Implementation of New Energy-efficient Secure Block Cipher for M2M CPSs

Chandrama Thorat[1†] **and** **Vandana Inamdar**[2††],

Electronics and Telecommunication Engineering,
College of Engineering Pune, Pune,
Maharashtra, 411005, India

Computer Engineering and
Information Technology Department,
College of Engineering Pune, Pune,
Maharashtra, 411005, India,

Summary

Often Cyber-physical systems collect valuable information while controlling or monitoring the systems. It is becoming essential to implement security and privacy to protect connected CPSs. Nevertheless, the resource-constrained nature of the CPSs creates hurdles to achieve a high level of protection. In this work, we propose an energy-efficient secure block cipher TED. TED encrypts 64-bit size plaintext through 128-bit size key in 26 rounds. TED cipher supports improved energy efficiency and compact memory footprint in comparison with the other state-of-the-art balanced Feistel ciphers on a software platform and satisfactory performance on a hardware platform. To optimize the memory footprint, we avoid the use of look-up tables for the permutation box and substitution box. Operating on half part of input text and simple round structure improve the energy-efficiency. The proposed cipher design needs only 1296 Bytes of the primary memory and 793pJ with UMC 90nm STM CMOS technology. Compared with the other ciphers, our implementation on ARM processors supports low latency. We have tested statistical, structural, and algebraic cryptanalysis attacks and shown the robustness of proposed cipher against these attacks. The TED cipher has a good substitution layer, which has a large number of active S-boxes to thwart linear and differential cryptanalysis attacks. Additionally, the effectiveness of Bit-sliced implementation of the substitution layer reduces the threat of timing and cache type SCA attacks. The proposed cipher design will have a strong influence in the field of lightweight security implementation for the connected CPSs.

Keywords:

Lightweight cryptography; Energy efficiency, Block cipher; Bit permutation; Bit-slice;

1. Introduction

As we move towards an era of smart cities and smart systems, the proliferation of machine to machine (M2M) connected cyber-physical systems (CPSs) is changing our lives. Resource-constrained nature makes CPSs more vulnerable to different security attacks. The conventional security algorithms like AES [1] or T-DES [2] cannot apply directly to an internet of things (IoT) devices due to their resource-intensive computations. Lightweight cryptography is a perfect solution for such devices. As per the National Institute of Standards and Technology (NIST), the recent report on lightweight cryptography [3], the devices bifurcation proposed is shown in Fig. 1.

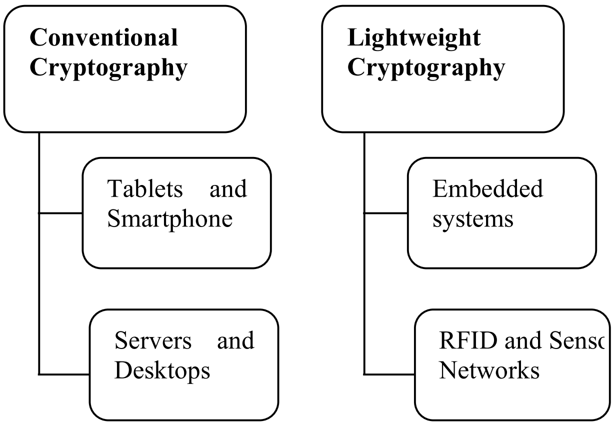


Figure. 1 Devices Bifurcation

Lightweight encryption block cipher techniques [4] constructed based on different design approaches such as a Substitution Permutation Network (SPN), Feistel Structure (FS), Generalized Feistel Structure (GFS) and Add Rotate and XOR (ARX). Along with these basic approaches recently, some cipher implementations have used a hybrid approach as used in [27]. Among these approaches, the oldest one is the FS approach used in the proposed cipher development. There are many other block ciphers which have used the Feistel structure approach, few of them are DESLX [5], GOST [6], ITUBee [7], RoadRunneR [8] and Mantra [9]. Among these lightweight encryption techniques, many cipher techniques primarily focus on ultra-low power and area design. For ultra-lightweight

Manuscript received December 5, 2025
Manuscript revised December 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.12.11

cipher design, researchers go with a smaller key size, which ultimately affects a security level of the proposed cipher technique. These kinds of implementations are mainly suitable for RFID devices or sensors. However, security requirements and resource availability differ in IoT devices than the RFID devices as they both work in different environments.

IoT devices are more vulnerable to Side-Channel Attacks (SCAs) and other attacks since they are more accessible to an attacker and connected to the Public network. The objective of this work is to design a cipher for the IoT devices, which needs a moderate security level along with energy efficiency. We aim to develop a cipher which will resist against SCAs and other known attacks with marginal energy consumption and memory footprint on primarily software platform.

The energy efficiency of any cipher technique depends on its Gate equivalence (GE) and CPU cycle count required for producing the ciphertext, as mentioned in [15]. Along with energy efficiency, robustness is also a valuable property to achieve by any block cipher. A secure block cipher is built-up with a proper selection of linear and non-linear layers in it. The proposed cipher titled Tiny Encryption Design (TED) uses two non-linear operations: 4×4 bit S-box and modular addition. The following are the main design issues of TED block cipher:

1. TED uses 4×4 bit S-box which meets all the criteria for a good substitution layer.
2. TED uses a 32-bit permutation operation, which is inspired by a PRESENT cipher.
3. Along with S-box, TED also uses a modular addition operation.
4. Proposed cipher can be implemented in parallel.
5. On ARM cortex A-53 processor, TED cipher implementation takes 27.5 cycles for the encryption of a byte and 30.1 cycles for the decryption of a byte.

Section 2 describes different Feistel-structure based block cipher techniques. Our contribution and TED specifications outlined in Section 3. Design rationale stated in Section 4, the results of the software and the hardware-based implementations are elaborated in Section 5. An extensive security analysis is done in Section 6 to prove TED supports adequate security. Section 7 presents concluding remarks.

2. Related Works

Over the last three decades, different lightweight cryptographic algorithms developed by the researchers with different objectives. Lightweight cryptographic algorithms include block ciphers, stream ciphers, hash functions, and the recent one authenticated encryption techniques. Among these, block ciphers more widely used. Lightweight block cipher designs are also adopted by ISO/OSI, where PRESENT and CLEFIA are the ISO/OSI standard lightweight block cipher algorithms [16]. Though PRESENT cipher is selected as an ISO/OSI standard block cipher, it has a weak substitution layer. While a permutation layer of the PRESENT block cipher is considered as one of the best P-layers used.

For the first time, Feistel-structure was used to develop Lucifer cipher designed by an IBM scientist Horst Feistel and Don Coppersmith in 1973. Later the US government adopted a DES cipher, which is based on Lucifer. Further, in 2007 DESLX cipher proposed, which is a lightweight version of the DES cipher in which eighht S boxes are replaced by a unique S-box for easy implementation. In [6], the author has proposed another cipher named GOST, which is developed with a simple structure, and the absence of a key-schedule makes it a tiny hardware footprint. ITUBee is a Feistel cipher developed mainly for a software environment that resulted in high efficiency and less memory requirement in software. Unlike other ciphers, ITUBee uses 80 bits of the plain text block size.

In [13], the author has claimed a minimal memory space for the cipher developed GRANULE. A lightweight cipher design MANTRA is proposed in [9], the author has claimed that it uses a less memory footprint. Linear trails and differential trails grouping prevented with the help of a secure permutation layer in MANTRA block cipher.

In [12], 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. It does not use any key scheduling algorithm. Notations used in TED cipher design are given below:

- PTH : 64- plaintext block bits
- CTH : 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
    - $\otimes$: Addition in modulo 2^{32}

3. TED Block Cipher

TED block cipher is an iterated Feistel-structure based cipher design. Fig.2 gives the insight of each round of a TED block cipher. Each round of TED block cipher uses two sub-functions are F1 and F2, which operate on the left halve of the 64-bit plaintext. The 64-bit plaintext is separated into two halves PLi and PRi, each having a size of 32 bits.

$$F_1 = F_2: \{0, 1\}^{32} \leftarrow \{0, 1\}^{32}$$

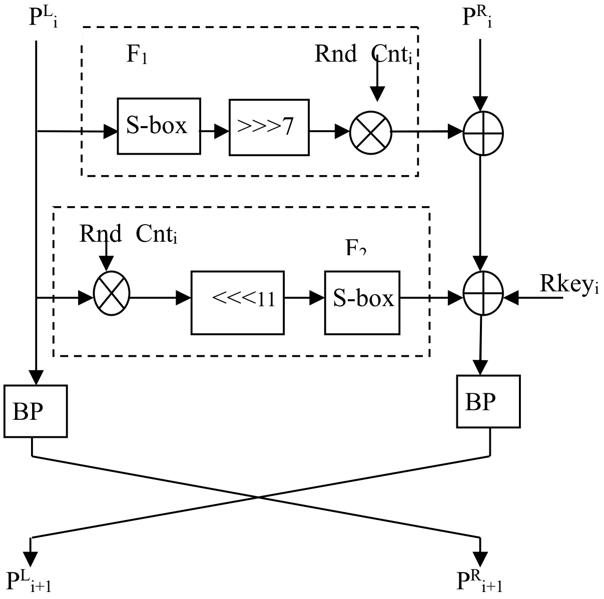


Figure.2 Single Round of TED

Function F_1 includes S-box operation, right circular shift operation, and modular addition operation, whereas function F_2 uses S-box operation, left circular shift operation, and modular addition operation. The output of function F_1 is XOR with P^R_i, and the result of this operation is XOR with the output of the function F_2 and 32-bit round-based key. Twenty-six different keys are produced through the 128-bit key and key scheduling algorithm.

3.1 Encryption Flow

A) The plaintext is divided into two 32-bit sub-blocks, which are referred to a P^L_i and P^R_i "(see (1))".

$$PT_H \leftarrow P^L_i \,||\, P^R_i \qquad (1)$$

B) The encryption algorithm is elaborated as described below:

1. Apply function F1 and F2 to 32-bit plaintext halve P^L_i

   $F_1 \leftarrow F\ (\boldsymbol{P^L_i})$

   $F_2 \leftarrow F\ (\boldsymbol{P^L_i})$

2. XOR with PR_i, apply key Rkeyi

   $F_X \leftarrow F_1 \oplus P^R_i$

   $F_Y \leftarrow F_2 \oplus F_X$

   $P_t \leftarrow F_Y \oplus Rkey_i$

3. Apply 32-Bit permutation (BP)

   P^R_{i+1} = BP $[P^L_i]$

   P^L_{i+1} = BP $[P_t]$

C) After 26 rounds, 64-bit cipher text 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

Like encryption function in decryption, the ciphertext is also partitioned to halves of 32-bit each as follows:

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

1. Perform 32-bit inverse Bit Permutation (BP) on P^R_{i+1} and P^L_{i+1}

   $P^L_i \leftarrow \mathrm{BP}^{-1}[P^R_{i+1}]$

   $P_t \leftarrow \mathrm{BP}^{-1}[P^L_{i+1}]$

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. Output of function F_2 is XOR with the current round key $Rkey_i$ and P_t

   $F_X \leftarrow F_2 \oplus Rkey_i \oplus P_t$

4. At last, F_x is XOR with F_1, results in 32-bit plaintext.

   $P^R_i \leftarrow Fx \oplus \mathrm{F1}$

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

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

3.3 Functions used in TED

The functions F_1 and F_2 of TED are derived from the right and left circular shift operation, S-box, and modular addition; the output of functions F_1 and F_2 are denoted by two different notations as F_X and F_Y respectively. The output of function F_1 is XOR with $P^R{}_i$ and the output of function F2 is XOR with the round key.

$$F_1: \{0, 1\}^{32} \neg \{0, 1\}^{32}$$
$$F_2: \{0, 1\}^{32} \neg \{0, 1\}^{32}$$

The mathematical illustration of function F_1 and F_2 is given as below:

$$F_1 \neg [\text{S-Box } (\boldsymbol{P^L}{}_i) >>> 7]] \oplus \text{Rnd\_Cnt}_i$$
$$F_2 \neg \text{S-Box } [((\boldsymbol{P^L}{}_i \oplus \text{Rnd\_Cnt}_i) <<< 11)]$$

Three different operations are used in function F1 and F2 of TED which are S-Box Layer, circular left-right rotations and modular additions. All of these operations are described in the following sections.

3.4 S-box

The S-box used in TED block cipher takes a 4-bit input and produces a 4-bit output. S-box used in TED block cipher is as shown in Table 1 in the form of hex representation. S-box of TED block cipher: $F_2^4 \rightarrow F_2^4$

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

3.4.1 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. It was the first time applied by Biham [28] to boost the performance of DES cipher in hardware. 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 lookup tables. The implementations carried out on embedded ARM cortex CPUs ranging from lower-end microcontrollers to fully-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.

$$B0 = \ !A_0\&!A_2\&!A_3 \ || \ !A_0\&!A_1\&!A_2 \ || \ A_0\&A_2\&A_3 \ || \ !A_1\&!A_2\&!A_3 \ || \ !A_1\&A_2\&A_3$$

$$B1 = \ A_0\&A_1\&A_3 \ || \ A_1\&A_2\&A_3 \ || \ A_0\&A_2\&A_3$$

$$B2 = \ !A_0 \ \& \ !A_1\&A_2 \ || \ !A_0 \ \& \ A_2 \ \& \ A_3 \ || \ A_1\&!A_2\&!A_3 \ || \ A_0\&A_1\&!A_2 \ || \ A_0\&!A_2\&A_3$$
$$B3 = \ ! \ A_0 \ \&!A_1\&! \ A_3 \ || \ A_0 \ \& \ A_2 \ \&! \ A_3 \ ||! \ A_1\&A_2\&! \ A_3 \ || \ A_1\&! \ A_2\&A_3 \ || \ A_0 \ \&! \ A_2 \ \&A_3$$

3.5 Bit Permutation

After applying functions F_1 and F_2, a bit permutation operation applied on LSB $\boldsymbol{P^L}{}_i$ and the XOR output of the function F_2. A bit at ith position moved to BP (i)th place. The combination of a non-linear S-box operation followed by the permutation operation increases the active S-box count. For permutation, many cipher techniques use a look-up table, but it raises a memory footprint of the cipher. Thus, we are calculating the position at which bit to be shifted in an algorithm itself. Algorithm for finding a bit-permutation position is given in Algorithm1. Resulted permutation value from Algorithm 1 shown in Table 2.

ALGORITHM 1: To Generate Bit Permutation positions.

```
Input: index
Output: Bit_permutation[ ]
for $i = 0$ to $31$ do
    if ($index == 0$ OR $index == 31$) then
        $Bit\_permutation[index] = index$
    else
        $Bit\_permutation[index] = (index * 8) \bmod 31$
    end if
end for
```

Table 2: TED Permutation Layer

i	0	1	2	3	4	5	6	7
BP(i)	0	8	16	24	1	9	17	25
i	8	9	10	11	12	13	14	15
BP(i)	2	10	18	26	3	11	19	27
i	16	17	18	19	20	21	22	23
BP(i)	4	12	20	28	5	13	21	29
i	24	25	26	27	28	29	30	31
BP(i)	6	14	22	30	7	15	23	31

3.6 Key Schedule of TED

A key-scheduling algorithm inspired by the PRESENT block cipher. The PRESENT block cipher considered to be one of the robust key scheduling designs among different key scheduling design algorithms. In the TED block cipher, the key scheduling algorithm produces 26 sub-keys of the size of 32 bits as follows:

A user gives a 128-bit key ((K_{127} K_{126} K_{125}…K_2 K_1 K_0) as input, which is referred as KEY. Out of these, 32- LSBs from KEY register used for each round of TED cipher that is K_i.

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

After extracting 32-bits, KEY is 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 a key scheduling algorithm, two S-boxes are used, and in step (4), *i* represent a round counter which is unique for each round.

3.7 Encryption Algorithm

ALGORITHM 2: Algorithm to generate 64-bit Cipher Text.

Input: Plaintext: $P_{64}=P_{63}P_{62}P_{61}.....P_3P_2P_1P_0$, S_Box[16],BP[32],$RKey_i$

Output: Cipher_Text:$C_{64} = C_{63}\,C_{62}C_{61}.....C_2C_1C_0$

```
1.  Divide the Plaintext into two halves $P^L_k$ and $P^R_k$
        $P^L_k = P_{63}P_{62}P_{61}.....\ P_{34}P_{33}P_{32}$
        $P^R_k = P_{31}P_{30}P_{29}.....\ P_3P_2P_1P_0$
2.  For k=0 to 26 do
    1) Apply function $F_1$ and $F_2$ on $P^L_k$
           $F_1 = [\text{S-Box}\ (P^L_k) >>> 7]\ ]\ \oplus\ k$
           $F_2 = \text{S-Box}\ [((\boldsymbol{P^L_k}\ \oplus\ k\ )\ <<<11)]$
    2) XOR the output of the function F1 and PRk and
       store it in temporary variable T1.
           $T1 = F1 \oplus PRk$
    3) XOR the output of the function F2, round key
       and T1 and store it in temporary variable T2.
           $T2 = F2 \oplus RKeyk \oplus T1$
    4) Apply bit permutation on both halves $P^L_k$ and
       $P^R_k$ and swap them.
    5) Update $RKey_k$
3.  end for
4.  Concatenate $P^L_k$ and $P^R_k$ to get ciphertext
        $C_{64} \rightarrow P^L_{26}\ ||\ P^R_{26}$
```

4. Security Analysis of TED Block Cipher

There are many different forms of cryptanalytic attacks applied to different lightweight cryptographic block ciphers till date. In this section of the paper, it is shown that how TED block cipher is secure against many foremost attacks. There are various linear and differential cryptanalysis techniques which can be applied to prove how the cipher is adequately secured. Moreover, S-box selection is also an important aspect in cipher design as it performs a nonlinear operation in cipher execution [18, 19]. S-box and modular additions are the two non-linear operations used in the whole cipher design. A computer based algorithm is used to get an active S-boxes count for the selected S-box as per the lemma stated by Matsui. Up-side count of the active S-box is one of the security measures for lightweight block ciphers. In this section, it has been proved that robustness of TED block cipher against different popular attacks.

4.1. Design Criteria of the S-box

Properties essential for a good S-box design are mentioned as follows:

1. For any nonzero input mask and output mask like p, q $\in F_2^4$, so there is LC (p, q) in the following manner:
   $LC\ (p, q) = \#\{x \in F_2^4 |\ p \bullet x = q \bullet S(x)\}\ - 8| \leq 4$
2. For any non zero input sum 'p' and output sum 'q' where p, q $\in F_2^4$, such that Hw (p) = Hw (q) = 1,
   Where, Hw stands for Hamming weight,
   Set $LC = LC\ (p, q) = \#\{x \in F_2^4 | p \bullet x = q \bullet S(x)\}\ - 8| \neq 0$
3. For any nonzero input and output differences ΔP, $\Delta Q \in F_2^4$ respectively, there is $DC(\Delta P,\Delta Q)$ as follows:
   $DC\,(\Delta P, \Delta Q) = \#\ \{m \in F_2^4\ |\ S\ (m) \oplus S\ (m \oplus \Delta P) = \Delta Q\} \leq 4$
4. For any non zero input differences ΔA and output differences ΔB where ΔA, $\Delta B \in F_2^4$ set Hw (ΔA) = Hw(ΔB) = 1,
   Set $DC = DC\,(\Delta A, \Delta B) = \#\ \{x \in F_2^4 | S(x) \oplus S\ (x \oplus \Delta A) = \Delta B\} = 0$

5. Bijective i.e. S (p) ≠ S(q) where $\forall$ p,q : { p,q ∈ F_2^4 | p ≠ q}
   For example: S [2] ≠ S [7]
6. For any input of the S-box 'c' there should not be the output as a 'c'. Where, $\forall c : \{ c \in F_2^4 \mid S(c) \neq c\}$
   For example: S [2] ≠ 2.

Let CarDC denote the cardinality of Set DC and CarLC denote the cardinality of Set LC. Different cipher's S-boxes CarDC and CarLC values are shown in Table 3.

Table 3: Comparison of S-box Used in Different Block Ciphers

Cipher Name	Year of publishing and Ref.	Size of S-Box	CarDC	CarLC
GRANULE	2018[13]	4x4	2	2
FEW	2014 [14]	4x4	2	2
PRESENT	2007 [10]	4x4	0	8
MANTRA	2018[15]	4x4	0	4
TED	**This paper**	**4x4**	**2**	**2**

4.2. Selection of the S-box of TED

While selecting S-box, two more properties accomplished as follows:

(a) Affine equivalence [17]: Given two S-boxes S1 and S2 are said as affine equivalent only when there is bijective linear mappings P, Q and constants a_1, b_1 ε F_2^4 exist as follows :

$$S_2(x) = Q(S_1 (P(x) + a_1)) + b_1$$

Further, the equivalence referred to as affine equivalence. When a selected S-box satisfies criteria 1, 3, and 5 (see Section 4), then the S-box meets an affine equivalence property.

(b) Permutation-then-XOR equivalence [17]: Given two S-boxes S1 and S2 are said as permutation-then-XOR equivalent, if there exists 4x4 permutation matrices A0, A1 and constants (p, q) ε F_2^4 as follows :

$$S_2 (x) = A_1 (S_1 (A_0 (x) + p)) + q.$$

The equivalence referred to as PE equivalence. If selected S-boxes fulfill the criteria 1-5, then its PE equivalent S-boxes also fulfill the criteria of 1-5 (see Section 4).
The TED block cipher design considered both the properties mentioned above while designing S-box.

4.3. Linear Cryptanalysis

Linear cryptanalysis attacks, also known as 'Known plaintext attack', mainly based on the high-probability occurrences of linear expression, which consists of key and plaintext and ciphertext. To apply a linear attack, the attacker requires access to the subset of the known plaintexts and its ciphertext and their correlation. The S-box used in our cipher examined by constructing the Linear Approximation Table (LAT). As per the lemma is given in [30] and LAT, the best linear bias (εL) calculated as follows:

For the linear probability P_L, bias is = $|P_L\text{-}1/2|$,
TED block cipher's S-box bias (εL) = 2^{-2}

A Maximum bias for a specific number of rounds is needed to calculate. The count of these active S-boxes plays a vital role in resistance against this attack. In linear trail, the S-box that has a non-zero input and output mask referred to as an active S-box. The active S-boxes count in each round, given in Table 4. Linear attack complexity is calculated from the number of minimum known plaintexts to be known by an attacker. To show the robustness for a linear attack, this required number of known plaintexts should be higher than 2^{64}.

Table 4: Active S-boxes Count for TED Cipher

# of rounds	#of minimum active S-boxes
1	0
2	2
3	6
4	11
5	12
6	19
7	24

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 given in [19]. For four rounds of TED cipher, the total bias calculated given as below:

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

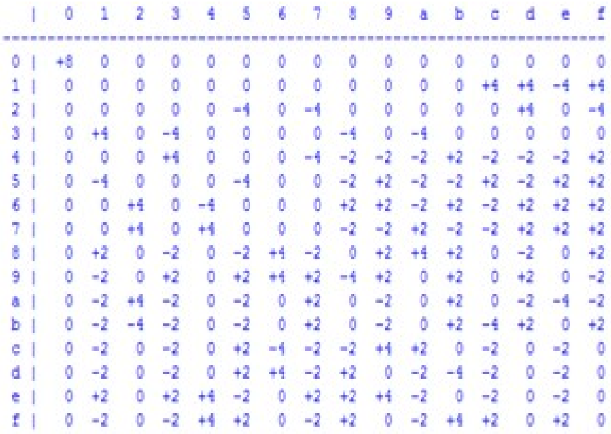
By applying the same lemma for 24 rounds, the total bias (ε) given as:

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

The complexity of linear attack : $N_L = 1/(\varepsilon)^2$
For 24 rounds of the TED cipher, the required number of known plaintext/ciphertext can be given as:

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

To apply a linear attack or a known-plaintext attack, the required number of known plaintext/ciphertext is 2^{134}, which is far greater than the available limit, i.e. 2^{64}. Hence, all the rounds of the TED cipher have a strong resistance against this attack. Figure 3 represents a linear approximation table (LAT) of the TED block cipher.



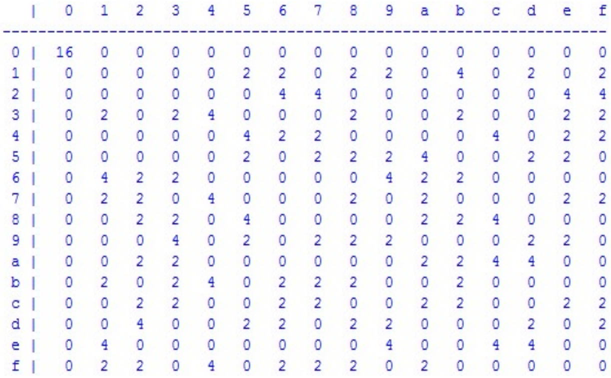
```
    |   0   1   2   3   4   5   6   7   8   9   a   b   c   d   e   f
----------------------------------------------------------------------
  0 |  +8   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  1 |   0   0   0   0   0   0   0   0   0   0   0   0  +4  +4  -4  +4
  2 |   0   0   0   0   0  -4   0  -4   0   0   0   0   0  +4   0  -4
  3 |   0  +4   0  -4   0   0   0   0  -4   0  -4   0   0   0   0   0
  4 |   0   0   0  +4   0   0   0  -4  -2  -2  -2  +2  -2  -2  -2  +2
  5 |   0  -4   0   0   0  -4   0   0  -2  +2  -2  -2  +2  -2  +2  +2
  6 |   0   0  +4   0  -4   0   0   0  +2  +2  -2  +2  -2  +2  +2  +2
  7 |   0   0  +4   0  +4   0   0   0  -2  -2  +2  -2  -2  +2  +2  +2
  8 |   0  +2   0  -2   0  -2  +4  -2   0  +2  +4  +2   0  -2   0  +2
  9 |   0  -2   0  +2   0  +2  +4  +2  -4  +2   0  +2   0  +2   0  -2
  a |   0  -2  +4  -2   0  -2   0  +2   0  -2   0  +2   0  -2  -4  -2
  b |   0  -2  -4  -2   0  -2   0  +2   0  -2   0  +2  -4  +2   0  +2
  c |   0  -2   0  -2   0  +2  -4  -2  -2  +4  +2   0  -2   0  -2   0
  d |   0  -2   0  -2   0  +2  +4  -2  +2   0  -2  -4  -2   0  -2   0
  e |   0  +2   0  +2  +4  -2   0  +2  +2  +4  -2   0  -2   0  -2   0
  f |   0  -2   0  -2  +4  +2   0  -2  +2   0  -2  +4  +2   0  +2   0
```

Figure 3: LAT for TED Block Cipher S-box

4.4 Differential Cryptanalysis

Differential cryptanalysis [18] is one more critical attack that was first applied by Biham and Shamir on DES cipher in 1990. In this type of attack, the attacker finds a pair of high probability input and output to get round keys. S-box examined through a Difference Distribution Table (DDT). This table is formed by finding differential trails with a high-probability input and output difference. In this type of cryptanalysis, S-box that has a non-zero input difference or non-zero output difference discovered. Such S-box is referred to as an active S-box here. The computer-based search algorithm used to get differential trails and active S-boxes in each round.

The Maximum differential probability (P_d) for the TED cipher S-box (4x4) is 2^{-2}. Difference Distribution Table (DDT) for the S-box of TED cipher shown in Figure. 4. To deal with differential cryptanalysis attacks, active S-box count needs to be more. Table 5 shows the active S-box count for TED cipher. In the fourth round, the minimum active S-boxes obtained are 12, which is a good count to defeat against differential cryptanalysis attack.



```
    |   0   1   2   3   4   5   6   7   8   9   a   b   c   d   e   f
----------------------------------------------------------------------
  0 |  16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  1 |   0   0   0   0   0   2   2   0   2   2   0   4   0   2   0   2
  2 |   0   0   0   0   0   0   4   4   0   0   0   0   0   0   4   4
  3 |   0   2   0   2   4   0   0   0   2   0   0   2   0   0   2   2
  4 |   0   0   0   0   0   4   2   2   0   0   0   0   4   0   2   2
  5 |   0   0   0   0   0   2   0   2   2   2   4   0   0   2   2   0
  6 |   0   4   2   2   0   0   0   0   0   4   2   2   0   0   0   0
  7 |   0   2   2   0   4   0   0   0   2   0   2   0   0   0   2   2
  8 |   0   0   2   2   0   4   0   0   0   0   2   2   4   0   0   0
  9 |   0   0   0   4   0   2   0   2   2   2   0   0   0   2   2   0
  a |   0   0   2   2   0   0   0   0   0   0   2   2   4   4   0   0
  b |   0   2   0   2   4   0   2   2   2   0   0   2   0   0   0   0
  c |   0   0   2   2   0   0   2   2   0   0   2   2   0   0   2   2
  d |   0   0   4   0   0   2   2   0   2   2   0   0   0   2   0   2
  e |   0   4   0   0   0   0   0   0   0   4   0   0   4   4   0   0
  f |   0   2   2   0   4   0   2   2   2   0   2   0   0   0   0   0
```

Figure 4: DDT for the TED block cipher's S-box

Table 5: Active S-boxes count of TED cipher S-box

Number of rounds	Number of minimum active S-boxes
1	0
2	2
3	5
4	12

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

Theorem 2: *TED has a total of 26 rounds out of 24 rounds of TED minimum 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 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 evaluated by formulating the number of chosen plain text required (N_d). The number of chosen plain text required (N_d) calculated as follows:

$$N_d = C/P_d$$

Where $C = 1$ and $P_d = 2^{-144}$, the total chosen-plaintext required is:

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

The required number of chosen plaintext/cipher text is 2^{144}, which is significantly more than 2^{64}. Hence, the TED cipher shows good resistance against a differential cryptanalysis attack.

4.5 Biclique Attack

The biclique attack [21] is a variation of the meet-in-the-middle (MITM) attack. It is a complete theoretical attack solely based on how the key is get chosen. Biclique attack cryptanalysis is applied to TED block cipher. A 3-dimensional biclique constructed for round 21 to 26 of TED. The partial keys used for these rounds are (RK_{23}, RK_{24}, $RK_{25,}$ and RK_{26}) described as follows:

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

From the above equations, it has been found that by applying the following sub-keys (K63, K62, K61) and (K33, K32, K31) on the above rounds gives a biclique attack on the complete TED block cipher.
The Δi-differential is constructed by considering the sub keys (K63, K62, and K61) and for the ∇j-differential, sub keys (K42, K41, K40) are considered. Let f be a sub-cipher from round 23 to round 26. Data complexity for TED cipher does not exceed 244.

The Computational complexity of biclique attack on TED is computed as follows:

$C_{Total} = 2^{k-2d} (C_{biclique} + C_{precomp} + C_{recom}p + C_{falsepos})$
$C_{Total} = 2^{128 -6} (2.46+ 6.77+ 30.62 + 2^2)$
$C_{Total} = 2^{122} (2^{1.3}+ 2^{2.76}+ 2^{4.94} + 2^2).$
$C_{Total} = 2^{127.45}$
where,
$C_{biclique} = 2^{d+1}$(Number of rounds in biclique/Total rounds)
$= 2^{3+1}(4/26) = 2.46$
$C_{precomp} = 2^d$ (Number of rounds in pre-computation/Total rounds)
$= 2^3(22/26) = 6.77$
$C_{recomp} = 2^{2d}$ (Number of active S- boxes in pre-computation/ Total number of maximum active S-boxes)
$= 30.62$
$C_{falsepos} = 2^{2d-(\text{no. of matching bits})} = 2^{2 \times 3- (4)} = 2^2.$

4.6 Algebraic Attack

Stream ciphers are more sensible to algebraic attacks [23] than block cipher algorithms. To elaborate 4x4-bit S-box, there is a need of minimum 21 equations with eight input or output variables. Whole cipher can be described by $m = x \times 21$ quadratic equations in $n = x \times 8$ variables. Where x is the total S-boxes used in a whole cipher along with key scheduling used in the cipher. For TED block cipher's single round, the total number of S-boxes used is sixteen S-boxes, and in key scheduling, two S-boxes used. For 26 rounds, there are 26×16 = 416 S-boxes in the encryption system, and 26×2 = 52 S-boxes in the key scheduling algorithm. Thus, total quadratic equations and required variables are calculated as follows:

$m = (416 + 52) \times 21 = 9828$
$n = (416 + 52) \times 8 = 3744$

Thus, for TED block cipher the total quadratic equations are formed with the help of 3744 variables which is a good count.

4.7 Avalanche Effect

The avalanche effect of a block cipher, a single bit alteration causes a considerable change in output bits. A robust block cipher should have a good avalanche effect, so it can defeat against many different attacks with higher probability [24]. The poor avalanche effect means poor randomization; such a block cipher does not support strong security characteristics. Table 7 (a) and (b) summarize the Avalanche effect for TED cipher. By keeping the key-value constant and changing a single bit from plaintext, the avalanche effect shown in Table 7 (b).

Table 7 (a): Avalanche Effect of TED Block Cipher with Constant Plaintext

Plaintext	0000 0000 0000 0000	No. of Bits Change
Key	0000 0000 0000 0000 0000 0000 0000 0000	-
Cipher Text	285b 7cd1 d34e 62af	
Key	0008 0000 0000 0000 0000 0000 0000 0000	35
Cipher Text	382a df61 bc73 81ae	
Key	0000 0000 0000 0000 0000 0000 0000 0020	34
Cipher Text	2671 3910 ade4 fb85	

Table 7(b): Avalanche Effect of TED Block Cipher with Constant Key Value

Key	0000 0000 0000 0000 0000 0000 0000 0000	No. of Bits Change
Plaintext	0000 0000 0000 0000	-
Cipher Text	7d83 43cf fb86 7dbd	
Plaintext	4000 0000 0000 0000	36
Cipher Text	21bd c533 2a54 fb34	
Plaintext	0000 0000 0000 2000	35
Cipher Text	8a26 7dbd 559e bc28	

4.8 Key Collision Attack

Any block cipher can be susceptible to a key collision attack. A key collision attack [25] is solely based on total key bits used to form a key. The attack is not dealing with key scheduling algorithm complexity. This attack creates a message having a complexity of 2k/2, where k represents the total length of the key. The complexity of the created message is given as $2^{128}/2=2^{64}$.

4.9 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 access to an encryption function with these keys [25]. There are two variants of this attack, namely a) Known related key attack and b) chosen related key attack. Till date, ciphers who unable to defend against related-key attacks are namely: AES, T-DES, GOST, etc. [1, 2, 7].

Having a complex relationship among the encryption keys is one of the approaches to fight against this attack. For this, each key can be generated through the key derivation function named as a key scheduling algorithm. Another attack named is slide attack, which analyzes the complexity of the key scheduling algorithm of the cipher and tries to get cyclic keys, if any.

For the PRESENT cipher's key scheduling algorithm is the most robust key-scheduling algorithm considered. 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.

The proposed cipher uses a circular left shift operation, XOR operation, and two S-boxes for designing a key scheduling algorithm.

4.10 Structural Attacks

Structural attacks are not as powerful attacks as Statistical Attacks for a given block cipher since they are less capable of making the use of weaknesses of integral functions. Integral attacks, high order differential attacks, and bottleneck attacks are all well-known forms of structural attacks [28]. Block ciphers which are having word-based operations are more susceptible to this attack. However, the TED block cipher design almost based on the bitwise operations, such as bitwise permutation, modular addition, and XOR operation.

5. Comparative Analysis

In this section, the security of TED block cipher compared with other state-of-the-art ciphers algorithms. The linear and differential cryptanalysis results are given in Table 8. In linear and differential cryptanalysis results, found that TED cipher has a sufficiently large number of active S-boxes than other ciphers.

Table 8: Linear and Differential Cryptanalysis Comparison

Cipher	# of rounds	# of active S-boxes	# of known plaintext	# of chosen Plaintext	Reference
TED	**24**	**66**	$\mathbf{2^{134}}$	$\mathbf{2^{144}}$	**This paper**
MANTRA	28	56	2^{114}	2^{112}	Bansod et al.,2018
ANU	18	54	2^{110}	2^{94}	Bansod et al.,2017
GRANULE	21	63	2^{140}	2^{138}	Bansod et al.,2018
FEW	27	45	2^{90}	2^{90}	Kumar et al., 2014
PRESENT	25	50	2^{102}	2^{100}	Bogdanov et al., 2007
PICCOLO	30	30	2^{120}	2^{120}	Shibutani et al., 2011

Data complexities and computational complexities of TED cipher compared with other ciphers are presented in Table 9. The Computation complexity of TED cipher is higher than other ciphers.

Table 9: Data Complexities and Computational Complexities Comparison

Cipher	Data complexity	Computational Complexity	Reference
TED-128	$\mathbf{2^{44}}$	$\mathbf{2^{127.45}}$	**This paper**
GRANULE-80	2^{40}	$2^{79.85}$	[13]
MANTRA-80	2^{59}	$2^{79.71}$	[9]
PRESENT-128	2^{19}	$2^{127.42}$	[10]
PRESENT-80	2^{23}	$2^{79.54}$	[10]
PICCOLO-128	2^{24}	$2^{127.35}$	[20]
PICCOLO-80	2^{48}	$2^{79.13}$	[20]
LED-128	2^{64}	$2^{127.37}$	[26]
LED-96	2^{64}	$2^{95.37}$	[26]

6. Results and Discussion

The performance parameters considered for comparison are CPU cycles, GE, Throughput, Execution time, memory requirement, and power consumption. All the ciphers considered for the comparison are implemented on the same hardware and software platforms to have a fair comparison. The proposed cipher evaluated on both the software and hardware platforms.

6.1. Hardware-Based Implementation of TED

For hardware-based benchmarking Application Specific Integrated Circuit (ASIC) approach is used. TED block cipher and the other block ciphers considered here for the comparison implemented in Verilog code. The functional verification carried out using Cadens CDS Encounter v11.10 - p003_1 (64 bit) simulation software. The designs synthesized using the 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 previous works [14, 15, 23]. A cipher implementation could be directed towards latency, area, or throughput optimized implementation. Among these, which optimization to be applied is based on the type of resource constraint the end device having in it. The benchmarking results presented in this paper are obtained from the area optimized implementations. Many devices or applications more often use only encryption than the encryption-decryption both. Hence the results are given for both the operations. The Comparative results of the TED cipher and other ciphers are shown in Table 10.

Table 10: GEs an Energy Consumption Comparison for a Single block of size 64-bit and key size128-bit

Cipher Name (Structure)	GE	Energy (pJ)
MANTRA (Feistel)	1045	605.8
HIGHT(ARX+GFN)	3048	1964.8
PRESENT(SPN)	1886	1523.2
PICCOLO (SPN)	1362	2099.2
TWINE(GFN	2285	1292.8
PRINCE (SPN)	3491	1555.2
PUFFIN (SPN)	2577	1236.4
RECTANGLE(SPN)	1787	1049.6
LED (SPN)	1265	1171.2
GRANULE(Feistel)	1690	908.8
TED(Feistel)	**1550**	**793.4**
RoadRunner(Feistel)	1123	755.2
FeW (Feistel)	2355	2260.4

*GFN-Generalized Feistel Network,

* SPN-Substitution and permutation Network

Comparative results of the TED cipher and other ciphers provided in Table. 10. The proposed cipher's results shown in bold numbers. TED cipher is the third one in the energy efficiency compared to other ciphers, except SPECK and SIMON ciphers. But SPECK and SIMON ciphers are the ciphers for which the security analysis not provided by cipher authors. It is evident from Fig. 2 that TED cipher has the most compact memory footprint compared to other lightweight ciphers. The Green color bar represents the smallest RAM requirement due to in-place bit permutation and bit-level operations used in the cipher design.

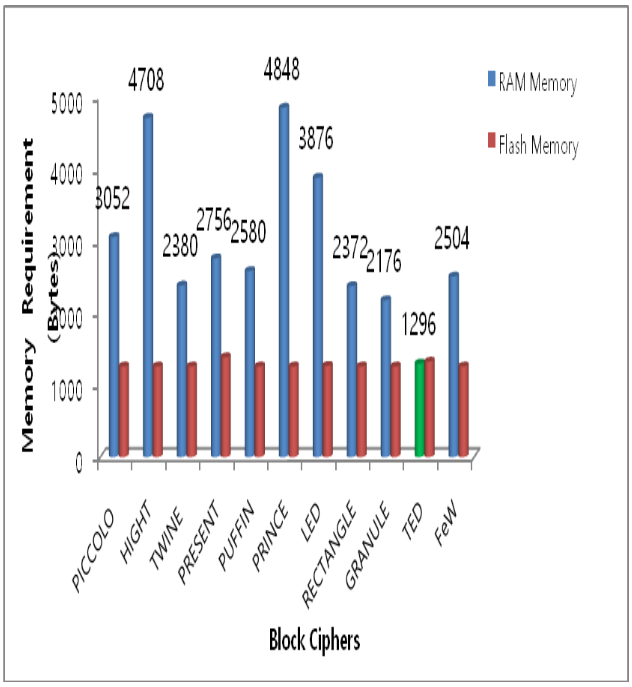


Figure 5 Memory requirements in bytes of different ciphers

6.2 Software-Based Implementation of TED

Software-based implementation carried out on 32-bit widely used embedded ARM processors like ARM cortex M0, M3, A5, and A15 processor. CPU cycles measured on the ARM processor by accessing the performance monitor control register. For compilation, the GCC compiler is used with the O3 optimization level. Code size and CPU cycles required for the processing of key-schedule, encryption, and decryption given.

From the obtained results, we realized that ARM cortex processor M0 supports the most compact code size, whereas the ARM cortex A53 processor gives the highest speed-up for the encryption and decryption of the plaintext.

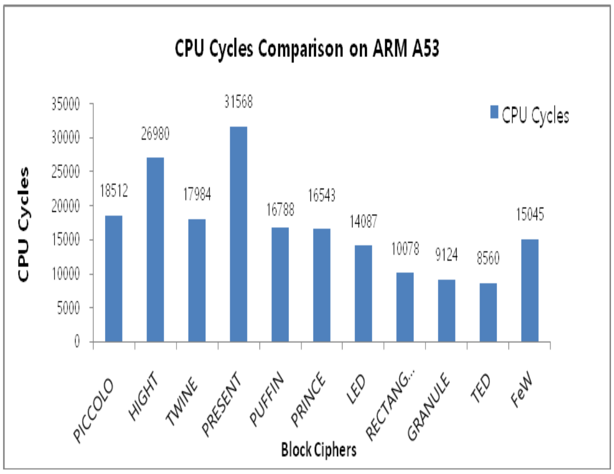


Figure 6 CPU Cycles results on ARM A53 of different ciphers

7. Conclusions

Proposed cipher is tested on most widely used embedded processors, it is found that ARM cortex A5 gives 2.8x times speed-up compared to other processors considered for the experimentation. From avalanche effect resultant values, it can be said that TED has strong randomization properties.

We propose a formal quantification of the number of known and chosen plaintexts required to perform linear and differential cryptanalyst is attack. Due to the power of the two-stage non-linear layer, we could derive a tight bound up to 66 active S-boxes. Therefore, we have proved the resistance of our algorithm to differential and linear attacks. Bit-slice implementation of S-box makes proposed cipher SCA attack resistant.

We achieved 14.85% improved energy efficiency with a compact primary memory footprint design. Finally, we have performed the Zero- correlation attack, biclique attack, structural attack, and Algebraic attacks on our proposed 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. This hybrid block cipher is very useful for the software accelerated lightweight cryptography community.

8. Acknowledgments

The authors would like to thank the Signal and Image Processing Centre of Excellence, College of engineering, Pune and JSPM's Rajarshi Shahu College of Engineering, Pune for giving all kind of support. We also express thanks to unknown referees for their helpful comments.

9. Funding

This research is supported by a Indian central government agency, Department of Science and Technology(DST) - Inter-Disciplinary Cyber Physical Systems(ICPS) under grant no - T108.

References

[1] Daemen, J., Rijmen, V.: 'The design of Rijndael, AES – the advanced encryption standard' , Springer, 2002

[2] US Department of Commerce National Bureau of Standards, D. E. :'Data encryption standard', Federal Information Processing Standards Publication (FIPS), Publication 46-3, Oct 1999

[3] Kerry, A., M., Larry, B., Meltem, T., et al.: 'Report on Lightweight Cryptography' , NIST, 2017, pp. 1-21

[4] Hatzivasilis, K., F., Papaefstathiou, I., and Manifavas C., 'A review of lightweight block ciphers', J. Cryptograph. Eng., vol. 8, no. 2, 2018, pp. 141–184

[5] Leander, G., Paar, C., Poschmann, A., et al.: 'New lightweight DES variants', In Fast Software Encryption, Springer, Berlin, Heidelberg, pp. 196-210

[6] Axel, P., San, L., Huaxiong, W.,: '256 Bit Standardized Crypto for 650 GE GOST Revisited', In CHES 2010, LNCS 6225, 2010, pp. 219-233

[7] Karakoc, F., Demirci, H., Harmanci, A.E.: 'ITUbee: A software oriented lightweight Block Cipher', Lightweight Cryptography for Security and Privacy, Springer, LNCS, 8162, 2013, pp.16-27

[8] Baysal, A., Suhap, S.: 'RoadRunneR: A small and fast bitslice block cipher for low cost 8-bit processors', Int. Workshop on Lightweight Cryptography for Security and Privacy - Volume 9542, LightSec 2015, 2015, pp. 58-76

[9] Bansod, G., Pisharoty, N., Patil, A. : 'MANTRA: an ultra lightweight cipher design for ubiquitous computing', Int. Journal of Ad Hoc and Ubiquitous Computing, 28(1), 2018, pp. 13-26

[10] Bogdanov, A., Knudsen, L.R., Leander, G., et al.: 'PRESENT: An ultra-lightweight block cipher', Proc. Cryptographic Hardware and Embedded Systems, CHESS 2007, Berlin, Germany: Springer, 2007, pp. 450-466

[11] Beaulieu, R., Shors, D., Smith, J., et al.: 'The SIMON and SPECK Families of Lightweight Block Ciphers',

Cryptology ePrint Archive, Report 2013/404, Available at http://eprint.iacr.org

[12] Lang, L., Botao, L., Hui, W.: 'QTL: A new ultra-lightweight block cipher', Journal of Microprocessors and Microsystems, Vol. 45 Issue- PA, August 2016, pp. 45-55

[13] Bansod, G., Patil, A., Pisharoty, N.: 'GRANULE: An Ultra lightweight cipher design for embedded security', IACR Cryptology ePrint Archive, 2018, pp. 600-612

[14] Kumar, M., Pal, S. K., Panigrahi, A. : 'FeW: a lightweight block cipher', IACR Cryptology ePrint Archive, 2014, pp. 1-18

[15] Kerckhof, S., Durvaux, F., Hocquet, C.: 'Towards green cryptography: A comparison of lightweight ciphers from the energy viewpoint', Cryptographic Hardware and Embedded Systems, Berlin, Germany: Springer, 2012, pp. 390–407

[16] 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

[17] Leander, G., Poschmann, A, "On the Classification of 4 bit S-boxes" In: Carlet, C., Sunar, B. (eds.) WAIFI 2007. LNCS, vol. 4547, Springer, Heidelberg (2007), pp. 159–176

[18] Biham, E., Shamir, A.:'Differential Cryptanalysis of DES-like Cryptosystems', Journal of Cryptology, vol. 4, no. 1,1991, pp. 372-391

[19] Matsui, M.: 'Linear Cryptanalysis Method for DES Cipher', Advances in Cryptology EUROCRYPT LNCS 765, Springer- Verlag,1994, pp. 386-397

[20] Shibutani, K., Isobe, T.,Hiwatari, H., et al. : 'Piccolo: An ultra-lightweight block cipher', Cryptographic Hardware and Embedded Systems, Berlin, Germany: Springer, 2011, pp. 342–357

[21] Jeong, K., Kang, H., Lee, C.,et al.: 'Biclique Cryptanalysis of Lightweight Block Ciphers PRESENT, Piccolo and LED', Cryptology ePrint Archive, Report 2012, pp. 621-639

[22] Bogdanov, A., Rijmen, V.: 'Zero Correlation Linear Cryptanalysis of Block Ciphers', IACR Eprint Archive Report,2011, pp. 123-141

[23] Albrecht, M., Cid, C.: 'Algebraic techniques in differential cryptanalysis'. FSE 2009, LNCS, vol. 5665, Springer, Heidelberg, 2009, pp. 193-208

[24] Shi, Z.,Lee, R. B. : 'Bit permutation instructions for accelerating software cryptography'.In Proc. IEEE Int. Conf. on Application Specific Systems, Architectures and Processors (ASAP 2000), July 2000, pp. 138-148

[25] Biryukov,A.,Wanger, D.:'Advanced slide attacks'.Proc. of Eurocrypt 2000, LNCS,1807,Springer-Verlag, 2000, pp. 589-606

[26] Guo, J., Peyrin, T., Poschmann, A., et al.:'The LED block Cipher, Cryptographic Hardware and Embedded Systems',CHES 2011. Springer, LNCS 6917, 2011, pp. 326–341

[27] Thorat, C. G., Inamdar, V. S.: 'Implementation of new hybrid lightweight block cipher', Applied Computing and Informatics. 2018