TED: A LIGHTWEIGHT BLOCK CIPHER FOR IoT DEVICES WITH SIDE-CHANNEL ATTACK RESISTANCE

Chandrama Thorat[1], *Vandana Inamdar*[1], *Bhagvat Jadhav*[2]

[1] College of Engineering Pune, Maharashtra
[2] Department of Electronics and Telecommunications, RSCOE, Pune
e-mails: chandrama1684@gmail.com ; vhj.comp@coep.ac.in ;
bdjadhav1979@gmail.com
India

Abstract: Lightweight cryptography has received significant attention in the field of pervasive and ubiquitous computing. The ultra-lightweight secure design proposed, which encrypts 64-bit size plaintext through 128-bit size key in 26 rounds. The proposed cipher TED is tested on both software and hardware platforms. TED cipher supports 5X reduced CPU cycles and 6X less memory footprint in comparison with the other state-of-the-art balanced Feistel ciphers. The rigorous security analysis shows that TED is capable of surviving against most of the prevalent security attacks.

Keywords: Lightweight cryptography, cryptanalysis, security, symmetric encryption

1. INTRODUCTION

The conventional security algorithms like AES or T-DES cannot apply directly to the Internet of Things (IoT) devices due to their resource-intensive computations. Lightweight cryptography mainly aimed at resource constraint devices. There are a diverse range of resource constraint devices, right from sensor nodes, RFID devices to IoT devices. The security requirements differ for IoT devices and the RFID devices as they both work in different environments. Many authors have come up with new ultra-low-power cipher designs which have a slightly low level of security. IoT devices are more prone to a higher number of attacks since they require more tight security than the RFID and sensor devices. They are also vulnerable to Side-Channel Attacks (SCAs) since they are more accessible to an attacker and are connected to the Public network. The objective of this work is to design a new secure and energy-efficient block cipher for the IoT devices titled Tiny Encryption Design (TED). The main aim of this work is to have

robustness against SCAs and other known attacks with marginal energy consumption on primarily software platform.

Any secure block cipher is designed and developed with a proper selection of linear and non-linear layers. In proposed cipher design along with 4×4 bit S-box, an additional non-linearity is provided with the modular addition operations. The proposed 4x4 bit S-box is tested against various cryptanalysis attacks and detailed results are presented in Section 3. The proposed cipher implementation is described in Section 2. Section 4 compares the security strength of the proposed cipher and other state-of-the-art cipher designs.

The results show that proposed cipher design TED performs better than an ISO standard cipher PRESENT and other Feistel-structure based cipher designs. For a software implementation, popular ARM processors are used, whereas hardware implementation results are obtained through ASIC based implementation with Cadence tool. Memory footprint is reduced by omitting the permutation table and by calculating the in-place bit-permutation positions. Hence, Section 5 gives the implementation results on both hardware and software platforms and memory efficiency of TED block cipher.

2. RELATED WORK

Over the last three decades, different lightweight cryptographic algorithms have been developed by researchers [1], which are based on substitution-permutation network (SPN) structure, Add-Rotate-Xor(ARX) structure, Feistel structure or hybrid structure [3]. Lightweight cryptographic algorithms include block ciphers, stream ciphers, hash functions and the recent one authenticated encryption techniques. Among these, block ciphers are more widely used than other techniques [2]. Lightweight block cipher designs are also adopted by ISO/OSI, where PRESENT and CLEFIA are standardized as ISO/OSI standard lightweight block cipher algorithms since 2012 to till date[17]. Though PRESENT cipher is adopted as an ISO/OSI standard block cipher, it has a weak substitution layer [4]. While a permutation layer of the PRESENT block cipher is considered as one of the best P-layers used.

Feistel structure is used in the development of a TED block cipher. In [7], the author has claimed a minimal memory space for the cipher developed GRANULE. For lightweight cipher design MANTRA [6], the author has claimed that it uses a compact memory footprint. A strong permutation layer is used to prevent the grouping of linear trails and differential trails in MANTRA block cipher.

In another lightweight block cipher FeW [8], the author aims to address the slow diffusion in the Feistel-network approach by changing the entire plaintext block in each round of encryption. QTL uses the same function for encryption and decryption [10]. It does not use any key scheduling algorithm. In [9], the author has proposed another cipher named Piccolo, which is developed with a simple structure and the absence of a key schedule makes it a tiny hardware footprint[9].

In this paper, an energy-efficient cipher design is proposed, which attains adequate security with fewer resources. The higher number of active S-boxes, strong in-place permutation layer, a less number of CPU cycles make the TED block cipher a suitable candidate for resource constraint devices.

3. TED BLOCK CIPHER

The proposed block cipher TED is an iterated Feistel-structure based cipher design. Figure 1 shows the insight of a single round of a TED block cipher.

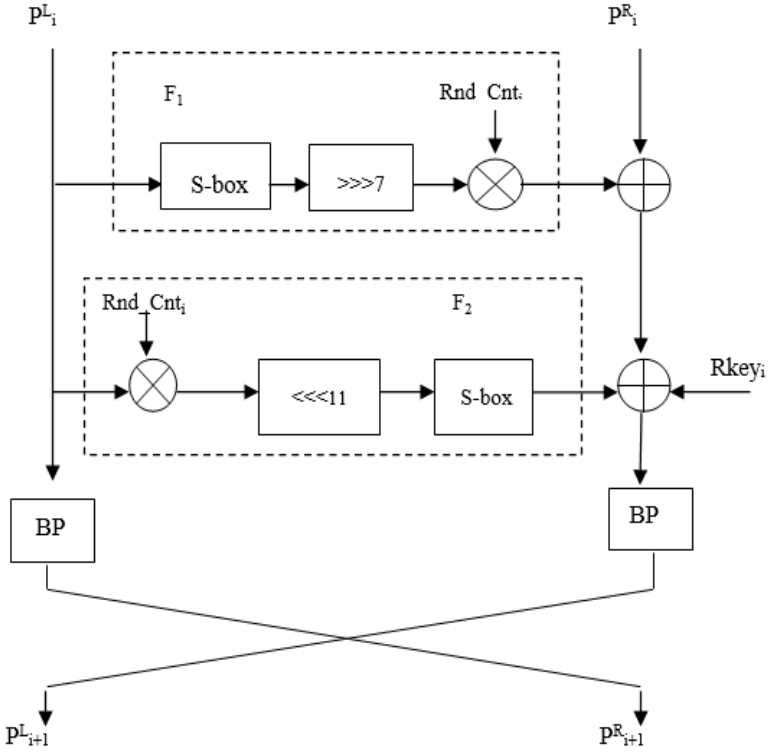

Fig. 1. Single Round of TED Block Cipher

Notations used in TED block cipher described as follows:

- PT_H : 64- plaintext block bits
- CT_H : 64-cipher text block bits
- RKeyi:128-bit Round sub-key used in each round *i*
- $\oplus$: XOR operation
- <<<m: Left cyclic rotation by m bits
- >>>m: Right cyclic rotation by m bits
- Rnd_cnti : Round counter i
- BP: Bit Permutation

- ⊞ : Modular addition in modulo 2^{32}

3.1. Encryption algorithm of TED cipher

The input 64-bit plaintext block PT_H is divided into two 32-bit sub-blocks, which are referred to as P^L_i and P^R_i, refer equation (1).

$$PT_H = P^L_i \,||\, P^R_i \quad (1)$$

Two different notations denote the output of functions F_1 and F_2 as F_X and F_Y, respectively. The output of function F_1 is XOR with P^R_i, and the output of function F_2 is XOR with round key and F_X.

The encryption algorithm described as below:

1. Apply function F_1 and F_2 to 32-bit plaintext halve P^L_i
   - $F_1 \leftarrow F(P^L_i)$
   - $F_2 \leftarrow F(P^L_i)$
   - $F_1 \neg [\text{S-Box}(P^L_i) >>> 7]] \oplus \text{Rnd\_Cnt}_i$
   - $F_2 \neg \text{S-Box}[((P^L_i \oplus \text{Rnd\_Cnt}_i) <<<11)]$
2. XOR with P^R_i, apply key $Rkey_i$
   - $F_X \leftarrow F_1 \oplus P^R_i$
   - $F_Y \leftarrow F_2 \oplus F_X$
   - $P_t \leftarrow F_Y \oplus Rkey_i$
3. Apply 32-Bit permutation(BP)
   - $P^R_{i+1} = \text{BP}[P^L_i]$
   - $P^L_{i+1} = \text{BP}[P^R_i]$

After 26 rounds, the 64-bit ciphertext can be obtained by concatenating P^L_{26} and P^R_{26}.

$$CT_H \leftarrow P^L_{26} \,||\, P^R_{26}$$

3.2. Decryption Flow

The ciphertext is partitioned into two halves of 32-bit each as follows:

$$CT_H \leftarrow P^R_{i+1} \,||\, P^L_{i+1}$$

1. Perform 32-bit inverse Bit Permutation (BP) on P^R_{i+1} and P^L_{i+1}
   - $P^L_i \leftarrow \text{BP}^{-1}[P^R_{i+1}]$
   - $P_t \leftarrow \text{BP}^{-1}[P^L_{i+1}]$
2. Apply F_1 and F_2 functions on P^L_{i+1} which results in F_X and F_Y, respectively,
   - $F_2 \leftarrow F(P^L_{i+1})$
   - $F_1 \leftarrow F(P^L_{i+1})$
3. The output of function F_2 is XOR with the current round key $Rkey_i$ and P^R_i
   - $F_X \leftarrow F_2 \oplus \text{Rkey}_i \oplus P_t$
4. F_x is XOR with F_1, which gives 32-bit plaintext.
   - $P^R_i \leftarrow \text{Fx} \oplus \text{F1}$

After 26 rounds, the ciphertext is converted into plaintext by concatenation of 32-bit LSB (P^L_i) and MSB (P^R_i).

$$PT_H \leftarrow P^L_i \;//\; P^R_i$$

3.3. S-box used in proposed cipher TED

The S-box used in TED block cipher is as shown in Table 1. S-box of TED block cipher is 4 bit S-box: $F_2^4 \rightarrow F_2^4$

Table 1. TED block cipher S-box

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

3.3.1. Bit-sliced Implementation of S-box

Bit slice algorithm performs N operations in parallel on a microprocessor with N-bit register width, facilitating improved performance boost and linear code. Along with the speed and parallelization, Bit-slice implementation supports constant-time implementation, which helps to thwart the cache-timing type of SCA attacks. Thus, Bit slice implementation of symmetric cipher has several advantages over the traditional one. Bit slice implementations convert the encryption algorithm into a series of logical bit operations using XOR, AND, OR and NOT logical gates. TED block cipher uses a bit sliced computation of the S-boxes using Boolean functions, not requiring look-up tables. The implementations are tested in embedded ARM cortex CPUs ranging from lower-end microcontrollers to full-featured processors, which supports vector instructions. The S-box of a TED block cipher described by the following Boolean equations.

Let, $A = A_0\ A_1\ A_2\ A_3$ be the input of the S-box and $B = B_0\ B_1\ B_2\ B_3$ be the output.

$B_0 = !A_0\&!A_2\&!A_3 \parallel !A_0\&!A_1\&!A_2 \parallel A_0\&A_2\&A_3 \parallel !A_1\&!A_2\&!A_3 \parallel !A_1\&A_2\&A_3$

$B_1 = A_0\&A_1\&A_3 \parallel A_1\&A_2\&A_3 \parallel A_0\&A_2\&A_3$

$B_2 = !A_0\ \&\ !A_1\&A_2 \parallel !A_0\ \&\ A_2\ \&\ A_3 \parallel A_1\&!A_2\&!A_3 \parallel A_0\&A_1\&!A_2 \parallel A_0\&!A_2\&A_3$

$B_3 = !A_0\ \&!A_1\&!\ A_3 \parallel A_0\ \&\ A_2\ \&!\ A_3 \parallel !\ A_1\&A_2\&!\ A_3 \parallel A_1\&!\ A_2\&A_3 \parallel A_0\ \&!\ A_2\ \&A_3$

3.4. Bit Permutation used in proposed cipher TED

The permutation is performed on LSB P^L_i and the XOR output of the function F_2. The combination of a non-linear S-box operation followed by the permutation operation increases the active S-box count. For permutation, many ciphers use a look-up table, but it improves the overall memory footprint as well as makes more prone to different attacks. Hence, the permutation position is calculated rather than getting it from a permutation table. The bit-permutation positions are calculated as per the following pseudo-code.

Algorithm 1 Algorithm for Bit-Permutation
Input : index Output: Bit_permutation[ ] 1. for i = 0 to 31 do 2. if (index = = 0 OR index = = 31) then 3. Bit_permutation[index] = index 4. else 5. Bit_permutation[index] = (index * 8) mod 31 6. end if 7. end for

3.5. Key Schedule of TED Block Cipher

The key-scheduling algorithm is inspired by the PRESENT block cipher [4]. The key-scheduling algorithm used in PRESENT block cipher is considered to be one of the most robust key-scheduling design. In TED block cipher, the key scheduling algorithm produces a total of 26 sub-keys of the size of 32 bits as described below:

$KEY = K_{127}\ K_{126}\ K_{125}\ldots K_2\ K_1\ K_0 \qquad K_i = K_{31}K_{30}\ldots\ K_2\ K_1\ K_0$

After extracting 32-bits, KEY updated as per the following operations:

1. KEY <<< 21.
2. $[K_3K_2K_1K_0] \leftarrow S[K_3K_2K_1K_0]$
3. $[K_7K_6K_5K_4] \leftarrow S\ [K_7K_6K_5K_4]$
4. $[K_{69}K_{68}K_{67}K_{66}K_{65}] \leftarrow [K_{69}K_{68}K_{67}K_{66}K_{65}] \wedge (i^2)$

In the key-scheduling algorithm, two S-boxes are used in which round counter i is unique for each round.

4. SECURITY ANALYSIS OF TED

This section describes the sustainability and robustness of the TED block cipher against different attacks. S-box and modular additions are the two non-linear operations used in this whole cipher design. A computer-based algorithm used to get an active S-box count.

4.1. Linear cryptanalysis

The Linear Approximation Table (LAT) is constructed to test the S-box used in the proposed cipher design. As per the lemma given in [11], and LAT the linear bias (εL) calculated as follows:

For the linear probability P_L, bias is = $|P_L-1/2|$,
Proposed S-box bias (εL) = 2^{-2}

It is required to calculate maximum bias for a specific number of rounds. In linear trail, the S-box has a non-zero input and output mask referred to as an active S-box. The linear attack complexity is calculated from the number of minimum known plaintexts to be known by an attacker. To defend against a linear attack, the required number of known-plaintexts should be greater than 2^{64}.

Theorem 1: TED has a total of 66 active S-boxes and 2^{-67} maximum bias over 24 rounds.

Proof: TED has at least eleven linearly active S-boxes over four rounds. The maximum bias for the TED cipher S-box is 2^{-2} by using Matsui's Pilling up Lemma [8]. For four rounds of TED cipher, the total bias calculated, as shown in equation (2):

$$2^{10} \text{ x } (2^{-2})^{11} = 2^{-12} \quad (2)$$

By applying the same lemma for 24 rounds, the total bias (ε) is given as:

$$\varepsilon = 2^{5} \text{ x } (2^{-12})^{6} = 2^{-67}$$

The complexity of linear attack [8] is computed in the following given equation (3):

$$N_L = 1/ (\varepsilon)^2 \quad (3)$$

For 24 rounds of the TED cipher, the required number of known-plaintexts to apply linear attack are calculated as shown in equation (4):

$$N_L = 1/ (\varepsilon)^2 = 1/ (2^{-67})^2 = 2^{134} \quad (4)$$

For a linear attack or a known-plaintext attack, the required number of known-plaintexts are $2^{134,}$ which is far greater than the available limit, i.e., 2^{64}.

4.2. Differential Cryptanalysis

The resistance of full round TED cipher against the differential attack is elaborated with the help of Theorem 2:

Theorem 2: TED block cipher has 26 rounds. Out of that, for 24 rounds, there are 72 active S-boxes. The total differential probability (Pd) is 2^{-144} for 24 rounds. The total chosen plaintext/ciphertext required is $2^{144,}$ which is higher than 2^{64}.

Proof: For four rounds of TED, it has a minimum of twelve differentially active S-boxes. Thus, for 24 rounds, there will be 12 x 6 = 72 active S-boxes. For 24 rounds of TED, the total differential probability (P_d) given as $(2^{-2})^{72} = 2^{-144}$.

The complexity of a Differential cryptanalysis attack is evaluated by formulating the number of chosen plain text required (N_d). The number of Chosen-Plaintexts required (N_d) are calculated [11], as follows:

$$N_d = C/P_d$$

(where, C = 1 and $P_d = 2^{-144}$). The total Chosen-Plaintexts required are:

$$N_d = 1/2^{-144} = 2^{144}$$

The required number of Chosen-Plaintexts is 2^{144}, which is significantly more than 2^{64}. Hence, the TED cipher has good resistive capacity against a differential cryptanalysis attack [10].

4.3. Biclique Attack

The biclique attack [12] is a variation of the meet-in-the-middle (MITM) attack. It is a complete theoretical attack and solely based on how the key is chosen. A 3-dimensional biclique is constructed for round 21 to 26 of TED. The partial keys are used for these rounds, which are described as follows:

$RK_{23} = K_{63}, K_{62}\ldots. K_{32}$
$RK_{24} = K_{12}\ldots.K_0, K_{108}... K_{61}$
$RK_{25} = K_{71}\ldots.. K_{40}$
$RK_{26} = K_{30}, K_{29}..... K_0, K_{127}$

The Δi-differential is constructed by considering the subkeys (K_{63}, K_{62}, and K_{61}) and for the ∇j-differential, subkeys (K_{42}, K_{41}, K_{40}) are considered. Data complexity for TED cipher does not exceed 2^{44}. The total computational complexity of the biclique attack on the full TED is computed as follows, where d is the Biclique dimension, $C_{biclique}$ is the complexity of single biclique construction, $C_{precomp}$ is pre-computation complexity, C_{recomp} is re-computation complexity and $C_{falsepos}$ is complexity caused by false positive.

$C_{Total} = 2^{k-2d}(C_{biclique} + C_{precomp} + C_{recomp} + C_{falsepos})$
$C_{Total} = 2^{128-6}(2.46+ 6.77+ 30.62 + 2^2)$
$C_{Total} = 2^{122}(2^{1.3}+ 2^{2.76}+ 2^{4.94} + 2^2).$
$C_{Total} = 2^{127.45}$

4.4. Algebraic Attack

Algebraic attacks (AA) maps block cipher into a system of equations. The attacker recovers the key by applying algebraic transformations to these equations. The higher number of equations means increased resistance against AA [13].

S-box of TED cipher is represented in equation form using a minimum of 21 equations with eight input-output variables.

The whole cipher is described by $m = x \times 21$ quadratic equations with,

$n = x \times 8$ variables

where x is the total S-boxes used in a whole cipher along with key scheduling. In TED block cipher, sixteen and two S-boxes used in a single round function and key-scheduling algorithm, respectively. Thus for 26 rounds, there are a total of 468 S-boxes used, 416 during encryption ($26\times16 = 416$), and 52 during key-scheduling ($26\times2=52$).

Thus, total quadratic equations and required variables are calculated as follows:

m = (416 + 52) x 21 = 9828
n = (416 + 52) x 8 = 3744

Thus, for TED block cipher the total quadratic equations are 9828 and required variables are 3744.

4.5. Avalanche Effect

A robust block cipher should have a higher avalanche effect. Poor avalanche effect results in poor randomization, and poor randomization implies the cipher has weak security characteristics [14]. Avalanche effect for TED cipher is tested by keeping the key-value constant and changing a single bit from plaintext and the average number of bits changed is 38. For constant plaintext and changing a single bit from the key, the average 39 number of bits are changed.

4.6. Related Key and Slide Attacks

In this type of attack, the attacker tries to know or to choose a relation among several keys, and the attacker has to access encryption function with these keys [15]. There are two variants of this attack, such as a) Known related-key attack and b) chosen related-key attack. The complex relationship among the encryption keys is one of the approaches to fight against this attack. To implement this approach, each key is generated through a key derivation function named as a key scheduling algorithm. Another attack is a slide attack that analyzes the complexity of the key scheduling algorithm of the cipher and tries to get cyclic keys. For the PRESENT cipher's key scheduling algorithm, the related-key attack has not found successful. Hence, the key-scheduling algorithm used in the proposed cipher is solely based on the PRESENT key-scheduling algorithm to fight against the related-key attack.

4.7. Structural Attacks

Structural attacks are not as powerful attacks as Statistical Attacks for a given block cipher, as they are less capable of making the use of weaknesses of integral functions. Integral attacks, high order differential attacks and bottleneck attacks are well-known forms of structural attacks [16]. Block ciphers which are having word-based operations are more susceptible to this attack. However, TED block cipher design is almost based on bitwise operations such as bitwise permutation, modular addition, and XOR operation.

5. SECURITY COMPARISON WITH STANDARD ALGORITHMS

In this section, the security of TED block cipher is compared with other state-of-the-art cipher algorithms. In each round of the cipher algorithm, S-box can be referred to as either 'active' or 'passive.' The Active S-box count for TED cipher is calculated in Section 4.3, and it is more than the other cipher algorithms. The number of known and chosen plaintexts required to perform linear and differential cryptanalysis attack and active S-boxes count comparison is described in Table 2.

Table 2. Linear and Differential Cryptanalysis

Cipher	*# of Rounds*	*# of active S-boxes*	*# of known PT*	*# of chosen PT*	*Reference*
TED	**24**	**66**	**2^{134}**	**2^{144}**	**Proposed Cipher**
PRESENT	25	50	2^{102}	2^{100}	[4]
MANTRA	28	56	2^{114}	2^{112}	[6]
QTL	18	54	2^{110}	2^{94}	[10]
GRANU-LE	21	63	2^{140}	2^{138}	[7]
FeW	27	45	2^{90}	2^{90}	[8]
PICCOLO	30	30	2^{120}	2^{120}	[9]

*PT: Plaintext

6. HARDWARE AND SOFTWARE PERFORMANCES OF THE TED CIPHER

The performance parameters considered for the comparison are CPU cycles, gate equivalents (GE), code-size, memory requirement, and power consumption. All ciphers considered for the comparison are implemented on the same hardware and software platforms to have a fair comparison. The proposed cipher is evaluated on both the software and hardware platforms.

6.1. Hardware-Based Implementation of TED

For hardware-based benchmarking ASIC approach is used. TED block cipher and the other block ciphers are implemented in Verilog. The functional verifications are carried out using Cadens CDS Encounter v11.10 - p003_1 (64 bit) simulation software. These designs are synthesized using RTL Compiler for Standard Cell library of the STM 90nm Logic Process. Cadens tool calculates GEs more accurately than a manual approach used in the literature [4,6, 7 and 8]. A lightweight block cipher technique construction can be directed towards a latency, area, or throughput optimized implementation. The benchmarking results presented in this paper are obtained from the area optimized implementation. Many applications more often use only encryption than the encryption-decryption both. Hence the results are given for both the operations. Comparative results of the TED cipher and other ciphers are provided in Table 3. From this table, it can be inferred that TED requires the lowest GEs and energy per bit for encryption operation as well as encryption and decryption as compared to other ciphers, except SPECK and SIMON ciphers. However, the security analysis is not fully provided by the SPECK and SIMON cipher's authors in their paper.

Table 3. GEs and Energy/bit Comparison

Cipher (Structure)	*BlockSize/ Key Size*	*Operation*	*GE*	*Energy (pJ)*	*Energy /bit (pJ)*
AES(SPN)	128/ 128	E+D	24234	816	6.4
		Enc	14895	462.7	3.6
PRESENT (SPN)	64/ 128	E+D	2286	274.2	4.3
		Enc	1518	169.3	2.6
SPECK (ARX)	64/ 128	E+D	1552	178.4	2.8
		Enc	856	90.5	1.4
SIMON (ARX)	64/ 128	E+D	1482	165.8	2.6
		Enc	758	84.7	1.3
ITUbee (Feistel)	80/ 80	E+D	4589	468.3	5.9
		Enc	3145	312.2	3.9
GRANULE (Feistel)	64/ 128	E+D	5084	429.4	6.7
		Enc	3589	394.1	5.4
MANTRA (Feistel)	64/ 128	E+D	2250	259.8	4.1
		Enc	1496	172.9	2.7
TED (Feistel)	**64/ 128**	**E+D**	**2168**	**250.7**	**3.9**
		Enc	**1280**	**185.2**	**2.4**

(*E+D: Encryption + Decryption)

Figure 2 shows the memory consumption in terms of bytes for different ciphers. It is observed that proposed block cipher TED has the most compact memory footprint as compared with other lightweight block ciphers due to in-place bit permutation technique and code optimization techniques used in the cipher design.

6.2. Software-Based Implementation of TED

Software-based implementation is carried out on 32-bit widely used embedded ARM processors like ARM cortex M0, M3, A5, and A15 processor. CPU cycles are measured on the ARM processor by accessing the performance monitor control register. For compilation, the GCC compiler is used with the O_3 optimization level. Code size and CPU cycles required for the processing of key schedule, encryption, and decryption are as shown in Table. 4. From the obtained results, it is realized that ARM cortex processor M3 supports the most compact code size, whereas the ARM cortex A5 processor gives the highest speed-up for the encryption and decryption of the plaintext.

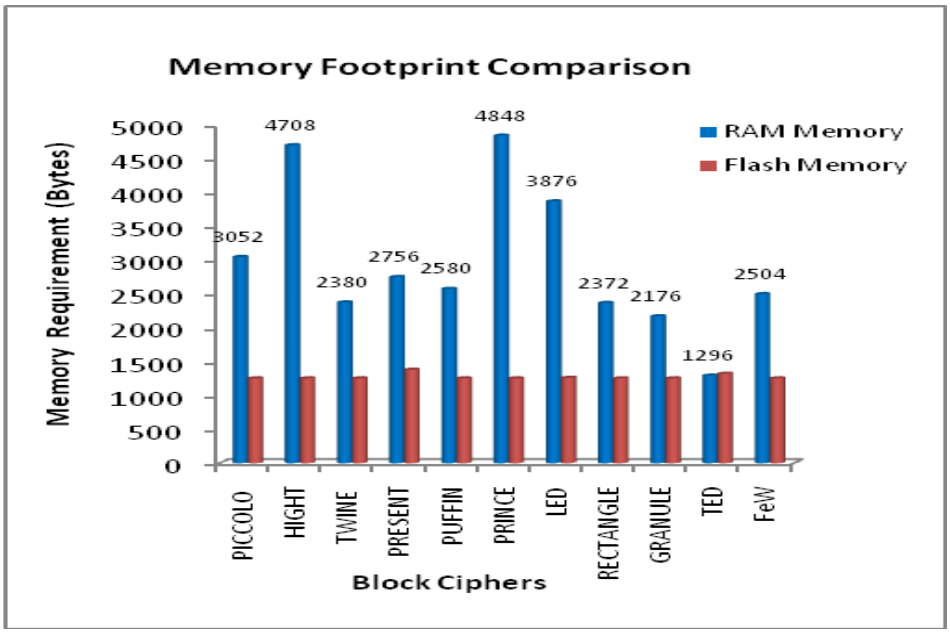

Fig. 2. Memory requirements in bytes of different ciphers

Table 4. Code size and CPU cycles result on ARM

Processor	*Code Size (Bytes)*	*Key Schedule (CPU Cycles)*	*Encryption+Decryption (CPU Cycles)*
ARM Cortex M0	1802	1428	4228
ARM Cortex M3	1768	1367	4056
ARM Cortex A5	1860	1040	3089
ARM Cortex A15	1908	1096	3369

7.CONCLUSIONS

Proposed cipher tested on most widely used embedded processors, and found that ARM cortex A5 gives 2.8X times speed-up compared to other processors considered for the experimentations. Avalanche effect resultant values, shows that TED has strong randomization properties. Due to the power of the two-stage non-linear layer, we could derive a tight bound up to 66 active S-boxes. Thus, resistance of proposed cipher design against differential and linear attacks is tested and results shows that proposed cipher design resist them at sufficiently high level. To make TED block cipher SCA-resistant with limited energy consumption, S-box is implemented with a bit-slice implementation.

We achieve 14.85% improved energy efficiency with area optimized design. Finally, we have performed the SCA attacks, Zero- correlation attack, Bicklique attack, structural attack, and Algebraic attacks on proposed cipher design and

proven its hard-edged security. As a possible direction for future research, one can investigate the energy-efficient design for masked S-boxes and inverse masked S-boxes to have robustness against SCA attacks one level up.

REFERENCES

[1] Prathiba A, Kanchana S., A Review on the Design of Lightweight Symmetric Block Ciphers for Cyber-Physical Systems. *Int'l Journal of Recent Technology and Engineering (IJRTE)*, ISSN: 2277-3878, **6** (Vol-7), March 2019, pp. 294-305.

[2] Dalmasso, L., Bruguier, F., Benoit, P. and Torres, L. Evaluation of SPN-based Lightweight Crypto-ciphers. *IEEE Access*, vol. 7, 2019, pp.10559-10567.

[3] Thorat, C., Inamdar, V. Implementation of the New Hybrid Lightweight Block Cipher. *Journal of Applied Computing and Informatics*, 2018, pp. 1-6.

[4] Bogdanov, A.et al. PRESENT: An Ultra-lightweight Block Cipher. *Proc. of the Cryptographic Hardware and Embedded Systems*, CHES 2007, Berlin, Germany: Springer, pp. 450-466.

[5] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B. and Wingers, L. The SIMON and SPECK Families of Lightweight Block Ciphers. *Proc. Of the IACR Cryptology ePrint Archive*, 2013 (1), pp.404-449.

[6] Bansod, G., Pisharoty, N., Patil, A., MANTRA: An Ultra-lightweight Cipher Design for Ubiquitous Computing. *Int'l Journal of Ad Hoc and Ubiquitous TableComputing*, **1** (vol. 28), 2018, pp. 13-26.

[7] Bansod, G., Patil, A., Pisharoty, N. GRANULE: An Ultra Lightweight Cipher Design for Embedded Security. *Proc. Of the IACR Cryptology ePrint Archive*, 2018, pp. 600-612.

[8] Kumar, M., Pal, S. K., Panigrahi, A., FeW: A Lightweight Block Cipher. *Proc. Of the IACR Cryptology ePrint Archive*, 2014, pp. 1-18.

[9] Shibutani, K., Isobe, T., Hiwatari, H., et al., Piccolo: An Ultra-lightweight Block Cipher. *Proc. Of the Cryptographic Hardware and Embedded Systems*, Berlin, Germany: Springer, 2011, pp. 342–357.

[10] Lang, L., Botao, L., Hui, W. QTL: A New Ultra-lightweight Block Cipher. *Int'l J. of Microprocessors and Microsystems*, vol. 45, August 2016, pp. 45-55

[11] C. Blondeau, G. Leander, and K. Nyberg, Differential-linear Cryptanalysis Revisited. *Springer Journal of Cryptology*, **3** (vol. 30), 2014, pp. 859-888.

[12] Han, G. and Zhang, W. Improved Biclique Cryptanalysis of the Lightweight Block Cipher Piccolo. *Security and Communication Networks*, 2017.

[13] K. Lisickiy, K. Kuznetsova, Y. Malenko, S. Kavun, O. Zavgorodnia and Y. Tarasenko, Accelerated Method for Calculating the Algebraic Immunity of S-Boxes. *IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, Lviv, Ukraine, 2019, pp. 899-905.

[14] Shi, Z.,Lee, R. B. Bit Permutation Instructions for Accelerating Software Cryptography, *Proc. of IEEE Int. Conf. on Application-Specific Systems, Architectures and Processors (ASAP 2000)*, 2000, pp. 138-148.

[15] Weize Y. and Selçuk K. A Lightweight Masked AES Implementation for Securing IoT Against CPA Attacks. *J. of IEEE Tra*ns. on Circuits and Systems, 2017, pp. 217-229.

[16] Wang Y. and Yajun H., A DFA-Resistant and Masked PRESENT with Area Optimization for RFID. *J. of ACM Trans. Embed. Comput. Syst.* **4** (vol. 16), Article 102, July 2017, pp. 302-334.

[17] Information Technology-Security Techniques-Lightweight Cryptography-Part 2: Block Ciphers, *Standard ISO/IEC 29192-2*, ISO, Geneva, Switzerland, 2012. [Online]. Available: https://www.iso.org/ standard/56552.html

Appendix: Test vectors

```
Plaintext: 0000 0000 0000 0000
Key: 0000 0000 0000 0000 0000 0000 0000 0000
Ciphertext: 54a1 50a7 d2f2  be42

Plain Text: 0123 4567 89ab cdef
Key: 0000 0000 0000 0000 0000 0000 0000 0000
Ciphertext: 3b69 e44b c5f2 68ad
```

Information about the authors:

Vandana Inamdar completed her Ph.D. in the area of image watermarking from the University of Pune, India. She is working as an Associate Professor in the College of Engineering, Pune, India. Her research interests include Signal and Image processing.

Chandrama Thorat is working as a research associate at the College of Engineering, Pune, India. Her research interests are in the area of cryptography, mobile devices security, lightweight cryptography, and wireless security.

Bhagvat Jadhav received the B.E. degree in Electronics Engineering from University of Pune, Pune, India, in 2002, M.Tech. from Pune University Pune, India, in 2008, and Ph.D. degrees in Electronics and Telecommunication engineering from S.P.P.U Pune, India, in 2017. Since January 2005, he has been with the Department of Electronics and Telecommunication engineering, RSCOE, Currently he is working as a Professor and head of the Department in E&TC Department. His current research interests include Signal, Image processing and Electronic Devices. He is a Life Member of the Indian Society for Technical Education (ISTE). He has filed two patents and published more than twenty papers in various IEEE conferences and reputed journals.

Manuscript received on 31 January 2020
Revised version re-submited on 09 March 2020