are described in Table 1 as per the modified logic expressions (i.e., S9 is used in conjunction with the zero-extended XOR operation), whereas lower-left LUTs (S9-5 $\sim$ S9-7) can be obtained by eliminating $(x_{10}, x_{11}, \dots, x_{16})$ bits from the given expressions.

The LUTs for (S7-1 $\sim$ S7-3) are employed as 4 bit and 5 bit input LUTs as described in [21]. In the steps shown in Figures 1(d) and 1(e), the XOR gates of Figure 1(c) are reordered to configure S9-4, S9-8, S7-4, and S7-5 LUTs. The proposed FI function has the primary advantage of reduced LUTs and can be executed in a maximum of 4 clock cycles. Table 2 summarizes the area reduction of 66.7% and 41.3% with the proposed FI function compared to [20, 22], respectively.

2.2. FO Function and 32-Bit XOR. MISTY1 FO transformation function is appended with the 32 bit XOR operation in odd and even rounds (except for the last round) as depicted in Figure 2(a). Therefore, the proposed LUT-based architecture of the FO function comprises $\{FO + 32 \text{ bit XOR}\}$. Figure 2(b) depicts a modified FO function indicating parallel operations for left/right 16 bits. The dotted lines are also mentioned in Figure 2(b), dividing the FO function into 4 sections with each section having side-by-side logic operations. The proposed FO function is deliberated in Figure 2(c) comprising 4 LUT blocks for left and right 16 bits, respectively.

The LUTs of the first and third section include the XOR operations, whereas the second and fourth sections comprise FI functions and XOR operations. However, the left-hand side of the second section symbolized by FI_1 is composed of $(FI + XOR)$, whereas the right-hand side of the second section includes only the FI function. Similarly, the left-hand side of the fourth section shown as FI_3 comprises $(FI + (2 \times XORs))$ as compared to the right-hand side XOR operation. Thus, the FI function described in Section 2.1 is modified as

FIGURE 1: LUT-based FI function formation. (a) FI function. (b) Modifications for the LUT-based FI function. (c) LUT-XOR structure. (d) Reordering of XORs for the formation of LUTs. (e) LUT-based FI function.

per the design requirements of FI_1 and FI_3 as shown in Figures 3 and 4, respectively.

It is evident from Figures 3(a)–3(c) and 4(a)–4(c) that changes required to incorporate XORs into the FI function will mainly require the alterations in the last part of the aforementioned FI function. Therefore, new LUTs are added in the lower right part shown as S7-6 and S7-7 for FI_1 and FI_3 , respectively. In addition, S9-8 of Figure 1(e) is replaced by newly formed LUTs S9-9 and S9-10 in the lower left section of FI_1 and FI_3 functions, respectively.

A uniformly distributed LUT-based FO function and inclusion of 32 bit XOR reduce the (initial) latency as well as the pipeline requirements of proposed MISTY1

architectures. The reduction in pipelines and latency thought is not evident from the figures, yet the proposed implementation significantly reduces the area. Table 3 summarizes the area of (FO + 32 bit XOR) showing 53.3% and 44.4% reduction compared to [20, 22]. The proposed FO function is based on the clock cycle operation required to execute FI_1 / FI_2 / FI_3 functions and will be explained in detail in Section 3.

2.3. Proposed FL Function and Area Estimation of MISTY1 Architectures. A reference FL function is shown in Figure 5(a) followed by Figure 5(b) showing FL-1 and FL-2 representing 4/3 bit input LUTs for left and right 16 bits,

TABLE 1: Logic formulation for S9 s-boxes as S9-1, S9-2, and S9-3.

y _i	S9-I	Logic expressions
y ₉	S9-1	$x_9x_5 \oplus x_9x_4 \oplus x_8x_4$
y ₈		$x_9x_7 \oplus x_6 \oplus x_8x_6 \oplus x_7x_6$
y ₇		$x_9x_8 \oplus x_8x_6 \oplus x_5 \oplus x_9x_5 \oplus x_7x_5 \oplus x_6x_5$
y ₆		$x_9 \oplus x_8x_7 \oplus x_7x_5 \oplus x_7x_1 \oplus x_5x_1$
y ₅		$x_9x_6 \oplus x_7x_6 \oplus x_9x_4 \oplus x_3 \oplus x_6x_4 \oplus x_4x_3 \oplus x_7x_3$
y ₄		$x_9x_6 \oplus x_8x_5 \oplus x_8x_3 \oplus x_6x_5 \oplus x_5x_3$
y ₃		$x_9x_8 \oplus x_8x_5 \oplus x_9x_1 \oplus x_6 \oplus x_5x_1 \oplus x_1$
y ₂		$x_9x_8 \oplus x_8 \oplus x_9x_5 \oplus x_9x_2 \oplus x_5x_2$
y ₁		$x_9 \oplus x_9x_8 \oplus x_9x_4 \oplus x_9x_2 \oplus x_9x_1$
y ₉	S9-2	$x_8x_3 \oplus x_7x_3 \oplus x_7x_2 \oplus 1$
y ₈		$x_9x_3 \oplus x_7x_3 \oplus x_9x_1 \oplus 1$
y ₇		$x_9x_3 \oplus x_5x_4 \oplus x_4x_3 \oplus x_1$
y ₆		$x_8x_4 \oplus x_8x_2 \oplus x_6x_4 \oplus x_5x_4 \oplus x_4$
y ₅		$x_5x_3 \oplus x_3x_2 \oplus x_2x_1$
y ₄		$x_6x_2 \oplus x_4x_2 \oplus x_3x_2 \oplus x_2$
y ₃		$x_7x_4 \oplus x_7x_2 \oplus x_5x_4 \oplus x_4x_2$
y ₂		$x_8x_7 \oplus x_7x_6 \oplus x_8x_3 \oplus x_8x_1 \oplus x_6x_3$
y ₁		$x_8x_7 \oplus x_7x_4 \oplus x_4x_3$
y ₉	S9-3	$x_6x_2 \oplus x_6x_1 \oplus x_5x_1$
y ₈		$x_6x_5 \oplus x_5x_4 \oplus x_2 \oplus x_6x_1 \oplus x_4x_1$
y ₇		$x_8x_2 \oplus x_6x_2 \oplus x_16$
y ₆		$x_4x_3 \oplus x_3x_2 \oplus x_15$
y ₅		$x_8 \oplus x_7x_1 \oplus x_14$
y ₄		$x_9x_1 \oplus x_7 \oplus x_2x_1 \oplus x_13$
y ₃		$x_3x_1 \oplus x_2x_1 \oplus x_12 \oplus 1$
y ₂		$x_4 \oplus x_3x_2 \oplus x_11 \oplus 1$
y ₁		$x_6x_3 \oplus x_6x_1 \oplus x_5 \oplus x_3x_1 \oplus x_10 \oplus 1$

TABLE 2: Area reduction of the proposed FI function implemented on Xilinx Virtex-7.

Method	Ftns	LUTs				Area (slices)	% reduction	
		2-1	3-1	4-1	5-1		With respect to [10]	With respect to [12]
Prop.	S9-1	—	—	3	6	27	66.7	41.3
	S9-2	—	—	6	3			
	S9-3	—	—	6	3			
	S9-4	—	—	9	—			
	S9-5	—	—	4	5			
	S9-6	—	—	9	—			
	S9-7	—	1	6	2			
	S9-8	—	—	9	—			
	S7-1	—	—	1	6			
	S7-2	—	—	6	1			
	S7-3	—	—	6	1			
	S7-4	—	—	—	7			
	S7-5	7	—	—	—			
	FI	7	1	65	34			
[10]	FI	—	—	—	—	81		
[12]	FI	—	—	—	—	46		

respectively. Thus, area for $n = 8$ -round MISTY1 architecture can be computed by summation of LUTs required for $10 \times \text{FL}$ functions, $8 \times (\text{FO} + 32 \text{ bit XOR})$ functions, and

extended key generation function, i.e., $8 \times \text{FI}_2$ functions. Table 4 summarizes the area for proposed MISTY1 architectures.

FIGURE 2: LUT-based architecture for the FO function. (a) FO function + 32 bit XOR. (b) Modified FO function. (c) LUT-based FO function.

FIGURE 3: FI_1 function. (a) FI_1 function. (b) LUT-XOR structure of the FI_1 function. (c) LUT-based FI_1 function.

FIGURE 4: FI₃ function. (a) FI₃ function. (b) LUT-XOR structure of the FI₃ function. (c) LUT-based FI₃ function.

3. Design Space Exploration for High-Speed MISTY1 Architectures

3.1. Architecture 1: DET Pipeline Architecture for High-Speed MISTY1. A high-speed MISTY1 pipelined architecture is shown in Figure 6, whereas the respective FO and FI functions (only the FI₂ function is shown for reference) are depicted in Figures 7(a) and 7(b). High-speed MISTY1 comprises 8-round architecture with 5-stage and 10-stage

pipelines in odd and even rounds, respectively. The number of pipelines in odd and even rounds of MISTY1 is based on the number of clock cycles required to execute FO/FI functions. A double-edge-triggered pipeline is employed with each LUT triggering on alternate clock cycles. This reduces the pipeline requirements of the MISTY1 architecture; however, it has a path delay of $2 \times \text{LUTs}$ as mentioned in [11]. The proposed MISTY1 architecture can process $41 \times$ plaintexts and outputs the

FIGURE 5: FL function. (a) FL function. (b) LUT-based FL function.

TABLE 3: Area reduction of the proposed FO function.

Method	FTN	LUTs					Area	% reduction w.r.t.	
		2-1	3-1	4-1	5-1	6-1		[10]	[12]
Prop.	S9-1 ~ S9-7	—	1	43	19	—	114	53.3	41.2
	S7-1 ~ S7-5	7	—	13	15	—			
	S9-8	—	—	9	—	—			
	S9-9	—	—	—	9	—			
	S9-10	—	—	—	2	7			
	S7-6	7	—	—	—	—			
	S7-7	—	7	—	—	—			
	KO _{i1}	16	—	—	—	—			
	KO _{i2}	16	—	—	—	—			
	KO _{i3}	16	—	—	—	—			
	KO _{i4} and I _{5L}	—	16	—	—	—			
	FI ₂ XOR	16	—	—	—	—			
	FI ₁	14	1	56	43	—			
	FI ₂	7	1	65	34	—			
FI ₃	7	8	56	36	7				
	{FO + XOR}	92	26	177	113	7			
[19]	FO			—			244		
[21]	FO			—			194	—	

TABLE 4: LUT area for MISTY1 architectures.

Function	LUTs					Remarks
	2-1	3-1	4-1	5-1	6-1	
FO	736	208	1416	904	56	8 × FO
FL	—	160	160	—	—	10 × FL
Key gen	112	8	440	352	—	8 × FI
Total	848	376	2016	1256	56	$n = 8$ -round MISTY1

required ciphertext of 64 bits per clock cycle. Thus, high-speed MISTY1 is obtained with DET pipelines and highly optimized FO/FI function implementations.

$$T_{\text{Path Delay}} \frac{T}{2} = T_{\text{C2Q(LUT)}} + T_{\text{P(LUT)}} + T_{\text{Setup(LUT)}}. \quad (1)$$

3.2. Architecture 2: MISTY1 SET Pipeline Architecture for Very High-Speed MISTY1. Very high-speed MISTY1 and its respective FO and FI functions (FI₁ and FI₃ functions are presented here for reference) employing single-edge-triggered pipelines are depicted in Figures 8 and 9.

It is evident that the FI₁ function requires 4 clock cycles, whereas the corresponding FO function is executed in 9 clock

FIGURE 6: DET pipeline architecture for high-speed MISTY1.

FIGURE 7: (a) FI₂ function with double-edge triggering accessing 2 × successive LUTs in 1 × clock cycle. (b) FO function with double-edge triggering.

cycles. The pipeline registers are inserted in the FO function as well as MISTY1 architecture to synchronize LSB and MSB bits. The path delay of the SET-based pipelined architecture is 1 × LUT, and therefore, the architecture achieves very high speed. By increasing the pipeline stages, the latency, i.e., the initial ciphertext generation, increases and is found as 77 clock cycles. The proposed architecture is highly suitable for high-speed applications of the order of 40 Gbps.

4. Hardware Implementation Results and Comparison

The proposed MISTY1 high-speed architectures are implemented on FPGA Xilinx Virtex-7, XC7VX690T. The performance comparison/analysis is carried out with existing high-speed Camellia, AES, and MISTY1 architectures. Table 5 depicts the performance parameters, i.e., throughput, area, and efficiency, of the proposed and existing design schemes.

The proposed MISTY1 architectures outperform all previous MISTY1 implementations indicating high speed with low area achieving high efficiency value. The throughput values obtained are 43/25.2 Gbps with a high efficiency of 28.5/18.9 Mbps/slices for very high-speed/high-speed MISTY1 architectures, respectively. For a fair comparison, the referred MISTY1 architectures [20, 22] are implemented using the same FPGA device, i.e., Xilinx Virtex-7. The architectures thus represent highly efficient and high-speed MISTY1 implementations to date. Besides, the proposed architectures have higher efficiency values

compared to the existing AES and Camellia architectures (as per our study). This signifies the optimizations made for proposed high-speed MISTY1 architectures.

5. Conclusion

In this paper, we proposed MISTY1 8-round pipelined architectures characterizing high-speed and efficient implementations. The structural optimizations and logic modifications in MISTY1 transformation functions readily reduced the LUTs and pipeline requirements. The proposed high-speed MISTY1 architectures using the SET and DET pipeline explore the speed/area tradeoffs for FPGA implementations. The design/optimization schemes can be extended for the high-speed implementation of the KASUMI algorithm. The high-speed designs have applications in wireless sensor networks, image encryption, and network controllers.

5.1. Future Work. This paper deals only with a high-speed MISTY1 block cipher. In the future, we shall make an energy-efficient MISTY1 block cipher using capacitance scaling, clock gating, clock enable, thermal scaling, voltage scaling, and other energy-efficient techniques. In the future, we shall check the thermal stability of MISTY1. The implementation of the MISTY1 block cipher is on 28 nm technology-based Virtex-7 FPGA in this paper. There is an open scope to reimplement this MISTY1 block cipher design on both 20 nm technology-based Ultrascale Virtex FPGA and 16 nm technology-based Ultrascale Plus Virtex FPGA.

FIGURE 8: SET pipeline architecture for very high-speed MISTY1.

FIGURE 9: (a) FI₁ function, (b) FI₃ function, and (c) FO function with single-edge-triggered pipeline implementation for very high-speed MISTY1.

TABLE 5: FPGA implementation and comparison.

Ref.	Algorithm	Area (slices)	Speed (Gbps)	Freq (MHz)	Eff. (Mbps/slices)
[24]	AES	35,328	260	508	7.36
[25]	AES	4339	75.92	593	17.50
[23]	Camellia	2805	28.4	221.6	10.12
[20]*	MISTY1	1865	0.56	79	0.3
[20]*	MISTY1	4732	7.2	96	1.52
[20]**	MISTY1	2920	21.9	342	7.5
[20]*	MISTY1	4039	12.6	168	3.12
[20]**	MISTY1	2920	21.9	342	7.5
[21]*	MISTY1	1265	16.3	254.5	12.9
[22]*	MISTY1	6322	10.18	159	1.61
[22]**	MISTY1	2506	38.9	607.5	15.5
[22]*	MISTY1	6322	19.4	303	3.07
[22]**	MISTY1	2506	38.9	607.5	15.5
Ours	MISTY1	1509	43	673.8	28.5
	MISTY1	1331	25.2	393.7	18.9

*Results published in papers cited. **Results obtained by authors with the implementation on the same FPGA.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] T. Kumar, B. Pandey, T. Das, and B. S. Chowdhry, "Mobile DDR IO standard based high performance energy efficient portable ALU design on FPGA," *Wireless Personal Communications*, vol. 76, no. 3, pp. 569–578, 2014.
- [2] B. Pandey, "Energy efficient design and implementation of ALU on 40nm FPGA," in *Proceedings of the International Conference on Energy Efficient Technologies for Sustainability*, IEEE, Nagercoil, India, April 2013.
- [3] B. Pandey, "Clock gating based energy efficient ALU design and implementation on FPGA," in *Proceedings of the International Conference on Energy Efficient Technologies for Sustainability*, IEEE, Nagercoil, India, April 2013.
- [4] B. Pandey, "FSM based green memory design and its implementation on ultrascale plus FPGA," *Journal of Critical Reviews*, vol. 7, pp. 454–458, 2020.
- [5] R. Sharma, B. Pandey, V. Jha, S. Saurabh, and S. Dabas, *Input-output standard-based energy efficient UART design on 90 nm FPGA System and Architecture*, Springer, Singapore, pp. 139–150, 2018.
- [6] I. Kaur, L. Rohilla, A. Nagpal, B. Pandey, and S. Sharma, *Different configuration of low-power memory design using capacitance scaling on 28-nm field-programmable gate array System and Architecture*, Springer, Singapore, pp. 151–161, 2018.
- [7] V. Thind, S. Pandey, D. M. Akbar Hussain, B. Das, M. F. L. Abdullah, and B. Pandey, *Timing constraints-based high-performance DES design and implementation on 28-nm FPGA System and Architecture*, Springer, Singapore, pp. 123–137, 2018.
- [8] S. H. A. Musavi, B. S. Chowdhry, T. Kumar, B. Pandey, and W. Kumar, "IoT's enable active contour modeling based energy efficient and thermal aware object tracking on FPGA," *Wireless Personal Communications*, vol. 85, no. 2, pp. 529–543, 2015.
- [9] E. Aerabi, M. Bohlouli, M. H. A. Livany, M. Fazeli, A. Papadimitriou, and D. Hely, "Design space exploration for ultra-low-energy and secure IoT MCUs," *ACM Transactions on Embedded Computing Systems*, vol. 19, no. 3, pp. 1–34, 2020.
- [10] J. Yang and T. Johansson, "An overview of cryptographic primitives for possible use in 5G and beyond," *Science China Information Sciences*, vol. 63, pp. 1–22, 2020.
- [11] M. Matsui, "New block encryption algorithm MISTY," *Fast Software Encryption*, vol. 1267, pp. 54–68, 1997.
- [12] A. Yasir, N. Wu, and X. Zhang, "Compact hardware implementations of MISTY1 block cipher," *Journal of Circuits, Systems and Computers*, vol. 27, no. 3, Article ID 1850037, 2017.
- [13] D. Yamamoto, J. Yajima, and K. Itoh, "Compact architecture for ASIC implementation of the MISTY1 block cipher," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E93-A, no. 1, pp. 3–12, 2010.
- [14] Yasir, N. Wu, X. Q. Zhang, and M. R. Yahya, "Highly optimised reconfigurable hardware architecture of 64 bit block ciphers MISTY1 and KASUMI," *Electronics Letters*, vol. 53, no. 1, pp. 10–12, 2017.
- [15] AbdoulRjoub, "Low power/high speed optimization approaches of MISTY algorithm," in *Proceedings of the 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, UAE, Silchar, India, July 2016.
- [16] S. Mathew, S. Satpathy, V. Suresh et al., "340 mV-1.1 V, 289 Gbps/W, 2090-gate NanoAES hardware accelerator with area-optimized encrypt/decrypt GF(2⁴) 2 polynomials in 22 nm tri-gate CMOS," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 4, pp. 1048–1058, 2015.
- [17] A. Yasir, N. Wu, X. Chen, and M. Rehan Yahya, "Area-efficient hardware architectures of MISTY1 block cipher," *Radio-engineering*, vol. 27, no. 2, pp. 541–548, 2018.
- [18] N. W. Yasir, A. A. Zain, M. Mujtaba Shaikh, M. RehanYahya, and M. Aamir, "Compact and high speed architectures of KASUMI block cipher," *Wireless Personal Communication*, vol. 106, no. 4, pp. 1787–1800, 2018.
- [19] Yasir, fnm Ning Wu, and A. A. Siddiqui, "Performance Comparison of KASUMI and hardware architecture optimization of f8 and f9 algorithms for 3g UMTS Networks," in

- Proceedings of the 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 420–424, Islamabad, Pakistan, January 2017.
- [20] P. Kitsos, M. D. Galanis, and O. Koufopavlou, “Architectures and fpga implementations of the 64-bit Misty1 block cipher,” *Journal of Circuits, Systems and Computers*, vol. 15, no. 6, pp. 817–831, 2006.
 - [21] Yasir, N. Wu, X. Chen, M. R. Yahya, and X. Zhang, “FPGA based highly efficient MISTY1 architecture,” *IEICE Electronics Express*, vol. 14, no. 18, Article ID 20170841, 2017.
 - [22] G. Rouvroy, F.-X. Standaert, J.-J. Quisquater, and J.-D. Legat, “Efficient FPGA implementation of block cipher MISTY1,” in *Proceedings of the International Parallel and Distributed Processing Symposium*, Nice, France, April 2003.
 - [23] A. F. Martínez-Herrera, C. Mancillas-López, and C. Mex-Perera, “GCM implementations of Camellia-128 and SMS4 by optimizing the polynomial multiplier,” *Microprocessors and Microsystems*, vol. 45, pp. 129–140, 2016.
 - [24] AbolfazlSoltani, “An ultra-high throughput and fully pipelined implementation of AES algorithm on FPGA,” *Microprocessors and Microsystems*, vol. 39, p. 7, 2015.
 - [25] Q. Liu, Z. Xu, and Y. Yuan, “High throughput and secure advanced encryption standard on field programmable gate array with fine pipelining and enhanced key expansion,” *IET Computers & Digital Techniques*, vol. 9, no. 3, pp. 175–184, 2015.
 - [26] M. Madani and C. Tanougast, “FPGA implementation of an enhanced chaotic-KASUMI block cipher,” *Microprocessors and Microsystems*, vol. 80, Article ID 103644, 2021.
 - [27] R. Muthalagu and S. Jain, “Improved KASUMI block cipher for GSM-based mobile networks,” *Journal of Cyber Security Technology*, vol. 4, no. 4, pp. 197–210, 2020.