ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse213942020.pdf
https://doi.org/10.30534/ijatcse/2020/213942020

FPGA Implementation of Simple Encryption Scheme for Resource-Constrained Devices

Kiran Kumar V G[1], Shantharama Rai C[2]
[1]Assistant Professor, Department of E & C Engineering, A J Institute of Engineering and Technology, Kottara Mangaluru, (Karnataka), INDIA, kiranvgk@gmail.com
[2]Professor and Principal, A J Institute of Engineering and Technology, Kottara Mangaluru, (Karnataka), INDIA, csraicec@gmail.com

ABSTRACT

Internet of things (IoT), where billions of devices are interconnected together, where a huge amount of data is being exchanged between conventional and resource constrained devices and the security of the data remains a huge concern. While conventional cryptographic algorithms, cannot fit into resource constrained devices, the design of such ciphers (hence the term Lightweight Cipher) is a major challenge, while the three principles of the security triad Confidentiality, Integrity and Availability of the data doesn't change.

In this paper, simple Lightweight ciphers based on ARX (Addition, Rotation and XOR) and MRX (Multiplication, Rotation and XOR) operations based on reversible logic and Vedic Mathematics are proposed. The addition and multiplication operations are implemented using Reversible Logic and Vedic Mathematics and a modified Montgomery algorithm is implemented to perform modular operation. The scheme is implemented using both software and hardware. The software implementation is done using MATLAB and the Histogram Analysis, Correlation Analysis and Entropy Analysis for the grayscale image are performed to verify the security of the image, and the simulations and synthesis are performed using Xilinx-Vivado verified on the Nexys-4 Artix-7 FPGA and compared with Virtex-6 FPGA and the performance of the ciphers is compared with the existing state-of-art work.

Key words: IoT, Lightweight cryptography, RFID, FPGA, ARX, reversible logic, modular multiplication.

1. INTRODUCTION

Internet-of-Things (IoT) where billions of devices ranging from the tiny devices like the sensors, actuators, RFID, or larger devices of the industrial machines or SCADA in the industry work together in tandem to perform a critical task of exchanging information. Security and integrity of the data has become one of the major concerns for the realization of the IoT [1]. Since, the IoT devices are comprised of lesser memory and lower power, hence the term resource-constrained devices. This has led NIST to start a lightweight cryptography project [2]. State-of-the art implementations of lightweight-cryptographic algorithms have been presented [3]. Many of these proposed ciphers are ARX based (Addition-Rotation-XOR). The implementations of ARX based ciphers are faster and are more optimized than SPN based ciphers [4], [5]. A design and implementation of ARX based simple lightweight cryptographic algorithm is presented in this paper. SPARX based ARX/MRX designs has been implemented with great efficiency across several embedded systems. It is in the top 6 among the most efficient software implementations due to its optimized code.

ARX/MRX represents symmetric-key algorithm implemented using the basic operations: modular-addition, bitwise-rotation and EX-OR while the latter implemented using the operation modular-multiplication, bitwise-rotation and EX-OR. Modular-addition/Modular-multiplication is the source of non-linearity for these ARX/MRX based algorithms compared to S-BOX based designs which uses S-Boxes as source of non-linearity.

2. RELATED WORK

A review of some State-of-the-art implementation of the existing cryptographic algorithms for the resource constrained algorithms is presented.

Paper [3], discusses the need for the lightweight cryptography and limitations in terms of security of the IoT and challenges in implementing them in the constrained devices are highlighted. It is found that a few of the existing lightweight cryptographic algorithms do not exploit the trade-offs between security and efficiency.

Paper [6] presents, an implementation of symmetric-key encryption methods based on ARX design which has proven resistance to differential and linear cryptanalysis. A Block-cipher SPARX – a family of ARX-based is designed based on the long trail design strategy methodology is presented. The 32-bit S-Boxes of SPARX are based on ARX

design has provably secured against differential and linear cryptanalysis. Further, SPARX can be implemented efficiently in number of embedded platforms. It is ranked as one among the top 6 along with LEA, SPECK, SIMON, and others, as the most efficient ciphers, due to its optimized software implementations.

Paper [7], proposes a faster method that implements an ARX-based encryption that encrypts block of data, a two-way operation technique-that computes two-modular-additions or two-rotations defined with 2^{16} in parallel with 32-bit variable. SPARX-64/128 and CHAM-64/128 are applied and the performance is estimated in terms of execution time (cycles per byte) on a 32-bit Advanced RISC Machines processor. A large amount of improvement in execution time has been achieved. An improvement in performance for SPARX-64/128 of about 31.31% in key-schedule 53.31% in encryption, while CHAM-64/128 has improved performance of 41.22% in encryption and 19.40% in key schedule

Paper [8], proposes a hybrid cryptographic algorithm that combines XTEA-IDEA- LFSR (i.e., Combination of XTEA (Extended-Tiny-Encryption-Algorithm), IDEA (International-Data –Encryption-Algorithm) and an LFSR (Linear-Feedback-Shift-Register) method for key generation. Performance evaluation for different Virtex devices for existing and ID-XT-EA-LFSR algorithm has been analyzed. It has been observed that FPGA performance has been efficiently reduced by LP-Virtex-6 device compared to Virtex-7. The performance of FPGA for the proposed algorithm on LP-Virtex-6 device has been improved by 21.27 %, 76.92% and 53.125% on LUT, Flip-Flop and Slices respectively compared to the QTL algorithm.

In paper [9], the image is encrypted and decrypted using AES-128-bit core and is implemented in FPGA. Here, first the image pixel values are converted into a hexadecimal values using the MATLAB code. Then, UART transmits the plain-text hexadecimal values to the FPGA for encryption operation. For decryption the reverse operation is performed. The simulation and synthesis for the AES 128-bit core is performed on Spartan-3E-1600E FPGA using Xilinx ISE 14.3. The performance parameters with respect to power, area and latency are analysed. For encryption is 6%, 2% and 5% of Slices, Flip-Flops and 4-input LUTS respectively, while the latency and power consumed for 128-bit core is 6.645 ns and 441.91 mW respectively. Similarly, for the decryption 7%, 2%, 7% slices, Flip-Flops and 4-input LUTS respectively with latency and power consumption of 7.770 ns and the total 442 mW respectively.

3. THE PROPOSED ENCRYPTION SCHEME

The objective behind the proposed encryption method is encryption digital images by simplest, easiest and highly secured method of encryption and decryption that generates a good diffusion between the plaintext and the cipher text. The algorithm operation is performed on on-byte and thus addition and multiplication modulo operations can be applied image pixels.

The proposed encryption scheme is a SPARX based ARX scheme [10], [11] comprises of four phases. The Keys required for the encryption are generated using the novel encryption scheme in the first phase, In the second phase the plaintext is added (or multiplied) with the generated key to perform the Addition-modulo (or Multiplication Modulo) operation, Shift operation or Rotation is performed with the result of the second phase while the third phase the result of second phase is XORed with the next set of keys.

The schemes proposed are based on Addition-modulo and multiplication-modulo which infuses source of nonlinearity when compared to the cryptographic algorithms that uses S-box (Substitution-Box) as a source of nonlinearity, hence creating more confusion (the relation between cipher-text and key is made more complex) and diffusion (change in one-bit at input may change n-output bits). The above operations are preferred over the substitution-permutation networks due to two main reasons. One is the use of look-up tables for the

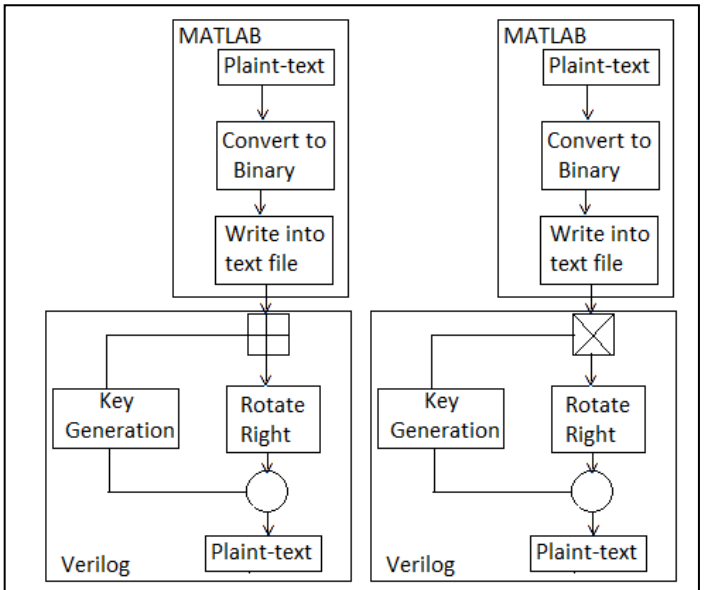

Figure 1: Block diagram of the proposed Encryption Scheme

S-Box based designs are eliminated.

Secondly the number of operations or rounds are minimized as compared to the lightweight ciphers like PRESENT, SIMON etc [4] which uses many rounds. Thus reducing the area and the power dissipation and increasing the speed for encryption.

Figure1 shows the proposed encryption scheme.

3.1 The Proposed Encryption Process

The encryption process for the algorithm comprises of two steps. The first step of the encryption process is the key generation. In this paper a unique key generation scheme is

developed. The algorithm uses two keys K1 and K2 for encryption. The second step is the encryption process. Here step it adds/multiplies the plaintext with K1 and then performs modulo operation. In the second step it rotates right by n/2 bits and in the third step the output of the second step is again XORed with K2 thus making it more secured.

3.2 Key Scheduling

The generation of random numbers is an essential element in the encryption and decryption process of the plaintext [12]. The security of the information depends on the key. The entire security of the information depends on the key, if the attacker gets to know the key, the secrecy of the information is lost. Hence the designer has to implement the key generation technique in such a way, that it shall be difficult to reveal the key K even by generating an estimate K'.

The novel key generation scheme is designed using swap, addition-modulo, Rotate, XOR and Bit shuffle operations (SARXS operations) [13], instead of using fiestel-structure or the S-BOX techniques this paper uses the above operations, so as to create significant confusion and diffusion thus making the encryption scheme computationally secure and its low power and low area making it suitable for secure IoT. Fig.2 shows the proposed key scheduling scheme.

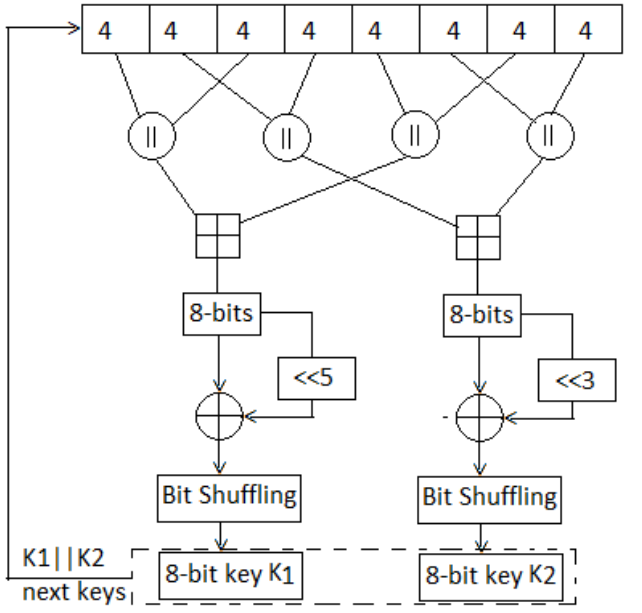

Figure 2: Key Scheduling Algorithm

Since an image pixel consists of 8-bits so in this scheme, to encrypt a plaintext of 8-bit, the key generation process starts with an initial seed of 64-bit and then after initial swap operation, an addition modulo n is performed where n is any prime number (e.g. if the plaintext is gray-scale image then n=257).resulting in an 8 bit value and then the result is XORed and next left shifted (in this case by 4-bit) the result and the keys K1 and K2 for encrypting an 8 bit data.(image pixel) The resulting 16 bit data is fed-back to generate next set of keys similar to LFSR.

3.3 Encryption using ARX scheme

In addition-modulo operation of the encryption step, each image pixel of the image is added with the corresponding key (random numbers (k1)) generated, and modulo operation is performed. [14]- [16].

$$Ca = (Pa + K1) \bmod n$$

Where Pa is the original image, K1 is the first set of keys generated and Ca is cipher-text after modulo-addition and n is a largest prime number of the block of plaintext considered.

The next step the output Ca is rotated by right by n/2 bits giving the output Ca2.

$$Ca2 = (Ca >> n/2)$$

this output is XORed with key K2 thus giving the cipher text

$$Cipher\ text = Ca2 \oplus K2.$$

3.4 Encryption using MRX scheme

In multiplication-modulo operation, the encryption step is performed on each image pixel of the image and the corresponding key (random numbers (k1)) generated using modular-multiplication method [14], [17].

$$Ca = (Pa \times K1) \bmod n.$$

The next step the output Ca is rotated by right by n/2 bits giving the output Ca2.

$$Ca2 = (Ca >> n/2)$$

this output is XORed with key K2 thus giving the cipher text

$$Cipher\ text = Ca2 \oplus K2.$$

4. SOFTWARE AND HARDWARE IMPLEMENTATION

The proposed encryption scheme has been implemented in software and hardware, the analysis and the results has been discussed in this section.

4.1 Software Implementation

The above ciphers are implemented on a MATLAB software platform, so that encryption scheme can be implemented efficiently and results in an optimized performance [18]-[20]. To exhibit the effectiveness and success of the proposed system, a standard 256×256 gray scale image is used as the plaintext image. The Histogram Analysis, Correlation Analysis and Entropy Analysis for the grayscale image are performed to verify the security of the image.

4.2 FPGA Implementation

ARX encryption scheme which consists of three simple operations: Addition-modulo, Shift and XOR. While the MRX encryption scheme consists of Multiplication-modulo, Shift and XOR. The implementation of Addition-modulo comprises of two steps first implementing adders using reversible logic and second designing modulo algorithm using Vedic-mathematics and reversible-logic. While the implementation of multiplication-modulo comprises of designing multiplication using reversible logic and Vedic-mathematics. Thus design of adders, multipliers and modulo algorithm are the basis of the processor design.

Reversible-logic is a highly promising computational method because of its ability to eliminate loss of information. Usage of reversible gates in the system supports retrieving the inputs from the outputs. The unique feature of reversible gates is that the number of inputs is equal to that of the outputs, hence making it a one on one mapping. Quantum cost of a reversible gate [21] is defined as the number of primitive reversible gates needed to form the desired gate

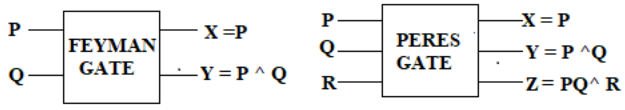

Figure. 3: Reversible Logic Gates

The design is hierarchal in structure. 1bit reversible adder and half adder using basic gates form the leaf cells in reversible adder and adder using basic gate respectively. The reversible 1bit adder uses a control signal that is assigned to 0 for addition and 1 for subtraction. Here, the circuit is modified to perform only addition, hence reducing the number of reversible gates from four to two. The prototype has two Feynman gates and two Peres gates, while the modified design has only two Peres gates. Feynman gate is a 2×2 reversible gate which can be used as an inverter by assigning the other input to 1 and has one quantum cost. Peres-gate is a 3×3 reversible gate with four quantum cost. g1 and g2 are the garbage outputs. Peres and Feynman gate is shown in Fig.3. Figure 4 depicts the block diagram of a full-adder using two reversible gates.

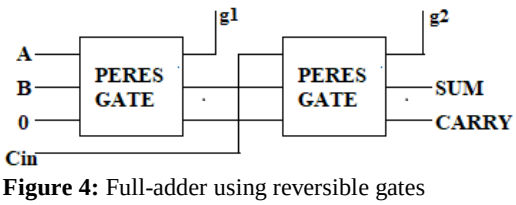

Figure 4: Full-adder using reversible gates

An N-bit reversible-adder is implemented by looping the adder units. Fig.5 depicts the use of two 1-bit adders for 2-bit addition. Hence has been observed that the number garbage outputs increase with the increase in the length of the inputs.

A 32-bit adder for performing the addition-modulo operation is implemented using reversible logic gates. The Vedic methodology is implemented using reversible and basic gates. Fig.6 describes the Vedic multiplier methodology. 2×2 Vedic multiplier is the leaf module for an N×N multiplication. For a 4×4 multiplication, the multiplicand and multiplier are divided into two groups of 2bits each. 2×2 multiplication is performed on these groups and the partial products are obtained. These partial products are then modified to get an 8bit product term.

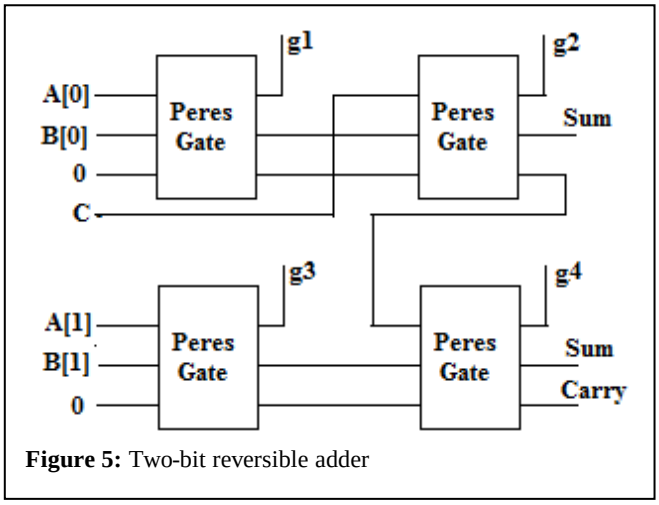

Figure 5: Two-bit reversible adder

Montgomery modular algorithm is modified and implemented. Security and speed are some of the aspects that need to be taken into consideration when designing a cryptographic system. Along with these aspects timing, area and power constraints also need to be considered. Even operations like exchanging of keys, is done using modular operations like exponentiation, addition, etc. which consume more time and area. To avoid these issues, Montgomery modular algorithm is used. The Montgomery algorithm uses large input values in the range of 512 bits and more.

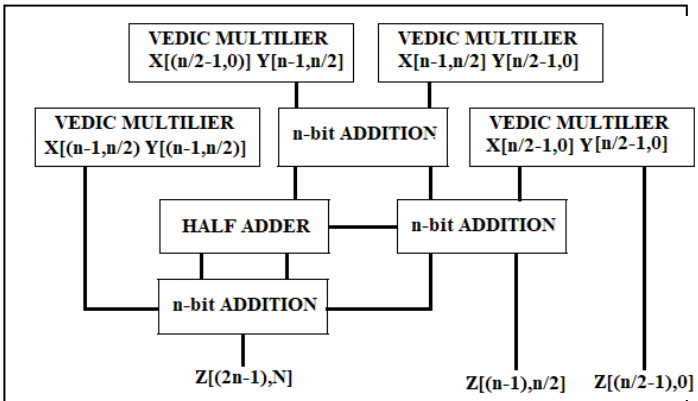

Figure. 6: Block Diagram Vedic-multiplier

Fig.7 is the flowchart of Montgomery Modular algorithm against the classical modular multiplication.

In the proposed algorithm, the inputs z, y and p are four hexadecimal digits wide i.e., 16-bits each. The condition for

value of R is that it should be greater than p. Here, the computations are done using hexadecimal digits, in order to reduce the complexity and understand algorithm. Unlike in, the value of R is taken to be equal to b^k where b is the base and k is the width of the input. Hence the value of R is equal to16^4. This way the process of converting the inputs into the Montgomery domain can done by appending four hexadecimal 0's and then dividing with p hence cutting down two multipliers. The converted inputs are then multiplied. Here it is done using an algorithm called Montgomery multiplication. Fig.8 shows the flowchart for stepwise implementation of Montgomery Multiplication.

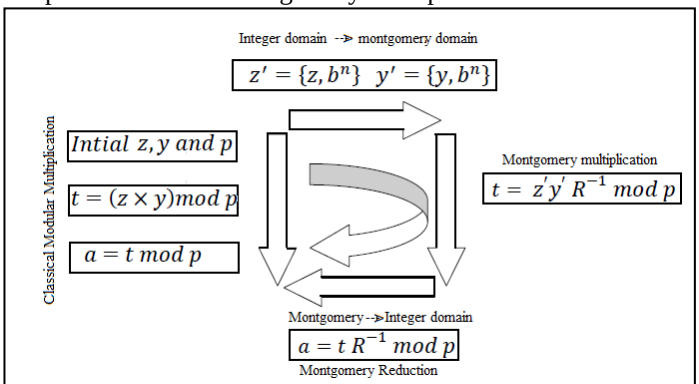

Figure. 7: Flowchart of Montgomery Modular algorithm

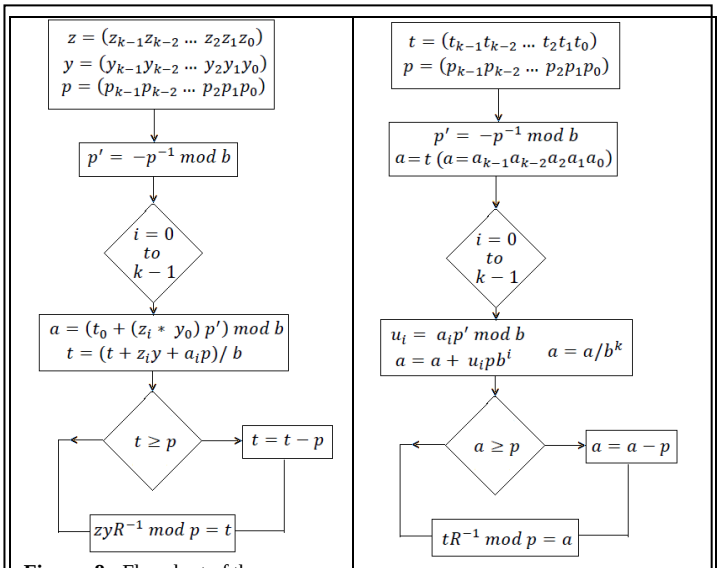

Figure 8: Flowchart of the Montgomery Multiplication

Figure 9: Flowchart of the Montgomery Reduction

Next step is the reduction of the product and returning the product to the integer domain. For this purpose, another algorithm called Montgomery Reduction is used which is also the last step of Montgomery modular operation. The additional data that is needed for the algorithm is inverse of p that can be calculated using an inverse algorithm. Fig.9 depicts the flowchart implementation of Montgomery Reduction.

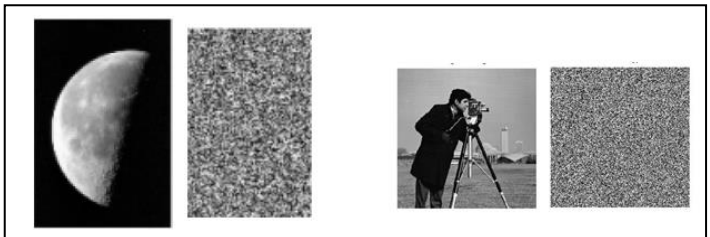

Figure 10: Encryption Results for moon and cameraman image

5. RESULTS AND ANALYSIS

The software implementation of the proposed system has been done using the MATLAB Tool and the simulations and synthesis are performed using Xilinx-Vivado verified on the Nexys-4 Artix-7 FPGA. The performance of the proposed system is evaluated on the basis of the following software parameters in the MATLAB tool.

- Visual Testing
- Histogram of the plaintext image and encrypted image
- Value of correlation coefficient of plaintext and encrypted image
- Entropy

5.1 Visual Testing

From the observation one can see that the encrypted images will not give any clue on plain images to the attacker's fig.10 represents the image encryption results for moon image and Cameraman image.

5.2 Histogram Analysis

It is of utmost importance, that the original image and encrypted image are not statistically identical to prevent information being leaked on to the attackers. The histogram analysis shows the distribution of the image's pixel values. The image histogram of an original image consists of distribution of different pixel values with sharp rises and declines i.e., the histogram of the original image has a non linear distribution as shown in figure 11a. While the histogram of the image after encryption comprises of uniform distribution in figure 11b.

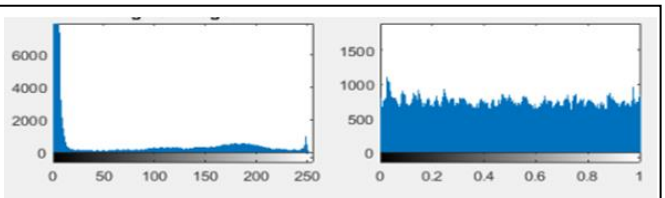

Figure 11: Histogram Analysis of the moon image

5.3 Entropy Analysis

The Entropy is defined as a measure of randomness or uncertainty associated with a random variable.
It is calculated using the formula

$$Entropy = \sum p_i(i)\, log_2\left(\frac{1}{p_i(i)}\right)$$

$p_i(i)$ is probability that pixel with grayscale value occurs

$p_i = \frac{n_i}{N}$

Ideal entropy value is 8 which correspond to absolute randomness. Table 1 shows the Entropy analysis for Cameraman-Image, Lena-Image and Panda-Image and is compared with [22]. It is seen that the average information entropy is 7.9 < E < 8 , that is, all the parts of the image have a higher degree of randomness that implies that encrypted images have a better security Higher the value of entropy, better the level of security.

Table 1: Entropy analysis

Methods/Images	Lena	Cameraman	Panda
Proposed ARX encryption	7.9947	7.9944	7.9938
Proposed MRX encryption	7.9894	7.9887	7.9938
Ref[22]	7.9973	7.9973	7.9971

5.4 Correlation Coefficient

The correlation between any two randomly selected neighbouring pixels is analyzed. Randomly 1000 adjacent pixel pairs are selected. The correlation coefficient between any two adjacent pixel pairs can be calculated as

$$cov(p,q) = \frac{1}{N}\sum_{i=1}^{N}(p_i - E(p))(q_i - E(q))$$

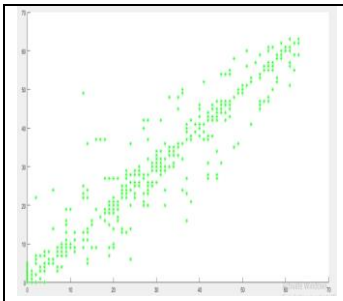

Figure 12: Correlation of an Original Image

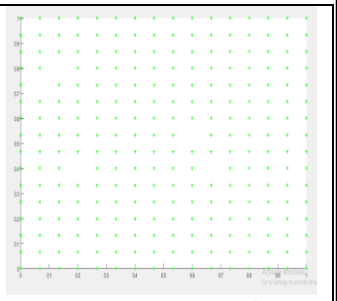

Figure 13: Correlation of an Encrypted Image

$$r_{pq} = \frac{cov(p,q)}{\sqrt{D(p)}\sqrt{D(q)}}$$

Where p and q are the values of two adjacent image pixels.

$$E(p) = \frac{1}{N}\sum_{i=1}^{N} p_i$$

$$D(p) = \frac{1}{N}\sum_{i=1}^{N}(p_i - E(p))^2$$

Table 2: Correlation Coefficient between the pixels of the cipher Image

Methods/Images	Lena	Cameraman	Panda
Proposed ARX encryption	-0.0022	0.0398	-0.0058
Proposed MRX encryption	-0.0023	-0.0250	-0.0138
Ref[22]	0.0012	0.0012	0.0022

Table 2 shows the correlation coefficients for different samples of encrypted images compared with Ref [22]. Table 2 shows the correlation analysis of three different encrypted images. The values obtained are close to zero and this shows that, the relation between image-pixels in encrypted image are not strongly related to each other. Hence, we conclude that the proposed ARX encryption scheme is secure. The Figures 12 and fig.13 are the correlation of original image and encrypted image respectively

5.5 FPGA Performances

The results design summary is obtained in Table 3 shows timing/ critical Path delay (logic delay + net delay) Slice LUTS, Registers and IOB's and total on chip power (Dynamic +Static) in terms of Watts as obtained using Xilinx Vivado Tool and Area in terms of micrometer square and power (internal +Switching + leakage power) in terms of microwatts as obtained using OASYS-RTL tool (45nm Technology).

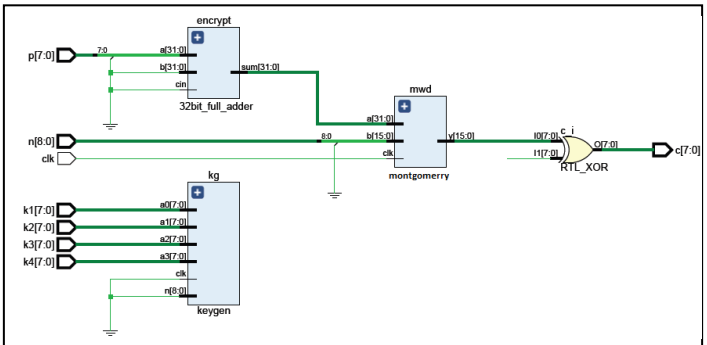

Figure 14: RTL schematic of ARX processor

- Area
- Power
- Timing

Fig.14 shows the RTL schematic of the ARX-processor implemented in Xilinx-Vivado Tool.
Fig.15 shows the RTL schematic of the MRX processor.

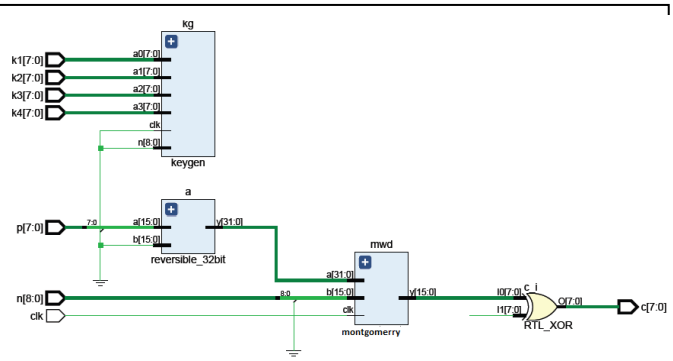

Figure 15: RTL schematic of MRX processor

To evaluate the performance parameters like Area, and Power, the ARX /MRX based SARXS key-generation scheme was implemented using Xilinx-Vivado tool by using Verilog code. Camera-man image is taken as input image. The image is converted into binary-text file format in MATLAB, which is given as input to Verilog. The proposed encryption scheme has been implemented in Artix-7 Nexys-4 FPGA and also in Virtex-7 FPGA. Due to its upward compatibility, low power, high performance and flexibility compared to the ASIC makes it much more suitable for VLSI implementations. The performances have been compared with state of the art implementations like Ref [8], Ref[9], Ref[11],Ref[25].

The FPGA performances for the proposed encryption scheme are evaluated on Virtex-7 Xc7vx330t, Artix-7 7a100tcsg324 devices. Table 3 shows the FPGA performance evaluation for the proposed encryption schemes. From the results it can be inferred that Artix-7 FPGA device has improved performance than Virtex-6 or Virtex-7 devices. The proposed ARX has been compared with conventional cryptographic algorithms like the AES Ref[9] and the Light-Weight cryptographic algorithms like hybrid Algorithm Ref[8] ARX Ref[11] and HIGHT Ref[25]and has been found that the proposed ARX/MRX encryption schemes based on reversible logic and Vedic mathematics achieved better results. Fig.16 shows the comparative analysis of LUT and Slices for Virtex-6, Spartan-6 and Artix-7 FPGA performance. And Fig.17 shows the comparative analysis of power and timing for Virtex-6, Spartan-6 and Artix-7 FPGA performance.

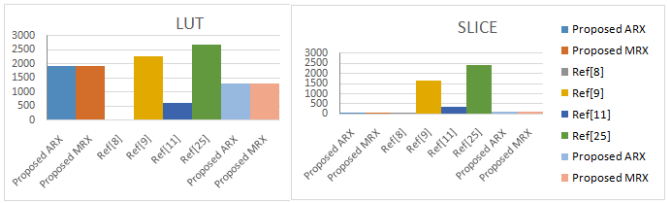

Fig.ure 16: Comparative analysis of Virtex-6, Spartan-6 and Artix-7 FPGA performance.

Table 3: Performance evaluation of the different FPGA devices for existing and proposed algorithm.

Target Devices	Cryptographic Algorithms	LUT	IOB	Slice Registers	Power(W)	Timing(ns)
Virtex 7 Xc7vx330t	Proposed ARX	1920/204000	51/408000	16/51000	0.143	74.336
	Proposed MRX	1920/204000	51/408000	32/51000	0.143	74.336
Artix 7 7a100tcsg324	Proposed ARX	1314/63400	26/210	98/126800	40.458	84.262
	Proposed MRX	1306/63400	0/210	98/126800	40.115	84.262
Virtex 7 Xc7vx330t	Ref[8]HCA	37/204000	18/408000	16/51000	--	121.4
Spartan-3E	Ref[9] AES	2255/29504	6/250	1661/14752	0.441	--
Spartan 6	Ref[11] ARX	604/9312	204/232	346/4656	--	--
Spartan 6	Ref[25]HIGHT	2689/27288	1/296	2409/54576	0.607	8.34

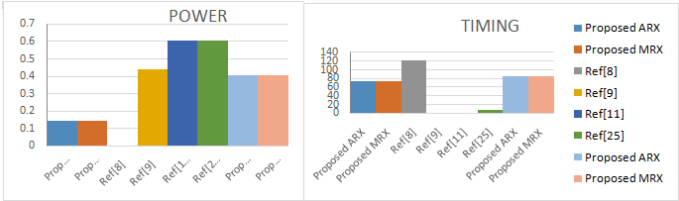

Figure 17: Power and Timing Analysis of the proposed scheme.

5. CONCLUSION

This paper presents the image encryption scheme by using the proposed ARX/MRX encryption scheme. The image was converted into binary format by using MATLAB version 2018a and the experimental analysis like histogram analysis, entropy analysis and correlation coefficients were applied, the results obtained inferred that the proposed encryption

schemes provided sufficient security. The binary value obtained from the MATLAB is given as input to Verilog and Area (in terms of LUT, FFs and IOB's), Power and timing reports are generated by using Xilinx-Vivado Tool. Hence it can be concluded that the above encryption scheme can provide sufficient security and less area and power and can be better suited for Lightweight Cryptography.

In future work, various other adders and different logic styles can be implemented and also with different for key generation schemes can be applied to get better security, improved performance and efficiency.

ACKNOWLEDGEMENT

The authors would like to thank the Department of Electronics and Communication and Engineering, Canara Engineering College Mangalore and Visvesvaraya Technological University, Belagavi for the support for carrying out the research work.

REFRENCES

1. Mario Weber, Marija Boban , ***Security challenges of the Internet of Things***, *MIPRO 2016*, May 30 - June 3, 2016, Opatija, Croatia. https://doi.org/10.1109/MIPRO.2016.7522219
2. Kerry A. McKay, Marry Bassham ,Meltem Sönmez Turan ,Nicky Mouha, "***DRAFT NISTIR 8114 Report on Lightweight Cryptography***" *National Institute of Standards and Technology Internal Report 8114* ,August 2016.
3. Katagi, M.; Moriai, S. ***Lightweight Cryptography for the Internet of Things***; *Sony Corporation, 2008*, pp. 7–10. http://dx.doi.org/10.1016/j.istr.2012.10.005.
4. Kiran Kumar V.G., Shantharama Rai C. (2019) ***Implementation and Analysis of Cryptographic Ciphers in FPGA.*** in: Abraham A., Dutta P., Mandal J., Bhattacharya A., Dutta S. (eds) *Emerging Technologies in Data Mining and Information Security. Advances in Intelligent Systems and Computing*, vol 755. Springer, Singapore.
5. S. T. Patel, N. H. Mistry, "***A survey: lightweight cryptography in WSN***," in *International Conference on Communication Networks (ICCN). IEEE, 2015*. https://doi.org/10.1109/ICCN.2015.3
6. D. Dinu, L. Perrin, "***SPARX: a family of arx-based lightweight block ciphers provably secure against linear and differential attacks***," in the *proceedings of Asiacrypt16*, 2017.
7. Byoungjin Seok and Changhoon Lee,"***Fast implementations of ARX-based lightweight block ciphers (SPARX, CHAM) on 32-bit processor***", *International Journal of Distributed Sensor Networks 2019*, Vol. 15(9).
8. Shailaja Acholli, and Krishnamurthy Gorappa Ningappa. "***VLSI Implementation of Hybrid Cryptography Algorithm Using LFSR Key***." *International Journal of Intelligent Engineering and Systems*, Vol.12, No.4, 2019.
9. M. P. Priyanka, E. L. Prasad and A. R. Reddy, "***FPGA implementation of image encryption and decryption using AES 128-bit core***," *2016 International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, 2016, pp. 1-5.
10. F. Bache, T. Schneider, A. Moradi and T. Giineysu, "***SPARX — A side-channel protected processor for ARX-based cryptography***," *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, Lausanne, 2017, pp. 990-995. https://doi.org/10.23919/DATE.2017.7927135
11. K SATHEESH, S SASI KIRAN, "***SPARX - A Side-Channel Protected Processor for ARX-based Cryptography,*** "International *Journal of Research*" Volume VIII, Issue I, January/2019.
12. Remya Justin, Binu K Mathew, Susan Abe, "***FPGA Implementation of High Quality Random Number Generator using LUT based Shift Registers***", *International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST 2015*, Science Direct Procedia Technology 24 (2016) 1155 – 1162.
13. Asmita Poojari , Nagesh HR , Kiran Kumar V G, Shantharama Rai C, "***A Novel Key Scheduling Algorithm for Lightweight Cryptographic Applications***", *International Journal of Advanced Trends in Computer Science and Engineering*, Volume 9, No.1, January – February 2020. https://doi.org/10.30534/ijatcse/2020/96912020.
14. R. Zimmermann, "***Efficient VLSI implementation of modulo 2 n ±1 addition and multiplication*** ", *Proc. 14th IEEE Symposium on Computer Arithmetic*, pp. 158-167, Apr. 1999.
15. H. T. Vergos, C. Efstathiou, D. Nikolos, " ***Diminished-one modulo 2n + 1 adder design*** ", *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1389-1399, Dec. 2002.
16. S. M. Dehnavi, A. M. Rishakani, M. M. Shamsabad, H. Maimani, E. Pasha, "***Cryptographic Properties of Addition Modulo 2n***" . *IACR Cryptology ePrint Archive 181 (2016*).
17. Zhongde Wang, G.A. Jullien and W.C. Miller, ***"An Algorithm for Multiplication Modulo (2^N-1)"*** *ASILOMAR '95 Proceedings of the 29th Asilomar Conference on Signals, Systems and Computers (2-Volume Set)* Page 956.
18. Xinpeng Zhang, "***Reversible data hiding in encrypted image***", *IEEE Signal Processing Letters*, vol. 18, No. 4, Apr. 2011.
19. Xinpeng Zhang, "***Separable reversible data hiding in encrypted image***" *IEEE Trans. on Information Forensics and Security*", vol. 7, No. 2, Apr. 2012.
20. Hayder Raheem Hashima, Irtifaa Abdalkadum Neamaa, "***Image Encryption and Decryption in A Modification of ElGamal Cryptosystem in MATLAB***", *International*

Journal of Sciences: Basic and Applied Research (IJSBAR) pp.141- 147, 2014.

21. Ruqaiya Khanam, Abdul Rahman and Pushpam, "***Review on Reversible Logic Circuits and its Application***", *2017 International Conference on Computing, Communication and Automation (ICCCA2017)*, 5-6 May 2017. https://doi.org/10.1109/CCAA.2017.8230046
22. Muhammad Usman, Irfan Ahmed, M. Imran Aslam, Shujaat Khan and Usman Ali Shah, "***SIT: A Lightweight Encryption Algorithm for Secure Internet of Things***" *International Journal of Advanced Computer Science and Applications,*Vol. 8, No. 1, 2017.
23. C. Yang, H. Wu and S. Su, "***Implementation of Encryption Algorithm and Wireless Image Transmission System on FPGA***," in *IEEE Access*, vol. 7, pp. 50513-50523, 2019.
24. B. H. Prasetio, et al., "**Image Encryption using Simple Algorithm on FPGA**," in *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol/issue: 13(4), 2015.
25. N. Sruthi, R. Nandakumar and P. Rajkumar, "***Design and characterization of HIGHT cryptocore,***" 2016 *International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES*), Paralakhemundi, 2016, pp. 205-209. https://doi.org/10.1109/SCOPES.2016.7955798
26. Sohel Rana, Saddam Hossain, Hasan Imam Shoun and Dr. Mohammod Abul Kashem, "***An Effective Lightweight Cryptographic Algorithm to Secure Resource-Constrained Devices***" *International Journal of Advanced Computer Science and Applications (IJACSA),* 9(11), 2018. http://dx.doi.org/10.14569/IJACSA.2018.091137.
27. Kiran Kumar V G, C Shantharama Rai, "***Low Power High Speed Arithmetic Circuits***", *International Journal of Recent Technology and Engineering (IJRTE*) ISSN: 2277-3878, Volume-8 Issue-2, July 2019.
28. Gunajit Kalita, Navajit Saikia, Amit Sravan Bora, ***"Design of Reversible Decoder with minimum Garbage Output"*** *International Journal of Advanced Trends in Computer Science and Engineering, Volume 9, No.3, May – June 2020. https://doi.org/10.30534/ijatcse/2020/150932020*.
29. Dr S.Sasikala, S.Gomathi, M.Kanimozhi, K.S.Lakshana , R.Karthik ***"Performance Analysis of a Low-Power High-Speed Hybrid Multiplier Circuit"*** *International Journal of Advanced Trends in Computer Science and Engineering, Volume 9, No.3, May – June 2020.* https://doi.org/10.30534/ijatcse/2020/197932020