Journal of Ambient Intelligence and Humanized Computing (2023) 14:5773–5787
https://doi.org/10.1007/s12652-020-01694-9

ORIGINAL RESEARCH

LRBC: a lightweight block cipher design for resource constrained IoT devices

A. Biswas[1] · A. Majumdar[1] · S. Nath[1] · A. Dutta[2] · K. L. Baishnab[1]

Received: 3 August 2019 / Accepted: 3 January 2020 / Published online: 11 January 2020

Abstract
The internet of things (IoT) is now an in-demand technology that has been adopted in various applications and includes various embedded devices, sensors and other objects connected to the Internet. Due to the rapid development of this technology, it covers a significant portion of the research interests nowadays. IoT devices are typically designed for collecting different types of data from various sources and transmitting them in digitized form. However, data security is the burning issue in the IoT technology, which can broadly impact the privacy of crucial data. In this regard, a new lightweight encryption method called LRBC has been proposed in this work for resource constraint IoT devices which can provide data security at the sensing level. The LRBC has used the structural advantages of both substitution–permutation network (SPN) and Feistel structure together to achieve better security. Furthermore, the proposed method has been tested on NEXYS 4 DDR FPGA (Artix-7) trainer kit and implemented for application specific integrated circuit (ASIC) chip on TSMC 65 nm technology. The proposed algorithm consumes very less power of 11.40 µW and occupies a 258.9 GE (Gate Equivalent) area. Besides, a thorough security analysis shows that the proposed scheme ensures high security against various attacks with robustness. Moreover, the average avalanche effect of LRBC is found to be 58% and 55.75% concerning plaintext and key, respectively.

Keywords Lightweight encryption · IoT · Block cipher · FPGA · ASIC chip

1 Introduction

Internet of things (IoT) is the extremely focused research topic that has revolutionized the life of human beings through its ability to provide numerous advanced computing facilities at the doorstep. The word 'things' in IoT refers to a physical object that has a unique identifier or an embedded system with processing and automatic data transmission capability dedicated to a specific task. It has the potential to form a competent and ubiquitous connectivity between all the physical objects around us and the digital world. The whole world is currently witnessing a remarkable growth in IoT deployments and solutions. IoT directs the modern world to become smarter progressively through different challenging application areas such as smart healthcare (Majumdar et al. 2018a, b), smart agriculture (Ray 2017), smart energy grids (Majumdar et al. 2018a, b), home and building automation (Hui et al. 2017), smart traffic (Zhou et al. 2017) etc. In this new world of data, IoT is quickly expanding; however due to insufficient security and privacy measures its sustainable development can be affected a lot. Typically, IoT devices are responsible for collecting different types of sensed data through various sensors, out of them some may be highly confidential to the user's perspective and should not be revealed to others. As per the various industry survey, security is the most promising concern for industrial IoT users today. So, it is utmost necessary to integrate more security features in both the hardware and the device software to enhance the IoT device layer security and ensure the protection of crucial sensed information. In this venture, incorporating the cryptography concept at the physical constraints of IoT devices can be a favorable option. Data encryption at IoT devices before transmitting to back-end systems (cloud server) using cryptographic algorithms can help to maintain data integrity and inhibit data sniffing by hackers. Though cryptography based security is increasingly important, supporting different cryptographic algorithms

✉ A. Biswas
arpita.nits@ieee.org

[1] Department of Electronics and Communication Engineering, NIT Silchar, Silchar, India

[2] Department of Electronics and Communication Engineering, Assam University, Silchar, India

and standards within an IoT device is still quite challenging due to having large area and power overhead. Furthermore, all IoT encryption must be followed by a suitable and efficient encryption key management processes, as poor key management can compromise to overall security. IoT devices are basically lightweight that means designed with less storage space. Besides, IoT devices are also battery driven and thus low power consumption is preferable. So, the symmetric key cryptography algorithms are generally advisable for designing IoT devices due to their less processing power, storage space, complexity and bandwidth as compared to the asymmetric key cryptography algorithms.

Block cipher is a variant of secret key cryptography which had been adopted for designing resource-constrained computing devices for the past decade due to its easier software and hardware implementation, higher diffusion and error propagation features. It requires extremely restricted hardware resources as compared to the stream cipher. While designing block ciphers for wireless devices, the traditional block cipher like AES occupies almost 3400 GEs of the area (Eisenbarth et al. 2007) which is quite large to be fit into the device. A gate equivalent (GE) is a unit for measuring the manufacturing-technology-independent complexity of digital electronic circuits. For today's CMOS technologies, the silicon area of a two-input drive-strength-one NAND gate usually constitutes the technology-dependent unit area commonly referred to as gate equivalent. A specification in gate equivalents for a certain circuit reflects a complexity measure, from which a corresponding silicon area can be deduced for a dedicated manufacturing technology.

On the other hand, while constructing block ciphers, the traditional Feistel ciphers take much time and consume a large amount of energy as they need far more round functions than the substitution–permutation networks (SPN). In this regard, the combination of Feistel and SPN structure has been suggested in various literatures. The combined structure provides simple and small round functions which result in faster diffusion with lesser energy and storage consumption. Considering all the difficulties and vulnerabilities of small IoT devices, a lightweight symmetric key cryptographic strategy has been proposed herein by improving all the susceptibilities present in the existing literatures.

Additionally, the proposed cryptographic strategy has been designed by combining the Feistel and SPN structure together for ensuring faster diffusion. Moreover, a round key management strategy has also been accompanied by the encryption approach. In addition, an energy-efficient hardware device using modern technology with low node power has been designed which can use more gate operations to provide adequate security with less power consumption. Meanwhile, the prototype has been built upon FPGA board as the test bed. Furthermore, ASIC chip tape out has also been realized on TSMC 65 nm technology. The proposed scheme is supposed to be used in the healthcare, agriculture and other sectors where secure transmission of sensed data is a crucial factor. In this context, Table 1 lists the details of some sensors with which the designed LRBC ASIC can be embedded together for confirming an enhanced grade of sensing level security. However, the applicability of the LRBC is not limited to these listed sensors only and can be used with a variety of sensors and IoT devices as well if appropriately configured. In addition, various analyses have been presented in this paper which analyses the strength of the proposed scheme.

The rest of the paper has been organized as follows: Sect. 2 presents the related works in the relevant field in detail. The proposed lightweight cryptographic strategy has been presented in detail in Sect. 4. Section 5 presents the testing environment with results and different kinds of analysis. The paper has been concluded in Sect. 6.

2 Literature survey

The lightweight cryptography is strongly needed to deal with the data size, device power and cost of computing devices to a minimal level. So, while designing a cryptography algorithm dedicated to any small computing device, the main

Table 1 Description of different sensors

Sensor name	Application area	Data produced	Feature
MAX30205	Healthcare	16-bits	Measures human body temperature
CJMCU-6701	Healthcare	12-bits	Measure galvanic skin response (GSR)
AK9750 (IR Sensor)	Healthcare/agriculture	16-bits	Detects the human in motion
CS526-L	Agriculture	8-bits	Checks PH-balance
EMG-muscle sensor	Healthcare	16-bits, 8-bits	Measures electrical activity of muscles
AD8232	Healthcare	16-bits, 8-bits	Measures ECG signal
BMP180	Agriculture	19-bits	Measures atmospheric pressure
MPXV7002DP	Agriculture	14-bits	Measures air speed
MAX30100	Healthcare	16-bits	Measures the oxygen saturation (SPO2) of haemoglobin in blood

motive should be made it lightweight in every aspect such as memory usage, chip size, power consumption etc. (Singh et al. 2017). Though they were lightweight, lack of adequate security is one of the main reason behind discarding those. However, still many algorithms such as PRESENT (Andrey et al. 2007), CLEFIA (Shirai et al. 2007), KATAN/KTANTAN (De Canniere et al. 2009), LED (light encryption device) (Guo et al. 2011a, b), PICCOLO (Shibutani et al. 2011), PRINCE (Borghoff et al. 2012), PHOTON (Guo et al. 2011a, b), SPONGENT (Bogdanov et al. 2011), SIMON/SPECK (Beaulieu et al. 2015), LEA (low power encryption algorithm) (Hong et al. 2013), PRIDE (Albrecht et al. 2014), MIDORI (Banik et al. 2014), and TEA (tiny encryption algorithm) (Wheeler and Needham 1994) etc. are being followed and applied in various application areas. Some of the most popular recent lightweight block ciphers have been summarized in this section.

Li et al. (2018) introduced SFN (substitution–permutation Fiestel Network) encryption technique by following both of the SPN and Feistel network (FN) structure. Their approach was based on 64-bit input block and 96-bit key length out of which the rear 32-bit key was acted as a control signal and the remaining 64-bit was used for the purpose of key expansion and round encryption. Each bit in the 32-bit control signal conducted one round operation and used for selecting the working mode of the two structures (SPN/FN) such as key expansion or encryption and decryption. The hardware implementation of SFN required an area of 1876.04 GEs. A group of authors proposed a family of lightweight block ciphers called CHAM which was based on 4-branch Feistel structure performing ARX (Addition, Rotation, XOR) operations (Koo et al. 2017). It had a simple key schedule which was implemented without updating key status and thus required smaller areas during hardware implementation. In this approach, different round functions were configured so that it could reuse the reduced set of round keys iteratively. Researchers also (Jagdish et al. 2017) designed a lightweight block cipher algorithm named LiCi in which 31 consecutive rounds associated with 4×4 lightweight S-boxes were considered. It was designed to support 64-bits plaintext content along with 128 bits key length. LiCi utilized 1153 gate equivalents (GEs) of area, 1944 bytes of memory and 30 mW power. Bansod et al. (2017) proposed a SPN-based lightweight cryptography technique and named it BORON. It supported 64-bit plaintext block along with a key length of 128/80 bits. It was based on total 25 number of rounds where the 4-bit to 4-bit S-boxes followed by round permutations, shift, and XOR operations were used. Moreover, it was resilient to linear and differential attacks. BORON was implemented on both software and hardware platform and required 1939 GEs for a 128-bit key and 1626 GEs for an 80-bit key. Banik et al. (2017) revised the design strategy of PRESENT and proposed an improved SPN-based block cipher named GIFT. It replaced the costly S-box of PRESENT with a smaller and cheaper S-Box by introducing bit permutation in association with Difference Distribution Table (DDT)/Linear Approximation Table (LAT) of the S-Box. In this approach, two variations of GIFT were proposed namely GIFT-64 and GIFT-128 with a predefined set of 28 and 40 rounds, respectively while keeping the key size constant with 128-bits. Researchers introduced a 64-bit block cipher algorithm named QTL in which a combination of both SPN and the feistel structure concept was followed for the purpose of encryption (Lang et al. 2016). Moreover, it supported two different key length i.e. 64-bit and 128-bit. For the decryption, they followed the same approach as encryption except considering the round constants and round sub-keys in a reverse order only. In their approach, the key scheduling was excluded for reducing the memory and power consumption in hardware design. The hardware implementation of QTL required 1025.52 and 1206.52 GE area for the 64 and 128 bits keys modes respectively. The QTL is not resilient to the standard statistical attacks on block ciphers which is the main limitation of this approach (Sadeghi et al. 2017). Zhang et al. (2015) suggested an SPN-based block cipher algorithm 'RECTANGLE' which was designed to accept 64-bit plaintext and keys of either 80 or 128 bits for producing 64-bit ciphertext. In case of the 80-bit key, it consumed 74.31 μw power and 1600 GEs to implement the block cipher. Whereas, for the 128-bit key mode, the area consumption was 2064 GEs with a power consumption of 72.15 μw. Some of the advantages of this approach are like very hardware friendly design, better competitive software speed etc. Another group of researchers proposed a lightweight block cipher, AKF by incorporating the alternate key concept into the traditional Feistel cipher (Karakoç et al. 2015). In addition, authors also reintroduced the software oriented lightweight block cipher ITUbee in respect of AKF with 80-bit block size and 80-bit security key (Karakoç et al. 2013). ITUBEE used 20 rounds and key whitening layers for ensuring better security. This cipher used S-box of AES and reduced the required memory, power and time. ITUBEE was claimed to be resistant against the related key attack. Yang et al. (2015) proposed a family of compact and lightweight Feistel block ciphers, named SIMECK (SIMECK 32/64, SIMECK 48/96 and SIMECK 64/128) by amalgamating the advantages of both SIMON and SPECK. The hardware implementation was realized on both CMOS 130 nm and CMOS 65 nm techniques. SIMECK had been proven to be vulnerable to bit-flip fault attack and random-byte fault attack (Nalla et al. 2016). Guo et al. (2011a; b) proposed an SPN-type symmetric block cipher supporting algorithm named LED. It used a block size of 64 bits as input and considered either 64 bit (LED-64) or 128 bit (LED-128) as the key length. The LED was based on a sequence of identical rounds of encryption where each round comprised of three

operations such as AddConstants, SubCells, ShiftRows, and MixColumns Serial similar to the round function of AES. Authors claimed that their approach was resistant against related and single key attack. Researchers also used elliptic curve cryptosystem-based proxy re-encryption schemes in lightweight devices for enhancing security features (Kim and Lee 2018). However, the additional computation in the polynomial equation made their scheme insignificant. Suzaki et al. (2011) proposed a 36-round lightweight block cipher named TWINE having block size of 64 bits and key size of 80/128 bits. Whereas, Wei et al. (2019) re-evaluated the security of TWINE-80 against impossible differential cryptanalysis in related-key model and improved the traditional impossible differential attack by one round.

3 System model

A lightweight cryptographic chip can be deployed inside a sensor for transforming the conventional flow of data into an encrypted form. This will help in configuring the encryption in the sensing layer. Figure 1 represents the system model for clearly understanding the significance of the cryptographic chip inside a sensor. As the sensors generate analog signal so a sensor compatible analog to digital convertor (ADC) is needed to transform the analog signal to digital form that will act as the input to the cryptographic chip. Based on the cryptographic functions coded inside the chip, the digitized sensed data is transformed into the encrypted form and further transmitted to the cloud server through data analytics center for storage. Later on, the authorized users will only be permitted to access the encrypted data from the cloud server through the cloud service provider. Designing this kind of lightweight chip can be beneficial in several security concerned applications like smart healthcare, smart agriculture, domotic and home automation etc. In this regard, a lightweight cryptographic algorithm named LRBC has been proposed and the corresponding ASIC chip tape out has also been realized. Various related algorithms have been scrutinized and so as to enhance the security aspect of the system, a new lightweight resource constrained block cipher strategy with a small and simple step has been designed and named LRBC. The proposed algorithm is the amalgamation of feistel and Substitution–Permutation–Linear network (SPLN) with a simple key combination. Generally, the feistel structure uses a large number of rounds and only operates on half of the block. SPN structure applies confusion, diffusion strategy which increases the redundancy of plaintext and confirms a strongly encrypted ciphertext. Thus, the mixture of this two structure has resulted in a more shielded system than using those strategies distinctively. The main constraints for designing lightweight encryption algorithm are storage, power, memory and speed.

4 Overview of LRBC

The proposed algorithm has been implemented using simple logical operations like Ex-OR operations, Ex-NOR operations, concatenation etc. Therefore, the data has been enciphered in such a way that it is made possible for an authorized user only to easily decrypt it. As the main goal is to

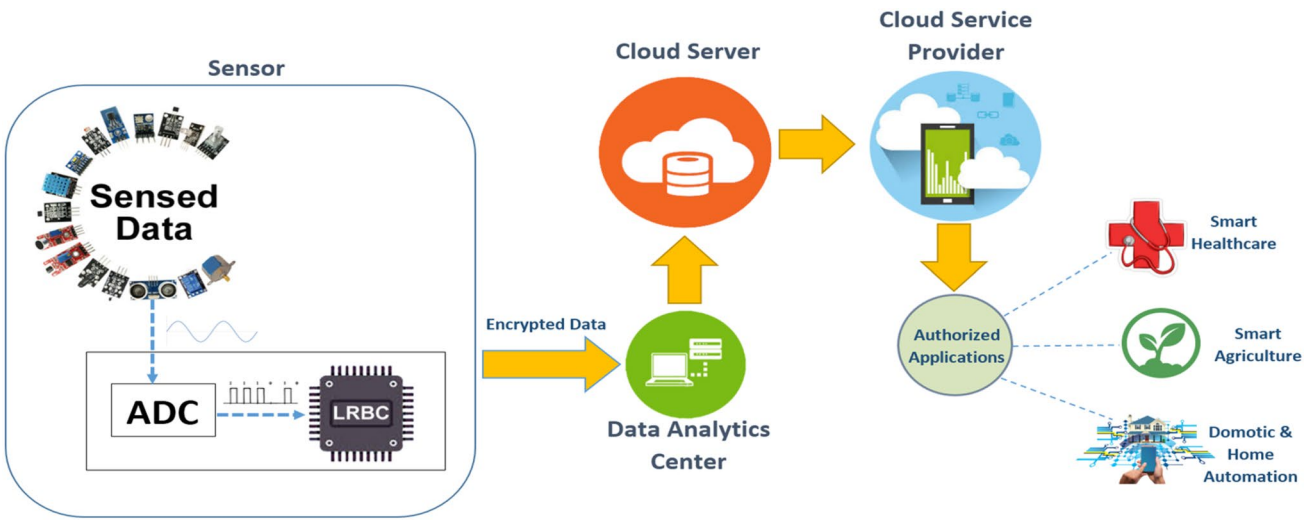

Fig. 1 System model

design a light weight encrypted system, thus, the proposed strategy always deals with 16-bit block of data. Hence, the long plaintext bit has been splitted into the several blocks each with 16 bit length of data which is being processed using a datapath and the same procedures will be repeated for consecutive data blocks. When all the data blocks are being processed, the results have been merged to produce the final encrypted text. The four number of keys with 4-bit each has been combined to each other and design 24 combinations. In this way the proposed algorithm consumes less power, occupies less storage space, memory and attain high speed for the resource constrained devices that means it achieves the solution against all the constrained mentioned above and provides better security against different cryptographic attacks.

4.1 Key consideration

In the proposed approach, four numbers of keys having 4-bit each (K^1, K^2, K^3 and K^4) have been used for both the encryption and decryption processes. As in this scenario, a lightweight encryption is desired, so a simple but strong combination of keys have been taken into consideration. From the abovementioned four keys, 24 number of possible combinations can be made which will be used during the round operation of encryption/decryption process. Hence, it guarantees as resilient to related key attacks. The design of the key combinations has been depicted in Fig. 2.

4.2 Encryption strategy

Encryption is the procedure to convert the data into indecipherable form by following confusion and diffusion strategies. So, it is being needed to make this scheme more robust such that it becomes difficult to decrypt the data for an unauthorized user. The interpretation of the proposed LRBC encryption technique has been illustrated in Algorithm 1 and the graphical representation for the same has been depicted in Fig. 3. The proposed algorithm goes through repeated steps including round transposition, F-function etc. for 24 number of consecutive rounds. Each round operation generates intermediate ciphertexts (ICT) that act as the input for the immediately next round. The output of the 24th round has been considered as the final ciphertext (CT).

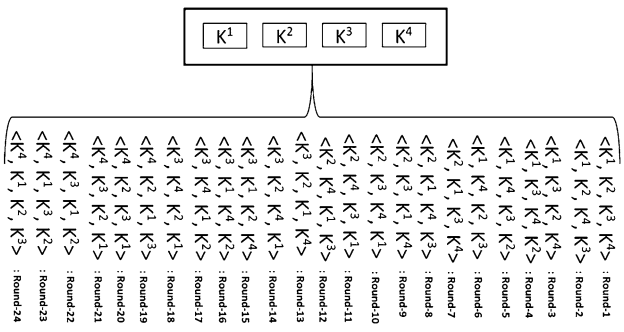

Fig. 2 Considered 4-bit key combinations

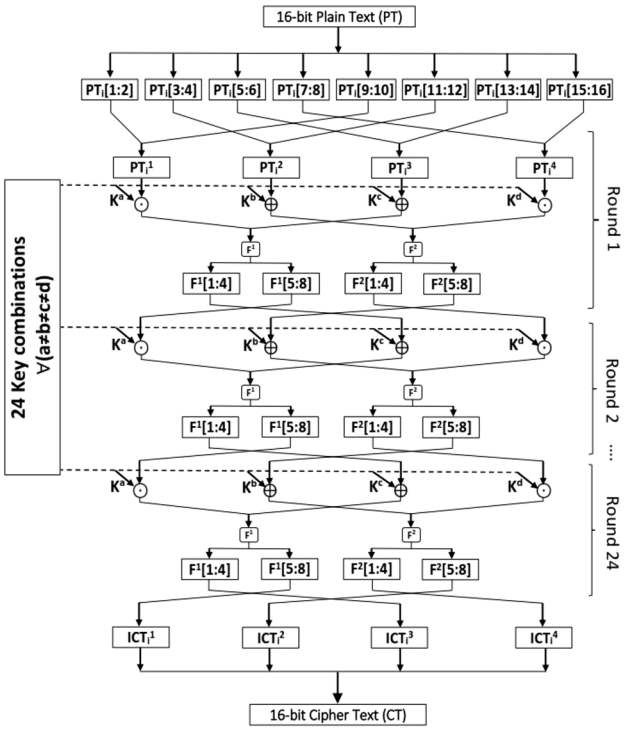

Fig. 3 Encryption process of LRBC

4.2.1 F-function generation

As per the proposed algorithm, every round has been configured with two F-functions which in turn comprised of three types of computing boxes namely, S-box, P-box, and L-box. Each type of box is responsible for performing different computations. Under every F-function, the computed output of S-box has been applied as the input of P-box and similarly the output of P-box has been applied as the input of L-Box. The computations performed by these boxes within an F-function has been illustrated in Algorithm 2. The operations performed inside each of this boxes can easily be understood using a suitable example.

Algorithm 1. LRBC Encryption

Input: 16-bit plaintext (PT), Four no's of 4-bit keys (K^a, K^b, K^c, K^d)
Output: 16-bit ciphertext (CT)

1. Read the byte values from the input file called plaintext (PT) and extract the byte values.
2. PT divided into 'n' number of blocks of 16 bits each (PT_i).
3. Initialize 'r' with value '1'.
4. Each PT_i is further sub-divided into 4 equal length parts $PT_i^k\big|_{\substack{1\le k\le 4\\ 1\le i\le n}}$ as,
$$PT_i^1 = PT_i[1] \parallel PT_i[2] \parallel PT_i[9] \parallel PT_i[10]$$
$$PT_i^2 = PT_i[3] \parallel PT_i[4] \parallel PT_i[11] \parallel PT_i[12]$$
$$PT_i^3 = PT_i[5] \parallel PT_i[6] \parallel PT_i[13] \parallel PT_i[14]$$
$$PT_i^4 = PT_i[7] \parallel PT_i[8] \parallel PT_i[15] \parallel PT_i[16]$$
5. Compute intermediate round cipher blocks as,
$$IC_i^1 = PT_i^1 \odot K^a \Big|_{\substack{1\le a\le 4\\ a\ne b\ne c\ne d}}$$
$$IC_i^2 = PT_i^2 \oplus K^b \Big|_{\substack{1\le b\le 4\\ b\ne a\ne c\ne d}}$$
$$IC_i^3 = PT_i^3 \oplus K^c \Big|_{\substack{1\le c\le 4\\ c\ne a\ne b\ne d}}$$
$$IC_i^4 = PT_i^4 \odot K^d \Big|_{\substack{1\le d\le 4\\ d\ne a\ne b\ne c}}$$
6. Generate F-Function as,
$F_i^1 = F\_Function(IC_i^1, IC_i^3)$;
$F_i^2 = F\_Function(IC_i^2, IC_i^4)$
7. Generate input for next round as,
$PT_i^1 = F_i^1[5:8]$; $PT_i^2 = F_i^2[5:8]$
$PT_i^3 = F_i^1[1:4]$; $PT_i^4 = F_i^2[1:4]$
$r = r + 1$
8. If (r < 24)
Go to step 5.
9. Else
Go to step 10.
10. $ICT_i^k\big|_{\substack{1\le k\le 4\\ 1\le i\le n}} = PT_i^k\big|_{\substack{1\le k\le 4\\ 1\le i\le n}}$
11. Generate Final Cipher as,
$CT = ICT_i^1 \parallel ICT_i^2 \parallel ICT_i^3 \parallel ICT_i^4$

4.2.2 Round transposition

After F-function operations, the data has to go through a round transposition operation which has been shown in Fig. 5. It has been used to provide an additional level of security over the data. As shown in figure, each input to the transposition method is of 4-bit. After each round of operation, the resultant bit values of the 4-bit output blocks have been treated as the input blocks for the next round.

Algorithm 2. F-Function

Input: Intermediate cipher blocks IC_i^1, IC_i^2, IC_i^3, IC_i^4.
Output: 16-bit ciphertext.

1. S-box computation,
$\mathbf{IS_i^1 = IC_i^1 \odot IC_i^3}$
$\mathbf{IS_i^2 = IC_i^1 \oplus 1}$
$\mathbf{IS_i^3 = IC_i^2 \odot IC_i^4}$
$IS_i^4 = IC_i^2 \oplus 0$
2. P-box computation,
$P_i^1 = IS_i^1[1] \parallel IS_i^2[4] \parallel IS_i^1[2] \parallel IS_i^2[3]$
$P_i^2 = IS_i^1[3] \parallel IS_i^2[2] \parallel IS_i^1[4] \parallel IS_i^2[1]$
$P_i^3 = IS_i^3[1] \parallel IS_i^4[4] \parallel IS_i^3[2] \parallel IS_i^4[3]$
$P_i^4 = IS_i^3[3] \parallel IS_i^4[2] \parallel IS_i^3[4] \parallel IS_i^4[1]$
3. L-box computation,
$\mathbf{T_i[1] = (P_i^1[1] \oplus P_i^2[4])}$; $\mathbf{X_i[1] = (P_i^1[1] \odot 0)}$
$\mathbf{T_i[2] = (P_i^1[2] \odot P_i^2[3])}$; $\mathbf{X_i[2] = (P_i^1[2] \oplus 1)}$
$\mathbf{T_i[3] = (P_i^1[3] \oplus P_i^2[2])}$; $\mathbf{X_i[3] = (P_i^1[3] \odot 0)}$
$\mathbf{T_i[4] = (P_i^1[4] \odot P_i^2[1])}$; $\mathbf{X_i[4] = (P_i^1[4] \oplus 1)}$
$\mathbf{T_i[5] = (P_i^3[1] \oplus P_i^4[4])}$; $\mathbf{X_i[5] = (P_i^2[1] \odot 0)}$
$\mathbf{T_i[6] = (P_i^3[2] \odot P_i^4[3])}$; $\mathbf{X_i[6] = (P_i^2[2] \oplus 1)}$
$\mathbf{T_i[7] = (P_i^3[3] \oplus P_i^4[2])}$; $\mathbf{X_i[7] = (P_i^2[3] \odot 0)}$
$T_i[8] = (P_i^3[4] \odot P_i^4[1])$; $X_i[8] = (P_i^2[4] \oplus 1)$
$\mathbf{L_i(1) = T_i[1] \parallel X_i[4] \parallel T_i[2] \parallel X_i[3] \parallel T_i[3] \parallel X_i[2] \parallel T_i[4] \parallel X_i[1]}$
$L_i(2) = T_i[5] \parallel X_i[8] \parallel T_i[6] \parallel X_i[7] \parallel T_i[7] \parallel X_i[6] \parallel T_i[8] \parallel X_i[5]$
$z = L_i(1) \parallel L_i(2)$
4. End.

4.3 Decryption strategy

It is undoubtedly important to have a successful decryption process while designing a strong encryption process. In this work, the decryption process of the proposed algorithm has been designed as the exactly same as encryption technique but in the reverse order.

5 Design principle

5.1 F-function

As discussed earlier, the 16-bit plaintext has been separated into four equal parts of four bits each. The F-function consists of S-box, P-box and L-box. The design of the proposed F-function is light and secure. According to literature, 8×8 S-box provides better security but takes more space for hardware implementation. In this regard, two numbers of

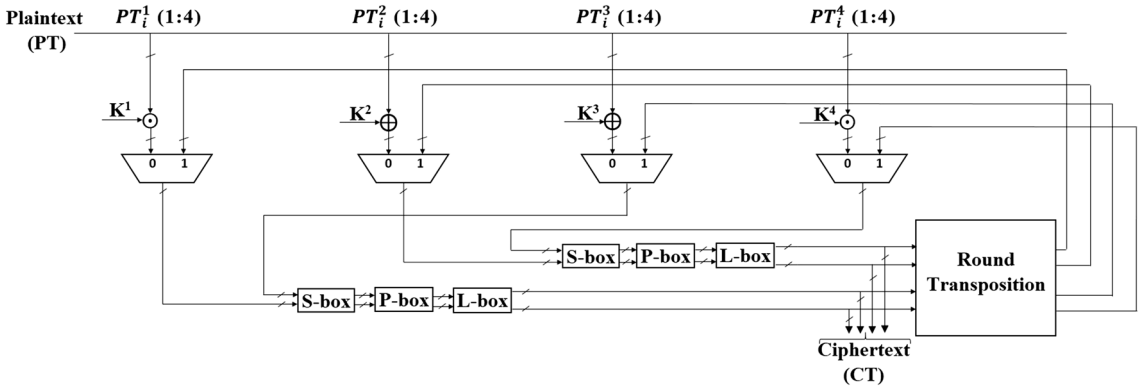

Fig. 4 Data path of LRBC scheme

Table 2 Area requirement of LRBC

Module function	Area (GE)
Key storage register	86
4 number of mux	2
16 number of 1-bit data Ex-ORs	28.3
8 number of 4-bit data Ex-ORs	56.6
Round transposition	0
Data register	86
Total	258.9

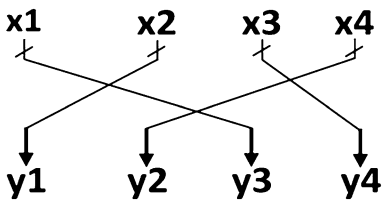

Fig. 5 LRBC round transposition process

4×4 S-boxes have been designed for the proposed algorithm because it takes less area and consumes less power as compared to the 8×8 S-box.

The P-boxes configured inside the F-functions have been designed by incorporating concatenation operations among the data bits. Similarly, the L-boxes have been arranged using four Ex-OR and Ex-NOR gate operations. Finally, the output data bits have been concatenated as directed in Algorithm 2. All the operations performed on these boxes are lightweight in nature since those consume less space as well as power for hardware implementation. Figure 4 shows the complete data path design of the proposed algorithm, which consists of four numbers of 2:1 multiplexor, twenty four numbers of data Ex-OR, data registers, key storage etc. Generally, the area of hardware is specified in gate equivalent (GE). Moreover, one GE is considered to be equivalent to the area of two input NAND gate. For example, a 4-bit 2:1 mux require 0.5 GE area (Lang et al. 2016). Each 4-bit Ex-OR operation requires 7.075 GE (Lang et al. 2016). The detailed area requirements for various components used while designing the proposed algorithm have been listed in Table 2. The proposed algorithm has been designed by considering all the possible constraints.

5.2 Round transposition

The output values of the f-functions in a round such as ×1, ×2, ×3 and ×4 have been directly sent as the next round input blocks y3, y1, y4 and y3 respectively. Each connection is performed by simple wiring to each other as per the Fig. 5. All the wiring shown in the figure is capable to transmit 4-bits of data at a time.

6 Simulation and hardware implementation

The overall workflow of this proposed lightweight encryption technique has been shown in Fig. 6.

In the 1st step the encryption algorithm has been developed and it is realized in Verilog Code. The Code is simulated in Xilinx-Vivado tool and the result has been shown in Fig. 7. As shown in the figure, the algorithm uses $(0006)_{16}$ as a sample plaintext and generates $(17eb)_{16}$ as the final ciphertext. The plaintext has been represented as 'X' and each intermediate ciphertext has been represented as 'Y_i'. The final ciphertext generated after 24 round operation has been represented as 'Y'. After successfully completion of the frontend part, it has been implemented on NEXYS 4 DDR FPGA (Artix-7) trainer kit to test the real time expediency of the proposed approach. Figure 8 shows the FPGA implementation result of the same sample. Each high output bit has been indicated by green LED.

Moreover, the code has been implemented on Synopsys Design-vision tool attached with TSMC 65 technology

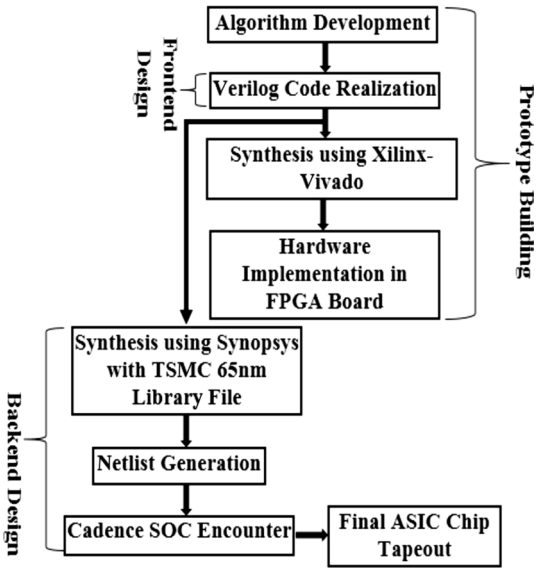

Fig. 6 Implementation flow

library file, which generates netlist file after synthesis. Further the netlist file has been used for backend design using Cadence Encounter Digital Implementation System (RTL to GDSII).Fig. 9 describes the final ASIC chip tapeout with the layout of the proposed algorithm. Power analysis has been performed from which it has been observed that power consumed by the proposed design is 11.40 µW which is less than most of the reported design while maintaining security. This is due to the use of 4-bit internal data operations in the proposed design. However, the comparison in terms of power has not been included in Table 6 as the power consumption mostly depends on the technology used. Moreover, the power comparison between different techniques designed in various technologies cannot be treated as justified. Furthermore, simulated values of power mostly depend on the simulation methods used, and the effort spent.

7 Security analysis

The security strength of the proposed approach has been evaluated through two kinds of analysis namely, avalanche effect analysis and attack analysis.

7.1 Avalanche effect

It is treated as an important security analysis in the cryptography that measures the goodness ratio of an encryption technique. This analysis is basically performed to realize the fact that even a change of 1 bit in plaintext or key may cause an enormous change in ciphertext. If the change affects at least half of the bits that means 50% of the ciphertext, then it will be treated as a good avalanche effect (Majumdar et al. 2019). Higher avalanche effect signifies higher security. The avalanche effect can be observed by following the Eq. (1). Here, for a particular plaintext block, the hamming distance has been changed randomly up to five bit during observation

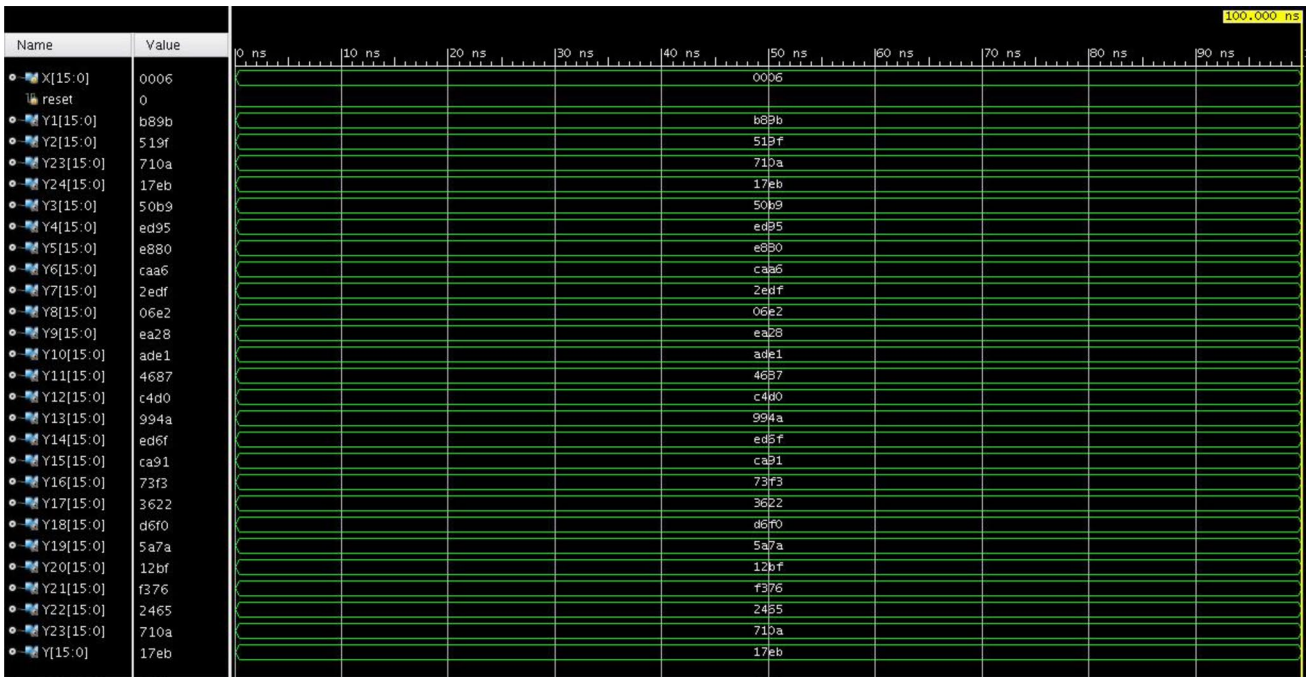

Fig. 7 Simulation result from Xilinx-Vivado

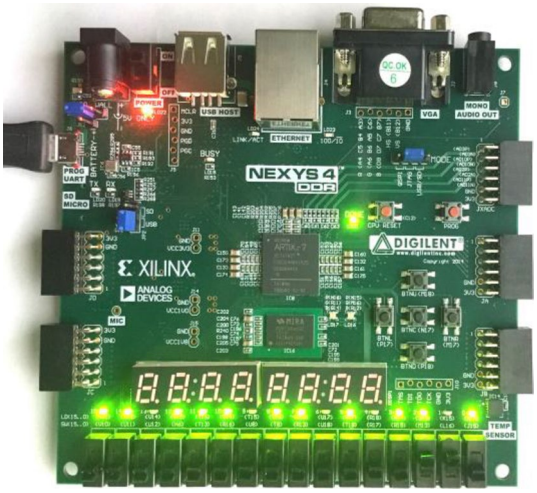

Fig. 8 FPGA implementation result of 16 bit Plaintext

and form fivedifferent test cases. Based on these test cases, a comparative study of LRBC with other state of the art lightweight encryption algorithms has been performed as shown in Table 3. Table 4 shows the similar observation considering the key while keeping a fixed plaintext. Figures 10 and 11 show the graphical representation of the avalanche effect for different test cases for LRBC with respect to plaintext and key respectively

$$\text{AE} = \frac{\text{Number of changed bit in ciphertext}}{\text{Number of bits in ciphertext}} \times 100\% \quad (1)$$

Moreover, a graphical comparative analysis on average avalanche effects w.r.t both plaintext and key has been represented in Fig. 12. After this analysis, the average avalanche effect for the proposed approach has been found to be 58% w.r.t plaintext and 55.75% w.r.t key which is better than the other state of the art algorithms.

8 Attack analysis

The proposed approach has been designed with simple but logically strong operational steps. The attack analysis of the proposed approach has been illustrated below, where the process of defense against various attacks has been explained.

8.1 Linear attack

This type of attack is known as known plaintext attack. In this case, attacker has the knowledge about some random plaintext and their corresponding ciphertext. The main goal is to recover the key, used in the encryption process. Basically, an attacker tries to make a linear relationship between the plaintext and their corresponding ciphertext using the Eq. (2) to figure out the probability of the equation to be satisfied (Heys 2002; Majumdar et al. 2019).

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

where x_1, $x_2 \ldots x_n$ are the random plaintext bit values and y_1, $y_2 \ldots y_n$ are their corresponding ciphertext bit values. For an encryption process, if the probability of satisfying the Eq. (2) is greater than or smaller greater than $1/2^n$, where

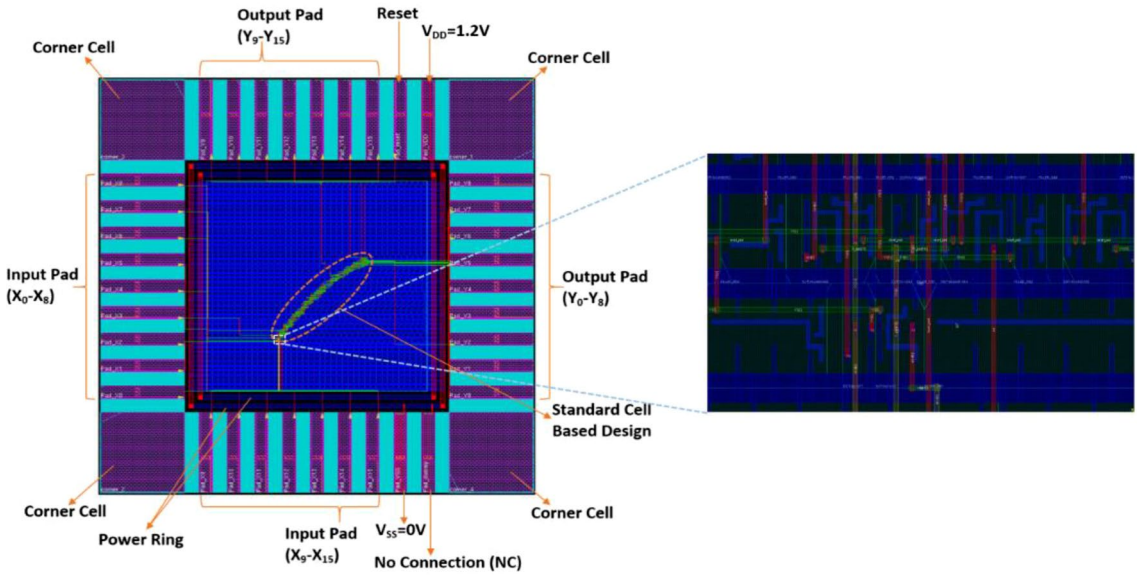

Fig. 9 ASIC chip tapeout of LRBC on TSMC 65 nm technology

Table 3 Average Avalanche effect analysis w.r.t plaintext

Algorithm	TC	HD-1bit (%)	HD-2bit (%)	HD-3bit (%)	HD-4bit (%)	HD-5bit (%)	Avg. AE (%)
LRBC	1	56.25	50.00	43.75	50.00	62.50	58.00
	2	50.00	56.25	68.75	75.00	62.50	
	3	62.50	56.25	ss	68.75	50.00	
	4	43.75	50.00	56.25	62.50	68.75	
	5	62.50	50.00	50.00	56.25	62.50	
QTL (Lang et al. 2016)	1	64.06	35.93	50.00	50.00	65.62	52.56
	2	50.00	54.68	45.31	46.87	46.87	
	3	59.37	51.56	45.31	59.37	48.43	
	4	48.43	50.00	51.56	70.31	78.13	
	5	37.50	48.43	53.13	65.62	37.50	
SIMECK (Yang et al. 2015)	1	50.00	35.93	50.00	18.75	78.13	53.00
	2	50.00	78.13	50.00	25.00	46.87	
	3	59.37	59.37	45.31	46.88	50.00	
	4	78.13	54.68	50.00	53.13	81.25	
	5	62.50	35.93	78.13	50.00	37.50	
PRINCE (Borghoff et al. 2012)	1	54.68	43.75	50.00	65.62	70.31	51.18
	2	64.06	43.75	50.00	50.00	67.18	
	3	54.68	37.50	64.06	48.43	50.00	
	4	54.68	34.37	39.06	46.87	50.00	
	5	50.00	37.50	34.37	50.00	68.75	
PRINT (Lars et al. 2010)	1	52.08	50.00	29.17	50.00	54.17	49.08
	2	52.08	37.50	31.25	45.83	58.33	
	3	72.91	43.75	50.00	56.25	50.00	
	4	50.00	50.00	50.00	56.25	50.00	
	5	50.00	20.83	50.00	54.17	62.50	
TEA (Abdelhalim et al. 2012)	1	50.00	34.37	42.18	57.81	46.87	49.12
	2	50.00	50.00	40.63	50.00	45.31	
	3	50.00	39.06	50.00	50.00	50.00	
	4	35.93	39.06	59.37	78.13	50.00	
	5	37.50	40.63	59.37	62.50	59.37	
LED (Guo et al. 2011a, b)	1	62.50	56.25	59.37	35.93	56.25	52.83
	2	48.43	50.00	50.00	59.37	65.50	
	3	50.00	43.75	78.13	50.00	57.81	
	4	59.93	62.50	45.31	50.00	40.63	
	5	48.43	50.00	46.87	50.00	43.75	

TC test case, *HD* hamming distance, *AE* Avalanche effect

n = number of bits in the plaintext, then the cipher is not secured under this than $1/2$, then it will be treated as vulnerable and can be possible to attack the cipher. The probability deviation from the value $1/2$ is called bias. A bias value near to $1/2$ represents better security against linear attack. Table 5 shows an example of linear probability analysis of the proposed S-box considering three sample linear expressions. It has been observed that the linear probability for most of linear expressions is exactly $1/2$ and the bias value is $1/2 - 1/2 = 0$. Therefore, the proposed technique provides better security against the known plaintext attack or linear attack.

8.2 Differential attack

This type of attack is called chosen plaintext attack. This type of attack focuses on the high probability of the occurrence of output differences with respect to a given input

Table 4 Average Avalanche effect analysis w.r.t key

Algorithm	TC	HD-1bit (%)	HD-2bit (%)	HD-3bit (%)	HD-4bit (%)	HD-5bit (%)	Avg. AE (%)
LRBC	1	50.00	43.75	50.00	56.25	50.00	55.75
	2	50.00	50.00	50.00	56.25	56.25	
	3	50.00	56.25	50.00	62.50	50.00	
	4	68.75	68.75	62.50	62.50	81.25	
	5	50.00	50.00	56.25	68.75	43.75	
QTL (Lang et al. 2016)	1	48.43	34.37	48.43	42.18	78.13	50.31
	2	39.06	32.81	48.43	57.81	48.43	
	3	39.06	57.81	48.43	69.37	53.13	
	4	58.68	50.00	64.06	56.25	48.43	
	5	39.06	39.06	53.12	56.25	46.87	
SIMECK (Yang et al. 2015)	1	46.88	34.37	50.00	18.75	78.13	51.25
	2	53.13	31.25	50.00	56.25	81.25	
	3	50.00	59.37	46.88	78.13	50.00	
	4	40.63	50.00	62.50	50.00	59.37	
	5	34.37	37.50	50.00	50.00	62.50	
PRINCE (Borghoff et al. 2012)	1	48.43	35.93	42.18	50.00	62.50	49.06
	2	46.87	45.31	42.18	50.00	56.25	
	3	45.31	50.00	32.81	48.43	62.50	
	4	50.00	48.43	50.00	46.87	64.06	
	5	50.00	50.00	50.00	48.43	50.00	
PRINT (Lars et al. 2010)	1	47.92	50.00	20.83	33.33	31.25	46.42
	2	41.67	43.75	37.50	35.42	29.17	
	3	41.67	52.08	50.00	56.25	50.00	
	4	62.50	50.00	72.91	62.50	50.00	
	5	50.00	50.00	41.66	50.00	50.00	
TEA (Abdelhalim et al. 2012)	1	42.18	50.00	68.75	50.00	67.18	47.12
	2	40.63	34.37	50.00	43.75	68.75	
	3	42.18	34.37	37.50	32.81	50.00	
	4	50.00	37.50	67.18	32.81	70.31	
	5	40.63	50.00	39.06	34.37	43.75	
LED (Guo et al. 2011a, b)	1	50.00	68.75	50.00	78.13	48.43	50.37
	2	59.37	50.00	39.06	50.00	70.31	
	3	57.81	48.43	39.06	50.00	40.63	
	4	40.63	50.00	50.00	50.00	50.00	
	5	40.63	50.00	42.18	35.94	50.00	

TC test case, *HD* hamming distance, *AE* Avalanche effect

difference (Heys 2002; Majumdar et al. 2019). For example, X is a set of input plaintext bits to the proposed S-box such as $X = [x_1, x_2 \ldots x_n]$ and Y is the corresponding set of ciphertext bits such as $Y = [y_1, y_2 \ldots y_n]$. The difference between two plaintexts is represented as $\alpha = x_1 \bigoplus x_2$ and the difference between their corresponding ciphertext is represented as $\beta = y_1 \bigoplus y_2$. An output difference β for a given input difference α is denoted as differential pair (α, β). If the occurrence of β for the given α is much particular type of attack. Generally it is necessary to reduce the difference pair probability to secure the cipher under differential attack or chosen ciphertext attack. Several combinations of 16-bit 'α' have been tested with various 16-bit plaintexts for generating the corresponding values of β. Consequently the (α, β) difference pair for the proposed approach has been observed. The maximum differential probability observed is $9/2^{16}$. On the other hand, the most satisfactory differential probability found is $2/2^{16}$ which is quite satisfactory for any encryption approach. Hence, it can be concluded that the proposed algorithm is protected against differential attack.

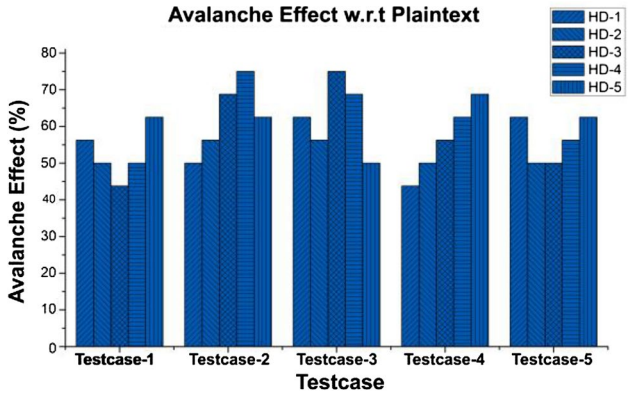

Fig. 10 Avalanche effect analysis w.r.t plaintext

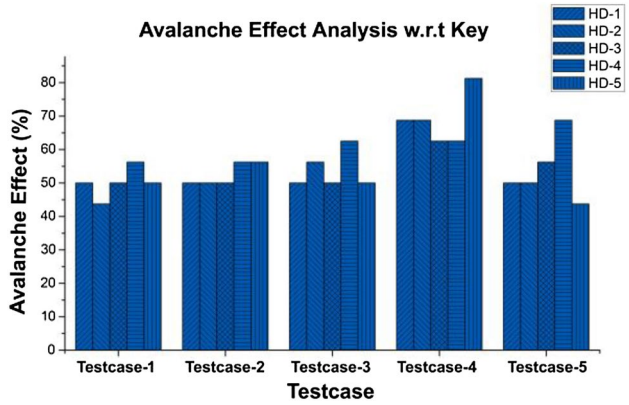

Fig. 11 Avalanche effect analysis w.r.t key

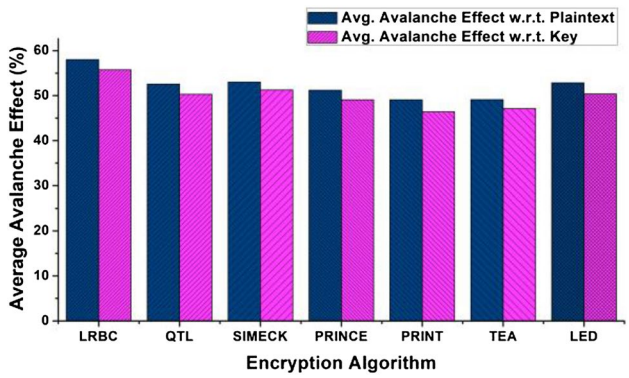

Fig. 12 Comparative analysis on average avalanche effect

8.3 Side channel attack

Side channel attack is based on the information yielded from the implementation of the encryption algorithm (Nikova et al. 2011). The proposed algorithm has been resisted from the side channel attack as it uses the random number to the encryption operation and the secret key which has been used is independent of the plaintext being choosen. The input output dependency is also very small for the proposed algorithm. Moreover, the proposed scheme is also able to provide security against other attacks. From the cryptanalysis point of view, the key and the round functions used in any cryptographic approach are two most attack prone parts. In this respect, round functions and the key(s) should be formed seriously while designing any cryptographic approach. In related key attack, attackers observe the bit patterns of ciphertext and identify the original key bit. It can only be possible whenever same key is used for all the rounds in an encryption process (Banik et al. 2017; Karakoç et al. 2013). Since different keys have been used in each round, the proposed encryption approach guarantees safety against this type of attack. Therefore, it would be harder for an attacker to realize the exact pattern of the keys used in the proposed approach.

9 Functionality analysis

Table 6 listed a brief overview of some well known state of the art lightweight cryptographic algorithms with their characterics and performance measures. The functionality based comparative analysis of the other algorithms with the proposed LRBC has been drawn based on the parameters such as block size, key size, structure, round, logic process, frequency, area etc. From Table 6, it can be noticed that most of the algorithms had been designed to support either 32 or 64 bit of block sizes with varying key sizes. Moreover, the structure of the algorithms are either SPN or Feistel based. Out of these, only QTL had availed the advantages of the mixed structural strategies.

Furthermore, after observing the listed values it can also be noticed that small block size and key size helps in fast diffusion of plaintext bit and provide better security. Similar to QTL, the proposed cipher also used the combined structures. However, the proposed LRBC used the combination of Feistel and SPN structures with 24 consecutive rounds which facilitates a more immune cipher. Moreover, similar to SIMECK, 65 nm technology file is used to design the proposed LRBC which allows more gates but occupies less area than others. As a result, LRBC fulfilled all the requirements related to light-weight resource constrained devices and outperforms the others in performance.

10 Conclusion

This work designs and implements a new lightweight block cipher named LRBC for resource constrained IoT devices. The proposed cipher enhances the data security

Table 5 Linear probability analysis

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

Table 6 Functionality analysis of lightweight cryptography approaches

Ciphers	Block size (bit)	Key size (bit)	Structure	Round	Logic process (μm)	Max. frequency (KHz)	Area (GE)
AES (Hamalainen et al. 2006)	8	128	SPN	16	0.13	153,000	3100
PRESENT (Andrey et al. 2007)	64	80	SPN	32	0.18	100	1570
LED (Guo et al. 2011a, b)	64	128	SPN	48	0.13	78,130	3407
PRINCE (Borghoff et al. 2012)	64	128	SPN	12	0.13	100	3491
TEA (Abdelhalim et al. 2012)	64	128	Feistel	32	0.35	100	1140
QTL (Lang et al. 2016)	64	64	Feistel, SPN	16	0.18	100	1026
RECTANGLE (Wentao et al. 2015)	64	80	SPN	26	0.13	100	1599.5
SIMON (Beaulieu et al. 2015)	32	64	Feistel	32	0.13	100	523
SPECK (Beaulieu et al. 2015)	32	64	Feistel	22	0.13	100	580
SIMECK (Yang et al. 2015)	32	64	Feistel	32	0.065	100	488
PRINT (Lars et al. 2010)	48	80	SPN	48	0.18	100	402
Proposed LRBC	16	16	Feistel, SPN	24	0.065	100	258.9

by incorporating the benefits of both feistel and SPN structure associated with an additional concept of linear box. Moreover, LRBC generates more number of active S-boxes which helps in resisting differential and linear attack. Minimal chip area and power are the two additional advantages of this scheme since it uses 4-bit internal data operations and optimized design. Additionally, the proposed design enhances the security in a great extent since it uses large number of gates. Rigorous security analysis shows that the proposed scheme guarantees high security with a balanced area and power consumption.

Acknowledgements This publication is an outcome of the R&D work undertaken project under the Visvesvaraya Ph.D Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation.

Compliance with ethical standards

Conflict of interest The authors have no conflict of interests to declare.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Abdelhalim M, El-Mahallawy M, Ayyad M, Elhennawy A (2012) Design and Implementation of an Encryption Algorithm for use in RFID System. Int J RFID Security Cryptogr (IJRFIDSC) 1(1/2):15–22

Albrecht MR, Driessen B, Kavun EB, Leander G, Paar C, Yalçın T (2014) Block ciphers-focus on the linear layer (feat. PRIDE). In: Proc of international cryptology conference. Springer, Berlin, Heidelberg, pp 57–76

Andrey B, Knudsen LR, Leander G, Paar C, Poschmann A, Robshaw MJB, Seurin Y, Vikkelsoe C (2007) PRESENT: an ultra-lightweight block cipher. Proceedings of international workshop on cryptographic hardware and embedded systems. Springer, Berlin, pp 450–466

Banik S, Bogdanov A, Isobe T, Shibutani K, Hiwatari H, Akishita T, Regazzoni F (2014) Midori: a block cipher for low energy. Proc of international conference on the theory and application of cryptology and information security. Springer, Berlin, pp 411–436

Banik S, Pandey SK, Peyrin T, Sasaki Y, Sim SM, Todo Y (2017) GIFT: a small PRESENT. Proc Int Conf Cryptogr Hardw Embedded Syst Springer Cham 2017:321–345

Bansod G, Pisharoty N, Patil A (2017) BORON: an ultra-lightweight and low power encryption design for pervasive computing. Front Inf Technol Electr Eng 18(3):317–331

Beaulieu R, Treatman-Clark S, Shors D, Weeks B, Smith J, Wingers L (2015) The SIMON and SPECK lightweight block ciphers. Proc of 52nd conference on design automation (DAC). ACM/EDAC/IEEE, San Francisco, pp 1–6

Bogdanov A, Knežević M, Leander G, Toz D, Varıcı K, Verbauwhede I (2011) SPONGENT: a lightweight hash function. Proc of international workshop on cryptographic hardware and embedded systems. Springer, Berlin, pp 312–325

Borghoff J, Canteaut A, Güneysu T, Kavun EB, Knezevic M, Knud-sen LR, Leander G, Nikov V, Paar C, Rechberger C, Rombouts P, Thomsen SS, Yalc T (2012) PRINCE—a low-latency block cipher for pervasive computing applications. In: Proc of ASIACRYPT 2012, Springer, pp 208–225

De Canniere C, Dunkelman O, Knežević M (2009) KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers. Proc of cryptographic hardware and embedded systems-CHES 2009. Springer, Berlin, pp 272–288

Eisenbarth T, Kumar S, Paar C, Poschmann A, Uhsadel L (2007) A survey of lightweight-cryptography implementations. IEEE Des Test Comput 24(6):522–533

Guo J, Peyrin T, Poschmann A (2011a) The PHOTON family of lightweight hash functions. Proc Annu Cryptol Conf Springer Berlin Heidelberg 2011:222–239

Guo J, Peyrin T, Poschmann A, Robshaw M (2011b) The LED block cipher. Proc of cryptographic hardware and embedded systems-CHES 2011. Springer, Berlin, pp 326–341

Hamalainen P, Alho T, Hannikainen M, Hamalainen TD (2006) Design and implementation of low-area and low-power AES encryption hardware core. In: Proc of 9th EUROMICRO conference on digital system design: architectures, methods and tools, DSD 2006, IEEE, pp 577–583

Heys HM (2002) A tutorial on linear and differential cryptanalysis. Cryptologia 26(3):189–221

Hong D, Lee JK, Kim DC, Kwon D, Ryu KH, Lee DG (2013) LEA: a 128-bit block cipher for fast encryption on common processors. Int Workshop Inf Secur Appl Springer Cham 2013:3–27

Hui TK, Sherratt RS, Sanchez DD (2017) Major requirements for building smart homes in smart cities based on internet of things technologies. Future Gen Comput Syst 76:358–369

Jagdish P, Bansod G, Kant KS (2017) LiCi: a new ultra-lightweight block cipher. In: Emerging trends and innovation in ICT (ICEI), international conference on IEEE, pp 40–45

Karakoç F, Demirci H, Harmancı AE (2013) ITUbee: a software oriented lightweight block cipher. Proc of international workshop on lightweight cryptography for security and privacy. Springer, Berlin, pp 16–27

Karakoç F, Demirci H, Harmancı AE (2015) AKF: a key alternating Feistel scheme for lightweight cipher designs. Inf Process Lett 115(2):359–367

Kim S, Lee I (2018) IoT device security based on proxy re-encryption. Journal of Ambient Intelligence and Humanized Computing 9(4):1267–1273

Koo B, Roh D, Kim H, Jung Y, Lee DG, Kwon D (2017) CHAM: a family of lightweight block ciphers for resource-constrained devices. Proc Int Conf Inf Secur Cryptol Springer Cham 2017:3–25

Lang L, Liu B, Wang H (2016) QTL: a new ultra-lightweight block cipher. Microprocess Microsyst Elsevier 45:45–55

Lars K, Leander G, Poschmann A, Robshaw MJB (2010) PRINTcipher: a block cipher for IC-printing. Proc of international workshop on cryptographic hardware and embedded systems. Springer, Berlin, pp 16–32

Li L, Liu B, Zhou Y, Zou Y (2018) SFN: a new lightweight block cipher. Microprocess Microsyst 60:138–150

Majumdar A, Debnath T, Sood SK, Baishnab KL (2018a) Kyasanur forest disease classification framework using novel extremal optimization tuned neural network in fog computing environment. J Med Syst 42(10):187

Majumdar A, Laskar NM, Biswas A, Sood SK, Baishnab KL (2018b) Energy efficient e-healthcare framework using HWPSO-based clustering approach. J Intell Fuzzy Syst 36(5):1–13

Majumdar A, Biswas A, Baishnab KL, Sood SK (2019) DNA based cloud storage security framework using fuzzy decision making technique. KSII Trans Internet Inf Syst 13(7):3794–3820

Nalla V, Sahu RA, Saraswat V (2016) Differential fault attack on SIMECK. In: Proc of the 3rd workshop on cryptography and security in computing systems, ACM, pp 45–48

Nikova S, Rijmen V, Schläffer M (2011) Secure hardware implementation of nonlinear functions in the presence of glitches. J Cryptol 24(2):292–321

Ray PP (2017) Internet of things for smart agriculture: technologies, practices and future direction. J Ambient Intell Smart Environ 9(4):395–420

Sadeghi S, Bagheri N, Abdelraheem MA (2017) Cryptanalysis of reduced QTL block cipher. Microprocess Microsyst 52:34–48

Shibutani K, Isobe T, Hiwatari H, Mitsuda A, Akishita T, Shirai T (2011) Piccolo: an ultra-lightweight blockcipher. Proc Int Workshop Cryptogr Hardw Embedded Syst Springer Berlin Heidelberg 2011:342–357

Shirai T, Shibutani K, Akishita T, Moriai S, Iwata T (2007) The 128-bit blockcipher CLEFIA. Proc of international workshop on fast software encryption. Springer, Berlin, pp 181–195

Singh S, Sharma PK, Moon SY, Park JH (2017) Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. J Ambient Intell Human Comput 1–18

Suzaki T, Minematsu K, Morioka S, Kobayashi E (2011) Twine: a lightweight, versatile block cipher. In: ECRYPT workshop on lightweight cryptography, pp 146–169

Wei Y, Xu P, Rong Y (2019) Related-key impossible differential cryptanalysis on lightweight cipher TWINE. J Ambient Intell Human Comput 10(2):509–517

Wentao Z, Bao Z, Lin D, Rijmen V, Yang B, Verbauwhede I (2015) RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Sci China Inf Sci Springer Verlag Berlin Heidelberg 58(12):1–15

Wheeler DJ, Needham RM (1994) TEA, a tiny encryption algorithm. Proc of international workshop on fast software encryption. Springer, Berlin, pp 363–366

Yang G, Zhu B, Suder V, Aagaard MD, Gong G (2015) The simeck family of lightweight block ciphers. Proc of international workshop on cryptographic hardware and embedded systems. Springer, Berlin, pp 307–329

Zhang W, Bao Z, Lin D, Rijmen V, Yang B, Verbauwhede I (2015) RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Sci China Inf Sci 58(12):1–5

Zhou G, Liu Z, Shu W, Bao T, Mao L, Wu D (2017) Smart savings on private car pooling based on internet of vehicles. J Intell Fuzzy Syst 32(5):3785–3796