Area-Efficient Hardware Architectures of MISTY1 Block Cipher

YASIR, Ning WU, Xin CHEN, Muhammad Rehan YAHYA

College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, 29 Yudao Street, Nanjing 210016, China

khizaryasir@nuaa.edu.cn, wunee@nuaa.edu.cn, xin_chen@nuaa.edu.cn, rehanyahya@nuaa.edu.cn

Submitted September 19, 2017 / Accepted December 23, 2017

Abstract. *In this paper, state-of-the-art hardware implementations of MISTY1 block cipher are presented for area-constrained wireless applications. The proposed MISTY1 architectures are characterized of highly optimized transformation functions i.e. FL and {FO-XOR-EKG}. The FL function re-utilizes logic AND-OR-XOR combinations whereas {FO-XOR-EKG} function explores 2 × compact design schemes for s-boxes implementation. A Combined Substitution Unit (CSU) and threshold area implementation are proposed for s-boxes based on Boolean reductions and Common Sub-expression Eliminations (CSEs). Besides, {FO-XOR-EKG} function is designed for manifold operations of FO / FI functions, 32-bit XOR operation and extended key generation thereby reducing the area. Hardware implementations on ASIC 180nm, 1.8V standard library cell realized compact and threshold MISTY1 designs constituting 1853 and 1546 NAND gates with throughput values of 41.6 Mbps and 4.72 Mbps respectively. A comprehensive comparison with existing cryptographic hardware designs establishes that the proposed MISTY1 architectures are the most area-efficient implementations till date.*

Keywords

MISTY1, Application Specific Integrated Circuit (ASIC), wireless communications, S-box, Common Sub-expression Elimination (CSE)

1. Introduction

With the wide-spread usage of wireless devices and embedded systems, cryptographic hardware circuits are proving as the critical component of modern day System-On-Chips (SOCs) laying the foundations for network security. However, the provision of security features in communication networks is materialized in the form of performance degradation thereby increasing the area or reducing the throughput. Considerable efforts are underway to optimize the hardware design / implementation of encryption algorithms for respective applications. MISTY1 block cipher characterizing repeated-loop structure is highly suitable for area-constrained applications. Taking this into account, a study has been carried out on compact MISTY1 implementations for mobile applications, hand-held devices and RFIDs.

Developed by Mitsubishi Electric Corporation, MISTY1 is an ISO / IEC standardized 64-bit block cipher algorithm designed to process smaller blocks of data e.g. PIN of 8-byte [1]. It has a proven probability parametric value of 2^-56 against "Linear and Differential Cryptanalysis" [2]. Many attacks have been proposed by researchers to break MISTY1 block cipher. The attacks though exposed several weaknesses of MISTY1; they could not compromise the full security for 8-rounds MISTY1 [3], [4]. Moreover, the complexities subject to time-domain and acquisition of large plain-text data for retrieval of the secret key made it practically impossible to undermine the security of MISTY1 block cipher. Therefore, MISTY1 is considered as a secure algorithm and is currently being employed for online-transactions and ATM networks.

A detailed study has been carried out on the existing hardware designs of MISTY1, KASUMI, AES, SHA-1, CAMELLIA and SAFER [5–22]. The review covered the comparison of performance parameters i.e. area, speed and frequency. For high speed architectures, the commonly employed schemes include Look-Up Tables (LUTs) or Block RAMs (BRAMs) implementation for s-boxes [10], [11], [19], [20]. In addition, optimizations are made for path delay reduction by effective pipe-lines implementation [11], [19]. As a result, very high throughput value of the order of 100 Gbps is obtained and is employed for image encryption, Ethernet devices, sensor networks, etc. [19]. Though high speed architectures have wide range of applications in sensor networks, the area they constitute make the devices power-hungry. Thus, high speed designs are not suitable for portable devices and mobile equipments.

In comparison to high throughput encryption cores, compact designs make use of the logic optimization techniques for transformation functions and s-boxes using combinational logic [5–10], [12–18], [21], [22]. Besides, re-utilization methodologies have also been implemented exploiting the rolling-feature of the architecture. The analysis carried out during this work implies that the most area-efficient cryptographic hardware circuit constitutes 1947

NAND gates for AES algorithm [15]. Moreover, for low area MISTY1, a very compact hardware architecture has been realized in [5] consisting of 2331 NAND gates. The silicon area though is very less, we discovered that architectural and logic optimizations can be carried out on MISTY1 algorithm for very compact hardware implementations. The major contributions / salient features of our work are as under:

a. Optimization of S9 and S7 s-boxes.
   i. Design and implementation of combined substitution unit for S9 and S7 for compact MISTY1.
   ii. Threshold area implementation of S9 / S7 s-boxes for very low area MISTY1.

b. Optimization of MISTY1 transformation functions.
   i. FL function implementation for 2 and 4 × clock cycles execution by logic re-utilization.
   ii. Designing of a new and novel function {FO-XOR-EKG} to encompass round transformation operations and extended key generation function.

c. Design / configuration of area-efficient MISTY1 by employing highly optimized transformation functions FL and {FO-XOR-EKG}.

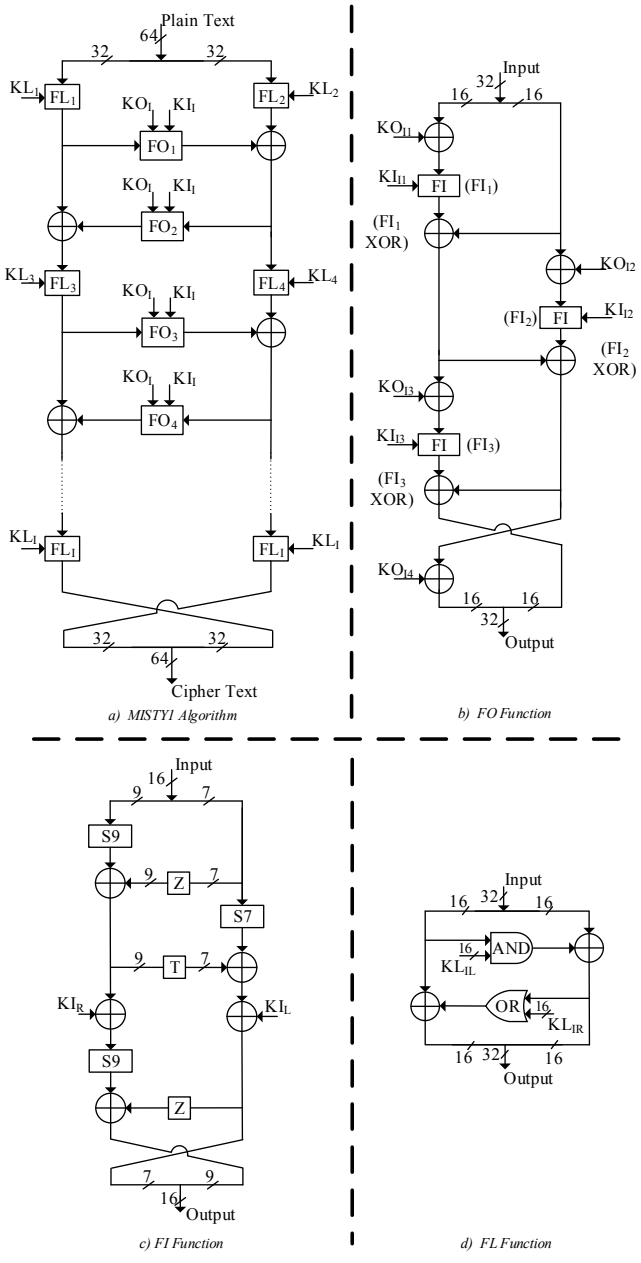

Fig. 1. MISTY1 algorithm and its transformation functions.

The paper is organized as follows. MISTY1 algorithm is briefly described in Sec. 2 followed by S9 / S7 s-boxes optimization in Sec. 3. The optimized MISTY1 transformation functions are explained in Sec. 4. A detailed explanation of proposed MISTY1 hardware architectures is presented in Sec. 5. Lastly, ASIC results and conclusion are summarized in Sec. 6 and 7 respectively.

2. 64-bit MISTY1 Block Cipher

MISTY1 algorithm is depicted in Fig. 1a and its constituent functions FO, FI and FL are shown in Figs. 1b–1d respectively. The algorithm transforms 64-bit plain-text to 64-bit cipher text using 128-bit secret key after n rounds operation. The specifications described in [1] recommend the value for the number of rounds as $n \geq 8$. Moreover, for n-rounds operation, the algorithm requires the generation of 128-bit extended key by using its FI function. MISTY1 odd rounds consist of 2 × FL functions, FO function and 32-bit XOR whereas the even rounds comprise of only FO function and 32-bit XOR. Furthermore, the last round is an exception consisting of only 2 × FL functions. The outputs of FL functions are finally concatenated to form 64-bit cipher text. The FO and FI functions have a feistel-like structure with FI consisting of substitution functions S9 and S7. The substitution functions S9 and S7 substitute the respective 9-bit and 7-bit inputs to 9-bit and 7-bit outputs by logic operations or LUTs [1].

3. S9 and S7 S-boxes Optimization

3.1 Scheme 1 – S9 / S7 Combined Substitution Unit (CSU)

In this scheme, a Combined Substitution Unit (CSU) is proposed for S9 and S7 s-boxes to substitute 9-bit and 7-bit inputs to 9-bit and 7-bit outputs respectively on alternate clock cycles. The first step involved in CSU design is the algebraic reductions of S9 / S7 logic expressions. Therefore, XOR gates are replaced by NOT gates (for both S9 and S7 logic expressions) and 3-input AND gates are reduced to maximize the use of 2-input AND gates (for S7 logic expressions only). The Common Sub-expression Elimination (CSE) is then carried out from combined S9 / S7 logic expressions thus eliminating the redundant 2-input ANDs and AND-XORs sub-expressions. The reduced algebraic expressions of S9 and S7 are expressed in Tab. 1 whereas common sub-expressions are shown in Tab. 2.

Reduced S9 Algebraic Expressions

$y_9 = x_8x_3 \oplus x_7x_2 \oplus (x_9x_4)' \oplus C_1 \oplus C_2 \oplus C_3$
$y_8 = x_9x_7 \oplus x_7x_6 \oplus x_5x_4 \oplus x_4x_1 \oplus x_2 \oplus C_1 \oplus C_4 \oplus C_5 \oplus C_6$
$y_7 = C_2 \oplus C_4 \oplus C_6x_5 \oplus C_7 \oplus C_8 \oplus C_9$
$y_6 = x_9 \oplus x_8x_7 \oplus C_3 \oplus C_6x_4 \oplus C_8 \oplus C_9 \oplus C_{10}$
$y_5 = x_8 \oplus x_7x_6 \oplus x_6x_4 \oplus x_7x_3 \oplus C_{10} \oplus C_{11}x_3 \oplus C_{12} \oplus C_{13}$
$y_4 = x_7 \oplus x_5x_3 \oplus C_5 \oplus C_6x_2 \oplus C_{13} \oplus C_{14} \oplus C_{15}$
$y_3 = x_7x_4 \oplus x_5x_4 \oplus x_7x_2 \oplus x_5x_1 \oplus x_2x_1 \oplus C_6 \oplus C_7 \oplus C_{14} \oplus C_{16}$
$y_2 = x_9x_5 \oplus x_8x_1 \oplus x_4 \oplus x_8' \oplus C_{15} \oplus C_{17} \oplus C_{18} \oplus C_{19}$
$y_1 = x_9 \oplus C_{11} \oplus C_{12} \oplus C_{16} \oplus C_{17} \oplus C_{18} \oplus C_{20}$

Reduced S7 Algebraic Expressions

$y_7 = x_3x_2 \oplus (x_6x_4)' \oplus x_7C_{21} \oplus x_2C_{22} \oplus C_{23} \oplus C_{24}$
$y_6 = x_4x_3 \oplus (x_7x_3)' \oplus C_{22} \oplus C_{24} \oplus x_1(C_{25} \oplus x_6x_3) \oplus x_5C_{26}$
$y_5 = x_6x_5 \oplus x_7(x_5x_4 \oplus x_6x_3 \oplus x_3x_2) \oplus x_4x_1 \oplus x_4C_{10} \oplus$
$x_3(C_6 \oplus x_1C_{11}) \oplus C_{27}$
$y_4 = x_2x_1 \oplus C_6 \oplus C_{23} \oplus x_6C_{26} \oplus x_3C_{28} \oplus C_{29}$
$y_3 = x_6(x_4x_3 \oplus x_5x_2) \oplus C_1 \oplus x_2(x_3x_1 \oplus C_{11} \oplus C_{20}) \oplus C_{30}$
$y_2 = x_6x_3 \oplus x_7(x_5x_3)' \oplus C_6(x_7x_2) \oplus C_{28} \oplus C_{29} \oplus x_6C_{30} \oplus C_{31}$
$y_1 = x_3x_1 \oplus x_4(x_7x_1 \oplus x_6x_2) \oplus C_{19} \oplus x_4C_{25} \oplus x_5C_{21} \oplus C_{27} \oplus C_{31}$

Tab. 1. Reduced algebraic expressions for S9 and S7.

CSEs	S9	S7
$C_1 = x_7x_3 \oplus x_6x_1$	y_9, y_8	y_3
$C_2 = x_6x_2 \oplus x_9x_5$	y_9, y_7	-
$C_3 = x_8x_4 \oplus x_5x_1$	y_9, y_6	-
$C_4 = x_8x_6 \oplus x_9x_3$	y_8, y_7	-
$C_5 = x_6x_5 \oplus x_9x_1$	y_8, y_4	-
$C_6 = x_6'$	y_8, y_7, y_6, y_4, y_3	y_5, y_4, y_2
$C_7 = x_9x_8 \oplus x_1$	y_7, y_3	-
$C_8 = x_7x_5 \oplus x_5x_4$	y_7, y_6	-
$C_9 = x_4x_3 \oplus x_8x_2$	y_7, y_6	-
$C_{10} = x_3x_2 \oplus x_7x_1$	y_6, y_5	y_5
$C_{11} = x_5'$	y_5, y_1	y_3, y_5
$C_{12} = x_9x_4 \oplus x_4x_3$	y_5, y_1	-
$C_{13} = x_9x_6 \oplus x_2x_1$	y_5, y_4	-
$C_{14} = x_8x_5 \oplus x_4x_2$	y_4, y_3	-
$C_{15} = x_8x_3 \oplus x_3x_2$	y_4, y_2	-
$C_{16} = x_9x_1 \oplus x_3x_1$	y_3, y_1	-
$C_{17} = x_9x_8 \oplus x_8x_7$	y_2, y_1	-
$C_{18} = x_6x_3 \oplus x_9x_2$	y_2, y_1	-
$C_{19} = x_7x_6 \oplus x_5x_2$	y_2	y_1
$C_{20} = x_7x_4 \oplus x_6x_1$	y_1	y_3
$C_{21} = x_4x_3 \oplus x_6x_1$	-	y_1, y_7
$C_{22} = x_4x_1 \oplus x_7x_5$	-	y_7, y_6
$C_{23} = x_7 \oplus x_5x_1$	-	y_7, y_4
$C_{24} = x_6x_2 \oplus x_7x_2x_1$	-	y_7, y_6
$C_{25} = x_7'$	-	y_6, y_1
$C_{26} = x_3x_2 \oplus x_4x_1$	-	y_6, y_4
$C_{27} = x_7x_2 \oplus x_6x_1$	-	y_5, y_1
$C_{28} = x_5 \oplus x_7x_1$	-	y_4, y_2
$C_{29} = x_7x_6x_5 \oplus x_7x_4$	-	y_4, y_2
$C_{30} = (x_5x_4)'$	-	y_3, y_2
$C_{31} = x_4x_2 \oplus x_5x_2x_1$	-	y_2, y_1

Tab. 2. CSEs for S9 and S7.

The AND gates of S9 and S7 reduced logic expressions are shown by respective bits for simplicity; however the implementation is carried out by permuting 9-bits to form 36 × combinations. The path delay of CSU using parallel AND-XORs hierarchy is expressed as (1) whereas the area reduction as compared to straight-forward s-boxes {2 × S9 + S7} is found as 60.8% illustrated in Tab. 3.

$$T_{\text{Path Delay}} = (2\text{-}1)\ \text{MUX} + 2 \times \text{AND} + 4 \times \text{XOR} \quad (1)$$

Method	Ftn	AND	XOR	NOT	MUX	Gates*	% Reduction
Prop.	CSU	58	128	9	9	438	60.8
{2 × S9 + S7}	S9	89	101	-	-	1120	-
	S7	104	77	-	-		
	S9	89	101	-	-		

Tab. 3. Area reduction of CSU *(AND = OR = 1.5 NAND, XOR = MUX = 2.5 NAND, Reg = 6 NAND, NOT = 1 NAND).

3.2 Scheme 2 – S9 / S7 Threshold Area Implementation

S9 / S7 threshold area implementation is depicted in Figs. 2 and 3 consisting of MUXes, AND, XORs and 1-bit high enabled registers. The proposed design scheme sets a threshold limit for area of S9 and S7 s-boxes to generate a throughput value $\geq$ 4 Mbps.

The substituted bits for S9 and S7 are produced after 45 and 58 clock cycles respectively and are based on the maximum possibilities of S9 and S7 logic sub-expressions. For instance, S9 s-box has 36 × combinations for AND

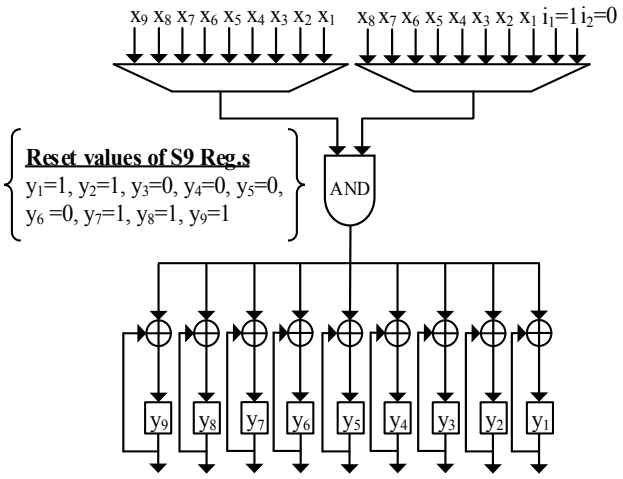

Fig. 2. Threshold area S9 s-box.

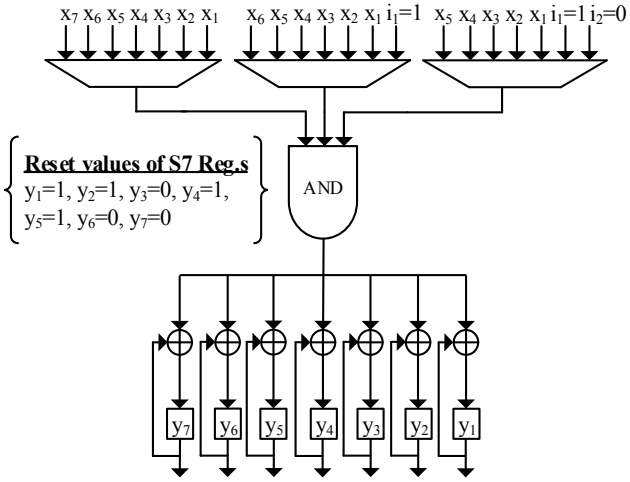

Fig. 3. Threshold area S7 s-box.

Method	Ftn	AND	XOR	MUX	Reg	Gates	% Reduction
Prop {S9 + S7}	S9	1	9	17	9	231	81
	S7	2	7	19	7		
{2 × S9 + S7}	S9	89	101	-	9	1216	-
	S7	104	77	-	7		
	S9	89	101	-	-		

Tab. 4. % area reduction of CSU based s-box.

gates and 9 × combinations for respective 9-bits. The additional input bit 1 in multiplexers is used to generate all the possible input values for S9 and S7 substitution functions whereas input bit 0 reproduces the registers output for certain clock cycle operations of {FO-XOR-EKG} function. The reset register values mentioned in Figs. 2 and 3 are based on the respective logic expression of S9 / S7 and reduce the clock cycle operations. Table 4 summarizes the percentage area reduction of 81% with the proposed design as compared to MISTY1 FI function with {2 × S9 + S7} s-boxes.

4. Optimized MISTY1 Transformation Functions

4.1 FL Function Implementation

Figures 4 and 5 depict FL functions generating 32-bit output after 2 and 4 clock cycles respectively. The input to the proposed FL function is a 32-bit plain text or the output of {FO-XOR-EKG} function and the outputs are saved in enabled registers. The design provides a reference for area reduction of FL function and can be configured for 8 and 16 clock cycles operation. The area reduction for compact MISTY1 architectures is mainly due to use of 1 × FL function; however the area reduction with the proposed methodologies FL - 1 and FL - 2 is found as 4% and 6.1% respectively as compared to straight-forward FL function (ref. Fig. 1d). The area for 8 / 16 clock cycles FL function is also mentioned in Tab. 5; since the NAND gates difference w.r.t. proposed FL – 2 is insignificant, they are not implemented in this paper.

4.2 Novel Design of {FO-XOR-EKG} Function

{FO-XOR-EKG} function is the core part of the proposed MISTY1 hardware architecture. The re-utilization methodology has widely been adopted for the optimum operation of {FO-XOR-EKG} function. The intended idea behind the design of proposed {FO-XOR-EKG} function is to perform the transformation operations including FO / FI and 32-bit XOR operation (appended with FO in rounds 1-8). Moreover, the design can generate the extended keys for onward use in MISTY1 8-rounds operation. The accumulation of the above mentioned functionalities in a single function reduces the circuit area considerably. The area reduction is complemented with optimized implementation of S9 and S7 s-boxes within {FO-XOR-EKG} function. Thus, 2 × design schemes for {FO-XOR-EKG} function implementing CSU based s-box and S9 / S7 threshold area s-boxes are shown in Figs. 6 and 7 respectively.

Imp	Clock Cycles	AND	OR	XOR	MUX	Reg	Gates	% Red.
Prop. FL -1	2	8	8	16	48	64	568	4
Prop. FL - 2	4	4	4	8	56	64	556	6.1
FL – 3 (for ref.)	8	2	2	4	60	64	550	7.1
FL – 4 (for ref.)	16	1	1	2	62	64	547	7.6
Straight-forward	1	16	16	32	32	64	592	-

Tab. 5. % area reduction of FL-1 and FL-2 functions.

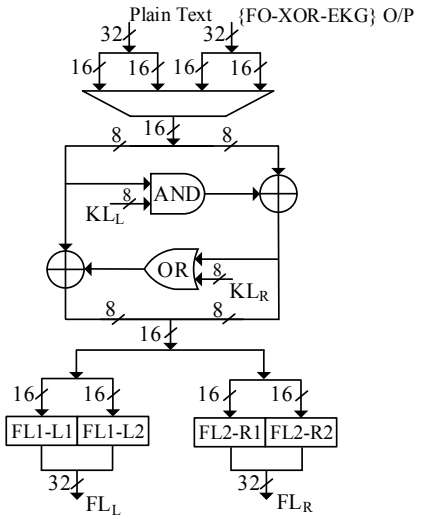

Fig. 4. FL-1 (2 clock cycles).

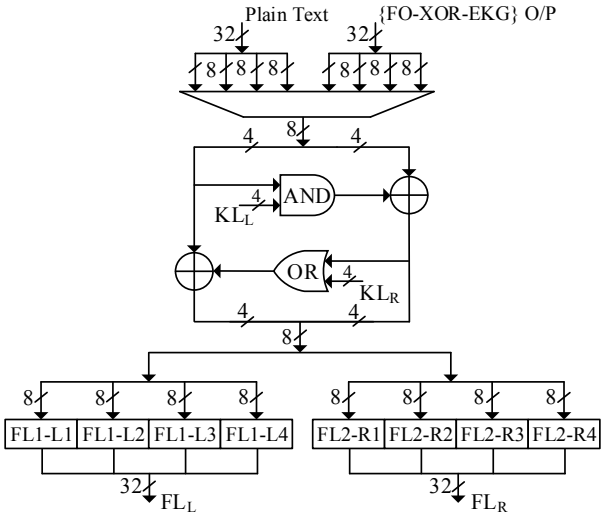

Fig. 5. FL-2 (4 clock cycles).

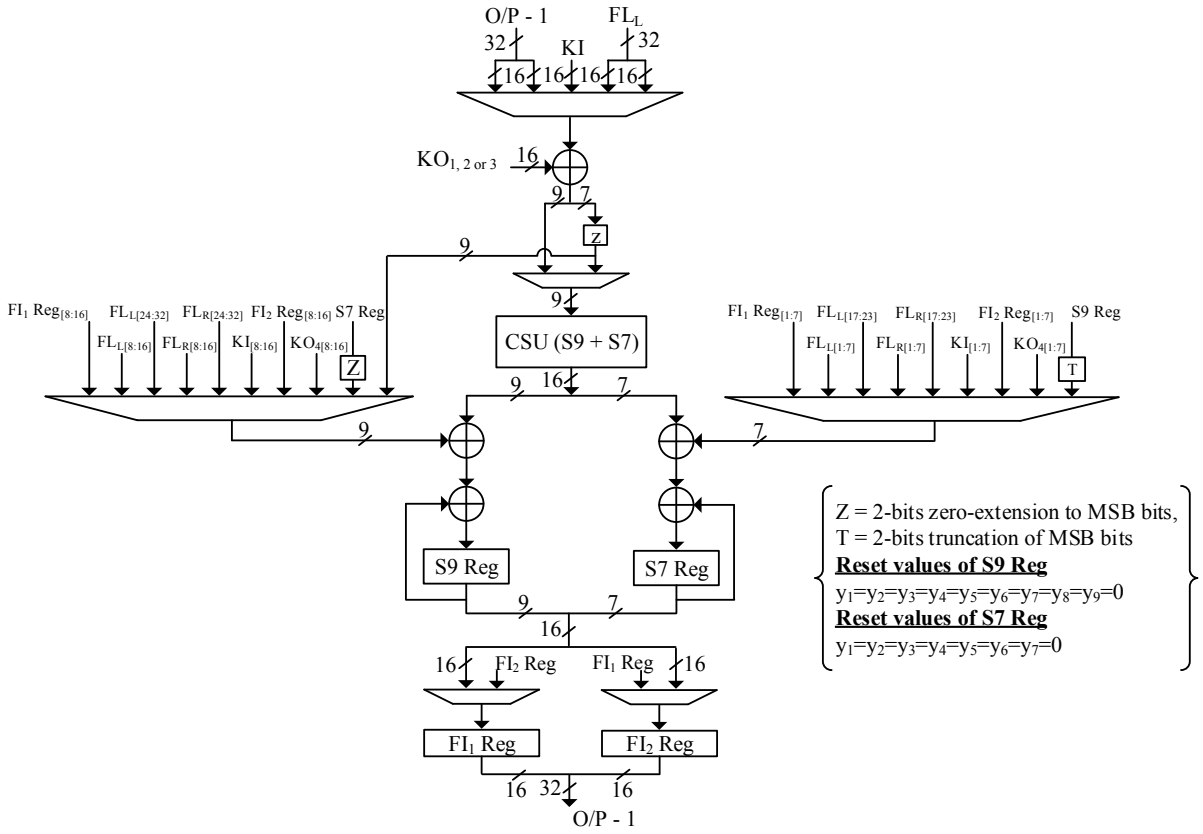

Fig. 6. FO – 1 function based on CSU based s-box.

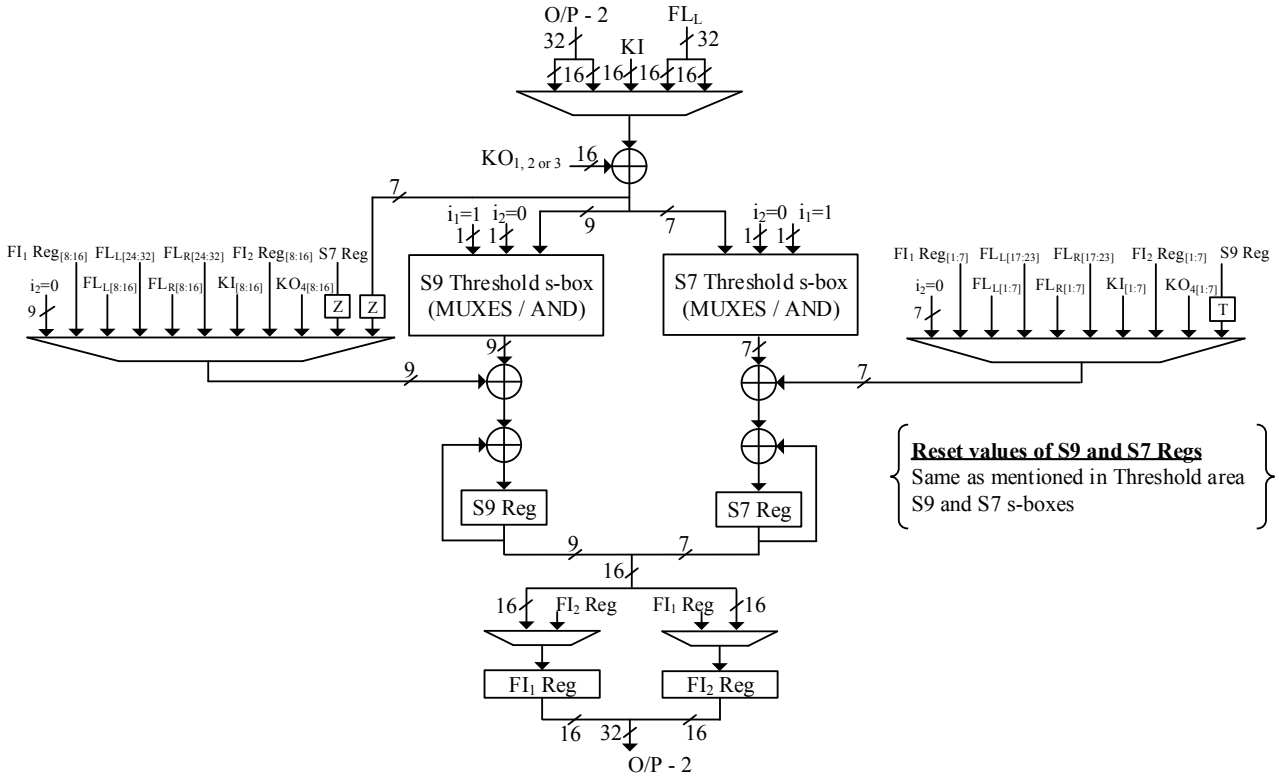

Fig. 7. FO – 2 function based on S9 / S7 Threshold Area s-box.

Functions	Steps	Operations
Extended key generation Step 7 is excluded for key generation and KO_i = 16-bit 0s	1	*S9 substitution, z-extended XOR operation (for Arch-1)*
	2	*S7 substitution, truncated XOR operation with S9 Reg (for Arch-1)*
	1	*S9 substitution, S7 substitution (for Arch-2)*
	2	*z-extended XOR operation for LSB bits, truncated XOR operation for MSB bits (for Arch-2)*
FO function for Round 1 FI_1 function, FI_1 XOR operation, KO_i = 16-bit SK	3	*KI_L XOR operation, KI_R XOR operation*
	4	*Reg values of S9 and S7 saved in FI_1 Reg*
	5	*Rst of S9 Reg*
	6	*S9 substitution, z-extended XOR operation*
	7	*S9 Reg ⊕ FL_L[24:32], S7 Reg ⊕ FL_L[17:23]*
	8	*S9 and S7 Regs values saved in FI_1 Reg*
	9	*Rst of S9 and S7 Regs*
FO function for Round 1 FI_2 function, FI_2 XOR operation, KO_4 XOR operation **32-bit XOR Op.** XOR operation with FL_R[1:16] for FO O/P [1:16]	-	***Repeat steps 1-6 with S9 and S7 Reg values saved in FI2 Reg***
	10	*S9 Reg ⊕ FI_1[8:16], S7 Reg ⊕ FI_1[1:7]*
	11	*S9 Reg ⊕ KO_4[8:16], S7 Reg ⊕ KO_4[1:7]*
	12	*S9 Reg ⊕ FL_R[8:16], S7 Reg ⊕ FL_R[1:7]*
	-	***Repeat steps 8-9 with S9 and S7 Reg values saved in FI_2 Reg***
FO function for Round 1 FI_3 function, FI_3 XOR operation, KO_4 XOR operation **32-bit XOR Op.** XOR operation with FL_R[1:16], XOR operation with FL_R[17:32] for FO O/P [17:32]	-	***Repeat steps 1-6 with S9 and S7 Reg values saved in FI_1 Reg***
	13	*S9 Reg ⊕ FI_2[8:16], S7 Reg ⊕ FI_2[1:7]*
	-	***Repeat steps 10-11 to nullify the effect of KO_4 and FL_R [1:16]***
	14	*S9 Reg ⊕ FL_R[24:32], S7 Reg ⊕ FL_R[17:23]*
	-	***Repeat steps 8-9 with S9 and S7 Reg values saved in FI_1 Reg***

Tab. 6. FO-1 and FO-2 algorithm.

The two architectures primarily have the same design basis but differ in terms of clock cycle operations. In order to incorporate all the functionalities, 2 × 9-bit XORs and 2 × 7-bit XORs are appended with optimized s-boxes with inputs being fed by multiplexers and registers. In addition, a 16-bit secret key is added in the input multiplexer and KO_i have a variable value of 16-bit SK or 0s. Table 6 describes the algorithm / steps involved for the execution of above mentioned functions.

The EK generation and FO function differs in the selection for input texts and KO_i XOR. For EK generation, the input and KO_i is assigned as 16-bit SK and 16-bit 0s respectively whereas the input and KO_i for FO is FL_L / FO O/P and SK respectively. Moreover, as compared to EK generation, the FO function has extended clock cycle operations carried out for 3 × FIs, 3 × XORs and 32-bit XOR (ref to Fig. 1b for FO function operations). A 32-bit XOR operation in FO is an exception such that 32-bit XOR is divided into 16-bit LSB and MSB XORs i.e. 16-bit LSB XOR is performed after {FO / FI_2 + KO_{i4}} whereas 16-bit MSB XOR is performed at the last step of FO / FI_3 execution. Since, KO_{i4} and 16-bit LSB XORs are performed before FO / FI_3 (as per algorithm these operations are to be carried out after FO / FI_3), therefore the inputs 16-bit LSB and KO_{i4} are re-XORed to MSB bits. Hence, 3 × functionalities are complemented in a single {FO-XOR-EKG} function with significant reduction in area. The percentage area reduction of FO-1 / FO-2 architectures compared with [7], [10] is depicted in Tab. 7.

Imp.	AND	NOT	XOR	MUX	Reg	Gates	% Red. w.r.t.	
							[7]	[10]
FO-1	58	9	176	242	48	1429	23.6	65.5
FO-2	3	-	48	303	48	1170	37.5	71.7
[7]	282	-	384	80	48	1871	-	-
[10]	845	-	1072	-	32	4141	-	-

Tab. 7. Area reduction of FO-1 and FO-2 functions.

5. Area-Efficient MISTY1 Hardware Architectures

The proposed hardware architecture of area-efficient MISTY1 8-rounds algorithm is depicted in Fig. 8.

The input to MISTY1 algorithm is a 64-bit plain-text (PT) and 128-bit secret key (SK) and the output is a 64-bit cipher-text. A 128-bit extended key (EK) is generated prior to MISTY1 8-rounds operation by {FO-XOR-EKG} function and is saved in an external 128-bit extended key register for onward round operations. The SK and EK in conjunction are later used for MISTY1 round transformation operations. The EK generation by {FO-XOR-EKG} function readily reduces the circuit area as it avoids the use of independent key generation module, i.e. FI function. However, the extended key generation by {FO-XOR-EKG} function reduces the throughput for MISTY1 8-rounds operation since EKs have to be generated in advance requiring multiple clock cycles. The speed i.e. throughput value (Mbps) of the proposed architectures can be calculated as:

$$\text{Throughput} = \text{Output (bits)} / \text{Clock Cycles (sec)}. \quad (2)$$

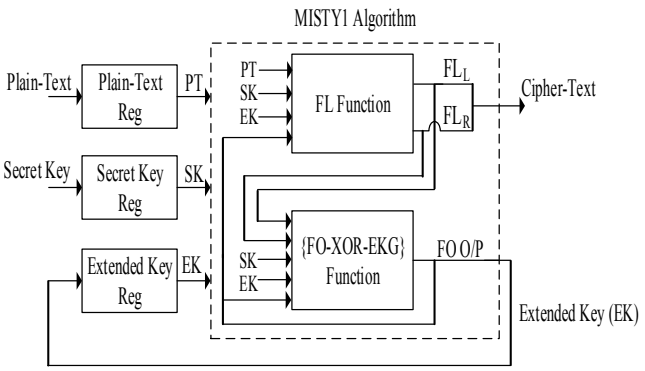

Fig. 8. Area-efficient MISTY1 hardware architecture.

6. Hardware Implementation of Proposed MISTY1 Architectures

Hardware implementation of the proposed MISTY1 architectures is performed on ASIC platform 180 nm, 1.8 V standard library cell using Synopsys Design Compiler and is optimized for area. A comprehensive analysis was carried out to obtain moderate speed MISTY1 - architecture 1 by integrating FO - 1 with FL - 1 whereas FO – 2 is configured with FL - 2 resulting into a threshold area MISTY1 – architecture 2. Table 8 summarizes the performance comparison in terms of area and throughput as follows.

Ref.	Algorithm	Technology (nm)	Area (Gates)	Throughput (Mbps)	Freq (MHz)
[14]	AES	90	3500	-	452.5
[15]	AES	22	1947	432	1133
[16]	AES	65	8100	119.2	149
[17]	AES	180	2421	0.06	0.1
[21]	CAMELLIA	130	4313	81	253
[22]	SHA-1	65	5469	-	625
[6]	KASUMI	180	3481	154	154.6
[12]	KASUMI	180	2990	110	97.6
[5]	MISTY1	180	3041	171	187
[5]	MISTY1	180	2331	166	187
[6]	MISTY1	180	3481	130	154.6
[7]	MISTY1	180	3400	70.3	39.1
[9]	MISTY1	180	3950	71.1	66.7
Prop	MISTY1 - Arch 1	180 nm	1853	41.6	210
Prop	MISTY1 - Arch 2	180 nm	1546	4.72	266.7

Tab. 8. ASIC implementation – results and analysis.

The proposed architectures outperform all previous implementations in terms of area depicting highly optimized MISTY1. The architecture 1 achieved throughput value of 41.6 Mbps with area of 1853 of gates depicting a compact and moderate speed MISTY1 implementation while architecture 2 sets a threshold value for both area and speed constituting area of 1546 gates with 4.72 Mbps. Compared with the lowest area of 1947 gates for AES and 2331 gates for MISTY1 implementations, the gate counts for the proposed MISTY1 architectures 1 and 2 is lesser. This represents MISTY1 architectures 1 and 2 as the most area-efficient block cipher implementations till date. The highly optimized MISTY1 architectures set a bench mark for low area implementation and can be employed for compact ASIC applications.

7. Conclusion

This paper proposes MISTY1 architectures for area-constrained embedded applications. The design / methodologies adopted for MISTY1 implementations by optimizing the transformation functions and s-boxes significantly reduced the hardware area. As per ASIC results / analysis with existing MISTY1 and other block cipher architectures like AES and KASUMI, the proposed MISTY1 implementations have the smallest gate-size to the best of our knowledge. The optimization techniques employed for MISTY1 can be adopted to 64-bit KASUMI block cipher and other encryption algorithms for low area implementation. The proposed architectures have high significance in cryptographic circuit design / optimization.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61774086, 61376025, 61404087), Natural Science Foundation of Jiangsu Province (No. BK20160806), Fundamental Research Funds for Central Univ. (No. NS2015045) and Aeronautical Science Foundation of China (No. 20152052025).

References

[1] MATSUI, M. New block encryption algorithm MISTY. *Lecture Notes in Computer Science*, 1997, vol. 1267, p. 54–68. DOI: 10.1007/BFb0052334

[2] MATSUI, M. New structure of block ciphers with provable security against differential and linear cryptanalysis. In *Proceedings of the 3rd International Workshop on Fast Software Encryption FSE*. Cambridge (UK), 1996, p. 205–218. ISBN: 3-540-60865-6

[3] DUNKELMAN, O., KELLER, N. Practical-time attacks against reduced variants of MISTY1. *Designs Codes and Cryptography,* 2015, vol. 76, no. 3, p. 601–627. DOI: 10.1007/s10623-014-9980-2

[4] TODO, Y. Integral cryptanalysis on full MISTY1. *Journal of Cryptology*, 2017, vol. 30, no. 3, p. 920–959. DOI: 10.1007/s00145-016-9240-x

[5] YASIR, WU, N., ZHANG, X. Compact hardware implementations of MISTY1 Block Cipher. *Journal of Circuit Systems and Computers*, 2017, vol. 27, no. 3. DOI: 10.1142/S0218126618500378

[6] YASIR, WU, N., ZHANG, X. Highly optimized reconfigurable hardware architecture of 64-bit block ciphers MISTY1 and KASUMI. *IET Electronics Letters*, 2017, vol. 53, no. 1, p. 10–11. DOI: 10.1049/el.2016.3982

[7] YAMAMOTO, D., YAJIMA, J., ITOH, K. Compact architecture for ASIC implementation of MISTY1 block cipher. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 2010, vol. E93-A, no. 1, p. 3–12. DOI: 10.1587/transfun.E93.A.3

[8] RJOUB, A., GHABASHNEH, E. M. Low power / high speed optimization approaches of MISTY algorithm. In *IEEE 5th International Conference on Electronic Devices, Systems and Applications*. United Arab Emirates, 2016. ISBN: 978-1-5090-5306-3

[9] YAMAMOTO, D., YAJIMA, J., ITOH, K. A very compact hardware implementation of MISTY1 block cipher. In *Proceedings of 10th International Conference on Cryptographic Hardware and Embedded Systems CHES 2008*. Washington (USA), 2008, p. 315–330. ISBN:978-3-540-85052-6

[10] KITSOS, P., GALANIS, M. D., KOUFOPAVLOU, O. Architectures and FPGA implementation of 64 bit MISTY1 block cipher. *Journal of Circuit Systems and Computers*, 2006, vol. 15, no. 6, p. 817–831. DOI: 10.1142/S0218126606003362

[11] YASIR, WU, N., CHEN, X., et al. FPGA based highly efficient MISTY1 architecture. *IEICE Electronics Express*, 2017, vol. 14, no. 18, p. 20170841. DOI: 10.1587/elex.14.20170841

[12] YAMAMOTO, D., ITOH, K., YAJIMA, J. Compact architecture for ASIC and FPGA implementation of KASUMI block cipher. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 2011, vol. E94-A, p. 2628–2638. DOI: org/10.1587/transfun.E94.A.2628

[13] YASIR, WU, N., SIDDIQUI, A. A. Performance comparison of KASUMI and H/W architecture optimization of f8 and f9 algorithms for 3g UMTS networks. In *Proceedings of 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. Islamabad (Pakistan), 2017, p. 420–424. DOI: 10.1109/IBCAST.2017.7868088

[14] VAN LAN DAO, ANH-THAI NGUYEN, VAN-PHUC HOANG. An ASIC implementation of low area AES encryption core for wireless networks. In *International Conference on Communications, Management and Telecommunications*. DaNang (Vietnam), 2015, p. 99–102. DOI: 10.1109/ComManTel.2015.7394268

[15] SANU, M., SATPATHY, S., SURESH, V. 340 mV–1.1 V 289Gbps/W, 2090-gate nano AES H/W accelerator with area-optimization encryption / decryption GF $(2^4)^2$ polynomial in 22nm tri-gate CMOS. *IEEE Journal of Solid-State Circuits*, 2015, vol. 50, no. 4, p. 1048–1058. DOI: 10.1109/JSSC.2014.2384039

[16] VAN-PHUC HOANG, THI-THANH-DUNG PHAN, VAN LAN DAO. A compact, ultra-low power AES-CCM IP core for wireless body area networks. In *2016 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*. Tallinn (Estonia), 2016, p. 1–4. DOI: 10.1109/VLSI-SoC.2016.7753566

[17] MORADI, A., POSCHMANN, A., LING, S., et al. Pushing the limits: A very compact and a threshold implementation of AES. In *30th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Tallinn (Estonia), 2011, p. 69–88. ISBN: 978-3-642-20464-7

[18] OMRAN, S. S., JUMMA, L. F. Design of SHA-1 & SHA-2 MIPS processor using FPGA. In *2017 Annual Conference on New Trends in Information and Communications Technology Applications (NTICT)*. Baghdad (Iraq), 2017, p. 268–273. DOI: 10.1109/NTICT.2017.7976113

[19] LIU, Q., XU, Z., YANG, Y. High throughput and secure AES on FPGA with fine pipelining and enhanced key expansion. *IET Computers & Digital Techniques*, 2015, vol. 9, no. 3, p. 175–184. DOI: 10.1049/iet-cdt.2014.0101

[20] MARTINEZ-HERRERA, A. F., MANCILLAS-LÓPEZ, C., MEX-PERERA, C. GCM implementation of Camellia-128 and SMS4 by optimizing the polynomial multiplier. *Microprocessors and Microsystems*, 2016, vol. 45, p. 129–140. DOI: 10.1016/j.micpro.2016.04.006

[21] KAVUN, E. B., YALCIN, T. A pipelined camellia architecture for compact hardware implementation. *In 2010 21st IEEE International Conference on Application-specific Systems Architectures and Processors (ASAP)*. Rennes (France), 2010, p. 305–308. DOI: 10.1109/ASAP.2010.5540987

[22] JUNG, E. G., HAN, D., LEE, J. G. Low area and high speed SHA-1 implementation. In *2011 International SoC Design Conference.* Jeju (South Korea), 2011, p. 365–367. DOI: 10.1109/ISOCC.2011.6138786

About the Authors ...

YASIR did his BE and MS from UET, Peshawar and Taxila, Pakistan. Currently, he is a PhD research scholar in the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), China. His research interests include circuit design and optimization, FPGA and ASIC implementations, cryptography and network security.

Ning WU received B.S. and M.S. degrees in 1982 and 1985 from the University of Science and Technology, China. She is currently a Professor in the College of Electronic and Information Engineering, NUAA. Her research interests include digital system design, electronic system integration and ASIC implementation.

Xin CHEN received his B.S. degree and Ph.D. degree in 2005 and 2010 from the Southeast University, China. He is currently an associate professor in the College of Electronic and Information Engineering, NUAA. His research interests mainly include digitally controlled phase-locked loop, low power circuit design.

Muhammad Rehan YAHYA did his BS and MS from COMSATS Institute of Information Technology, Islamabad and UET Taxila, Pakistan. Currently, he is a doctoral student in the College of Electronic and Information Engineering, NUAA. His research interests include FPGA and ASIC implementations, reconfigurable hardware design and Network-on Chip (NOC).