GRANULE: An Ultra lightweight cipher design for embedded security

Gaurav Bansod, Narayan Pisharoty, and Abhijit Patil

Abstract— **In this paper we proposed an ultra-lightweight cipher GRANULE. It is based on Feistel network which encrypts 64 bits of data with 80/128 bits of key. GRANULE needs very less memory space as compared to existing lightweight ciphers .GRANULE needs 1288 GEs for 80 bit and 1577 GEs for 128 bit key size. It also shows good resistance against linear and differential cryptanalysis. GRANULE needs very small footprint area and provides robust secure design which thwart attacks like biclique attack, zero correlation attack, meet in the middle attack ,key schedule attack and key collision attack. GRANULE is having a strong S-box which is the key designing aspect in any cipher design. In this paper GRANULE is proposed with 32 rounds which are enough to provide resistance against all possible types of attacks. GRANULE consumes very less power as compared to other modern lightweight ciphers. We believe GRANULE cipher is the best suited cipher for providing robust security in applications like IoT.**

Index Terms— **Lightweight Cryptography, Feistel cipher, Block cipher, IoT, Encryption, Embedded security**

I. INTRODUCTION

Recently many lightweight ciphers has been designed with robust design to make the fields like IoT, pervasive computing feasible. Ciphers like PRESENT [1], HUMMINGBIRD-2 [2], SIMON and SPECK [3], PICCOLO [4], RECTANGLE [5] have compact design and needs less GEs which is the key parameter in the field of lightweight cryptography. It is necessary the new cipher design should meet all the constraints like less gate count, memory size and robust secure design. Motivated from the latest cipher designs, in this paper we have proposed a cipher which is compact in design, needs very less footprint area, less gate count, low power consumption and last but not least should have strong secure architecture. The newly designed ciphers like PRESENT have a strong permutation layer. PRESENT is designed by keeping in mind compactness while RECTANGLE is designed with strong cryptanalysis properties. RECTANGLE has a strong S-box. In recent years ciphers like SIMON, SPECK, TWINE [6] are considered being most compact cipher and have good performance on both hardware and software platforms. GRANULE, the proposed cipher in this paper also designed to give netter performance on hardware and software platform. A nonlinear operation in all cipher design plays a very crucial role. In our design also we have chosen a strong S-box such that the overall design should be robust. In RECTANGLE cipher a strong S-box is implemented with the asymmetric permutation layer which prevents clustering of linear and differential trails. PRESENT have a problem of clustering of trails due to the improper selection of S-box and symmetric permutation. In GRANULE cipher design a strong S- box is accompanied with the strong and asymmetric permutation layer which results not only in preventing clustering of trails but also gives more number of active S-boxes.

We have done extensive computer based search for strong S-box in such a way that it should meets all the requirements of a strong substitution layer. Different circular shifts has been carried and tried out in GRANULE design to get resistance against all possible types of attacks. We believe that GRANULE's strong S-box followed by a robust F function is the robust and secure architecture which is best suited for providing security in applications like IoT. GRANULE has shown resistance against all possible types of basic attacks. GRANULE cipher design is motivated from parameters like less gate count, robust architecture and less power dissipation

For GRANULE cipher we used following notations

- PTm 64-bit input plaintext block
- CTm 64-bit output ciphertext block
- K 128-bit key register
- RKi 32-bit sub keys for round i
- F Function
- $\oplus$ Bitwise exclusive-OR operation
- <<<n Left cyclic shift by n bits
- >>>n Right cyclic shift by n bits
- $[i]^2$ Binary representation of integer i
- || Concatenation of two strings
- ! Bitwise NOT operation

II. THE BLOCK CIPHER GRANULE

GRANULE is Balanced Feistel Structure which consists of 32 rounds. The block length is of 64-bits and it support two key lengths of 80 and 128-bits.In each round of GRANULE, 32-bit round key RKi which is extracted from 128-bits key register is xored with the plaintext PT^0 and with the output of F function shown in Fig. 1.

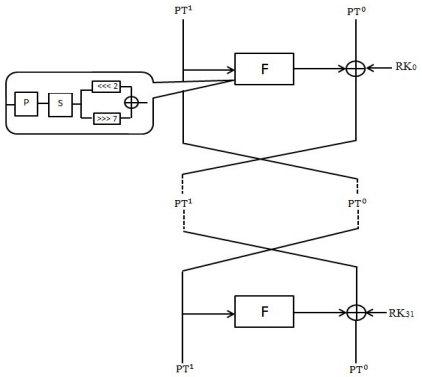

Fig. 1: GRANULE Feistel Structure

The F function in GRANULE is a combination of permutation layer, an S-box and the shift operators. Permutation layer in F function is block permutation, where each block size of 4-bits. The non-linear layer in GRANULE is a single 4-bit S-box. In F function left circular shift and right circular shift are used. Pseudo-code for cipher is given below:

PTm = PT^1 || PT^0
generateRoundKeys()
for i = 0 to 31 do
    PLayer(PT^1)
    SBoxLayer(PT)
    RPLayer(PT)
    addRoundKey(PT,PT^0, Ki)
    PT^0 = PT^1
    PT^1 = PT
end for
CTm = PT^1 || PT^0

After 25 round we get cipher text which is concatenation of PT^1 and PT^0.

A. Encryption Algorithm

In GRANULE, a 64-bit plaintext Pm is divided in two halves namely PT^1 and PT^0. Each of these halves is of size 32-bits follow:

$$PTm \leftarrow PT^1 \parallel PT^0$$

Encryption design in GRANULE is described below.

1. For i =0 to 24, apply Function F on 32-bit word PT^1 and XOR it with PT^0 and key Ki.
$$PT \leftarrow PT^0 \oplus F(PT^1) \oplus Ki$$

2. As shown in fig. 1 the plaintext will be interchanged
$$PT^0 \leftarrow PT^1$$
$$PT^1 \leftarrow PT$$

After 32 rounds, we obtain the 64-bit cipher text CTm which is concatenation of PT^1 and PT^0

$$CTm \leftarrow PT^1 \parallel PT^0$$

B. 'F' Function

F Function of GRANULE is shown in fig. 2 It takes 32-bit input PT^1 and produces 32-bit output.

$$F: \{0,1\}^{32} \leftarrow \{0,1\}^{32}$$

We have used three different layers in F function of GRANULE which are PLayer, SBoxLayer and Round Permutation Layer (RPLayer). All these functions take 32-bit input and produces 32-bit output. These are described in following sections.

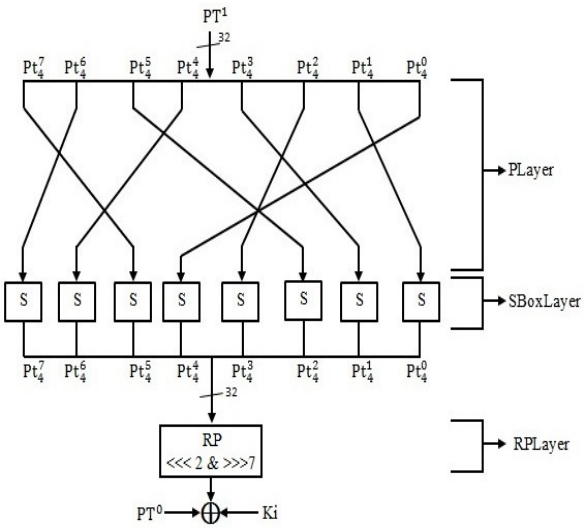

Fig. 2: 'F' Function of GRANULE

1. PLayer

The block permutation which is PLayer used in GRANULE is given by the Table 1. Block i of size 4 bits is substituted to block position P[i] of 4 bit size. The permutation P: $\{0,1\}^{32} \leftarrow \{0,1\}^{32}$ divides a 32-bit input PT^1 into eight 4-bit data as $PT^1 = Pt_4^0 \parallel Pt_4^1 \ldots Pt_4^6 \| Pt_4^7$ then permutes them by the following manner shown in fig. 2.

TABLE 1
PERMUTATION LAYER OF GRANULE

x	0	1	2	3	4	5	6	7
P[x]	4	0	3	1	6	2	7	5

P: $Pt_4^0 \parallel Pt_4^1 \| Pt_4^2 \| Pt_4^3 \| Pt_4^4 \| Pt_4^5 \| Pt_4^6 \| Pt_4^7 \leftarrow Pt_4^4 \parallel Pt_4^0 \| Pt_4^3 \| Pt_4^1 \| Pt_4^6 \| Pt_4^2 \| Pt_4^7 \| Pt_4^5$.

2. SBoxLayer

The S-box used in GRANULE is a 4-bit to 4-bit S-box S: $F_2^4 \rightarrow F_2^4$.The S-box of GRANULE is described in the Table 2.

TABLE 2
S-BOX OF GRANULE

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S[x]	e	7	8	4	1	9	2	f	5	a	b	0	6	c	d	3

3. RPLayer

The RPLayer in GRANULE takes 32-bit input and produces 32-bit output. RPLayer consists of application of cyclic shifts and exclusive-or operation. RPLayer is described below:

$$RP: \{0,1\}^{32} \leftarrow \{0,1\}^{32}$$
$$Temp0 \leftarrow IN <<< 2$$
$$Temp1 \leftarrow IN >>> 7$$
$$Y \leftarrow Temp0 \oplus Temp1$$

C. Key Schedule 80-bit and 128-bit

GRANULE supports 80 and 128-bit key size, GRANULE key design is motivated from the PRESENT [1] key schedule. The 128-bit key is stored in a key register K and is represented as $K_{127}\ K_{126}\ K_{125}\ldots K_2\ K_1\ K_0$. In each round of GRANULE, 32-bit round key Ki is extracted from 80/128- bits key register. The key scheduling functions for 128 bit is described below

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

After extracting 32-bit round key Ki, the key register K = K127 $K_{126}\ K_{125}\ldots K_2\ K_1\ K_0$ updated as described below

1. K <<< 31.
2. $[K_3\ K_2\ K_1\ K_0] \leftarrow S\ [K_3\ K_2\ K_1\ K_0]$
3. $[K_7\ K_6\ K_5\ K_4] \leftarrow S\ [K_7\ K_6\ K_5\ K_4]$
4. $[K_{70}\ K_{69}\ K_{68}\ K_{67}\ K_{66}] \leftarrow [K_{70}\ K_{69}\ K_{68}\ K_{67}\ K_{66}] \oplus [i]^2$

Round counter i of 5-bit is xor with 5-bit of the K register these are $K_{70}\ K_{69}\ K_{68}\ K_{67}\ K_{66}$

The key scheduling functions for 80-bit is also described below

$$K = K_{79}\ K_{78}\ K_{77}\ldots K_2\ K_1\ K_0$$
$$RKi = K_{31}\ K_{30}\ldots K_1\ K_0$$

After extracting 32-bit round key Ki of, the key register K = $K_{79}\ K_{78}\ K_{77}\ldots K_2\ K_1\ K_0$ is updated as described below

1. K<<< 31.
2. $[K_3\ K_2\ K_1\ K_0] \leftarrow S\ [K_3\ K_2\ K_1\ K_0]$.
3. $[K_7\ K_6\ K_5\ K_4] \leftarrow S\ [K_7\ K_6\ K_5\ K_4]$
4. $[K_{70}\ K_{69}\ K_{68}\ K_{67}\ K_{66}] \leftarrow [K_{70}\ K_{69}\ K_{68}\ K_{67}\ K_{66}] \oplus [i]^2$

III. Security Analysis GRANULE

There are large varieties of cryptanalytic attacks which can be applied on Feistel based design. In this paper we have shown GRANULE cipher resistance against major attacks. Linear and differential cryptanalysis is the important techniques to decide whether cipher is vulnerable to attacks or not. Moreover S-box selection also plays an important role as it adds a nonlinear element in cipher designing. We have adopted computer based techniques not only to find a good S-box but also to find minimum number of active S-boxes. GRANULE shows good resistance against linear and differential attacks which is described in the following sections.

A. Design Criteria of the S-box

A 4-bit to 4-bit S-box is more compact in hardware than an 8-bit to 8-bit S-box. Serpent [7] uses 8 different S-boxes, to provide robust architecture, but that result in more gate counts and the use of different S-boxes for different rounds does not result in a reasonable improvement of the resistance against known attacks [8].

In GRANULE we chose 4x4 S-box to be hardware efficient and also it should provide robust secure design.

The S-box used in GRANULE is a 4-bit to 4-bit S-box S: $F_2^4 \leftarrow F_2^4$. The two conditions which are essential for designing good S- box are mentioned below

Condition 1: Assume x is the input to S-box. Let ΔX, ΔY are the inputs and output differences respectively and ΔX, $\Delta Y \in F_2^4$ so $D_C(\Delta X, \Delta Y)$ is define as:

$$D_C(\Delta X, \Delta Y) = \#\{x \in F_2^4 \mid S(x) \oplus S(x \oplus \Delta X) = \Delta Y\}$$

where D_C represents differential cryptanalysis [9] [10]. This property helps us to create difference distribution table.

Condition 2: Assume a and b are the inputs and output constants and a, b$\in F_2^4$ so $L_C(a, b)$ is define as:

$$L_C(a, b) = \#\{x \in F_2^4 \mid a \bullet x = b \bullet S(x)\} - 8|$$

Where L_C represent linear cryptanalysis [9] [11], this property helps us to create linear approximation table

Where • denotes the inner product on F_2^4 , # indicate number of matches in linear approximation table for the constants a and b minus 8.

Complete Design Criteria of GRANULE S-box is as follows

1. For any non-zero input difference $\Delta X \in F_2^4$ and non-zero output difference $\Delta Y \in F_2^4$ we have

$$D_C(\Delta X, \Delta Y) = \#\{x \in F_2^4 \mid S(x) \oplus S(x \oplus \Delta X) = \Delta Y\} \leq 4$$

2. For any non-zero input difference $\Delta X \in F_2^4$ and non-zero output difference $\Delta Y \in F_2^4$ such that $Hw(\Delta X) = Hw(\Delta Y) = 1$, where Hw(a) denote Hamming weight of a, we have

$$SetD_C = D_C(\Delta X, \Delta Y) = \#\{x \in F_2^4 \mid S(x) \oplus S(x \oplus \Delta X) = \Delta Y\} = 0$$

Let $CarD_C$ denote the cardinality of $SetD_C$, we have $CarD_C = 2$.

3. For any non-zero input constant a $\in F_2^4$ and non-zero output constant b $\in F_2^4$, we have

$$L_C(a, b) = \#\{x \in F_2^4 | a \bullet x = b \bullet S(x)\} - 8| \leq 4$$

4. For any non-zero input constant $a \in F_2^4$ and non-zero output constant $b \in F_2^4$, such that Hw(a) = Hw(b) = 1, we have

$$SetL_C = L_C(a, b) = \#\{x \in F_2^4 | a \bullet x = b \bullet S(x)\} - 8| \neq 0$$

Let $CarL_C$ denote the cardinality of $SetL_C$, we have $CarL_C = 2$.

5. Bijective i.e. $S(x) \neq S(y)$ for all values of $x \neq y$.

6. No static point i.e. $S(y) \neq y$ for all values of $y \in F_2^4$.

$CarD_C = 0$ and $CarL_C = 8$ for the PRESENT S-box [5] and CarDC = 2 and $CarL_C = 2$ for the RECTANGLE S-box [5]. This indicates that RECTANGLE S-box shows good strength. In GRANULE cipher S-box we have $CarL_C = 2$ and $CarD_C = 2$ which also indicates that our S-box is robust in design.

B. Selection of the S-box of GRANULE

For selecting S-box we use following two definitions

Definition 1:- Affine equivalence [5] [12]

Two S-boxes are referred as affine equivalent if there exist bijective linear mappings X, Y and constants x, $y \in F_2^4$ such that $S'(a) = B(S(X(a) + x)) + y$ then the equivalence is called affine equivalence.

When S-boxes satisfies criteria 1, 3 and 5(*see section A*) then the S-box also meets Affine equivalence property.

Definition 2:-Permutation-then-XOR equivalence [5] [12]

Two S-boxes are referred as permutation-then-XOR equivalent if there exist 4-bit to 4-bit permutation matrices M_0, M_1 and constants x, $y \in F_2^4$ such that $S'(a) = M_1(S(M_0(a)+x))+y$. The equivalence is called PE equivalence for short.

PE equivalent S-boxes also satisfies criteria 1-5 when any S-box satisfies criteria 1-5(*see section A*).

Both these definitions are considered while designing GRANULE S-box.

C. Differential cryptanalysis

Differential cryptanalysis [9] [10] is the basic cryptanalytic attack and cipher has to resist this attack, which successfully applied on DES by Biham and Shamir. In encryption system this attack exploits high probability differences in the input and output and certain pairs of high probability input and output occurrences are used to recover round sub keys from last rounds. Linear component in encryption design produces certain outputs with probability 1, but this is not applicable for nonlinear component in design i.e. S-box in our design. Nonlinear component is examined by forming difference distribution table (DDT). DDT has high probability input and output differences and are used to form differential trails of the cipher. Table 3 represents Difference Distribution Table for GRANULE S-box**.** Maximum differential probability (P_d) in Difference Distribution Table (DDT) for our S-box is 4/16 = 1/4 = 2^{-2}.

TABLE 3
DIFFERENCE DISTRIBUTION TABLE FOR GRANULE S-BOX

	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	0	0	0	2	2	2	2	2	2	2	2
2	0	0	0	4	0	0	4	0	0	0	2	2	0	0	2	2
3	0	4	0	0	0	4	0	0	0	0	2	2	0	0	2	2
4	0	0	0	4	0	0	4	0	0	0	2	2	0	0	2	2
5	0	0	0	0	0	0	4	4	2	2	0	0	2	2	0	0
6	0	0	0	0	0	0	0	0	4	4	0	0	4	4	0	0
7	0	4	0	0	0	4	4	4	0	0	0	0	0	0	0	0
8	0	0	0	2	2	2	0	2	0	0	0	2	2	2	0	2
9	0	2	4	0	2	0	0	0	2	0	0	0	0	2	0	4
A	0	0	0	2	2	2	0	2	0	0	2	0	2	2	2	0
B	0	2	4	0	2	0	0	0	0	2	0	0	2	0	4	0
C	0	0	0	2	2	2	0	2	2	2	0	2	0	0	0	2
D	0	2	4	0	2	0	0	0	0	2	0	4	2	0	0	0
E	0	0	0	2	2	2	0	2	2	2	2	0	0	0	2	0
F	0	2	4	0	2	0	0	0	2	0	4	0	0	2	0	0

Number of differentially active S-boxes plays an important role for evaluation of security against differential cryptanalysis. Basic concept is to find the structure that maximizes number of active S-boxes. We have performed a rigorous computer-based search for differential trails and also to find out minimum number of active S-boxes in each round.

Maximum differential probability (P_d) of the 4-bit to 4-bit S-box of GRANULE is 2^{-2}. Table 4 shows the minimum numbers of active S-boxes calculated from differential trail.

TABLE 4
MINIMUM NUMBERS OF ACTIVE S-BOXES FROM DIFFERENTIAL TRAIL

#Round	# Min. active S-boxes
1	0
2	1
3	3
4	6
5	10
6	17
7	23

The theorem which shows the resistance of full round GRANULE against the Differential attack is described below.

Theorem 1:

GRANULE has total 25 rounds and for 21 rounds of GRANULE, we got minimum 69 active S-boxes. The total differential probability (P_d) is 2^{-138} for 21 rounds. Total chosen plaintext required is 2^{138} which is greater than 2^{64}.Our experimentation on GRANULE shows that 25 rounds are enough to deploy differential attack [20].

Proof:

For 7 rounds of GRANULE, it has minimum 20 differentially active S-boxes. Calculation of total number of active S-boxes for 21 rounds will be 23 x 3 = 69 differentially active S-boxes. For 21 rounds of GRANULE, the total

differential probability (P_d) is given as $(2^{-2})^{69} = 2^{-138}$.

Complexity of Differential attack evaluated by finding number of chosen plain text required (N_d), number chosen plain text required (N_d)

$$N_d = C/P_d$$

Where C = 1 and $P_d = 2^{-138}$, so the number of chosen plaintext required are

$$N_d = 1/2^{-138} = 2^{138}.$$

Table 5 represents differential trails of GRANULE cipher. The values in differential trails are represented in hexadecimal format. We have used computer based program to find out the differential trails. Non-zero input value to 'F' function in the table indicates number of active S-boxes.

TABLE 5
DIFFERENTIAL TRAIL FOR GRANULE

#Round	Input to F Function i.e. PT^1	Output of F Function xor With PT^0
1	0000 0000	4000 0000
2	4000 0000	00c0 6000
3	00c0 6000	4000 0e07
4	4000 0e07	0004 c260
5	0004 c260	3c05 ce87
6	3c05 ce87	8630 20cc
7	8630 20cc	5bca ac4b

D. Linear cryptanalysis

Another basic attack in block cipher is linear cryptanalysis [9] [11]. High probability occurrences of linear expressions containing plaintext bits, cipher text bits and sub key bits are used to mount Linear cryptanalysis attack. This attack also referred as known plain text attack. S-box used in our design examined by using Linear Approximation Table (LAT). Table 6 represents Linear Approximation table for GRANULE S-box. From this table we found best linear bias (ε_L) for nonlinear component. Best linear bias (ε_L) is given as $|P_L-1/2|$ where P_L is the linear probability. Best linear bias (ε_L) for GRANULE S-box is 2^{-2}. From differential trials, for 7 rounds of GRANULE we got minimum 20 active S-boxes. For the calculation of complexity of attack against linear cryptanalysis, we need to calculate maximum bias for number of rounds. Maximum bias can be given by using Matsui`s Piling-up lemma [9].

Optimizing S-box i.e. minimizing largest bias in LAT and finding structure that maximizes the number of active S-boxes can show resistance against linear cryptanalysis.

Matsui`s Pilling up Lemma:

For 'n' independent random binary variables $X_1, X_2 \ldots X_n$, the equation can be given as,

$$\Pr(X_1 \oplus \ldots \oplus X_n = 0) = 1/2 + 2^{n-1} \prod_{i=1}^{n} \varepsilon_{iL}$$

Or

$$\varepsilon_{(1L,2L,\ldots,nL)} = 2^{n-1} \prod_{i=1}^{n} \varepsilon_{iL}$$

Where $\varepsilon_{(1L, 2L,\ldots,nL)}$ represents the bias of $X_1 \oplus \ldots \oplus X_n = 0$.

Theorem1:

GRANULE has a minimum 60 active S-boxes for 21 rounds and maximum bias for 21 rounds is 2^{-61}.

Proof:

For 7 rounds it has total 23 active S-boxes. For our S-box, maximum bias is 2^{-2}. Maximum bias can be given as

$$2^{22} \times (2^{-2})^{23} = 2^{-24}$$

Maximum bias for 21 rounds of GRANULE is

$$2^{2} \times (2^{-24})^{3} = 2^{-70}$$

Complexity of attack can be evaluated by finding number of known plain text required. Number of known plaintext required (N_L) can be calculated as,

$$N_L = 1/(\varepsilon_L)^2$$

Where ε_L describes maximum bias for the GRANULE S-box, for 21 rounds, number of known plaintext required for the GRANULE is given as

$$N_L = 1/(2^{-70})^2 = 2^{140}$$

Known plaintext data requirement are 2^{140} which is more than the available data limit i.e. 2^{64}.

TABLE 6
LINEAR APPROXIMATION TABLE FOR GRANULE S-BOX

	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	-4	0	0	0	4	0	0	0	-4	0	0	0	-4
2	0	0	0	-4	-2	-2	2	-2	0	-4	0	0	-2	2	2	2
3	0	0	0	0	-2	-2	2	2	4	0	4	0	2	-2	-2	2
4	0	2	0	2	0	2	0	2	0	-2	4	2	-4	2	0	-2
5	0	-2	0	2	0	-2	0	2	-4	-2	0	-2	0	2	-4	2
6	0	-2	-4	-2	-2	4	-2	0	0	-2	0	2	2	0	-2	0
7	0	2	4	-2	-2	0	-2	0	0	2	0	2	2	4	-2	0
8	0	0	0	0	0	0	4	-4	0	0	0	0	0	0	-4	-4
9	0	4	0	0	4	0	0	0	0	-4	0	0	4	0	0	0
A	0	-4	0	0	2	-2	2	2	0	0	0	4	2	2	2	-2
B	0	0	0	0	-2	-2	-2	-2	-4	0	4	0	2	-2	2	-2
C	0	2	0	2	-4	-2	0	2	0	-2	-4	2	0	-2	0	-2
D	0	2	0	-2	0	2	4	2	-4	2	0	2	0	-2	0	2
E	0	2	-4	2	-2	0	2	0	0	2	0	-2	2	4	2	0
F	0	2	-4	-2	2	-4	-2	0	0	2	0	2	-2	0	-2	0

E. Biclique attack

Biclique attack [13] [14] is an extension of meet-in-the-middle attack.

In this paper, we have applied biclique cryptanalysis on GRANULE-80. Based on the attack results, comparison is also made with the most popular lightweight block ciphers like PRESENT, Piccolo and LED.

We have constructed a 3-dimensional biclique for round 26 ~ 31 of GRANULE-80. For these rounds the partial keys used is (RK_{26}, RK_{27}, RK_{28}, RK_{29}, RK_{30}, RK_{31}) which is described as follows.

$RK_{28} = K_{43}, K_{42}, \ldots\ldots K_{12}$
$RK_{29} = K_{12} \ldots K_0, K_{79}, \ldots\ldots K_{61}$
$RK_{30} = K_{61}, \ldots\ldots K_{30}$
$RK_{31} = K_{30}, K_{29}, \ldots\ldots K_0, K_{79}$

From above equations we found that by applying following sub keys (K_{63}, K_{62}, K_{61}) and (K_{33}, K_{32}, K_{31}) on above rounds gives biclique attack on the full GRANULE-80.

To construct the Δi-differential, we have considered the sub keys (K_{63}, K_{62}, K_{61}) and for the ∇j-differential, we have considered the sub keys (K_{32}, K_{31}, K_{30}). Let f be a sub-cipher from round 28 to round 31. Data complexity does not exceed 2^{40} as shown in Fig 3. The red blocks in Fig. 3. at 31st round represents the active blocks where each block is of 4 bits so the data complexity for GRANULE-80 does not exceed than 2^{40}.

A total computational complexity of our attack on GRANULE-80 is computed as follows.

$C_{total} = 2^{k-2d} (C_{biclique} + C_{precomp} + C_{recomp} + C_{falsepos})$.
$C_{total} = 2^{80-2 \times 3} (2 + 2^{2.81} + 2^{5.46} + 2^{2})$.
$C_{total} = 2^{79.85}$.

Where

$C_{biclique} = 2^{d+1}$(number of rounds in biclique/total rounds) $= 2^{3+1}(4/32) = 2$.

$C_{precomp} = 2^{d}$(number of rounds in precomputation/total rounds) $= 2^{3}(28/32) = 2^{2.81}$.

$C_{recomp} = 2^{2d}$(number of active S- boxes in precomputation/total number of maximum active S-boxes) $= 2^{2 \times 3}(178/32 \times 8) = 2^{5.46}$.

$C_{falsepos} = 2^{2d-(\text{no. of matching bits})} = 2^{2 \times 3 - (4)} = 2^{2}$.

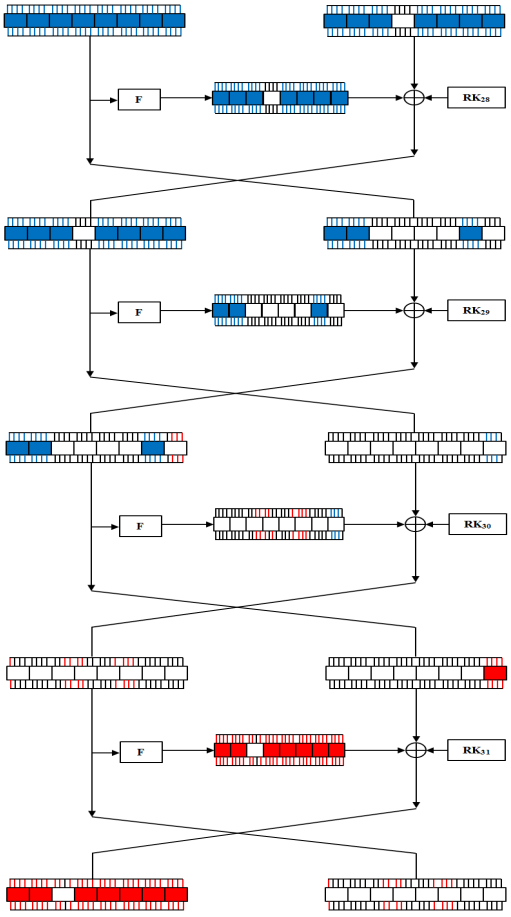

Fig. 3: Three - dimensional biclique for GRANULE-80.

Fig.4 (a) and (b) represents the Re-computations in forward and backward directions for GRANULE-80.

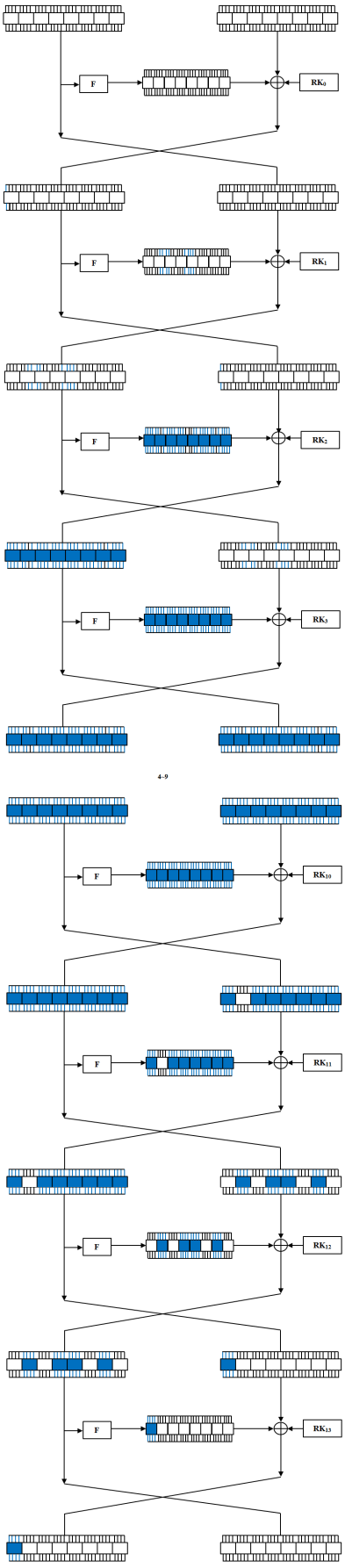

Fig. 4 (a) Recomputations in forward directions for GRANULE-80

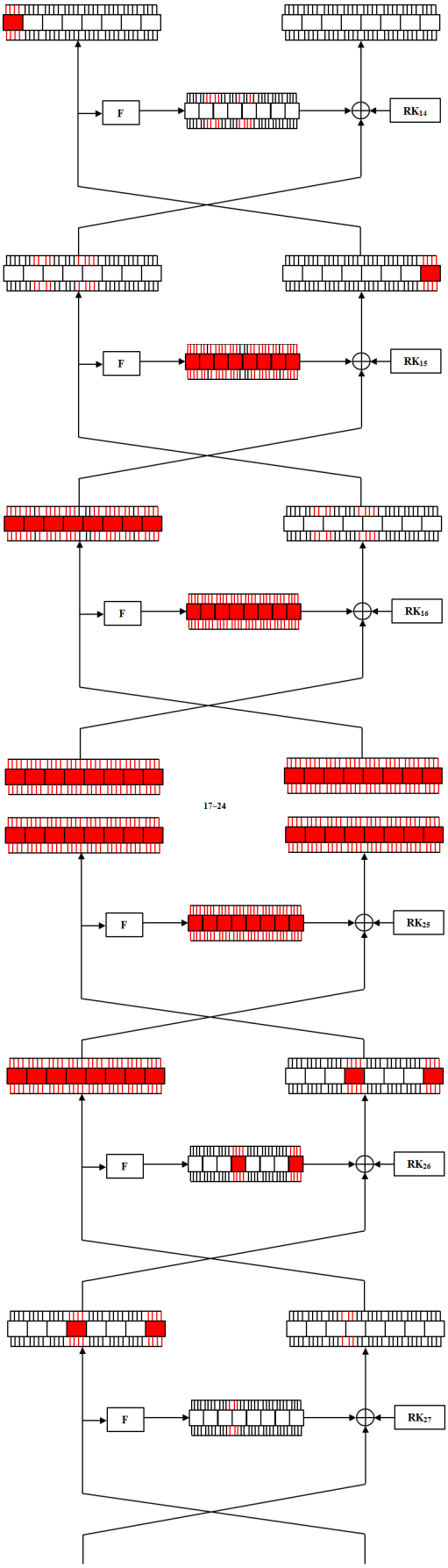

Fig. 4 (b) Recomputations in backward directions for GRANULE-80

F. Zero-Correlation attack

Zero-correlation attack [15] [16] is the extension of Linear cryptanalysis. The block ciphers should resist zero correlation attack. Zero-correlation Attack is based on linear approximations with a correlation value of zero. Zero-correlation attack is considered as a counter part of impossible differential cryptanalysis in domain of linear cryptanalysis. We have applied matrix method [16] to mount Zero-correlation attack which is explained below. Following three Lemmas are used to find contradiction.

Lemma 2: XOR approximation

Either the three linear selection patterns at an XOR ⊕ are equal or the correlation over ⊕ is exactly zero.

Lemma 3: Branching approximation

Either the three linear selection patterns at a branching point • sum up to 0 or the correlation over • is exactly zero.

Lemma 4: Permutation approximation

Over a permutation φ, if the input and output selection patterns are neither both zero nor both nonzero, the correlation over φ is exactly zero.

1) The Matrix Method [16]

Impossible differential characteristic can be constructed by using Miss-in-the-middle approach. Contradiction in middle is formed by two differential paths with probability.

Matrix method is used to find linear approximation with correlation zero which is mentioned below,

The linear masks applied to the words can be of the following five types:

1. Zero mask denoted by 0,
2. An arbitrary non-zero mask denoted by $\overline{0}$,
3. Non-zero mask with a fixed value a,
4. The exclusive-or of a fixed non-zero mask a and an arbitrary non-zero mask, denoted by $\bar{a}$,
5. Any other mask is denoted by ∗.

TABLE 7
ARITHMETIC RULES MULTIPLICATION BY 0, 1 AND 1F.

	0	**1**	**1F**
0	0	0	0
$\overline{\mathbf{0}}$	0	$\overline{0}$	$\overline{0}$
a	0	a	$\overline{0}$
$\overline{\mathbf{a}}$	0	$\overline{a}$	∗
∗	0	∗	∗

The matrix shows that how a linear mask of each output word is affected by the linear mask of an input word. Table 7 and 8 illustrate arithmetic rules for multiplication and addition.

TABLE 8
ARITHMETIC RULES ADDITION BETWEEN TWO MASK

+	**0**	$\overline{\mathbf{0}}$	**a**	$\overline{\mathbf{a}}$	∗
0	0	$\overline{0}$	a	$\overline{a}$	∗
$\overline{\mathbf{0}}$	$\overline{0}$	∗	$\overline{a}$	∗	∗
b	b	$\overline{b}$	a+b	∗	∗
$\overline{\mathbf{b}}$	$\overline{b}$	∗	∗	∗	∗
∗	∗	∗	∗	∗	∗

2) Zero-Correlation for 4 rounds of GRANULE

For (000a000000000000) → (0000000000000000b) has correlation exactly zero for which the values a and b are non-zero. We found contradiction for 6 rounds of GRANULE.

TABLE 9
TRAILS FOR ZERO-CORRELATION FOR GRANULE CIPHER

#Rounds	PT_1	PT_0
0	0 0 0 0 0 0 0 0 0 0 0 0 a a a a	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 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 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 a a a a
	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2	0 0 0 0 0 0 0 0 0 0 0 0 a a a a	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 $\overline{0}$ $\overline{0}$ 0 0 0 0 0 0 0 0 $\overline{0}$ $\overline{0}$
3	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**	**0 0 0 0 $\overline{0}$ $\overline{0}$ 0 0 0 0 $\overline{0}$ $\overline{0}$ a a a a**
	0 0 0 0 $\overline{0}$ $\overline{0}$ 0 0 0 0 0 0 0 0 0 0 $\overline{0}$ $\overline{0}$	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
3	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 $\overline{0}$ $\overline{0}$**	**0 0 0 0 $\overline{0}$ $\overline{0}$ 0 0 0 0 0 0 0 0 0 0**
	0 0 0 0 0 0 0 0 0 0 0 0 $\overline{0}$ $\overline{0}$ b b b b	**0 0 0 0 0 0 0 0 0 0 0 0 $\overline{0}$ $\overline{0}$ 0 0 0 0**
4	0 0 0 0 $\overline{0}$ $\overline{0}$ 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0 0 0 $\overline{0}$ $\overline{0}$ 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 b b b b
5	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0 0 0 0 0 b b b b	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 b b b b

G. Avalanche Effect [17]

When a single bit change in the input changes the output considerably, this results in an avalanche effect. For example by flipping a single bit in the input or in a key could change half of the bits in cipher text. Cipher with good avalanche effect has higher probability to resist all possible types of attacks.

In case of robust design of block ciphers, drastic change in the cipher text is visible when a small change in the key or the plaintext takes place. The poor randomization occurs when a block cipher does not show the avalanche effect to a significant degree.

Table 10 summarizes the Avalanche effect for GRANULE cipher.

TABLE 10
AVALANCHE EFFECT FOR GRANULE-128

Plaintext	0000 0000 0000 0000	# Bits Change
Key	0000 0000 0000 0000 0000 0000 0000 0000	--
Ciphertext	7d83 43cf fb86 7dbd	
Key	8000 0000 0000 0000 0000 0000 0000 0000	37
Ciphertext	e2b5 46b2 271d 0a5d	
Key	0000 0000 4000 0000 0000 0000 0000 0000	40
Ciphertext	b70c 8d71 55e9 bc82	

H. Algebraic attack

Algebraic attack [18] is more successful when it gets applied on stream cipher rather than block cipher. For any four bit S-box is described by a minimum of 21 equations in 8 input/output variables. Entire cipher can be explained by a = x × 21 quadratic equations in b = x × 8 variables. Where x represents number of S-boxes used in encryption algorithm and in key scheduling algorithm of the cipher. GRANULE uses 8 S-boxes in encryption algorithm for single round, so for 32 rounds of GRANULE it has 32 × 8 = 256 S-boxes. While in 128-bit key scheduling for single round of GRANULE uses 4 S-boxes, so for 32 rounds 32 × 2 = 64 S-boxes are used. Number of quadratic equations can be formed and is given as

$$a = (256 + 64) \times 21 = 6720$$

And number of variables can be given as

$$b = (256+64)_{\times} 8 = 2560$$

By applying same method for 80-bit key schedule algorithm there will be 6720 number of quadratic equations can be formed in 2560 variable.

I. Key schedule attacks

For designing of key scheduling, there are no specific established guidelines. Wide varieties of key scheduling techniques are possible. Related key attack [19] and slide attack [15] can give weakness in key scheduling algorithm. Related Key attack is referred as chosen key attack. Related key attack successfully applied on reduced round AES-256 [21]. Slide attack is independent on the number of rounds of the cipher.

We have adopted PRESENT style key-scheduling algorithm and no key related attack has been found on PRESENT key scheduling algorithm. Both these attack depends on finding recognizable relationships between different sets of sub keys. In GRANULE key scheduling, nonlinear operations are used and 5-bit round constant is added which thwart the key schedule attacks.

J. Key Collision Attack

Key collision attack [7] is used to create message with complexity of $2^{k/2}$, where k denotes length of key size. This attack can be mounted on any block cipher and it depends only on key length, regardless of its key scheduling algorithm design. For GRANULE cipher key scheduling under 128-bit key complexity of created message is $2^{128/2} = 2^{64}$ which is sufficient to prove resistance against key collision attack.

IV. SECURITY COMPARISON WITH STANDARD ALGORITHM

In this section we have compared the security analysis of GRANULE with the other standard lightweight ciphers. The comparison is represented in Table 11 and 12. Table 11 compares the linear complexity and differential complexity by considering the number of active S-boxes for particular rounds.

TABLE 11
LINEAR AND DIFFERENTIAL ATTACK COMPARISON

Cipher Name	**GRANULE**	**PRESENT**	**L-Block**	**FEW**	**PICCOLO**
#Rounds	21	25	15	27	30
# Active S-box	63	50	32	45	30
#Known Plaintext	2^{140}	2^{102}	2^{66}	2^{90}	2^{120}
#Chosen Plaintext	2^{138}	2^{100}	2^{64}	2^{90}	2^{120}
Reference	This Paper	[1]	[23]	[24]	[3]

Table 12 compares the data complexity and computational complexity of GRANULE with other ciphers.

TABLE 12
BICLIQUE ATTACK COMPARISON

Cipher Name	**Rounds**	**Data Complexity**	**Computational Complexity**	**Reference**
GRANULE-80	Full(32)	2^{40}	$2^{79.85}$	This Paper
PRESENT-80	Full(31)	2^{23}	$2^{79.54}$	[14]
PRESENT-128	Full(31)	2^{19}	$2^{127.42}$	[14]
PICCOLO-80	Full(25)	2^{48}	$2^{79.13}$	[14]
`PICCOLO-128	Full(31)	2^{24}	$2^{127.35}$	[14]
LED-64	Full(48)	2^{64}	$2^{63.58}$	[14]
LED-80	Full(48)	2^{64}	$2^{79.37}$	[14]
LED-96	Full(48)	2^{64}	$2^{95.37}$	[14]
LED-128	Full(48)	2^{64}	$2^{127.37}$	[14]

Table 13 compares the S-box design consideration comparison with the lightweight ciphers. GRANULE S-box have CAR_{DC} = 2 and CAR_{LC} = 2 which illustrate that GRANULE cipher S-box is robust in design and provides good security than other lightweight ciphers.

TABLE 13
S-BOX DESIGN CONSIDERATION

Cipher Name	Max. Val. in DDT	Max. Val. in LAT	CAR_{DC}	CAR_{LC}
GRANULE	4	4	2	2
PRESENT	4	4	0	8
RECTANGLE	4	4	2	2
TWINE	4	4	5	7

V. HARDWARE AND SOFTWARE PERFORMANCE OF GRANULE

GRANULE is designed to give optimum performance both on hardware and software. Its compact structure results in very small footprint area in hardware and less flash size in software. GRANULE has least power consumption as compared to compact ciphers like PRESENT, LED [22]. In hardware we have tried and achieved a reduced pin count (Input/Output) for cipher GRANULE. For software performance of GRANULE cipher we have used 32 bit ARM 7, LPC2129 processor [23]. For hardware we have used Verilog on XILINX ISE Design Suit 14.2 version. For calculation of power we have used XPower Analyzer of Xilinx. Clock frequency used is 10 MHz. GEs are calculated with the standard cell library based on UMCL180 0.18µ logic process (UMCL18G212T3) [24]. Memory size required for GRANULE on 32 bit processor is 2104 bytes as Flash memory and 1256 as RAM memory. All other ciphers are written in embedded C and implemented on 32 bit processor. GRANULE turns out to be the cipher of least memory requirement as compared to other existing cipher. Fig 3 shows the comparison of ciphers on memory requirement.

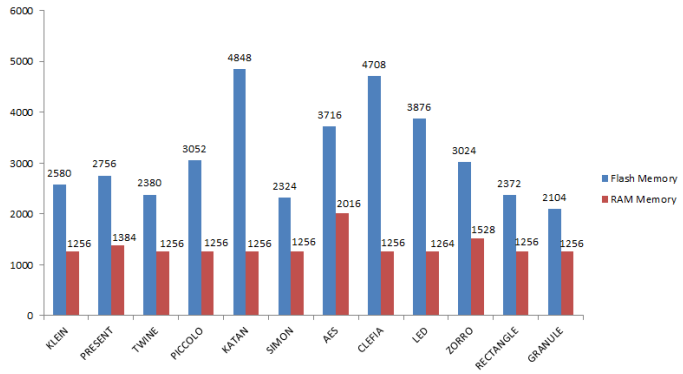

Fig. 5: Memory Comparison of Standard algorithms with GRANULE implemented on LPC2129

Fig. 5 represents memory comparison of the lightweight ciphers with GRANULE. GRANULE needs 2104 bytes of Flash memory which is least as compared to the existing ciphers shown in Figure. Its small code size results in less footprint area which also results in less GEs.

Fig. 6 shows datapath for GRANULE cipher. We have implemented datapath by using round based architecture.

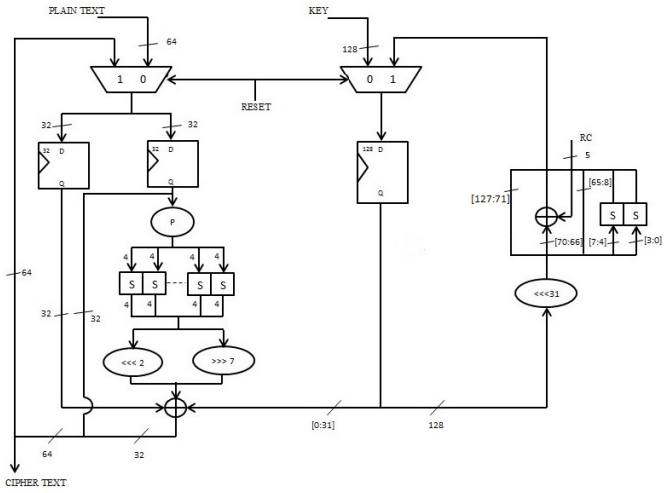

Fig. 6: Datapath for GRANULE for 64-bit plaintext and 128-bit key

Fig. 7 shows comparison of other existing ciphers with GRANULE ciphers on GEs [23].

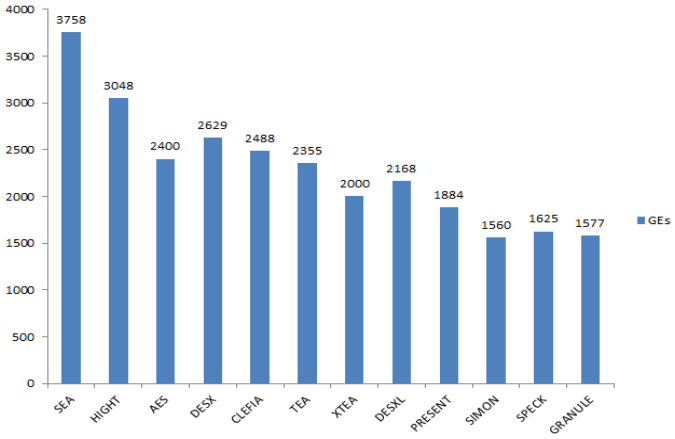

Fig. 7: GEs Comparison of Standard algorithms vs GRANULE for UMCL180 standard cell library.

Table 14 shows gate count for UMCL180 standard cell library which we have used to count the total gate equivalents [24].

TABLE 14
GATE COUNT OF UMCL18G212T3 LIBRARY

Standard Cell	Process	GE
NOT	0.18µm	*0.67*
AND	0.18µm	*1.33*
XOR	0.18µm	*2.67*
D-Reg.	0.18µm	*6*

Table 15 shows the area requirement for GRANULE. GRANULE needs 1577 GEs for 128-bit key and 1288.23 for 80-bit key length.

TABLE 15
CALCULATION OF GES FOR GRANULE

DATA Layer	GEs	KEY Layer	GEs
D Reg.	*384*	K Reg.	*768*
P Layer	*0*	Shift Operator	*0*
S-box Layer	*192*	S-box	*48*
XOR	*85.44*	XOR RC	*13.35*
		Key XOR	*85.44*
Total	*661.44*	Total	*914.79*
Total number of gates required for 128-bit key = *1576.23 = 1577*			

The consumption of power is mainly depends on switching frequency and the technology used. We analyzed GRANULE on 10MHz frequency. Dynamic power consumption of GRANULE is 27mW which us lesser than PRESENT. PRESENT consumes 38mW of power.

Table 16 shows the comparison of lightweight ciphers with GRANULE based on parameters like execution time, throughput and number of cycles required to convert plain text to cipher text. Throughput computed at 12 MHz Frequency on software platform.

TABLE 16
COMPARISON WITH RESPECT TO THROUGHPUT, EXECUTION TIME AND NUMBER OF CYCLES

Ciphers	Block Size	Key Size	Execution Time (In uSec)	Throughput (In Kbps)	No. of Cycles
SP NETWORK					
ZORRO	128	128	913.21	140	10958.52
KLEIN	64	96	887.51	72	10650.12
HUMMINGBIRD-2	16	128	316.51	51	3798.12
PRESENT	64	128	2648.65	24.16	31783.8
FEISTEL STRUCUTRE					
SPECK	64	128	49.02	1305	588.24
SIMON	64	128	105.67	605	1268.04
PICCOLO	64	128	227.68	281	2732.16
GRANULE	**64**	**128**	**296.1**	**216.14**	**3553.2**
CLEFIA	128	128	1048.01	122	12576.12
TWINE	64	128	592.87	108	7114.44

Table 17 and 18 shows comparison of GRANULE cipher with standard algorithm.

TABLE 17
COMPARISON WITH STANDARD ALGORITHMS ON BASIS OF POWER

	PRESENT	RECTANGLE	LED
GRANULE	-28.94%	-12.90%	-73%

TABLE 18
COMPARISON WITH STANDARD ALGORITHMS ON BASIS OF FLASH MEMORY

	GRANULE
PRESENT	-26.65%
RECTANGLE	-11.29%
LED	-45.71%
TWINE	-11.60%
PICCOLO	-31.06%
SIMON	-9.46%

VI. CONCLUSION

In this paper we present a GRANULE, an ultra-lightweight cipher. GRANULE has compact design which results in less foot print area and less power consumption. GRANULE performs efficiently both on hardware and software platform. We have shown resistance of GRANULE mainly against Linear, Differential, Biclique, zero correlation, Meet in the middle, Key schedule and Key collide attacks. During the cipher design we have done extensive computer search for good S-box, minimum number of active S-boxes and calculation of hamming weight for a specific entries in LAT and in DDT. GRANULE has a strong S-box and robust permutation layer which prevents a cipher design to undergo clustering of linear and differential trails. In GRANULE, we have achieved a less gate count so that it can be implemented for any small scale embedded systems. Through GRANULE, we have tried designing a smallest block cipher in terms of footprint area and it is competitive with the SIMON and SPECK cipher design. Applications like RFID tags, Wireless sensor nodes where GEs and power consumption plays a crucial role, we believe GRANULE is the best design suited for such applications.

TEST VECTORS(FOR 128 BIT KEY)

Plain Text:	0000 0000 0000 0000
Key:	00000000000000000000000000000000
Cipher Text:	7d8343cffb867dbd

Plain Text:	0123 4567 89ab cdef
Key:	0123456789abcdef0123456789abcdef
Cipher Text:	b3684a657634012f

References

[1] A. Bogdanov, G. Leander, L.R. Knudsen, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT - An Ultra-Lightweight Block Cipher," In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems — CHES 2007*, Vol. 4727 in *LNCS*, pages 450-466, Springer Berlin Heidelberg, 2007.

[2] Daniel Engels, Markku Juhani O. Saarinen, Peter Schweitzer, and Eric M. Smith: "The Hummingbird-2 Lightweight Authenticated Encryption Algorithm" pages 19-31, Volume-7055, Springer Berlin Heidelberg, 2012.
[3] Beaulieu, R., Shors, D., Smith, J., Clark, S.T., Weeks, B., Wingers, L, "The SIMON and SPECK Families of Lightweight Block Ciphers" *Cryptology ePrint Archive*, Report 2013/404, Available at http://eprint.iacr.org
[4] Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai,"Piccolo: An Ultra-Lightweight Blockcipher", pages 342-357, Volume-6917 Springer Berlin Heidelberg, 2011.
[5] Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., Verbauwhede, "RECTANGLE: A bit-slice ultra-lightweight block cipher suitable for multiple Platforms" *Cryptology ePrint Archive*. Available at https://eprint.iacr.org/2014/084.pdf
[6] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, "TWINE: A Lightweight, Versatile Block Cipher" *Cryptology ePrint Archive*. Available at www.nec.co.jp/rd/media/ code/research/images/twine_LC11.pdf
[7] R. Anderson, E. Biham and L. Knudsen, "Serpent: a proposal for the advanced encryption standard," *NIST AES proposal 174*, June 1998. available at ftp://dijkstra.urgu.org/crypto/Serpent/v1/res/serpent.pdf
[8] National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES)," Federal Information Processing Standards Publication 197, November 26, 2001. Available at http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.
[9] Howard M. Heys, "A Tutorial on Linear and Differential Cryptanalysis" ,http://citeseer.nj.nec.com/443539.html
[10] Biham, E., Shamir, A. "Differential Cryptanalysis of DES-like Cryptosystems" Journal of Cryptology, vol. 4, no. 1, pp. 372, 1991
[11] Matsui, M., "Linear Cryptanalysis Method for DES Cipher" Advances in Cryptology EUROCRYPT 1993, LNCS 765, pp. 386-397, Springer-Verlag, 1994
[12] Leander, G., Poschmann, A, "On the Classification of 4 bit S-boxes" In: Carlet, C., Sunar, B. (eds.) WAIFI 2007. LNCS, vol. 4547, pp. 159–176. Springer, Heidelberg (2007)
[13] Jeong, K., Kang, H., Lee, C., Sung, J., Hong, S.: "Biclique Cryptanalysis of Lightweight Block Ciphers PRESENT, Piccolo and LED", Cryptology ePrint Archive, Report 2012/621.
[14] A. Bogdanov, D. Khovratovich and C. Rechberger,: "Biclique Cryptanalysis of the Full AES", ASIACRYPT 2011, LNCS 7073, pp. 344–371, IACR, 2011.
[15] Bogdanov, A., Rijmen, V.: "Zero Correlation Linear Cryptanalysis of Block Ciphers" IACR Eprint Archive Report 2011/123 (March 2011)
[16] Hadi Soleimany and Kaisa Nyberg. "Zero-correlation linear cryptanalysis of reduced-round lblock" Cryptology ePrint Archive, Report 2012/570, 2012. http://eprint.iacr.org/.
[17] Z. Shi and R. B. Lee, "Bit permutation instructions for accelerating software cryptography," In Proceedings of the IEEE International Conference on Application Specific Systems, Architectures and Processors (ASAP 2000), pages 138-148, July 2000.
[18] M. Albrecht, C. Cid. "Algebraic techniques in differential cryptanalysis" FSE 2009, LNCS, vol. 5665, pp. 193-208. Springer, Heidelberg. 2009
[19] E. Biham. "New Types of Cryptanalytic Attacks Using Related Keys". Proceedings of Eurocrypt 93, LNCS. vol. 765, pp 398-409, Springer-Verlag. 1994
[20] A. Biryukov and D. Wagner, "Advanced Slide Attacks", Proceedings of Eurocrypt 2000, LNCS. vol. 1807, pp. 589-606, Springer-Verlag. 2000
[21] A. Biryukov, D. Khovratovich and I. Nikoli´c. "Distinguisher and Related-Key Attack on the Full AES-256". http://eprint.iacr.org/2009/241. 2009
[22] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED Block Cipher," In *Cryptographic Hardware and Embedded Systems CHES 2011, LNCS*, Vol. 6917/2011, pages 326-341, Springer, 2011.
[23] Gaurav Bansod, Nishchal Raval, Narayan Pisharoty, "Implementation of a New Lightweight Encryption Design for Embedded Security", IEEE Transactions on Information Forensics and Security, Issue 1,Vol 10,Jan 2015.
[24] A. Poschmann. "Lightweight cryptography: cryptographic engineering for a pervasive world," In PhD Thesis, Faculty of Electrical Engineering and Information Technology, Ruhr-University Bochum, Germany, February 2009.
[25] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. S. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee, "HIGHT: A New Block Cipher Suitable for Low-Resource Device," In L. Goubin and M. Matsui, editors, Cryptographic Hardware and Embedded Systems — CHES 2006, Vol. 4249 in LNCS, pages 46–59, Springer Berlin Heidelberg, 2006.
[26] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit blockcipher CLEFIA," In Fast Software Encryption- FSE'07, Vol. 4593 in LNCS, pages 181-195, Springer Berlin Heidelberg, 2007.
[27] D. Wheeler and R. Needham, "TEA, a Tiny Encryption Algorithm," In B. Preneel, editor, *Fast Software Encryption — FSE 1994*, Vol. 1008 in *LNCS*, pages 363–366, Springer Berlin Heidelberg, 1995.

Gaurav Bansod received the P.hD degree from Symbiosis International University, Pune. He has received his M.Tech. Degree in Embedded Sytems from Jawaharlal Nehru Technological University, Hyderabad, India in 2008. He is having total 8 years of teaching experience and has worked with 2 to 3 Universities across India. Currently he is also working as an Associate Professor in Pune Institute of Computer Technology(PICT), Pune. He has publications in reputed IEEE Transactions on Information Forensics and Security and also in reputed SCI Indexed journals. His research area includes low power cryptographic design, embedded system and hardware and software design.

Narayan Pisharoty received B.E. degree from IIT Bombay in 1966, M.Tech. degree from IIT Kanpur in 1968 and Ph.D from Carnegie Mellon University, Pittsburgh, USA in 1971. He held the post of Managing Director in Systech Ltd, Pune from 1972 to 2004 and Business Development Consultant in Persistent Systems Ltd from 2008 to 2010. Currently he is the Research Mentor for Engineering at Symbiosis International University, Pune, India and a Professor in the Electronics & Telecommunication Department of SIT. He has published many papers in reputed journals including IEEE transactions on Biomedical Engineering, IEEE transactions on Information Forensics and Security. He is currently guiding 7 Ph.D. students on different topics like matrix topology for multimode converters, UBW microwave antenna, and performance enhancement using dynamic partial reconfiguration. His research area includes RFID Applications, Alternate energy sources and Applications of microcontrollers in Agriculture.

Abhijit S Patil received B.E. Degree in Electronics from Shivaji University, Kolhapur in 2012. He is currently pursuing the M.Tech. degree in Electronics and Telecommunication from Symbiosis Institute of Technology, Pune. He had work experience in Quality Analyst Field from year 2012 to 2013 in "EMERSON NETWORK POWER". His research interests are in the embedded security system, lightweight cipher, embedded automotive systems and embedded real time system.