Copyright ⓒ 2021 KSII

# LCB: Light Cipher Block An Ultrafast Lightweight Block Cipher For Resource Constrained IOT Security Applications

**Siddhartha Roy[1*], Saptarshi Roy[2], Arpita Biswas[1] and Krishna Lal Baishnab[1]**
[1] Department of Electronics and Communication, National Institute of Technology Silchar
Silchar, Assam 788010 India
[e-mail: sroy7710@gmail.com; arpita.abiswas.biswas8@gmail.com; klbaishnab@gmail.com]
[2] Department of Computer Science and Engineering, National Institute of Technology Rourkela
Rourkela, Odisha 769001 India
[e-mail: saproy7710@gmail.com]
[*]Corresponding author: Siddhartha Roy

*Received November 24, 2020; revised July 26, 2021; accepted September 12, 2021; published November 30, 2021*

## Abstract

In this fast-paced technological world, the Internet of Things is a ground breaking technology which finds an immense role in the present electronic world which includes different embedded sensors, devices and most other things which are connected to the Internet. The IoT devices are designed in a way that it helps to collect various forms of data from varied sources and transmit them in digitalized form. In modern era of IoT technology data security is a trending issue which greatly affects the confidentiality of important information. Keeping the issue in mind a novel light encryption strategy known as LCB is designed for IoT devices for optimal security. LCB exploits the benefits of Feistel structure and the architectural benefits of substitution permutation network both to give more security. Moreover, this newly designed technique is tested on (Virtex-7) XC7VX330T FPGA board and it takes much little area of 224 GE (Gate Equivalent) and is extremely fast with very less combinational path delay of 0.877 ns. An in-depth screening confirms the proposed work to promise more security to counter cryptographic attacks. Lastly the Avalanche Effect (AE) of LCB showed as 63.125% and 63.875% when key and plaintext (PT) are taken into consideration respectively.

***Keywords:*** Avalanche Effect, Feistel structure, FPGA, Internet of things, Light encryption, Substitution Permutation Network (SPN).

## 1. Introduction

Internet of things, an enormously focused topic for research which has reformed the human life by its capability to provide uncountable number of advanced computational provisions. In IoT, the 'things' denote a physical entity which has an exclusive identifier with automatic processing and data transmitting capacity for a particular task. It can seamlessly build an excellent and universal link with the physical matters surrounding the digital world and us. Tremendous leap in the advancement of IoT dispositions and solutions is the greatest good the globe is seeing nowadays. The present world is changing rapidly and is becoming smart with the blessing of IoT through various challenging areas of application like mobile edge computing [1], [2], intelligent vehicular transport [3], [4] intelligent automation in buildings and houses [5], intelligent agriculture and farming [6], intelligent energy and IoT enabled smart grids [7], power constrained systems [8], healthcare systems [9], [10] etc. IoT is growing at a very fast pace in the world of information and data but it's sustained growth can suffer a lot due to the absence of sufficient privacy and security measures. Main function of IoT devices is to collect various kinds data sensed through different sensors among which few are very sensitive and confidential and should not be disclosed. According to survey of various industries, the most important concern is the data security in today's IoT users [11]. Therefore, integration of many features in terms of security for hardware and software both is utmost important to enhance the security of IoT device and to make sure that the vital sensed data are protected. For this reason, applying the concept of cryptography is a good option for IoT device's physical constraints. Before sending to the cloud server in the back end cryptographic algorithms are used in IoT devices by using authentication mechanisms like in [12] to maintain integrity of data and prevent hackers to sniff the data. Although data security based on cryptography is finding importance at a very increasing rate, but still it is more of a challenge to fit various cryptographic algorithms with huge power and area overheads in an IoT device. An efficient and suitable key management is required for encryption of IoT devices as security is compromised if the key management is poor. IoT devices are named so because they are meant to be designed with low storage. Also, IoT devices are run on battery and hence power requirement must be low. Hence symmetric key cryptography is preferred to design IoT devices because they take less power, space, bandwidth and complexity when compared to asymmetric key cryptography.

Over the past decade to achieve ease of implementation in hardware and software the block cipher which is a type of secret key cryptographic approach proved to be most useful due to high diffusion and error propagating property. In comparison with the stream cipher the block cipher uses very low hardware resources. The old block cipher for example the Advanced Encryption Standard (AES) requires about 3400 Gate Equivalent (GE) of area [13]. It is very hard for traditionally designed block ciphers for wireless equipment to be implanted in the device. Gate equivalent is a unit by which the complexity of manufacturing-technology-independent digital circuits is measured. In complementary metal oxide semiconductor (CMOS) technology of today, a gate equivalent is the covered area by silicon of a simple 2-input NAND gate which is the fundamental technology dependent unit area. In a particular circuit the gate equivalent count gives a measure of complexity with the help of which the corresponding area of the silicon can be calculated for a particular manufacturing process.

Traditional Feistel structured ciphers require huge amount of time and power when making block ciphers because they go through many numbers of round functions as compared to the substitution-permutation network (SPN). For this reason, Feistel structure and Substitution Permutation Network both are used combinedly in many studies. When combined together the

structures yield small and simple round functions for less use of storage and energy with a fast diffusion process. With vulnerabilities and difficulties taken into account a light symmetrical key cryptography method is proposed here for light IoT devices overcoming all the shortcomings in previous works.

This new design discussed here exploits the joint benefits of Feistel and Substitution Combination Network (SPN) together to achieve a fast diffusion process. In addition, the circular key management strategy is also compatible with the encryption method. Additionally, an energy efficient hardware which uses recent technological advancements with less nodal power is developed, which uses more gate operations for providing sufficient protection with lower power usage. A FPGA board is used as a testing platform of the prototype model to verify its correctness of application. The proposed work will find it's true potential when applied in the fields of smart agriculture and farming, healthcare systems and other domains too where encrypted and secure transmission of data is of top priority. **Table 1** listed here contains the detailed information of some practical sensors mentioned in [56] with their area of application in which the designed LCB can be implanted for improved level of security in sensing capacity. If correctly configured, this LCB can find its application in IoT devices and wide range of sensors and will not be restricted in these mentioned sensors. Various analyses are shown here to support the strength and feasibility of the designed scheme.

The remaining paper is ordered as follows: Section 2 discusses relevant works in related fields in details. Section 4 contains a detailed explanation about the working and mechanism of the proposed light cryptography algorithm. Section 5 brings up a discussion about the results of different analyses performed in the test environment. Section 6 finally concludes the paper.

**Table 1.** Details of some practical sensors [56]

| Sensor Name | Data Width | Area of Application | Function |
|---|---|---|---|
| MAX30100 | 16 bits | Healthcare | Detects pulse oximetry and heart-rate signals. |
| MPXV7002DP | 14 bits | Agriculture | Determines air speed |
| BMP180 | 19 bits | Agriculture | Determines atmospheric pressure |
| CJMCU-6701 | 12 bits | Healthcare | Measures skin electrical response (GSR) |
| AK9750 (Infrared Sensor) | 16 bits | Agriculture/ Healthcare | Human Sensing |
| AD8232 | 8/16 bits | Healthcare | Measures ECG signal |
| MAX30205 | 16 bits | Healthcare | Measures human body temperature |
| EMG-muscle sensor | 8/16 bits | Healthcare | Measures muscle's electrical activity |
| CS526-L | 8 bits | Agriculture | Checks PH-balance |

## 2. Key Challenges and Motivation

The main problems faced during implementation of standard cryptographic algorithms in IoT devices are given below [14]:

- Inadequate memory (ROM, RAM, registers)
- Minimized computational power
- Less physical area for implementation
- Little power of battery
- Response in real time

IoT devices like sensors and RFIDs mostly have limitation in size [15], memory, processing power [15] with the need of speedy, accurate results for real-time applications with necessary security becomes a very challenging task altogether [16]. Numerous risks are faced by the designers of IoT devices which includes protection of data [17] and energy capacity [18]. The above points are enough for finding motivation in pursuit of designing a new, robust lightweight block cipher. The design is attempted to make use of as less number of Gate equivalent area (GE) possible by minimizing the logical operations throughout the encryption rounds which make unnecessary propagation delays, larger area consumption and also least possible computing power yet maintaining decent security of data. All the other proposed widely used lightweight block ciphers use much greater number of logical operations like XOR along with many rounds which eventually increases the resource utilization.

Because of their simplicity, lightweight cryptography is vulnerable to attacks [19]. Different attacks like linear cryptanalysis [20], algebraic attack along with differential attack [21] and saturation attack [22] weakens PRESENT cipher mostly due to the use of small S-Boxes. This becomes a point for getting strong motivation for designing a new kind of light cipher block with increased security by introduction of strong F-function composed of 8-bit P-boxes and 16-bit L-boxes which overcomes the shortcomings of use of 4-bit S-boxes in PRESENT.

## 3. Literature Survey

Power, size of data and computational cost are the primary concerns which are to be taken care of by light cryptographic algorithm and reducing them to minimal levels. For this reason, the main objective to make it lightweight for specific small IoT devices the parameters like power usage, chip area, memory requirement etc. [23] should be in the designer's focus. In spite of being lightweight, weak security led to the disposal of these algorithms. Few popular algorithms are still followed in various application areas like Tiny Encryption Algorithm (TEA) [24], HIGHT [25], mCRYPTON [26], CLEFIA [27], PRESENT [28], KTANTAN and KATAN [29], GOST [30], PICCOLO [31], SPONGENT [32], Light Encryption Device [33], PHOTON [34], KLEIN [35], PRINCE [36], Low power Encryption Algorithm (LEA) [37], MIDORI [38], PRIDE [39], SIMON and SPECK [40].

Few very common new lightweight ciphers blocks have been taken into discussion [41] which announced the Substitution Permutation Feistel Network (SFN) method of encryption by combining both Feistel structure and Substitution Permutation Network (SPN) together. The aforementioned method used 64-bit input with a key length of 96 bits where the last 32 bits of the key functioned as a control signal and the left 64 bits served the purpose of expansion of key and encryption process in round manner. In that control signal of 32 bits, every single bit is used for performing one round and to select the mode of working for two structures (Substitution Permutation Network and Feistel Network) like encryption and decryption and key expanding. Hardware implementation of Substitution Feistel Network took Gate Equivalent area of 1876.04. HIGHT [25] is an ultra-light weight block cipher intended to be used for low resource consuming devices like in sensors in USN or a RFID tag which use simple operations to be energy efficient with 3048 GEs on 0.25 μm technology. Another 64-bit block cipher introduced for tiny resource constrained devices named as mCRYPTON [26] which encrypts data in twelve rounds by nibble wise substitution, column wise bit permutation, transposition of column to row and addition of key. It takes 2420 GEs in hardware implementation. A very well-known Russian encryption algorithm GOST 28147-89 revisited by Poschmann et al. [30] optimizing its design by selecting only one S-box instead of eight

different one to greatly reduce the area requirement constraint to 651 GE. Researchers invented a new member of the lightweight block cipher family for limited resourced devices like sensor nodes and RFIDs called as KLEIN [35] which used SPN structure for encryption and decryption and consumed 2629 gates of area. Some researchers suggested a collection of lightweight cipher block named as CHAM [42] that depended on a Feistel structure of four branch and containing Addition, Rotation and XOR (ARX) operation. This design used small area when implemented in hardware because of its simple key scheduling without updating the status of the key. To use the reduced set of keys of rounds in iteration repeatedly in this method the different round functions are designed in such a way. LiCi, a lightweight algorithm of block cipher design developed by Patil et al. [43] consisted of lightweight Substitution boxes of 4 × 4 dimension passed through 31 rounds successively. This design was for supporting 64-bits input as plaintext with key length of 128 bits. Gate Equivalent area, memory and power utilized by it was 1153 GEs, 1944 Bytes and 30mW. In 2017 a researcher [44] presented a light approach for cryptographic encryption based on SPN called BORON which had input as 64-bit plaintext and 128 or 80 bits of key length. This method comprised of 25 stages of encryption with 4 cross 4 substitution boxes and operations like round permutation, bit shifting and XOR making it resistant to differential and linear attacks. For 128-bit key and 80-bit key the area required for software and hardware implementation of BORON are 1939 GE and 1626 GE respectively. The architecture of PRESENT was revised by Banik et al. [45] and an improved version of block cipher based on SPN was proposed named as GIFT. This new method basically introduced cheaper and smaller S-Box by presenting bit permutation along with Linear Approximation Table (LAT) and Difference Distribution Table (DDT) of the substitution box resulting in less circuit complexity and faster computation. With a constant key size of 128 bits, the GIFT had two variations GIFT 64 and GIFT 128 with 28 and 40 encryption rounds respectively. A new algorithm of 64-bit block cipher called QTL which is introduced with the amalgamation of substitution permutation network and the regular Feistel network by researchers [46] for encryption. Moreover, this method could operate with both 64-bit key and 128 bits key. Same method was followed for decryption just like the encryption but taking round sub keys and round constants only in opposite manner. For the reduction of power and memory in hardware of QTL scheduling of key was omitted. On hardware for 128 bits and 64 bits keys this QTL takes 1206.52 GE area and 1025.52 GE area respectively. Main drawback of QTL [47] is its inability to protect against normal statistical attacks. RECTANGLE, a substitution permutation-based block cipher suggested by Zhang et al. [48] accepts input as 64-bit block and 80/128 bits keys for giving ciphertext (CT) of 64-bit. Power and area requirement for block cipher implementation are 74.31 μw and 1600 gate equivalents for 80-bit key case and 72.15 μw and 2064 gate equivalents for 128-bit key case. Few plus points of this method are speed and simple and friendly hardware design etc. A new architecture of lightweight block cipher was introduced by a group of researchers [49] which had integrated alternate key idea in regular Feistel structure. They also brought into effect again a software based lightweight block cipher featuring 80-bit input block and key of 80-bits length named ITUBEE. For improved security ITUBEE was made for 20 encryption rounds and whitening of key layers. Substitution box from the AES encryption technique was used in this architecture for the sake of reduction of memory, power and delay. ITUBEE proved its worth for its resistance against different key related attacks. The same authors of ITUBEE again introduced a new system of block cipher by altering the key in different rounds in Feistel structure called AKF [50]. A family of lightweight compact regular cipher block proposed by Yang et al. [51] by combining the benefits of SIMON and SPECK both known as SIMECK. The implementation in hardware was done on 130 nm and 65 nm CMOS process.

SIMECK developed by Yang et al. [51] showed its weakness against random byte fault attack and bit flip fault attack [52]. A symmetric Substitution Permutation Network type of block cipher named as LED was designed by Guo et al. [33]. It had made use of 64 bits input and 64 bits or 128 bits key length for 64-bit LED and 128-bit LED respectively. Similar to the rounds in AES, LED also had a series of similar encryption rounds which was made up of three functions like Constant addition, Substitution of Cells, Row shifting and Column mixing. The researchers promised their scheme to be impervious to single key and related attack. Researchers made use of ECC (elliptic curve cryptography) for designing proxy re-encryption system for improving the security in lightweight devices [53]. But the scheme proved to be insignificant because of extra calculation in polynomial equation. A lightweight round cipher block of 36 rounds was proposed by Suzaki et al. [54] called TWINE which has its input as 64 bits and 80- or 128-bits key length. TWINE-80 was improved in terms of security by Wei et al. [55] to make it resistant against differential cryptographic analysis and by one round improvement against conventional differential attacks. Biswas et al. [56] had proposed a 16-bit key length cryptographic strategy with 24 rounds of operation by uniting the SPN and Feistel structure together for confirming a fast diffusion process which consumes 258.9 GE area on 65 nm process technology.

## 4. System model

Sensors can easily be deployable by lightweight cryptographic chips inside them for conversion of normal unmasked data to masked one helping to configure the encrypting mechanism of sensing layer. **Fig. 1** shows a system model that depicts the importance of cryptographic circuitry in a sensor similar to the model built in [56]. The analog signal generated through the sensors needs to be converted to digital signal by a sensor compatible Analog to Digital Converter which becomes an input for the security chip. Then the digital sensed information is encrypted on the basis of the hardcoded cryptographic algorithms on the chip and is sent for storage in cloud server after passing through data analytics center. This encrypted data provided by cloud service provider can only be accessed by an authorized user from the cloud server. This type of light security chip design will be of great help for many security specific applications like home automation, smart healthcare system and smart agriculture etc. For this reason, a light cryptographic method called LCB has been proposed. Many related strategies have been inspected and for enhancing the system's system a novel resource restricted lightweight block cipher is designed called LCB. This newly designed algorithm is an incorporation of Substitution Permutation Network and Feistel structure together with combination of key kept simple. Normally a regular Feistel structure makes use of huge number of encryptions rounds but performs operation on half of the block only. The substitution permutation network uses confusion and diffusion tactic to increase the plaintext's redundancy ensuring a strong encrypted ciphertext. Hence, a combination of two structures has promised a more robust and protected system compared to when those structures are used individually. Speed, power, memory and storage are the key parameters to be kept in mind for the design of lightweight cryptographic system.

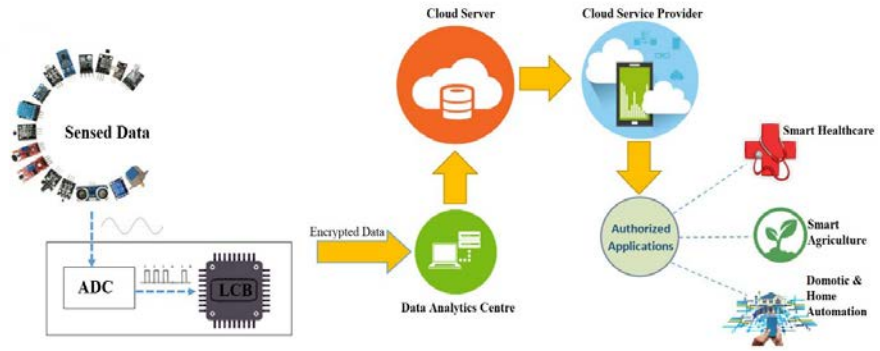



**Fig. 1.** System Model

## 5. Outline of LCB

The proposed algorithm presented here uses simple logical Ex-OR operations, bit scrambling and concatenation. Hence only an authorized user can decrypt the enciphered data for use. The proposed method operates with 32 bits of plaintext data keeping in mind to design lightest weight possible encryption system. For this reason, the lengthy block of data is divided into several blocks of 32 bits which in turn is treated by a data path and the process repeats for next blocks of data also. After processing all blocks of data, the encrypted data is obtained by merging the end results. Four subkeys each of 16 bits derived from the main key are used in different rounds of encryption. Thus, the proposed approach of enciphering uses less power, consumes less memory and storage space to achieve high performance for devices with tight resource budget which means it can achieve the solution without breaching all the constraints mentioned before and delivering better protection from different cryptanalysis techniques.

### 5.1 Key formation

Four subkeys each of 16-bit width ($K_1$, $K_2$, $K_3$ and $K_4$) are derived from the main 64-bit key in the proposed method of encryption and decryption. The need for building a lightweight encryption process has led to build a strong but simple key generation tactic by feeding different subkeys in different F-functions of different rounds of encryption process. As mentioned earlier ten different rounds of encryption uses four different subkeys sequence generated from the main key which are used in different levels of round operations guarantees resilience against related key attacks. The subkey generation strategy is depicted in Key generation algorithm.

### 5.2 Encryption policy

Encryption is the process of conversion of readable data to unreadable form by adopting different confusion and diffusion strategies. Hence it is very important to make this approach more protective to different cryptanalyst's attacks. The proposed LCB is well explained and its encryption process is depicted clearly in Algorithm 1 and for block diagram representation **Fig. 2** is shown. The newly designed algorithm passes over repetitive number of steps which includes F-function and round transposition for ten successive rounds of operation. Intermediate ciphertext outputs are generated after each round which are fed immediately to the next round. The output generated after completion of ten consecutive rounds is the final ciphertext output (CT).

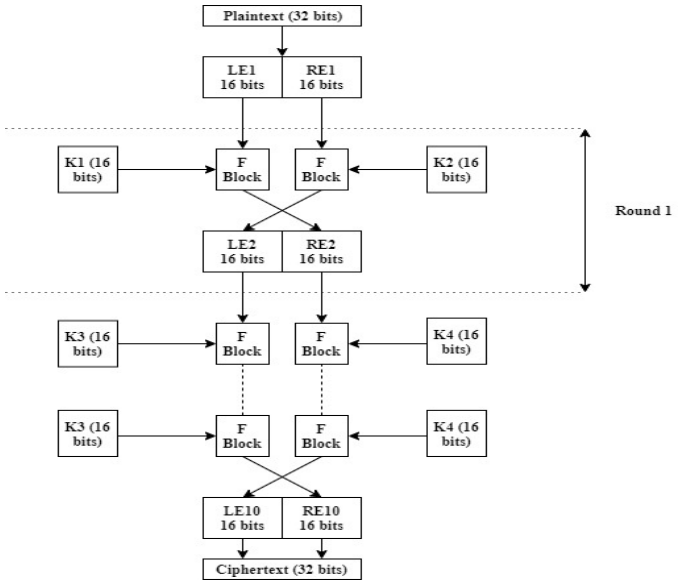



**Fig. 2.** Encryption Process of LCB

**Algorithm 1:** LCB Encryption

1. 32-bit Plaintext is divided into two parts of 16 bits each i.e. ($LE_n$ and $RE_n$)
2. Initialize 'n' with value '1'.
3. The $LE_n$ and $RE_n$ is passed through F-function and the outputs become $RE_{n+1}$ and $LE_{n+1}$ respectively.
4. $K_1$, $K_2$, $K_3$ and $K_4$ are the input subkeys to the first four F-blocks respectively and the sequence repeats for next F-Blocks also.
5. If (n<10) then go to step 3.
6. Else go to step 6.
7. Ciphertext = Output of last right F-Box concatenated with output of last left F-Box.

**Algorithm 2:** F-Function

1. Input of 16 bits is divided into four parts of 4 bits each.
2. These four 4 bits are fed into four S-Boxes.
3. Outputs of first and second S-Boxes are fed to first P-Box which gives output of 8 bits and outputs of third and fourth S-Boxes are fed to second P-Box.
4. The outputs of P-Boxes are eventually fed to the L-Box whose 16-bit output is eventually XORed with the subkey giving the final output.

**S-Box Algorithm**

1. Input a[n] is 4 bits, Output b[n] is 4 bits
2. Output=a [1], a [4], a [2], a [3]

**P-Box Algorithm**

1. Input 1 c[n] is 4 bits, Input 2 d[n] is 4 bits, Output e[n] is 8 bits
2. Output= c [1], d [1], c [4], d [4], c [2], d [2], c [3], d [3]

**L-Box Algorithm**

1. Input 1 f[n] is 8 bits, Input 2 g[n] is 8 bits, Output h[n] is 16 bits
2. Output= f [1], g [1], f [8], g [8], f [2], g [2], f [7], g [7], f [3], g [3], f [6], g [6], f [4], g [4], f [5], g [5]

**Key Generation**

1. 4 subkeys $\{K_1, K_2, K_3, K_4\}$ are needed in both encryption as well as decryption process and the same subkeys are used for both the processes
2. The subkeys are generated as:
   - $K_1$ = First 16 bits of the 64-bit key
   - $K_2$ = Second 16 bits of the 64-bit key
   - $K_3$ = Third 16 bits of the 64-bit key
   - $K_4$ = Fourth 16 bits of the 64-bit key

### 5.2.1 F-function generation

According to the proposed method, each round consists of two F-functions which is again built up of three different functional boxes called S, P and L-Box. All these blocks perform different types of computations as described in the algorithm section. Inside the F-Block the incoming data is split up into four parts and each one fed to each of the four S-Boxes present inside which in turn gives the output to the next P-Box which again feeds its output as input to the L-Box. The exact operations going on inside these individual blocks is shown in Algorithm 2. A suitable example will make it understandable about the operations going on inside each box.

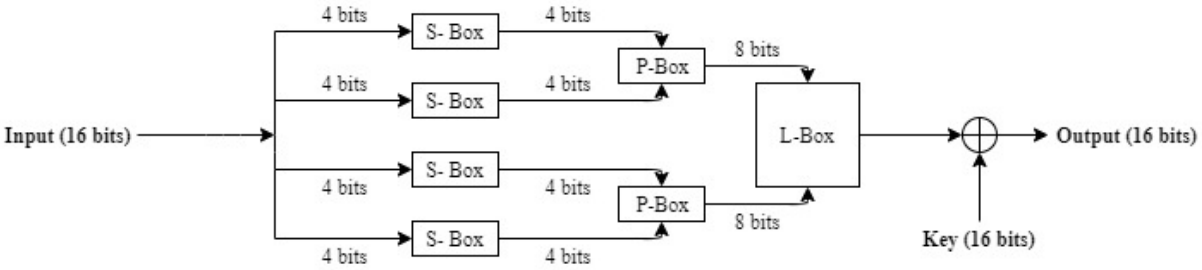



**Fig. 3.** Architecture of F-Block.

### 5.2.2 Round transposition

The data goes through round transposition after passing through F-function where the incoming data bits are swapped and fed as input to the next round which is demonstrated in **Fig. 4**. This process is to ensure an extra layer of security on the data. According to the figure shown the input for this transposition process is 16 bits. The resultant output of 16 bits obtained after transposition is made as input to the next round.

### 5.3 Decryption strategy

Designing an equally strong decryption algorithm is unquestionably important when developing an encryption process. Here as the structure of the algorithm is Feistel in nature so regularity is maintained in both encryption and decryption flow but the whole process is inverted for the decryption technique.

## 6. Design technique

### 6.1 F-Block

As previously said, input plaintext of 32 bits is split up into two blocks of 16 bits each. The components of light and secure designed proposed F-Box are S, P and L-Boxes. As seen in the literature the 8×8 S-Boxes use much more resource in hardware implementation while providing better security which is a challenge for lightweight devices. For this reason, four 4×4 S-Boxes are introduced in this algorithm for its low power consumption and less area requirement when compared with the 8×8 S-Box.

On the other hand, the P-Box doubles the data width by concatenating the input bits from previous S-Boxes in a particular manner as shown in Algorithm 2 thus producing an output of 8 bits. L-Block also does the same operation just like P-Block taking two inputs of 8 bits and producing output of 16 bits which is in turn XORed with the subkey produced and the final output of the F-Block is given as directed in the algorithm section. To make the design ultra-lightweight for very low consumption of area and power in hardware and to achieve maximum speed of computation minimum number of logical operations are used and maximum focus was kept on bit jumbling and scrambling. The F-Block algorithm is made so simple that it uses only one 16-bit XOR gate in each F-Function. The block diagram of the proposed algorithm is shown in **Fig. 2** and that of a single F-Block in **Fig. 3**. The whole algorithm is made up of twenty 16-bit XOR gates, data registers and key storage etc. Normally gate equivalent (GE) is the measure of area consumed which is equivalent to the area required by a NAND gate having two inputs. A 2:1 Multiplexer of 4 bits takes 0.5 gate equivalents of area and for 4-bit XOR operation it takes 7.075 GE area [46] as an example. The novel approach is designed by satisfying all possible necessary constraints.

### 6.2 Round swapping

The generated values of output from the F-Box of each stage like X1, X2 are sent to the next round inputs as Y2, Y1 respectively. The connections are done by simple wires as shown in **Fig. 4** which are capable of transmitting 16 bits of data at a time.

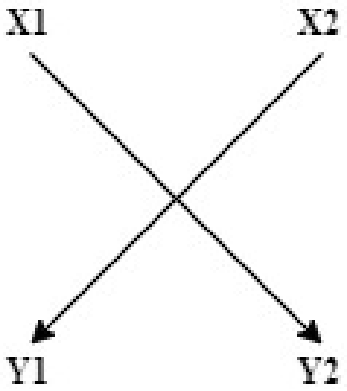



**Fig. 4.** Round transposition process of LCB

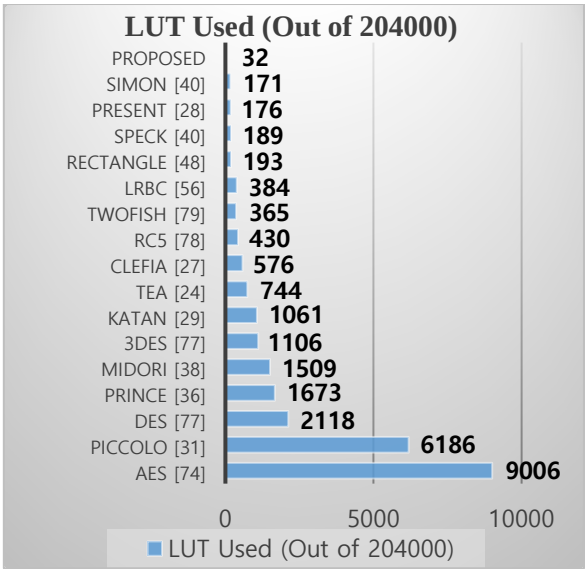
## 7. Simulation, Synthesis and Implementation

The suggested lightweight cryptographic model is designed and written in Verilog and its rightness is verified in ModelSim 2020.1.1 by running the behavioral simulation taking one

input as the 32-bit Plaintext another input as the 64 bits Key and getting 32-bit Ciphertext output as shown in the simulation window in **Fig. 6**.

As shown here in this figure $eed4b555_{16}$ is taken as plaintext input and $cdba38fb_{16}$ is the output with $d622529ecc92d353_{16}$ as key. The plaintext represented here is 'X', the intermediate outputs are called '$Y_i$' and final output obtained after running ten rounds of algorithm is Y. After this the design is synthesized on (Virtex-7) XC7VX330T FPGA board on 28 nm process using Xilinx ISE 9.2i and Xilinx ISE 14.7 for verification of delay and area parameters. The proposed algorithm is proved to be extremely fast with a combinational path delay of 0.877ns and has a very low look up table (LUT) count as 32 only out of 20400 and consuming only 224 GE of area which is lesser than all previously reported algorithms. The area and resources consumed by the algorithm is directly proportional to the LUT count of the design. This design is made in such a way that it uses very minimal number of logical operations and gates to reduce the resources used to maximum extent. Only twenty 16-bit XOR gates are used in the entire design. The main confusion and diffusion for this algorithm depends on bit jumbling and scrambling. The number of rounds is also kept low to avoid unnecessary process delay as increase in the number of rounds only increase the critical path delay of the architecture. As the logic slices used here is significantly low so it can be told without any question that the power requirement should be dramatically low also. A power comparison is not given because different reported designs used different process technologies and it will not be justified to compare different algorithms implemented on different process technologies. A good Avalanche Effect is met after only ten rounds of operation so there stands no reason to increase the number of rounds for more security compromising the area, power and delay constraints.

**Fig. 5** shows a comparison of LUT count and delay between the popular encryption algorithms introduced over the years. **Table 2** and **Table 3** shows the detailed test vectors taken to test the algorithm once by keeping key constant and once by keeping plaintext constant respectively. Five test cases are taken for each of different hamming distances (HD) and an average avalanche effect is shown.



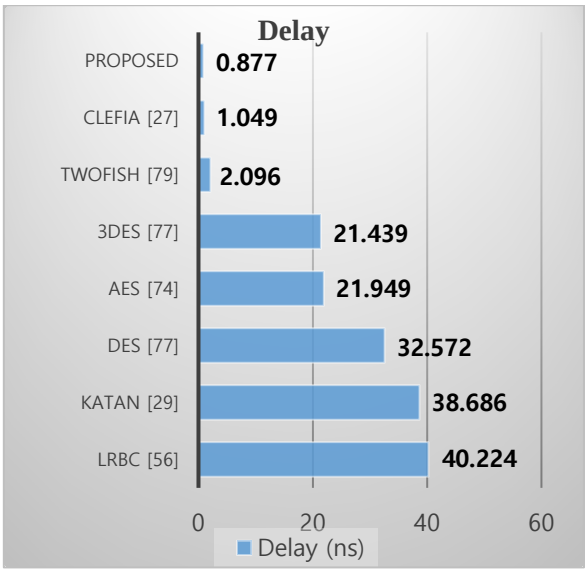



**Fig. 5.** LUT count and delay comparison

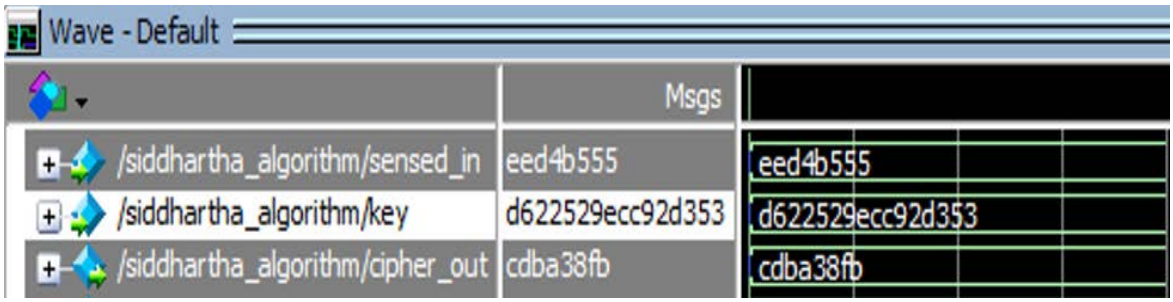



**Fig. 6.** Simulation result in ModelSim 2020.1

**Table 2.** Test vectors taken keeping PT variable and Key constant with AE for each case

| | | HD-1 | HD-2 | HD-3 | HD-4 | HD-5 | Average |
|---|---|---|---|---|---|---|---|
| 1 | **PT** | EED4B555 | 6ED4B555 | 2ED4B555 | 2ED4B755 | 2ED4B757 | 63.125 |
| | **CT** | CFA34FA8 | 4FA34FA8 | 6FA34FA8 | 6FA34F28 | 6FA34F2C | |
| | **AE** | 65.625 | 65.625 | 65.625 | 59.375 | 59.375 | |
| 2 | **PT** | CED4B555 | CED4B5D5 | CED4B5D7 | DED4B5D7 | FED4B5D7 | 65.625 |
| | **CT** | CFAB4FA8 | CFAB4BA8 | CFAB4BAC | CFA94BAC | CFA14BAC | |
| | **AE** | 65.625 | 65.625 | 65.625 | 65.625 | 65.625 | |
| 3 | **PT** | CED4B475 | CED6B475 | CED6B474 | EED6B474 | EAD6B474 | 61.875 |
| | **CT** | CFAB4FC8 | CFAF4FC8 | CFAF4FC9 | CFA74FC9 | CDA74FC9 | |
| | **AE** | 65.625 | 59.375 | 59.375 | 59.375 | 65.625 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | **PT** | CED4B5C5 | CED4B5C4 | CED0B5C4 | CAD0B5C4 | CAD0BDC4 | 68.125 |
| | **CT** | CFAB4BB8 | CFAB4BB9 | DFAB4BB9 | DDAB4BB9 | DDAB43B9 | |
| | **AE** | 65.625 | 65.625 | 65.625 | 71.875 | 71.875 | |
| 5 | **PT** | CED4B5C6 | CEF4B5C6 | CEF4B5D6 | CEF4B7D6 | 4EF4B7D6 | 60.625 |
| | **CT** | CFAB4BBD | CFEB4BBD | CFEB4BAD | CFEB4B2D | 4FEB4B2D | |
| | **AE** | 65.625 | 59.375 | 59.375 | 59.375 | 59.375 | |
| **Key:** D622529ECC92D353 | | | | | | | 63.875 |

**Table 3.** Test vectors taken keeping Key variable and PT constant with AE for each case

| | | **HD-1** | **HD-2** | **HD-3** | **HD-4** | **HD-5** | **Average** |
|---|---|---|---|---|---|---|---|
| 1 | **Key** | D63A529ECC92D353 | 563A529ECC92D353 | 163A529ECC92D353 | 163A5296CC92D353 | 163A5296CC92DB53 | 69.375 |
| | **CT** | CDAA4BBD | 4DAA4BBD | 4D2A4BBD | 4D2A49BD | 452A49BD | |
| | **AE** | 65.625 | 68.75 | 71.875 | 68.75 | 71.875 | |
| 2 | **Key** | 163AD296CC92DB53 | 167AD296CC92DB53 | 967AD296CC92DB53 | D67AD296CC92DB53 | D67AD296EC92DB53 | 63.75 |
| | **CT** | 452AC9BD | 450AC9BD | C50AC9BD | C58AC9BD | C58AC9B5 | |
| | **AE** | 65.625 | 65.625 | 65.625 | 65.625 | 56.25 | |
| 3 | **Key** | D63A569ECC92D353 | D63A769ECC92D353 | D63A769E8C92D353 | D63A629E8C92D353 | D63A629E8C92F353 | 60 |
| | **CT** | CDAA4BB5 | CDAA4FB5 | CDAA6FB5 | CDAA6FB9 | CDA26FB9 | |
| | **AE** | 62.5 | 59.375 | 56.25 | 62.5 | 59.375 | |
| 4 | **Key** | D6BA629E8C92F353 | D6BA6A9E8C92F353 | D2B29A9E8CD2F353 | D2B29ADE8CD2F353 | 56BA569ECC92D351 | 62.5 |
| | **CT** | EDA26FB9 | EDA267B9 | EFAAE23D | EFAAE21D | 6DAE4BB5 | |
| | **AE** | 62.5 | 62.5 | 62.5 | 59.375 | 65.625 | |
| 5 | **Key** | 76BA569EDC93D353 | 76BA569EDC9BD353 | F6BA569EDC9BD353 | D6BA569EDC9BD353 | F4AB5DB7D4EBF352 | 60 |
| | **CT** | 69AA4BB6 | 69AA0BB6 | E9AA0BB6 | EDAA0BB6 | A9B249B6 | |
| | **AE** | 65.625 | 62.5 | 59.375 | 56.25 | 56.25 | |
| **Plaintext:** CED4B5C6 | | | | | | | 63.125 |

## 8. Security analysis

Two different kinds of analyses are employed to evaluate the strength of security of the proposed algorithm namely: Attack analysis and Avalanche Effect analysis.

### 8.1 Avalanche effect

In cryptography avalanche effect plays a very important role when determining the degree of goodness of an encryption algorithm. It is the measure of the number of bits changed in a ciphertext if only one bit is changed in the plaintext or key. The greater the number of bits changed for very little change in plaintext or key the more secure that encryption technique is. Generally, if that change is 50% or half of the number of bits gets affected if only one bit is changed that avalanche effect is considered a good one [57]. The more that value of avalanche effect is tighter that security is. The following expression in Equation (1) expresses Avalanche effect.

$$\text{Avalanche Effect (AE)} = \frac{\text{Number of flipped bits in ciphertext}}{\text{Number of bits in plaintext}} \times 100\ \% \quad \text{.....(1)}$$

The study here on Avalanche effect is done by taking a fixed block of plaintext and by changing the hamming distance of the key randomly up to 5 bits in five different test cases. The same process of observation is done for variable plaintext and fixed key also. **Fig. 7** displays a graphical interpretation of avalanche effect for different test cases for LCB keeping plaintext or key constant in each case and vice versa. Depending on this test a comparative analysis is done for LCB with previous state of art works which is shown in **Fig. 8**.

It is found after in depth security analysis that the newly designed approach attains a decently high avalanche effect of 63.875 % with respect to plaintext and 63.125 % with respect to key which is much improved than any previously reported algorithms taken into consideration.

## 9. Attack analysis

The newly designed algorithm follows some simple but hard logical operating steps. The section below covers a clear explanation on how the proposed method is resistant to different types of cryptographic attacks. Some commonly used attacks are: Linear Attack, Differential Attack, Key related side channel attack, Impossible Differential Attack, Structural Attack, Algebraic Attack and Slide Attack.

### 9.1 Linear Attack

Known plaintext attack also known as Linear Attack is the type of cryptographic attack where the cryptanalyst knows some portion of the plaintext and ciphertext in which his main objective remains to find the key used to encrypt the data. In this type of attack the cryptanalyst tries to find a linear relation between plaintext and ciphertext using Equation (2) to find the possibility of the equation to be fulfilled [57][58].

$$x_1 \oplus x_2 \oplus \cdots \oplus x_n \oplus y_1 \oplus y_2 \oplus \cdots \oplus y_n = 0 \quad \text{.....(2)}$$

where $x_1, x_2 \ldots x_n$ are the bit values of plaintext and $y_1, y_2 \ldots y_n$ are their corresponding bit values of ciphertext. The cipher is unsecured from this type of attack if the probability of

satisfying Equation (2) is not equal to $\frac{1}{2n}$, where n is the number of bits in plaintext or ciphertext. The difference of probability with the value $\frac{1}{2}$ is known as bias. The more the value of bias is closer to $\frac{1}{2}$ the better the security is from linear attack. A linear probability analysis example for proposed S-Box taking five linear equations is shown in **Table 4**. From the table it is noted that for most linear expressions the value is $\frac{1}{2}$ and the bias value is 0. Hence the proposed method of encryption is resistant to linear attacks or known plaintext attacks.


**Fig. 7.** Avalanche effect with respect to plaintext and key

**Table 4.** Analysis of linear probability of S-Box of proposed algorithm

| X | | | | Y | | | | $X_2 \oplus X_3$ | $Y_1 \oplus Y_3 \oplus Y_4$ | $X_1 \oplus X_4$ | $X_3 \oplus X_4$ | $Y_1 \oplus Y_4$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_1$ | $X_2$ | $X_3$ | $X_4$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |

### 9.2 Differential Attack

Differential Attacks are also called chosen plaintext attacks as they stress mainly on high chance of occurrence of difference of output with respect to a given difference of input [58][57]. For an example, for the proposed S-Box X = $[x_1, x_2, x_{3,\ldots\ldots,} x_n]$ is the input and Y = $[y_1, y_2, y_{3,\ldots\ldots,} y_n]$ is the output. Difference between two different plaintexts is denoted by α = $x_1 \oplus x_2$ and difference between their ciphertext correspondingly is denoted by β = $y_1 \oplus y_2$. A differential pair (α, β) is formed by a given difference of input α and its corresponding output difference β. For an α value, if the occurrence of β is large then the cipher is vulnerable to differential attacks. Commonly, it's important to decrease the probability of difference pair to protect the cipher from chosen ciphertext attack or differential attack [56]. In this regard many different combinations of 'α' of 16 bits have been examined with different plaintexts of 16-bits for producing the corresponding β values. Accordingly, the difference pair (α, β) for this newly designed approach is noted. Highest differential probability noticed is $\frac{9}{216}$ and most suitable differential probability obtained is $\frac{2}{216}$ which is quite acceptable for all types of enciphering techniques. Therefore, the proposed algorithm can be said is secure from differential attacks.

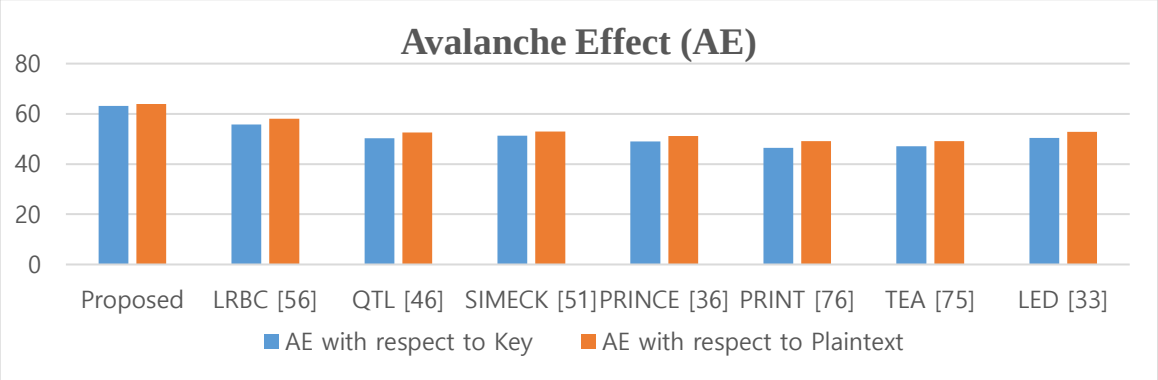



**Fig. 8.** Comparison of Avalanche Effect of different Encryption Algorithms

### 9.3 Side channel Attack

This type of attack depends on data obtained when the encryption algorithm is realized [59]. This proposed algorithm can be said that it is side channel attack proof because the key used here is not dependent on the plaintext chosen and it uses random number for encryption process. The input and output dependency are negligible for this proposed algorithm. Again, the proposed technique provides security from other types of cryptanalyses. The two most weak parts in any cryptographic algorithm are the round functions and the key used as seen by an attacker. For this reason, the key generation technique and the round function design are to be

done with sheer seriousness when designing any encryption scheme. Cryptanalysts try to recover the original key by closely observing the bit patterns of the output ciphertext when doing related key attacks. This can be possible only when a fixed key is being used in all the rounds of the algorithm [45][49]. The safety is maintained from this type of attack in this proposed method because it uses different subkeys in different rounds of operation. Thus, it can be said that this proposed method is well protected from side channel attacks as it becomes very hard for an intruder to determine the correct pattern of subkeys used in the proposed technique.

### 9.4 Impossible Differential Attack

One of the most used cryptanalytic techniques on lightweight block ciphers specifically fit for software platforms [60-65] is Impossible differential attack [66]. Because of the Feistel nature of the cipher structure the highest number of rounds on which impossible characteristics is seen is five [67] for which reason this kind of attack is not valid for this cipher as the number of rounds is ten.

### 9.5 Structural Attack

Like Bottleneck attack [68] and integral attack [69] structural attacks use word like structures in the encrypted text. The permutation used in LCB is bitwise which is the reason for its impermeability against structural attacks as word like structures cannot be derived in this case.

### 9.6 Algebraic Attack

LCB uses F-Box that is comprised of 4-bit S-Box, 8-bit P-Box and 16-bit L-Box. Although the 4-bit S-Boxes are attackable like in PRESENT [28] but the presence of 8-bit P-Box which can be described by eight algebraic equations of degree seven and 16-bit L-Box which is represented by sixteen algebraic equations with degree fifteen where the number of terms is far more than 8-bit S-Box used in HALKA [70]. The HALKA cipher is still more protective against algebraic attacks than PRESENT with number of terms ranging from 139 to 118 for its S-Box. This is not a system which uses small number of terms to describe itself. Therefore, a detailed analysis is not needed like PRESENT cipher to prove its security against algebraic attacks.

### 9.7 Slide Attack

Some limitations of key schedule are related-key [71] and Slide attacks [72]. The subkey generation mechanism of LCB is simple but because of the round function with high avalanche effect and strong non linearity it is very much frustrating to attack the cipher with this type of attack. The vulnerability of a cipher to slide attacks is when it uses similar structure and same subkey in the round functions, but since LCB makes use of dissimilar subkeys in each round it can be said to be resistant to slide attack.

## 10. Functional study

Some previously reported encryption algorithms are shown and compared with their performance measures and features in **Table 5**. Parameters like key length, block size, operational rounds, nature of the structure, process technology, frequency and Gate equivalent

area are used to draw a fine comparison between previous state of art works and this proposed work. Most designed block cipher algorithms have 32 bit or 64-bit plaintext data having different key lengths with either Feistel or SPN like structure as shown in **Table 5**. QTL and LRBC are the only two which used mixed structure of Feistel and SPN.

After observation of the shown values, it can be said that the smaller the key and block size is the better and fast is the confusion and diffusion process giving better security. The novel contribution uses a mixed structure approach for exploiting both the benefits of SPN and Feistel structure like QTL and LRBC with ten consecutive rounds to build a more immune cipher. Additionally, the proposed system used 28 nm process technology in Virtex-7 FPGA board [73] which permitted a greater number of gates but occupying lesser area than others. Thus, LCB fulfilled every aspect of design constraint for achieving better performance than all others to become a worthy name for light-weight resource constrained device's encryption system.

**Table 5.** Functionality analysis of different Encryption Algorithms

| Ciphers | Block Size (bit) | Key Size (bit) | Structure | Round | Process technology (*nm*) | Area (GE) |
|---|---|---|---|---|---|---|
| PRINCE [36] | 64 | 128 | SPN | 12 | 130 | 3491 |
| LED [33] | 64 | 128 | SPN | 48 | 130 | 3407 |
| AES [74] | 8 | 128 | SPN | 16 | 130 | 3100 |
| HIGHT [25] | 64 | 128 | Feistel | 32 | 250 | 3048 |
| KLEIN [35] | 64 | 80 | SPN | 16 | 180 | 2629 |
| mCRYPTON [26] | 64 | 64 | SPN | 12 | 130 | 2420 |
| RECTANGLE [48] | 64 | 80 | SPN | 26 | 130 | 1599.5 |
| PRESENT [28] | 64 | 80 | SPN | 32 | 180 | 1570 |
| TWINE [54] | 64 | 80 | Feistel | 36 | 90 | 1503 |
| TEA [75] | 64 | 128 | Feistel | 32 | 350 | 1140 |
| QTL [46] | 64 | 64 | SPN, Feistel | 16 | 180 | 1026 |
| PICCOLO [31] | 64 | 80 | Feistel | 25 | * | 683 |
| GOST [30] | 64 | 256 | Feistel | 32 | 180 | 651 |
| SPECK [40] | 32 | 64 | Feistel | 22 | 130 | 580 |
| SIMON [40] | 32 | 64 | Feistel | 32 | 130 | 523 |
| SIMECK [51] | 32 | 64 | Feistel | 32 | 65 | 488 |
| KATAN [29] | 32 | 80 | Stream | 254 | 130 | 462 |
| PRINT [76] | 48 | 80 | SPN | 48 | 180 | 402 |
| LRBC [56] | 16 | 16 | SPN, Feistel | 24 | 65 | 258.9 |
| Proposed LCB | 32 | 64 | SPN, Feistel | 10 | 28 | 224 |

## 11. Conclusion

A novel light cipher block termed as LCB for resource restricted IoT devices is depicted and realized here in this work. The suggested work increases the security of data by combining the positive sides of both SPN and Feistel structure connected by the idea of linear box. LCB also produces a greater number of S-boxes which actively help to resist differential and linear attack. Use of 4-bit data for internal operation and enhanced design takes very less delay and area of the chip. The use of a smaller number of gates in the new design greatly improve the computational delay and security is maintained by bit jumbling and repeating. The proposed system promises security which is decently high with a reasonable time and area consumption after going through rigorous security checks.

## Acknowledgement

Authors would like to acknowledge to the reviewers for their valuable feedback. This publication is an outcome of the R&D work undertaken project under the Ph.D. Scheme of Ministry of Human Resource and Development, Government of India, being implemented by Digital India Corporation.

## References

[1] S. Jangirala, A. K. Das, and A. V. Vasilakos, "Designing Secure Lightweight Blockchain-Enabled RFID-Based Authentication Protocol for Supply Chains in 5G Mobile Edge Computing Environment," *IEEE Trans. Ind. Informatics*, vol. 16, pp. 11, 2020. Article (CrossRef Link).

[2] J. Ni, K. Zhang, and A. V. Vasilakos, "Security and Privacy for Mobile Edge Caching: Challenges and Solutions," *IEEE Wirel. Commun*., vol. 28, pp. 77-83, 2021. Article (CrossRef Link).

[3] G. Zhou et al., "Smart savings on private car pooling based on internet of vehicles," *J. Intell. Fuzzy Syst.*, vol. 32, pp. 3785-3796, 2017. Article (CrossRef Link).

[4] J. Srinivas, A. K. Das, M. Wazid, and A. V. Vasilakos, "Designing Secure User Authentication Protocol for Big Data Collection in IoT-Based Intelligent Transportation System," *IEEE Internet Things J*., vol. 8, pp. 7727-7744, 2021. Article (CrossRef Link).

[5] T. K. L. Hui, R. S. Sherratt, and D. D. Sánchez, "Major requirements for building Smart Homes in Smart Cities based on Internet of Things technologies," *Futur. Gener. Comput. Syst.*, vol. 76, pp. 358-369, 2017. Article (CrossRef Link).

[6] P. P. Ray, "Internet of things for smart agriculture: Technologies, practices and future direction," *J. Ambient Intell. Smart Environ.*, vol. 9, pp. 395-420, 2017. Article (CrossRef Link).

[7] B. Bera, S. Saha, A. K. Das, and A. V. Vasilakos, "Designing blockchain-based access control protocol in iot-enabled smart-grid system," *IEEE Internet Things J*., vol. 8, pp. 5744-5761, 2021. Article (CrossRef Link).

[8] M. Hao, W. Zhang, Y. Wang, G. Lu, F. Wang, and A. V. Vasilakos, "Fine-Grained Powercap Allocation for Power-Constrained Systems Based on Multi-Objective Machine Learning," *IEEE Trans. Parallel Distrib. Syst*., vol. 32, pp. 1789-1801, 2021. Article (CrossRef Link).

[9] A. Majumdar, T. Debnath, S. K. Sood, and K. L. Baishnab, "Kyasanur Forest Disease Classification Framework Using Novel Extremal Optimization Tuned Neural Network in Fog Computing Environment," *J. Med. Syst*., vol. 42, 2018. Article (CrossRef Link).

[10] A. Majumdar, N. M. Laskar, A. Biswas, S. K. Sood, and K. L. Baishnab, "Energy efficient e-healthcare framework using HWPSO-based clustering approach," *J. Intell. Fuzzy Syst.*, vol. 36, pp. 3957-3969, 2019. Article (CrossRef Link).

[11] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the Internet of Things: perspectives and challenges," *Wirel. Networks*, vol. 20, pp. 2481-2501, 2014. Article (CrossRef Link).

[12] M. Wazid, A. K. Das, V. Bhat K, and A. V. Vasilakos, "LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment," *J. Netw. Comput. Appl*., vol. 150, 2020. Article (CrossRef Link).

[13] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel, "A survey of lightweight-cryptography implementations," *IEEE Design and Test of Computers*, vol. 24, pp. 522-533, 2007, Article (CrossRef Link).

[14] B. J. Mohd, T. Hayajneh, and A. V. Vasilakos, "A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues," *J. Netw. Comput. Appl*., vol. 58, pp. 73-93, 2015. Article (CrossRef Link).

[15] K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha, "Report on lightweight cryptography," *Natl. Inst. Stand. Technol*., 2017.

[16] O. Toshihiko, "Lightweight cryptography applicable to various IoT devices," *NEC Tech. J*., 2017.

[17] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. McCann, and K. Leung, "A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities," *IEEE Wirel. Commun*.,

2013. Article (CrossRef Link).
[18] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions," *J. Ambient Intell. Humaniz. Comput.*, 2017. Article (CrossRef Link).
[19] M. A. Abdelraheem, C. Blondeau, M. Naya-Plasencia, M. Videau, and E. Zenner, "Cryptanalysis of ARMADILLO$_2$," in *Proc. of ASIACRYPT 2011*, pp. 308-326, 2011. Article (CrossRef Link).
[20] J. Y. Cho, "Linear cryptanalysis of reduced-round PRESENT," 2010. Article (CrossRef Link).
[21] M. Albrecht and C. Cid, "Algebraic techniques in differential cryptanalysis," in *Proc. of FSE 2009*, pp. 193-208, 2009. Article (CrossRef Link).
[22] B. Collard and F. X. Standaert, "A statistical saturation attack against the block cipher present," in *Proc. of CT-RSA 2009*, pp 195-210, 2009. Article (CrossRef Link).
[23] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions," *J. Ambient Intell. Humaniz. Comput.*, 2017. Article (CrossRef Link).
[24] D. J. Wheeler and R. M. Needham, "Tea, a tiny encryption algorithm," in *Proc. of International Workshop on Fast Software Encryption*, pp. 363-366, 1994. Article (CrossRef Link).
[25] D. Hong et al., "HIGHT: A new block cipher suitable for low-resource device," in *Proc. of International Workshop on CHES 2006*, pp 46-59, 2006. Article (CrossRef Link).
[26] C. H. Lim and T. Korkishko, "mCrypton - A lightweight block cipher for security of low-cost RFID tags and sensors," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3786 LNCS, pp. 243–258, 2005. Article (CrossRef Link).
[27] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit blockcipher CLEFIA," in *Proc. of International Workshop on FSE 2007*, pp. 181-195, 2007. Article (CrossRef Link).
[28] A. Bogdanov et al., "PRESENT: An ultra-lightweight block cipher," in *Proc. of International Workshop on CHES 2007*, pp. 450-466, 2007. Article (CrossRef Link).
[29] C. De Cannière, O. Dunkelman, and M. Knežević, "KATAN and KTANTAN - A family of small and efficient hardware-oriented block ciphers," in *Proc. of International Workshop on CHES 2009*, pp. 272-288, 2009. Article (CrossRef Link).
[30] A. Poschmann, S. Ling, and H. Wang, "256 Bit standardized crypto for 650 GE - GOST revisited," in *Proc. of International Workshop on CHES 2010*, pp. 219-233, 2010. Article (CrossRef Link).
[31] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, "Piccolo: An ultra-lightweight blockcipher," in *Proc. of International Workshop on CHES 2011*, pp. 342-357, 2011. Article (CrossRef Link).
[32] A. Bogdanov, M. Knežević, G. Leander, D. Toz, K. Varici, and I. Verbauwhede, "Spongent: A lightweight hash function," in *Proc. of International Workshop on CHES 2011*, pp. 312-325, 2011. Article (CrossRef Link).
[33] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED block cipher," in *Proc. of International Workshop on CHES 2011*, pp. 326-341, 2011. Article (CrossRef Link).
[34] J. Guo, T. Peyrin, and A. Poschmann, "The PHOTON family of lightweight hash functions," in *Proc. of Annual Cryptology Conference CRYPTO 2011*, pp. 222-239, 2011. Article (CrossRef Link).
[35] Z. Gong, S. Nikova, and Y. W. Law, "KLEIN: A new family of lightweight block ciphers," in *Proc. of International Workshop on RFIDSec 2011: RFID. Security and Privacy*, pp. 1-18, 2011. Article (CrossRef Link).
[36] J. Borghoff et al., "PRINCE - A low-latency block cipher for pervasive computing applications," in *Proc. of Advances in Cryptology – ASIACRYPT 2012*, pp. 208-225, 2012. Article (CrossRef Link).
[37] D. Hong, J. K. Lee, D. C. Kim, D. Kwon, K. H. Ryu, and D. G. Lee, "LEA: A 128-bit block cipher for fast encryption on common processors," in *Proc. of International Workshop on Information Security Applications*, pp. 3-27, 2014. Article (CrossRef Link).
[38] S. Banik et al., "Midori: A block cipher for low energy," in *Proc. of Advances in Cryptology – ASIACRYPT 2015*, pp. 411-436, 2015. Article (CrossRef Link).

[39] M. R. Albrecht, B. Driessen, E. B. Kavun, G. Leander, C. Paar, and T. Yalçin, "Block ciphers - Focus on the linear layer (feat. PRIDE)," in *Proc. of Annual Cryptology Conference CRYPTO 2014*, pp. 57-76, 2014. Article (CrossRef Link).
[40] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," in *Proc. of of the 52nd Annual Design Automation Conference*, pp. 1-6, 2015. Article (CrossRef Link).
[41] L. Li, B. Liu, Y. Zhou, and Y. Zou, "SFN: A new lightweight block cipher," *Microprocess. Microsyst.*, vol. 60, pp. 138-150, 2018. Article (CrossRef Link)
[42] B. Koo, D. Roh, H. Kim, Y. Jung, D. G. Lee, and D. Kwon, "CHAM: A family of lightweight block ciphers for resource-constrained devices," in *Proc. of ICISC 2017*, pp. 3-25, 2018. Article (CrossRef Link).
[43] J. Patil, G. Bansod, and K. S. Kant, "LiCi: A new ultra-lightweight block cipher," in *Proc. of 2017 International Conference on Emerging Trends & Innovation in ICT (ICEI)*, 2017. Article (CrossRef Link).
[44] G. Bansod, N. Pisharoty, and A. Patil, "BORON: an ultra-lightweight and low power encryption design for pervasive computing," *Front. Inf. Technol. Electron. Eng.*, vol. 18, pp. 317-331, 2017. Article (CrossRef Link).
[45] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "GIFT: A small present: Towards reaching the limit of lightweight encryption," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10529 LNCS, pp. 321–345, 2017. Article (CrossRef Link).
[46] L. Li, B. Liu, and H. Wang, "QTL: A new ultra-lightweight block cipher," *Microprocess. Microsyst.*, vol. 45, pp. 45-55, 2016. Article (CrossRef Link).
[47] S. Sadeghi, N. Bagheri, and M. A. Abdelraheem, "Cryptanalysis of reduced QTL block cipher," Microprocess. Microsyst., vol. 52, pp. 34-48, 2017. Article (CrossRef Link).
[48] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, "RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms," *Sci. China Inf. Sci.*, vol. 58, pp. 1-15, 2015. Article (CrossRef Link).
[49] F. Karakoç, H. Demirci, and A. E. Harmanci, "ITUbee: A software oriented lightweight block cipher," in *Proc. of International Workshop on LightSec 2013*, pp. 16-27, 2013. Article (CrossRef Link).
[50] F. Karakoç, H. Demirci, and A. E. Harmanci, "AKF: A key alternating Feistel scheme for lightweight cipher designs," *Inf. Process. Lett.*, vol. 115, pp. 359-367, 2015. Article (CrossRef Link).
[51] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, "The simeck family of lightweight block ciphers," in *Proc. of International Workshop on CHES 2015*, pp. 307-329, 2015. Article (CrossRef Link).
[52] V. Nalla, R. A. Sahu, and V. Saraswat, "Differential fault attack on SIMECK," in *Proc. of of the Third Workshop on Cryptography and Security in Computing Systems*, pp. 45-48, 2016. Article (CrossRef Link).
[53] S. H. Kim and I. Y. Lee, "IoT device security based on proxy re-encryption," *J. Ambient Intell. Humaniz. Comput.*, vol. 9, pp. 1267-1273, 2018. Article (CrossRef Link).
[54] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, "Twine: A lightweight, versatile block cipher," in *Proc. of ECRYPT Work. pn Light. Cryptogr. LC11*, pp. 146–169, 2011. [Online]. Available: http://www.nec.co.jp/rd/media/code/research/images/twine_LC11.pdf.
[55] Y. Wei, P. Xu, and Y. Rong, "Related-key impossible differential cryptanalysis on lightweight cipher TWINE," *J. Ambient Intell. Humaniz. Comput.*, vol. 10, pp. 509-517, 2019. Article (CrossRef Link).
[56] A. Biswas, A. Majumdar, S. Nath, A. Dutta, and K. L. Baishnab, "LRBC: a lightweight block cipher design for resource constrained IoT devices," *J. Ambient Intell. Humaniz. Comput.*, 2020. Article (CrossRef Link).
[57] A. Majumdar, A. Biswas, K. L. Baishnab, and S. K. Sood, "DNA based cloud storage security framework using fuzzy decision making technique," *KSII Trans. Internet Inf. Syst.*, vol. 13, pp.

3794-3820, 2019. Article (CrossRef Link).
[58] H. M. Heys, "A tutorial on linear and differential cryptanalysis," *Cryptologia*, vol. 26, pp. 189-221, 2002. Article (CrossRef Link).
[59] S. Nikova, V. Rijmen, and M. Schläffer, "Secure hardware implementation of nonlinear functions in the presence of glitches," *J. Cryptol.*, vol. 24, pp. 292-321, 2011. Article (CrossRef Link).
[60] J. Chen, M. Wang, and B. Preneel, "Impossible differential cryptanalysis of the lightweight block ciphers TEA, XTEA and HIGHT," in *Proc. of International Conference on AFRICACRYPT 2012*, pp. 117-137, 2012. Article (CrossRef Link).
[61] F. Karakoç, H. Demirci, and A. E. Harmanci, "Impossible differential cryptanalysis of reduced-round LBlock," in *Proc. of IFIP International Workshop on Information Security Theory and Practice*, pp. 179-188, 2012. Article (CrossRef Link).
[62] Y. Liu, D. Gu, Z. Liu, and W. Li, "Impossible differential attacks on reduced-round LBlock," in *Proc. of International Conference on ISPEC 2012*, pp. 97-108, 2012. Article (CrossRef Link).
[63] Y. Liu, D. Gu, Z. Liu, and W. Li, "Improved results on impossible differential cryptanalysis of reduced-round Camellia-192/256," *J. Syst. Softw.*, vol. 85, pp. 2451-2458, 2012. Article (CrossRef Link).
[64] O. Özen, K. VarIcI, C. Tezcan, and Ç. Kocair, "Lightweight block ciphers revisited: Cryptanalysis of reduced round PRESENT and HIGHT," in *Proc. of ACISP 2009*, pp. 90-107, 2009. Article (CrossRef Link).
[65] W. Wu, L. Zhang, and W. Zhang, "Improved impossible differential cryptanalysis of reduced-round Camellia," in *Proc. of International Workshop on SAC 2008*, pp. 442-456, 2008. Article (CrossRef Link).
[66] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials," in *Proc. of International Conference on EUROCRYPT 1999*, pp. 12-23, 2005. Article (CrossRef Link).
[67] Knudsen, "The security of feistel ciphers with six rounds or less," *J. Cryptol.*, vol. 15, pp. 207-222, 2002. Article (CrossRef Link).
[68] H. Gilbert and M. Minier, "A collisions attack on the 7-rounds Rijndael," *Natl. Inst. Stand. Technol.*, pp. 230–241, 2000. [Online]. Available: http://perso.citi.insa-lyon.fr/mminier/papers/GilbertMinier_AES00.pdf.
[69] M. I. Mihailescu and S. L. Nita, "Integral Cryptanalysis," in *Proc. of Cryptography and Cryptanalysis with C++20*, 2021.
[70] S. Das, "Halka: A Lightweight, Software Friendly Block Cipher Using Ultra-lightweight 8-bit S-box.," *IACR Cryptol. ePrint Arch.*, 2014.
[71] E. Biham, "New types of cryptanalytic attacks using related keys," *J. Cryptol.*, vol. 7, pp. 229-246, 1994. Article (CrossRef Link).
[72] A. Biryukov and D. Wagner, "Slide attacks," in *Proc. of International Workshop on FSE 1999*, pp., 245-259, 1999. Article (CrossRef Link).
[73] "Xilinx Inc, Form 10-K, Annual Report, Filing Date May 31, 2011," [Online]. Available: http://secdatabase.com. (Accessed August 21, 2020)
[74] P. Hämäläinen, T. Alho, M. Hännikäinen, and T. D. Hämäläinen, "Design and implementation of low-area and low-power AES encryption hardware core," in *Proc. of 9th EUROMICRO Conference on Digital System Design (DSD'06)*, 2006. Article (CrossRef Link).
[75] M. B. Abdelhalim, M. El-Mahallawy, and M. A. A. Elhennawy, "Design and Implementation of an Encryption Algorithm for use in RFID System," *Int. J. RFID Secur. Cryptogr.*, vol. 2, pp. 51-57, 2013. Article (CrossRef Link).
[76] L. Knudsen, G. Leander, A. Poschmann, and M. J. B. Robshaw, "PRINTcipher: A block cipher for IC-printing," in *Proc. of International Workshop on CHES 2010*, pp. 16-32, 2010. Article (CrossRef Link).
[77] Ratnadewi, R. P. Adhie, Y. Hutama, A. Saleh Ahmar, and M. I. Setiawan, "Implementation Cryptography Data Encryption Standard (DES) and Triple Data Encryption Standard (3DES) Method in Communication System Based Near Field Communication (NFC)," in *Prco. of J. Phys.: Conf. Ser*, vol. 954, 2018. Article (CrossRef Link).

[78] A. B. Mohamed, G. Zaibi, and A. Kachouri, "Implementation of RC5 and RC6 block ciphers on digital images," in *Proc. of Eighth International Multi-Conference on Systems, Signals & Devices*, 2011. Article (CrossRef Link).

[79] D. Smekal, J. Hajny, and Z. Martinasek, "Hardware-Accelerated Twofish Core for FPGA," in *Proc. of 2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, 2018. Article (CrossRef Link).

**Siddhartha Roy** received the B.E. degree from Visvesvaraya Technological University, Belgaum, India and M.Tech. degree from National Institute of Technology Meghalaya, Shillong, India. He is currently pursuing the Ph.D. degree in Electronics and Communication Engineering with the National Institute of Technology Silchar, Assam, India. His research interests include Cryptographic Algorithms, Digital system design and Computer Arithmetic.

**Saptarshi Roy** received the B.E. degree in Computer Science and Engineering from Maulana Abul Kalam Azad University of Technology, Kolkata, West Bengal, India and M.Tech. degree in Software Engineering from National Institute of Technology Rourkela, Odisha, India. He is currently working as a system engineer in Tata Consultancy Services, Kolkata, India. He has research interests in Cryptography, Natural Language Processing and Machine Learning.

**Arpita Biswas** did her B. Tech. in Computer Science Engineering from Tripura Institute of Technology, Narsingarh, India in 2012 and M.Tech from Punjab Technical University, India in 2014. Currently, she is pursuing her Ph.D. from National Institute of Technology Silchar, Assam, India. Her research areas are information security, integration of IoT with cloud computing and optimization. She is currently serving as an Assistant Professor at Karunya University, Coimbatore, India in Department of Computer Science and Engineering.

**Krishna Lal Baishnab** received the B.E. degree from Regional Engineering College Silchar (presently National Institute of Technology Silchar) in the year 1995 and M.Tech. degrees from Indian Institute of Technology Kharagpur, West Bengal, India in the year 2004. He completed Ph.D. from NIT Silchar, Assam. He started his professional carrier as a Production Engineer in "WIDECOM FAX and Plotter" Noida export processing zone- Noida Phase-II in the year 1996-1997. After that he became a Lecturer in Regional Engineering College Silchar from the year 17$^{th}$ March 1998. He is currently an Associate Professor with National Institute of Technology Silchar, India. He got awarded twice within 5$^{th}$ position in prestigious CADENCE Design contest 2011 and 2012. His current research interests include Analog/Mixed VLSI for Visual Computation, VLSI Design for Machine Learning, Cryptography, Machine learning for Medical science.