Journal of Ambient Intelligence and Humanized Computing (2023) 14:11625–11637
https://doi.org/10.1007/s12652-022-03726-y

ORIGINAL RESEARCH

# FPGA implementation novel lightweight MBRISI cipher

**Asmita Poojary**[1] · **V. G. Kiran Kumar**[2] · **H. R. Nagesh**[3]

Received: 8 June 2021 / Accepted: 19 January 2022 / Published online: 12 February 2022


**Abstract**
The advancement Internet of things has led to an increasing exchange of information. Privacy and security has become a major concern. In this emerging ubiquitous computing world, Lightweight cryptographic algorithms are tailor made to secure the information in low resource devices for the Internet of Things (IoT) applications. In this paper an area efficient, high performance lightweight cipher, MBRISI cipher is proposed. The cipher is a combination of BRIGHT family of ciphers comprises of Addition-Modulo, Rotation and EX-OR (ARX) operations and the modification of SIMON cipher and also a novel lightweight key generation algorithm is used. The proposed cipher is better than the state of art existing lightweight ciphers and can be extended to support different block sizes and key sizes for the low resource environments like the IoT. The proposed MBRISI cipher encrypts a 32-bit plaintext employing a 64-bit key is implemented and then analyzed. The cipher is implemented in MATLAB tool (for software implementation) and is analyzed for correlation coefficient, entropy and histogram, avalanche criterion and Key-sensitivity. The Verilog code is written and simulated using Xilinx-Vivado tool and synthesized using FPGA's Artix-7 and Basys-3.

**Keywords** ARX · Lightweight cipher · IoT · Encryption · Decryption · FPGA

## 1 Introduction

The Communication Technology has revolutionized the world and the daily lives in the recent years. Modern day technology being Internet of Things (IoT), which comprises of millions of people around the world and smart-devices communicating with each other and their environment. The communication between two entities may vary from benign data like casual exchange of messages to most confidential data like banking or military information. Thus the data-confidentiality and data-security are increasingly significant to protect the data. State of the art cryptographic ciphers for encrypting and decrypting the data has been studied, implemented and analysed (Mohd et al. 2015). With an enormous information being exchanged between these devices the privacy and security of the same has become a major concern (Li et al. 2019). With the rapid developments of Internet of Things(IoT), there has been increase in requirement of low-power and energy efficient devices and, security being the major concern (Li et al. 2019). In terms of FPGA context, the problem in hand is to develop security to such resource-constrained devices has led to lightweight cryptography. Thus the need for design and development of such low-power, tiny devices and lower cost is a challenge for the designer. Thus the vital requirement of the encryption scheme is the very ingenious design and security of an algorithm is essential.

Encryption is the technique of converting the original/plain-text using a key into unintelligible text is called as cipher-text. Retrieving the original-text back from the cipher-text using the key is called as decryption. The key

Both Asmita Poojary and Dr. V.G. Kiran Kumar are first authors while Dr. H.R. Nagesh is second author.

✉ V. G. Kiran Kumar
kiranvgk@gmail.com

Asmita Poojary
asmitapoojari@nitte.edu.in

H. R. Nagesh
nageshhr@ajiet.edu.in

[1] Department of Computer Science and Engineering, N M A M Institute of Technology, Karkala, Karnataka, India

[2] Department of E & C Engineering, A J Institute of Engineering and Technology, Kottara, Mangaluru, Karnataka, India

[3] Department of Information Science and Engineering, A J Institute of Engineering and Technology, Kottara, Mangaluru, Karnataka, India

used for the encryption and decryption of the data can be same (symmetric Encryption) or different (asymmetric) i.e. keys used for encryption and decryption are different (Stallings 2005). The Encryption process consists of three phases (i) Key-Scheduling phase (ii) Encryption phase and (iii) Decryption phase. The key scheduling algorithm generates the sub-keys required for encryption or decryption.

The IoT comprises of several devices of different sizes like the tiny devices such as RFID, smart-cards, sensors (resource constrained) to the high end devices like the cloud servers or supercomputers working in tandem, exchanging information. The conventional algorithms may work well for traditional devices but may not give better performance when embedded in Low resource devices. Thus to improve the performance, with security and privacy of the data intact, led to the existence of Lightweight cryptography (Fan et al. 2013). This has led the researchers to propose various lightweight cryptographic algorithms (Cazorla et al. 2013). The ciphers designed for resource constrained devices are symmetric ciphers. The ciphers may be stream ciphers or block ciphers. In stream-cipher plaintext is encrypted on bit-by-bit basis, while in block-cipher plaintext is encrypted on a block of 32 bits,64 bits. and so on. The block ciphers can be fiestel structure (like SIMON Cipher) or SPN structure (like the PRESENT) or Lai-Massey (IDEA) (Chaitra et al. 2017; McKay and Bassham 2016) or ARX based. Earlier State of the art implementations where based on SPN (SUBSITUTION AND PERMTUTATION NETWORK) which used S-BOX as a source of non-linearity and required to form the LOOK-UP-TABLE which could be vulnerable to time-cache attack. ARX based ciphers are more compact and faster than SPN based ciphers. At the same time to suit the need of IoT applications so that they can easily embedded to IoT devices, ARX based ciphers are now widely researched upon.

The ciphers can be software or hardware based. The software implementations provide the advantages like (i) ease of use (ii) ease of upgrade (iii) portability, and (iv) flexibility. The conventional ciphers can be engineered fit into the resource constrained devices but the performance may not be acceptable. The lightweight ciphers have the advantages like lesser area, lesser power consumption with no compromise in security and privacy. The FPGA implementations of lightweight ciphers have advantages over ASICs like the (i) ease of upload (ii) ease of modification (iii)algorithm agility (iv) throughput (v) architecture efficiency and (vi) cost efficiency (Abed et al. 2019).

A lightweight cipher called MBRISI cipher suited for light-weight devices is proposed in this paper. The cipher is a fusion of modified SIMON Cipher and family of BRIGHT ciphers that uses ARX operations. Section 2 gives the background and state of the art implementations. Section 3 gives the detailed architecture and the operation of the proposed MBRISI cipher and the key generation scheme used in the proposed cipher, Sect. 4 analyzes the results of both software and hardware implementations. The conclusion is presented in Sect. 5.

## 2 Background

McKay and Bassham (2016) presented a NIST report that gives an overview of the lightweight cipher project, the standards for the design and development of the lightweight ciphers are described. The report mentions about the profiles which comprises of design goals, target devices, physical characteristics like area, memory and implementations like software and hardware, performance parameters like latency and throughput and security characteristics (like attacks).

An optimized model for SIMON-cipher is designed by Abed et al. (2019) for the resource constrained devices. The evaluation parameters are power and energy. The paper examines scalar and pipelined implementations. The implementation results show that 39% lesser resources and 45% lesser power by scalar implementations. Two round and four round pipelined cipher implementations that has been implemented resulted in better throughput and consume lesser energy than scalar designs.

Pandey et al. (2018) presented the implementation of modified PRESENT, ECDH, RSA and DH ciphers. The PRESENT cipher has been augmented and the S-BOX design has been improved and optimized. The algorithms are implemented in standard gate library of UMC-90 nm. The modified PRESENT cipher has an area improvement of $1.7\times$ and a power improvement of $63\times$ when compared to AES. For a 128-bits key, the proposed PRESENT algorithm has a break attack time complexity of 2127.

Wajih El Hadj Youssef et al. (2020) presented the optimized lightweight cipher designs of 32-bit for different ciphers like LED, SIMECK and SIMON. The designs were implemented on Spartan-3, Spartan-6, and Zynq boards and analyzed in terms of area power, speed and efficiency. The ciphers implemented on Zynq-7000 baord had a better throughput of 891.99 Mbps for SIMECK, 210.13 for LED while SIMON had 838.95 Mbps. and also comparisons were made by implementing it on different FPGA boards. The software implementation results on image was performed and statistical analysis was performed. Both encryption and decryption results show that the results are secure against attacks.

Deepti Sehrawat et al. (2020) proposed a highly optimized lightweight encryption algorithm BRIGHT. The structure comprises of pre-key whitening followed by Addition, Rotation and XOR operations, and permutation of bits thus making it a highly efficient structure than the existing

ones as compared. The cipher supports range of key sizes and block-sizes. The algorithm fulfils the key sensitivity test, randomness test and Strict Avalanche Criterion (SAC). When compared to the existing ones the BRIGHT cipher has lower memory utilization, lower cost and high speed compared with the state of the art lightweight ciphers.

A novel compact lightweight cipher is presented by Gookyi Dennis et al. (2017), the cipher encrypts 128-bit block using a key of 64-bit length using eight rounds. The proposed cipher is a fiestel structure and comprises of S-BOX and P-BOX to achieve confusion and diffusion. The algorithm is implemented in iNEXT-V6(virtex-6) FPGA board. The synthesis results show that it requires 196 slices at a clock frequency of 337 MHz. The proposed cipher has a better throughput and requires lesser resources compared to the existing ciphers.

Bansod et al. (2018), proposed a lightweight encryption algorithm VAYU. A fiestel cipher that encrypts a plaintext of 64-bit using a key of 80/128-bits using 31 rounds. The cipher has two F-functions with S-BOX for the diffusion. The cipher has a Permutation layer with a data complexity of 260 takes minimum rounds to generate maximum S-BOXes. the cipher uses lesser memory size compared to existing algorithms. It has an adequate security against Algebraic attacks, Bicyclique attack, zero correlation attack, differential and linear cryptanalysis.

Biswas et al. (2020) proposed a cryptographic algorithm based on fiestel structure called LRBC Cipher (Light Weight Resource Constraints Block Cipher). It is based on Substitution and Permutation network (SPN) to achieve better security. Artix7 NEXYS4 DDR FPGA is used to implement the algorithm. The algorithm has a power consumption of 11.40 μW and area occupied is 258.9 Gate Equivalent (GE). The proposed scheme exhibits a high security with robustness against various attacks with an avalanche effect of 58% for plaintext and 55.75% for key.

Mohd et al. (2016) implemented an optimized HIGHT cipher in FPGA. HIGHT requires 128-bit key takes 32 rounds and one final transformation round to encrypt plaintext of 64-bits. Scalar and pipelined design of HIGHT cipher along with other designs were designed and implemented. The analysis of the designs find that scalar designs require lesser area and consume lesser power dissipation while pipelined designs require lower energy and better throughput. The area occupied by scalar design is 18% lesser and power consumed is 10% lesser than pipelined design. While the pipelined design has a throughput of 18 times higher and energy consumption is 60% lesser than scalar designs.

Reversible-logic gates cryptography with LFSR (RLCD-LFSR) a lightweight cipher was proposed by Saranya Karunamurthi et al. (2019). Fredkin gate, Toffoli gate Feynman gate and SCL gates were used to design the proposed cipher. A $128 \times 128$ image was taken as input plain-text and the text-file was converted to binary. LFSR was used to generate keys. Histogram analysis reveals improved security. CADENCE-ASIC 180 nm technology implementation observed an area reduction of 8%, reduced power dissipation by 2.32% and delay reduction of 26.4%, while 45 nm technology implementations observed reduction in area by 9.6%, and reduction in power by 8.33% and delay reduced by 20.46%.

## 3 The MBRISI Block cipher

The MBRISI cipher is a Feistel-based block cipher Fig. 1 shows the block diagram representation. It comprises of 10 rounds so as to increase efficiency. To increase the security strength, the number of rounds can be increased. It takes an initial 64-bit key to encrypt a plain-text block of 32-bits. It uses 16-bit round key obtained from initial 64-bit key (master-key). The algorithm uses a novel key scheduling technique using simple XOR, Shift and Addition-Modulo operations. Figure 1 shows the Block diagram of the MBRISI Encryption Process.

Table 1 depicts the notations used for the MBRISI cipher.

The encryption process of the MBRISI cipher comprises of two parts

(i) Key Generation Process.
(ii) The Encryption mechanism.

### 3.1 The MBRISI key generation

The design of key generation process is an important part in any Encryption and Decryption algorithm. The security strength of any encryption algorithm banks on the design of the Key-generation Process. Figure 2 shows the process of the key generation for the proposed cipher, is derived by the fusion of two key generation algorithms the SIT (Rabab et al. 2018), Simple Encryption Scheme for IoT and Modified Fibonacci and Scrambling Algorithm (Amiruddin et al. 2019). The confusion and diffusion to the key generation process is created by the use of Permutation, Shift, XOR operations thus increasing the security (Alkamil and Perera 2019).

The steps of the key generation process are as follows

(i) The initial input key of 64-bits [K(63:0)]is permuted.
(ii) The permuted segments are concatenated as (ie kc1 = K(3:0)|| K(19:16)|| K(35:32)|| K(51:48); (ie kc2 = K(7:4)|| K(23:20)|| K(39:36) || K(55:52); (ie kc3 = K(11:8) || K(27:24|| K(43:40)|| K(59:56); (ie kc4 = K(15:12)|| K(31:28)|| K(47:44)|| K(63:60);
(iii) (ks1 = kc1 < < 5, ks2 = kc2 < < 3 ks3 = kc3 < < 5, ks4 = kc4 < < 3);

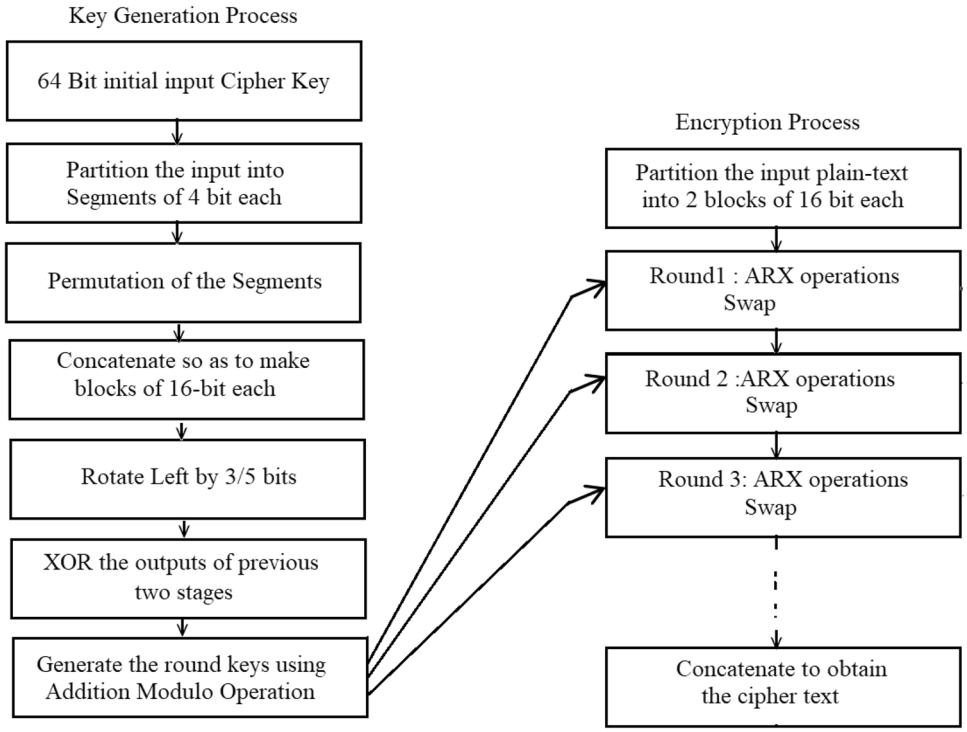



**Fig. 1** The block diagram of the proposed algorithm

**Table 1** MBRISI Cipher notations

| PTi | 32-bit input plain-text |
|---|---|
| CTo | 32-bit output cipher-text |
| Ri, Li | Right-half i-bits, Left half i-bits |
| MKi | 64-bit Master key |
| Rki | 16-bit round key |
| ⊕ | Bitwise-XOR |
| ⊞ | Modulo-Addition |
| ⊗ | Modulo-Multiplication |
| x < < < k | Circular Left-Shift by k bits |
| ‖ | Concatenation |

(iv) (ks1 = kc1 < < 5, ks2 = kc2 < < 3 ks3 = kc3 < < 5, ks4 = kc4 < < 3);

(v) kq1 = kc1 ⊕ ks1, kq2 = kc2 ⊕ ks2, kq3 = kc3 ⊕ ks3, kq4 = kc4 ⊕ ks4 (XORing the outputs of Step ii with Step iii)

(vi) Apply the Addition-Modulo Algorithm to generate the required number of round keys

a. The key SK(1) = mod (kq1 + kq2, n)
b. Similarly, the key SK(2) = mod (kq3 + kq4, n)
c. The remaining keys are determined as (n = 65,537 the largest prime is selected)

for j=3 to 10 do
SK(j) = mod (SK(j-1) + SK(j-2) ,n)
end for
End

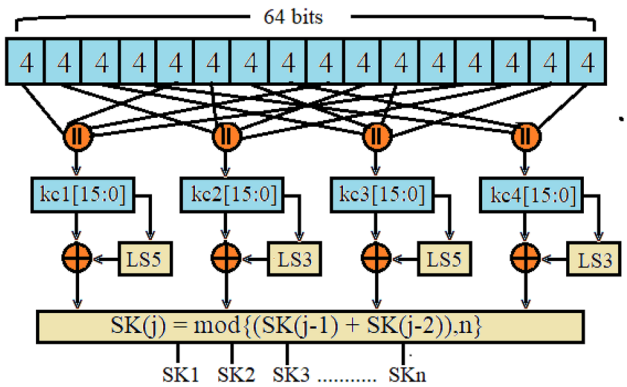



**Fig. 2** Key generation process

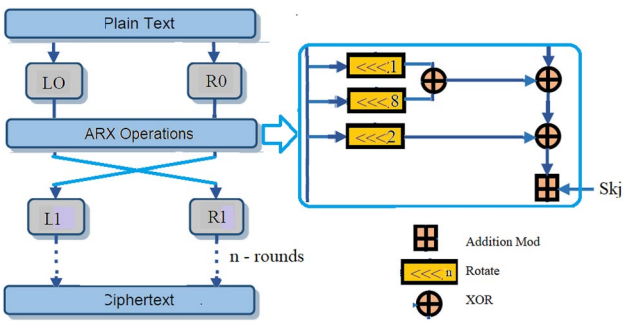
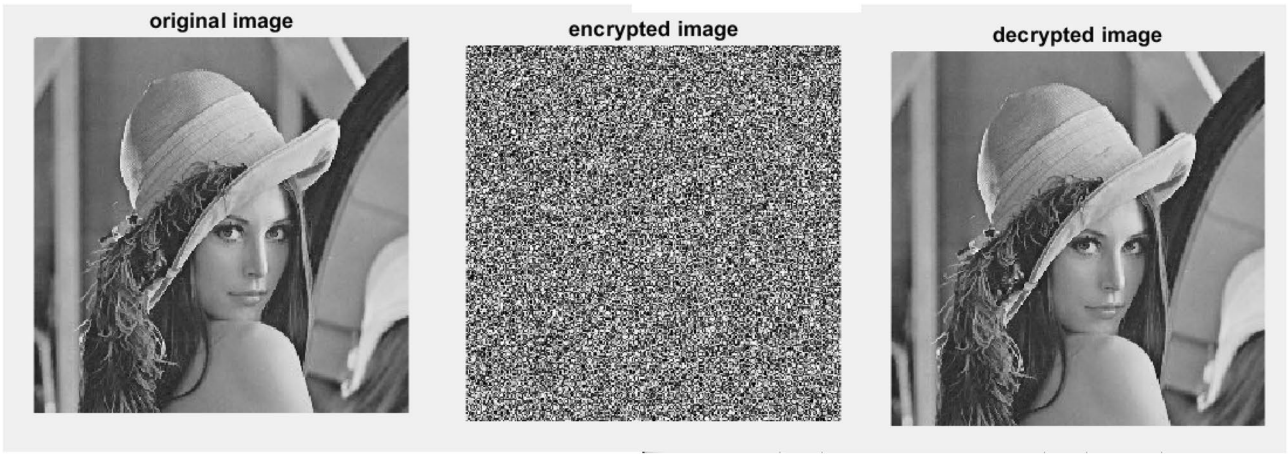
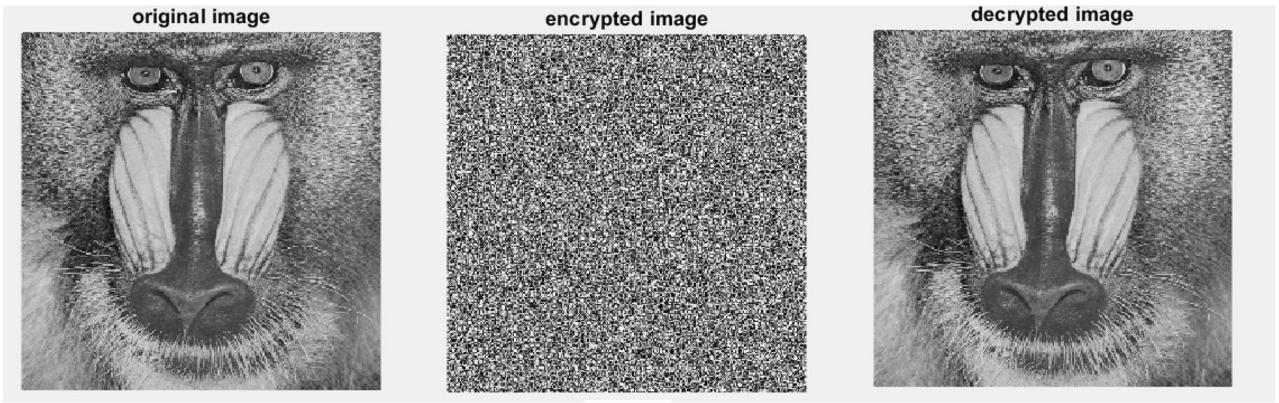


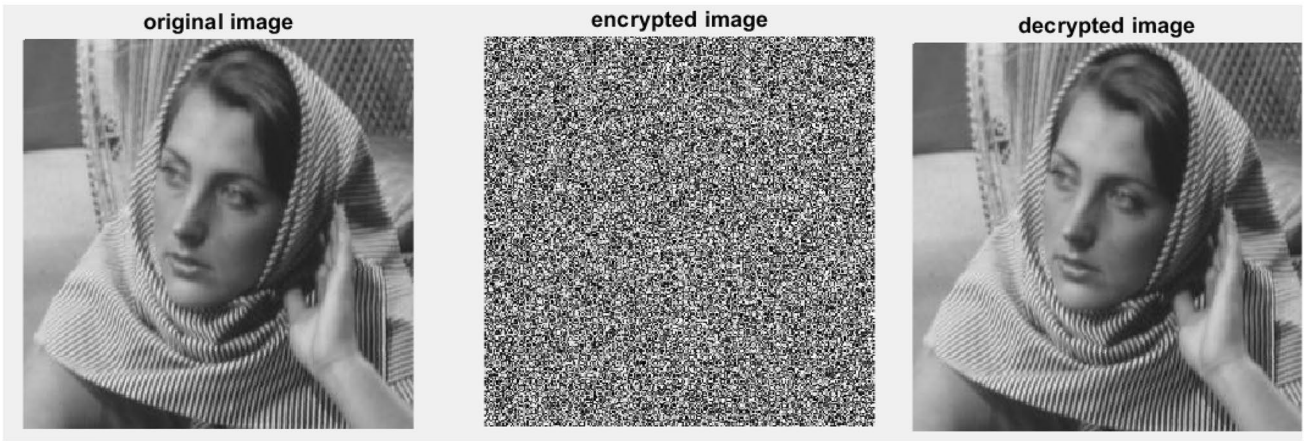

**Fig. 3** The MBRISI one round function




(a) Encryption and Decryption of Lena image




(b) Encryption and Decryption of Baboon image




(c) Encryption and Decryption of Barbara image

**Fig. 4** **a** Encryption and decryption of Lena image. **b** Encryption and decryption of Baboon image. **c** Encryption and decryption of Barbara image

## 3.2 The MBRISI encryption process

The Proposed MBRISI cipher is a fusion of SIMON algorithm that is modified and the ARX based (Addition-Modulo, Rotate and XOR) BRIGHT cipher.

### 3.2.1 The MBRISI encryption process

The MBRISI cipher uses the following arithmetic operations for Encryption Process.

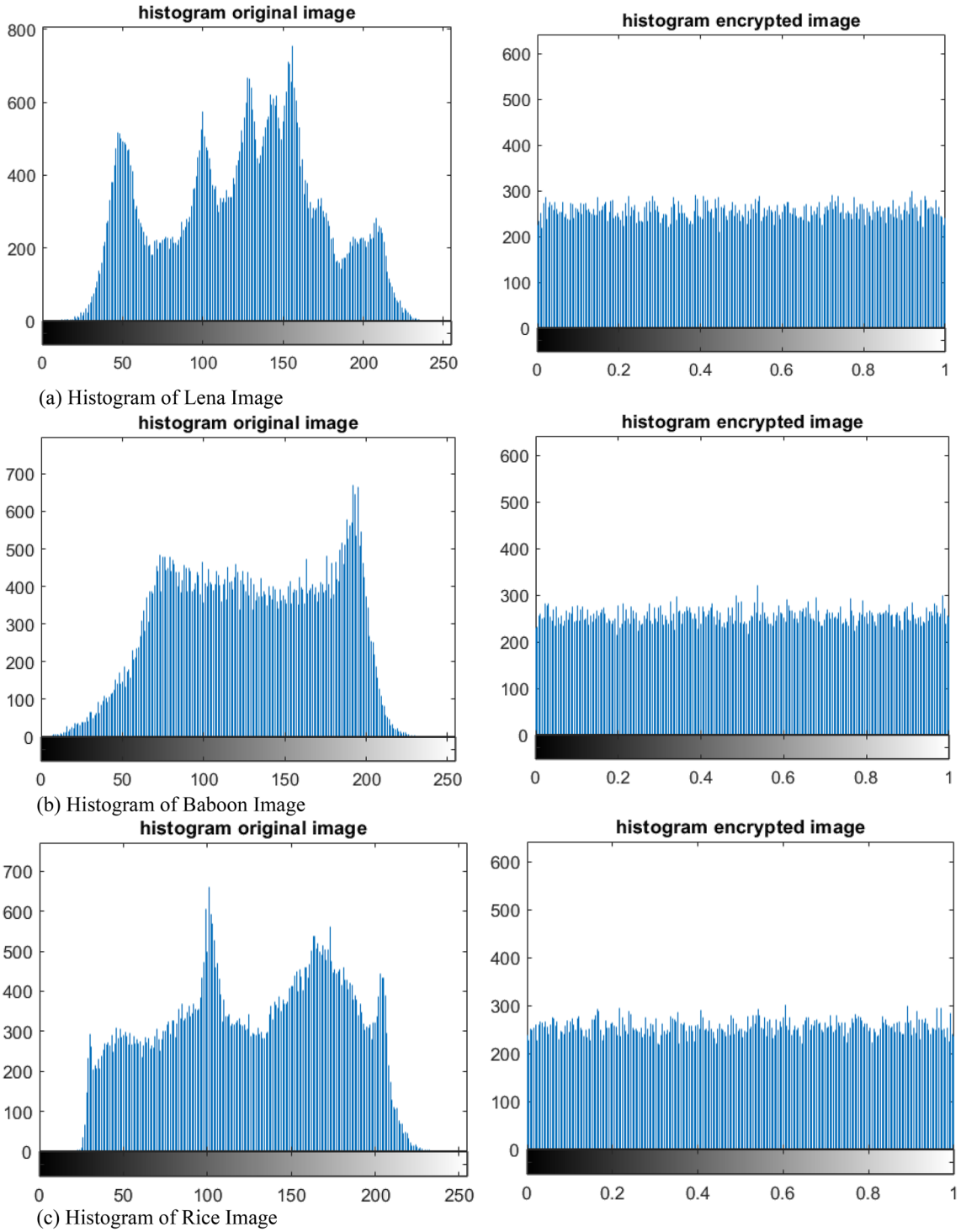



**Fig. 5** **a** Histogram of Lena image. **b** Histogram of Baboon image. **c** Histogram of rice image

- Addition-Modulo
- Rotate (Circular Shift by i-bits)
- XOR

The One Round Encryption Process comprises of following Steps

(i) the plain-text is split into two-halves L0(left) and R0(Right) (16-bits each).

ARX operations :

(ii) Left-Circular-shift L0 by 1-bit($<<<1$(Ls1)), left-circular-shift by 2-bits($<<<2$(Ls2)) and Left-circular-shift 8-bits($<<<8$(Ls8)).
(iii) $Ls18 = Ls1 \oplus Ls8$
(iv) $R01 = Ls18 \oplus R0$
(v) $R02 = Ls2 \oplus R01$
(vi) $RRO = R02 \oplus SKj$
(vii) $L1 = RRO; R1 = L0$.(Swapping of Left and Right half)

Continue the process for 10 rounds.

The MBRISI one round function for encryption, F, is given by

$$F(R,L,k) = ((Ls1 \oplus Ls8) \oplus R0) \oplus Ls2) \boxplus SKi)||L0 \tag{3.1}$$

where k is the round key. Figure 3 depicts the encryption process for MBRISI Cipher.

#### 3.2.2 The MBRISI decryption process

Since the MBRISI cipher is designed using fiestel structure the decryption operation is reverse operation of the encryption process but uses Subtraction-modulo operation for decryption.

The One Round Decryption Process comprises of following Steps

(i) the input cipher is split into two-halves of 16-bits, L1 – Left-half and R1 as Right-half.
(ii) $L1 = RR0; R1 = L0$. (Swapping of Left and Right half)
(iii) $R02 = RR0 \boxminus SKj \boxminus represents subtraction modulo$).
(iv) Left circular shift L0 by 1-bit(Ls1), Left circular shift by 2-bits(Ls2) and Left circular shit 8-bits(Ls8).
(v) $Ls18 = Ls1 \oplus Ls8$
(vi) $R01 = Ls2 \oplus R02$
(vii) $R0 = Ls18 \oplus R01$

Continue the process for 10 rounds

## 4 Experimental setup

The implementation of the MBRISI cipher has been carried out in software and hardware. The software implementation is carried out in MATLAB and FPGA implementation in Xilinx-Vivado tool.

### 4.1 The MATLAB implementation

The software implementation of the proposed cipher is performed in MATLAB tool. The following analysis has been performed.

#### 4.1.1 Visual testing

Figure 4 illustrates encryption and decryption for three images, it can be seen that the encrypted images will not reveal any information of original plaintext image.

#### 4.1.2 Histogram of the cipher images

The histogram analysis of the cipher image visualizes the strength of the security of a proposed cipher. Figure 5 depicts the measure of randomness of the encrypted images. From the Fig. 5a–c it can be seen that the histogram of cipher images has a nonlinear distribution of pixel values (i.e. uneven distribution of the different pixel values having crests and troughs) while the histogram of the encrypted images having a uniform distribution of pixel values, hence it can be concluded that the encryption is secure.

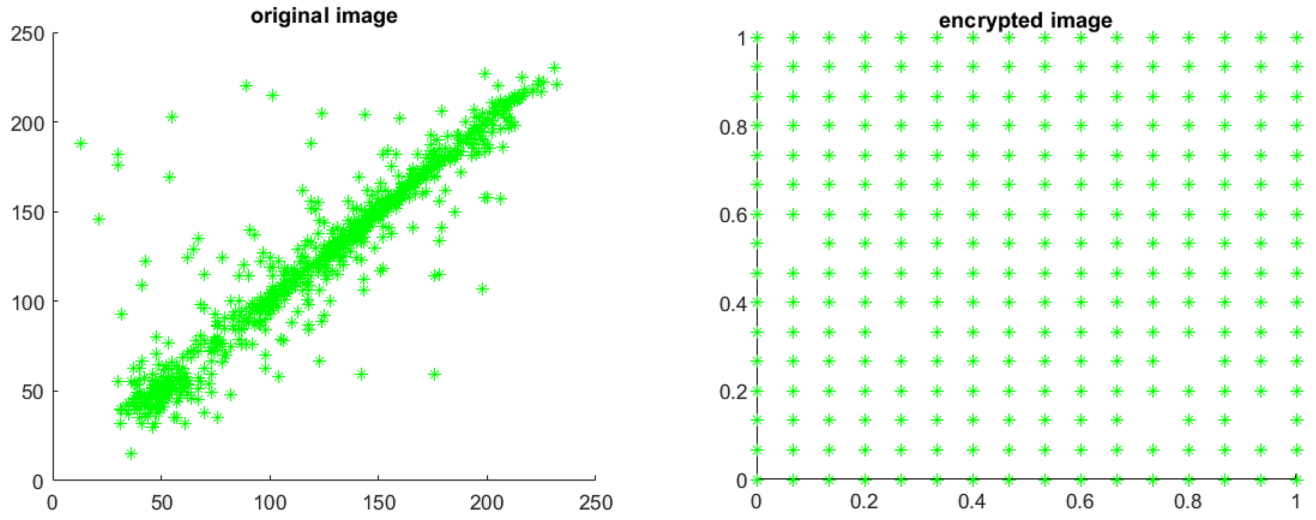
#### 4.1.3 Correlation coefficient analysis

Correlation coefficient calculates the dependency between two values and measures the security strength of a cipher algorithm.

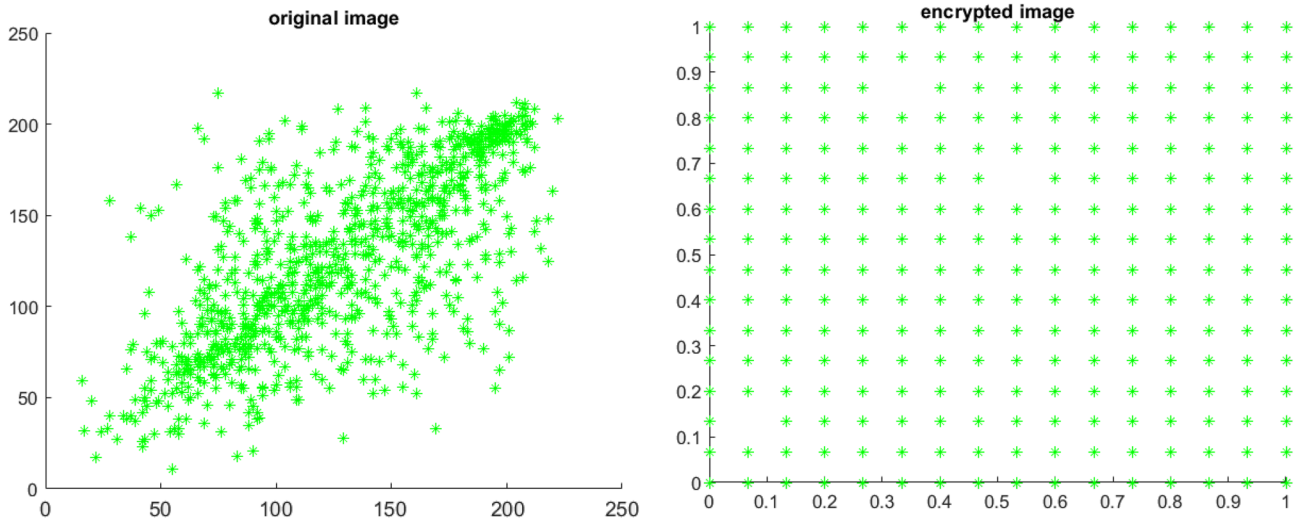
The correlation coefficient of the encrypted images/plaintext is given by

**Table 2** Correlation coefficient of the encrypted images

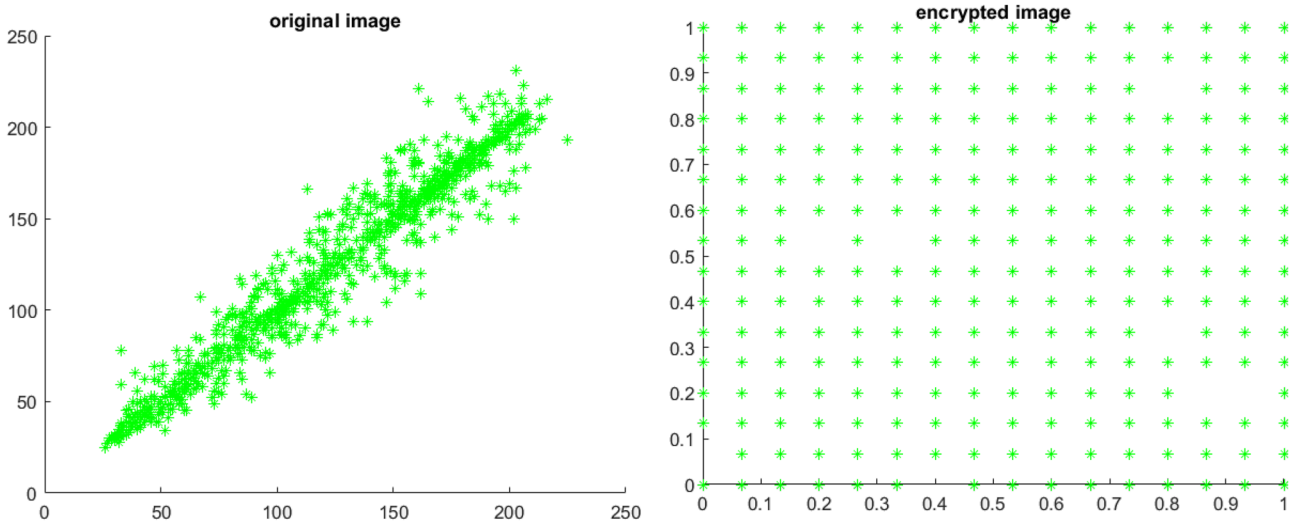
| Test image | Correlation coefficient | | | | | |
|---|---|---|---|---|---|---|
| | Horizontal | | Vertical | | Diagonal | |
| | Original | Encrypted | Original | Encrypted | Original | Encrypted |
| Lena | 0.9310 | −0.0015 | 0.9716 | −0.0042 | 0.9035 | 0.0232 |
| Baboon | 0.7131 | −0.0113 | 0.7112 | 0.0037 | 0.7881 | −0.0147 |
| Barbara | 0.9071 | 0.0042 | 0.9682 | −0.01292 | 0.8905 | −0.0039 |




(a) Correlation of Lena Image




(b) Correlation of Baboon Image




(c) Correlation of Barbara Image

**Fig. 6** **a** Correlation of Lena image. **b** Correlation of Baboon image. **c** Correlation of Barbara image

**Table 3** Results of information entropy

| Test image | Information entropy | |
|---|---|---|
| | Plain-text | Encrypted-image |
| Lena | 7.4750 | 7.9969 |
| Baboon | 7.4637 | 7.9970 |
| Barbara | 7.5302 | 7.9971 |

$$r_{pq} = \frac{\rho(\alpha, \beta)}{\sqrt{V(\alpha)}\sqrt{V(\beta)}}$$

and V() represents the variance and the ρ covariance.

The Variance V() is defined as

$$V(\alpha) = \frac{1}{N}\sum_{i=1}^{N}\left(\alpha_i - E(\alpha)\right)^2$$

While covariance is given by

$$\rho(\alpha, \beta) = \frac{1}{N}\sum_{i=1}^{N}(\alpha - E(\alpha) * (\beta - E(\beta)$$

The If X and Y represents the gray values of two neighbouring pixel values of the image.

$$E(\beta) = \frac{1}{N}\sum_{j=1}^{N}\beta_j$$

The correlation coefficient of an image is the contrast between original and encrypted image. From the Table 2 it is seen that the correlation coefficient of the cipher images is nearer to zero. Thus it can be concluded that it is infeasible to conduct an attack against the proposed cipher and the plaintext is secured (Gookyi et al. 2017; Biswas et al. 2020).

Figure 6a–c indicates that the images are highly correlated, thus algorithm is secured

**Table 4** NIST statistical test results for the key generation scheme

| Test | P-value | Result |
|---|---|---|
| Frequency | 0.035174 | Successful |
| BlockFrequency | 0.122325 | Successful |
| CumulativeSums(forward) | 0.035174 | Successful |
| CumulativeSums(inverse) | 0.066882 | Successful |
| Runs | 0.213309 | Successful |
| LongestRun | 0.035714 | Successful |
| Rank | 0.739918 | Successful |
| FFT | 0.017912 | Successful |
| NonOverlappingTemplate | 0.066882 | Successful |
| OverlappingTemplate | 0.213309 | Successful |
| Universal | 0. 534,146 | Successful |
| ApproximateEntropy | 0.350485 | Successful |
| RandomExcursions | 0.122325 | Successful |
| RandomExcursionsVariant | 0.739918 | Successful |
| Serial | 0.066882 | Successful |
| LinearComplexity | 0.452851 | Successful |

#### 4.1.4 Information entropy

The degree of randomness of an image can be calculated by and is given by

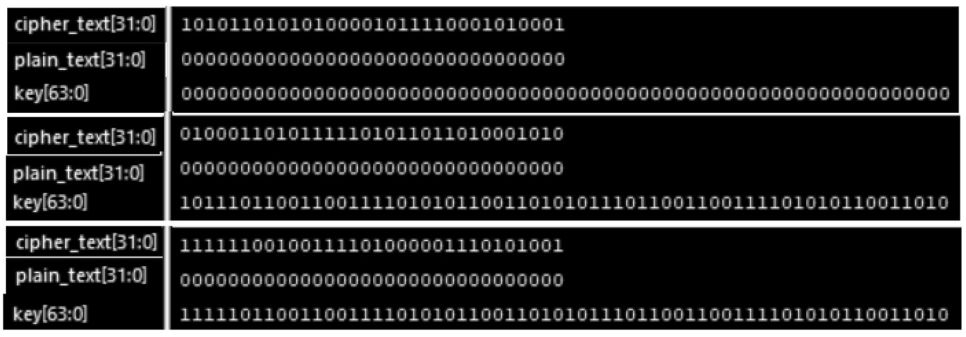



**Fig. 7** Avalanche effect for MBRISI cipher

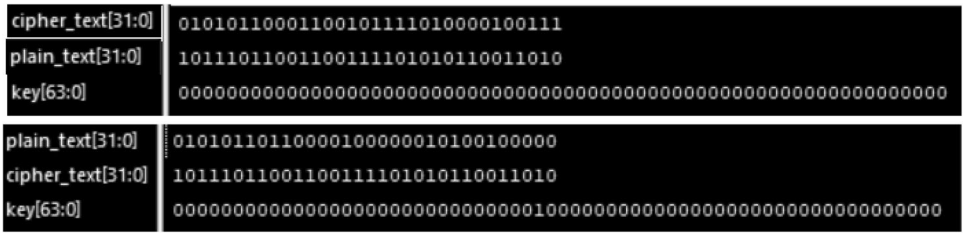



**Fig. 8** Key sensitivity for MBRISI cipher

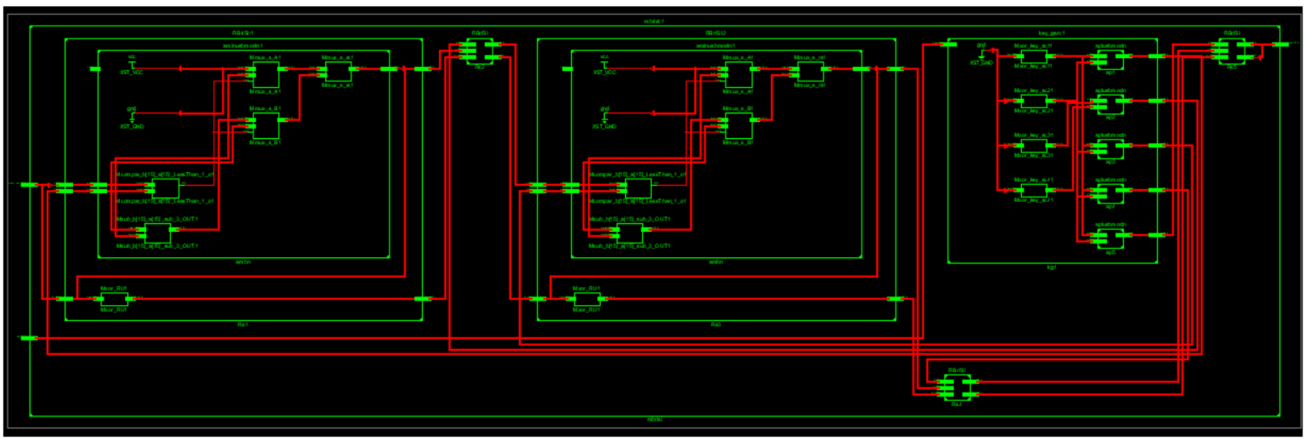


**Fig. 9** RTL schematic of MBRISI Cipher

**Table 5** Implementation results of MBRISI cipher in two FPGA families

| Elements | Virtex 7 (XQ7VX330T) | | Artix7 (XC7A100T) | |
|---|---|---|---|---|
| | Available | Used | Available | Used |
| Slice LUTs | 218,800 | 492 | 63,400 | 492 |
| LUT FFs | 492 | 0 | 492 | 0 |
| Bonded IOBs | 300 | 128 | 210 | 128 |
| Latency | | 6 | | 6 |
| Max frequency (MHz) | | 57.94 | | 23.667 |
| Throughput (Mbps) | | 309 | | 226 |
| Power (W) | | 0.143 | | 0.042 |

$$H(X) = \sum p(xi) log_2 \left( \frac{1}{p_i(xi)} \right)$$

where x represents a discrete random variable, p(xi) represents the probability of the intensity value xi.

The maximum entropy of an 8-bit grey scale picture is 8 bits. From Table 3 it can be observed that the entropy is $7.9 < E < 8$ and hence it can be inferred that the plaintext is secured.

#### 4.1.5 Avalanche effect

The security of the cryptographic algorithm is determined by Avalanche effect. Figure 7 depicts the Avalanche effect. By keeping key constant, a 1-bit change in the cipher-text yields at least 50% change in the cipher-text thus a considerable diffusion and confusion is achieved (Gookyi et al. 2017; Biswas et al. 2020).

**Table 6** Performance comparison

| Design | Year | LUT's | Registers | IOBs | Timing (nS) | Power (W) | FPGA |
|---|---|---|---|---|---|---|---|
| Alkamil et al. (Korobeynikov 2019) SIMON | 2019 | 570 | 594 | 180 | 13.65 | . | Virtex -6 |
| Zeesha Mishra LEA (Rabab et al. 2018) | 2019 | 360 | 382 | 122 | 4.44 | 0.566 | Virtex-5 |
| Pandey (2016) RECTANGLE | 2020 | 251 | 281 | 210 | 2.55 | 0.721 | Virtex-5 |
| Mishra et al. (2019) SIT | 2020 | 760 | 345 | 128 | 2.09 | 0.185 | Virtex-7 |
| Um e Rabab (Kiran Kumar and Shantharama Rai 2020) Secure IoT | 2018 | 711 | 796 | 813 | 12.11 | . | Cyclone II |
| Vijay (Pandey et al. 2020) ANU Cipher | 2020 | 270 | 103 | 199 | 1.78 | 0.050 | Virtex-5 |
| Ayesha (Mishra et al. 2021) PICO Cipher | 2021 | 259 | 77 | 210 | 2.8 | 1.295 | Virtex-6 |
| Proposed MBRISI | 2021 | 492 | 0 | 128 | 23.667 | 0.042 | Artix-7 |


**Fig. 10** Comparison of the ciphers

#### 4.1.6 Key sensitivity

Key sensitivity is a cryptographic property to determine the security of the cryptographic algorithm. Figure 8 shows, keeping plaintext constant, a 1-bit change in input-key the cipher-text changes by a considerable amount. A 1-bit change in the key yields more than 50% change in cipher-text, thus achieving considerable amount of confusion and diffusion (Gookyi et al. 2017; Biswas et al. 2020).

### 4.2 Key generation

The NIST random test (Thakor et al. 2021) was performed by generating $10^6$ bit streams from TRNG, the level of significance $P \geq 0.01$ is set for acceptance of randomness. The results as seen in Table 4 confirms that the key generated is random in nature.

### 4.3 The Hardware implementation

The FPGA implementation of the proposed MBRISI cipher is designed in Verilog HDL and has been implemented in Xilinx Vivado for the Virtex- 7 and Artix-7 FPGA. The RTL schematic for the one round encryption is shown in Fig. 9.

The proposed cipher is implemented in Virtex-7 and Artix-7 FPGA the hardware utilization is shown in Table 5.

The comparison of the Proposed cipher with the various lightweight encryption algorithms (Korobeynikov 2019; Rabab et al. 2018; Amiruddin et al. 2019; Failed 2016; Mishra et al. 2019, 2021; Kiran Kumar and Shantharama Rai 2020; Pandey et al. 2020) implemented in FPGA is depicted in Table 6, the comparison chart is shown in Fig. 10.

## 5 Conclusion

In this paper, MBRISI cipher, that encrypts 32-bit plaintext using 64-bit key using ten rounds is implemented in software and is been analyzed for histogram, entropy, correlation, avalanche criterion and Key Sensitivity. The key generation scheme passed all the 15 NIST statistical tests, it is found that proposed algorithm is secure by fulfilling these criteria. The FPGA implementation requires 492 LUTs in both Virtex-7 and Artix-7 FPGA. The throughput and power consumption of 309 and 0.143 in Virtex-7 and 150 and 0.043 was achieved in Artix-7. The cipher is also compared with various other cipher FPGA implementations and thus can be inferred that the proposed cipher is suited for resource constrained devices. With Internet of Things playing a significant role in the mere future, the security and privacy issues increasing the proposed cipher design of lightweight algorithm will be one of the possible candidate that can be adopted in IoT applications.

**Acknowledgements** The authors would like to thank the Department of Electronics and Communication and Engineering, AJ Institute of Engineering, Department of Computer Science and Engineering NMAMIT Nitte and Visvesvaraya Technological University, Belagavi for the support for carrying out the research work.

**Data availability** No Data was used in this project as it is a simulation based project. Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

## References

Abed S, Jaffal R, Mohd BJ, Alshayeji M (2019) FPGA modeling and optimization of a SIMON lightweight block cipher. Sensors (basel) 19(4):913. https://doi.org/10.3390/s19040913

Alkamil A and Perera DG (2019) Efficient FPGA-based reconfigurable accelerators for SIMON cryptographic algorithm on embedded platforms. In: 2019 International Conference on ReConFigurable Computing and FPGAs (ReConFig), Cancun, Mexico, pp 1–8. https://doi.org/10.1109/ReConFig48160.2019.8994803

Amiruddin A, Ratna AAP, Sari R (2019) Construction and analysis of key generation algorithms based on modified Fibonacci and scrambling factors for privacy preservation. Int J Netw Secur 21:250–258

Ayesha N, Acharya B (2021) FPGA implementation of PICO cipher. In: Nath V, Mandal JK (eds) Proceedings of the fourth international conference on microelectronics, computing and communication systems. Lecture notes in electrical engineering. Springer, Singapore. https://doi.org/10.1007/978-981-15-5546-6_43

Bansod G, Patil A, Pisharoty N (2018) GRANULE: an ultra lightweight cipher design for embedded security. IACR Cryptol Eprint Arch 2018:600

Biswas A, Majumdar A, Nath S et al (2020) LRBC: a lightweight block cipher design for resource constrained IoT devices. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-020-01694-9

Cazorla M, Marquet K, Minier M (2013) Survey and benchmark of lightweight block ciphers for wireless sensor networks. In: Proceedings of the 2013 international conference on security and cryptography (SECRYPT), Reykjavik, Iceland, 29–31 July 2013, pp 1–6

Chaitra B, Kumar VGK, Shatharama RC (2017) A survey on various lightweight cryptographic algorithms on FPGA. IOSR J Electron Commun Eng 12(1):45–59

Choi P, Lee M-K, Kim DK (2017) Fast compact true random number generator based on multiple sampling. Electron Lett 53(13):841–843

Dahiphale V, Bansod G, Zambare A et al (2020) Design and implementation of various datapath architectures for the ANU lightweight cipher on an FPGA. Front Inform Technol Electron Eng 21:615–628. https://doi.org/10.1631/FITEE.1800681

El Hadj YW, Abdelli A, Dridi F, Machhout M (2020) Hardware implementation of secure lightweight cryptographic designs for IoT applications. Secur Commun Netw 2020:1–13

Fan X, Mandal K, Gong G (2013) Wg-8: A lightweight stream cipher for resource-constrained smart devices. In International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, vol. 115. Springer: Berlin, pp 617–632

Gookyi DAN, Park S, Ryoo K (2017) The efficient hardware design of a new lightweight block cipher. Int J Control Autom 1(1):431–440

Gupta R, Pandey A, Baghel RK (2019) FPGA implementation of chaos-based high-speed true random number generator. Int J Numer Model 32:e2604. https://doi.org/10.1002/jnm.2604

Karunamurthi S, Natarajan VK (2019) VLSI implementation of reversible logic gates cryptography with LFSR key. Microprocess Microsyst 69:68–78

Kiran Kumar VG, ShantharamaRai C (2020) FPGA implementation of simple encryption scheme for resource-constrained devices. Int J Adv Trends Comput Sci Eng. https://doi.org/10.30534/ijatcse/2020/213942020

Kiran Kumar VG, ShantharamaRai C (2021) Design and implementation of novel BRISI lightweight cipher for resource constrained devices. Microprocess Microsyst 84:104267. https://doi.org/10.1016/j.micpro.2021.104267

Korobeynikov A (2019) Effective implementation of "Kuznyechik" block cipher on FPGA with OpenCL platform. IEEE Conf Russ Young Res Electric Electron Eng (EIConRus) 2019:1683–1686

Li S, Song H, Iqbal M (1935) Privacy and security for resource-constrained IoT devices and networks: research challenges and opportunities. Sensors 2019:19

McKay KA, Bassham M, Turan MS, Mouha N (2016) DRAFT NISTIR 8114 report on lightweight cryptography. National Institute of Standards and Technology Internal Report 8114

Mishra Z, Ramu G, Acharya B (2019) Hight Speed low area VLSI architecture for LEA encryption algorithm. In: Nath V, Mandal J (eds) Proceedings of the third international conference on microelectronics, computing and communication systems. Lecture notes in electrical engineering, vol 556. Springer, Singapore. https://doi.org/10.1007/978-981-13-7091-5_14

Mishra Z, Mishra S, Acharya B (2021) High throughput novel architecture of SIT cipher for IoT application. In: Nath V, Mandal J (eds) Nanoelectronics, circuits and communication systems. Lecture notes in electrical engineering, vol 692. Springer, Singapore. https://doi.org/10.1007/978-981-15-7486-3_26

Mohd BJ, Hayajneh T, Vasilakos AV (2015) A survey on lightweight block ciphers for low-resource devices: comparative study and open issues. J Netw Comput Appl 58:73–93

Mohd BJ, Hayajneh T, Khalaf ZA, Ahmad Yousef KM (2016) Modeling and optimization of the lightweight HIGHT block cipher design with FPGA implementation. Secur Commun Netw 9:2200–2216. https://doi.org/10.1002/sec.1479

Pandey JG, Goel T, Karmakar A (2018) A high-performance and area-efficient VLSI architecture for the PRESENT lightweight cipher. In: 2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID), Pune, pp 392–397. https://doi.org/10.1109/VLSID.2018.96

Pandey JG, Laddha A and Samaddar SD (2020) A lightweight VLSI architecture for RECTANGLE cipher and its implementation on an FPGA. In: 2020 24th International Symposium on VLSI Design and Test (VDAT). pp 1–6. https://doi.org/10.1109/VDAT50263.2020.9190623

Rabab UE, Ahmed IU, Aslam MI, Usman M (2018) FPGA implementation of secure internet of things (SIT) algorithm for high throughput area ratio. Int J Future Gener Commun Netw 11(5):63–72

Rana S, Hossain S, Shoun HI, Abulkashem M (2018) An effective lightweight cryptographic algorithm to secure resource-constrained devices. Int J Adv Comput Sci Appl. https://doi.org/10.14569/IJACSA.2018.091137

Sehrawat D, Gill N (2020) Ultra BRIGHT: a tiny and fast ultra lightweight block cipher for IoT. Int J Sci Technol Res 9:1063

Sruthi N, Nandakumar R and Rajkumar P (2016) Design and characterization of HIGHT cryptocore. In: 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES). Paralakhemundi, pp 205–209

Stallings W (2005) Cryptography and network security: principles and practice. Prentice Hall, Inc., Upper Saddle River

Thakor VA, Razzaque MA, Khandaker MRA (2021) Lightweight cryptography algorithms for resource-constrained IoT devices: a review, comparison and research opportunities. IEEE Access 9:28177–28193. https://doi.org/10.1109/ACCESS.2021.3052867

Xufan W and Shuguo L (2017) A new digital true random number generator based on delay chain feedback loop. IEEE Conference 978-1-4673-6853-7/17/$31.00