International Journal of Applied Science and Engineering

FPGA implementation of a lightweight simple encryption scheme to secure IoT using novel key scheduling technique

Kiran Kumar V. G.*, Shantharama Rai C.

Department of Electronics and Communication Engineering, A J Institute of Engineering and Technology, Kottara Mangaluru, India

ABSTRACT

Internet of things (IoT), being the technology of this generation and several billions of electronic devices exchanging a huge secure information. With the low resource devices like the sensors, RFIDs etc. to the super computers and the clouds the security and privacy issues remain a concern. While the conventional cryptographic algorithms approved by the National Institute of Standards and Technology, could be embedded into the Low-resource devices their performance may be reduced, then the design of Ciphers for such resource constrained devices become a challenge with the security principles confidentiality, Integrity and Availability remains the same.

This paper proposes, simple encryption schemes based on arithmetic operations Addition-Modulo/Multiplication Modulo, Rotation and XOR hence the name ARX/MRX. The cipher schemes have been implemented using reversible logic and Vedic Mathematics. The adders have been implanted using reversible logic and multipliers and the modular algorithms have been implemented using the combination of Vedic maths and Reversible logic. The software and hardware implementations are presented. The Histogram, Correlation coefficient and Entropy are found for the grayscale plaintext image using MATLAB to evaluate the security, and the hardware implementation is done writing Verilog code using Xilinx-Vivado and is verified using Nexys-4 Artix-7 FPGA the performance of the encryption schemes are analysed and compared with the existing literature.

Keywords: Low resource devices, FPGA, Internet of Things, ARX/MRX, Reversible logic.

OPEN ACCESS

Received: September 5, 2020

Accepted: December 12, 2020

Corresponding Author:
Kiran Kumar V. G.
kiranvgk@gmail.com

Publisher:
Chaoyang University of Technology
ISSN: 1727-2394 (Print)
ISSN: 1727-7841 (Online)

1. INTRODUCTION

With an increasing growth of Internet-of-Things (IoT) in the recent years and the number of devices connected to the network increasing, the with different types of devices like the low-resource devices like the RFID tags or sensors etc, or larger devices like the supercomputers or Cloud work together in concert to exchange the highly secure information. With the concerns in Security and integrity also in the rise, the realization lightweight ciphers for IoT (Mohd et al., 2015) is a challenge. Such IoT devices are low resource devices that are comprised of low memory and low power requirement (Weber and Boban, 2016), hence the term Low-resource devices. Thus prompting NIST to start a project (McKay et al., 2016) for implementing cryptographic algorithms for low-resource devices hence the term Lightweight Cryptography. In this paper, Lightweight-cryptographic algorithms have been presented (Katagi et al., 2008). The ciphers proposed are based on Addition-Modulo/ Multiplication Modulo, Rotation and XOR. The ARX based cipher implementations are simple, optimized and faster than SPN based cipher implementations (Kumar and Rai, 2019; Patel and Mistry, 2015). An effectively efficient ARX/MRX based simple lightweight cipher is presented in this research. The term

effective means secure implementation of the simple encryption scheme in terms of parameters like Histogram, Correlation-coefficient and Entropy and efficient implementation considers the parameters like power area and timing.

The Simple Encryption scheme is implemented using the basic arithmetic operations with reversible logic and Vedic-mathematics: modular-addition / modular-multiplication, bitwise-rotation and EX-OR. In SPN based cryptographic algorithms the source of nonlinearity is introduced by the S-BOX and Permutation Networks while in the proposed scheme the source of non-linearity is applied by Modular-addition/Modular-multiplication operation and XOR and Rotation applies diffusion and confusion.

2. RELATED WORKS

This section discusses few of the state of the art lightweight cipher implementations for the low resource devices.

The need for cryptographic ciphers for resource constrained devices is discussed (Katagi et al., 2008), the drawbacks in terms of security for Internet of Things and the challenges in implementing them in the low resource devices are presented. It is found that a few of the trade-offs between the efficiency and the security of the lightweight ciphers are not exploited by few of the implemented ciphers is observed.

ARX based symmetric key encryption methods have proven resistance to differential and linear cryptanalysis (Dinu et al., 2017). This paper presents SPARX – a family of ARX-based block cipher. Its design is based on the long trail design strategy technique. The encryption method has proved secure against linear and differential cryptanalysis due to the design of 32-bit SBOX based on ARX method. Due to its optimized structure it is implemented efficiently in number of embedded systems. Its effective software implementation its ranked along with SPECK, SIMON, LEA and others as top 6 as the most efficient cryptographic cipher.

An ARX-based encryption method that encrypts message faster based on a two-way encryption method (Seok et al., 2019) is implemented using 2 Addition- Modulo and 2 Rotate operations defined with 2^{16} in parallel with a 32-bit variable. SPARX-64/128 and CHAM-64/128 are implemented in a 32-bit Advanced RISC Processor Machine. The performance parameters like the execution time is observed in SPARX-64/128 and an improvement is achieved by 31% in implementing the key expansion and improvement in encryption by 53% is achieved. While CHAM-64/128 achieves an improvement of 19.4% in key expansion and 41.2% in encryption.

A combination of three algorithms XTEA-IDEA- LFSR and is termed hybrid cryptographic algorithm (Acholli and Ningappa, 2019), is implemented. The hybrid cryptographic algorithm is implemented in Virtex device. The implementations of the proposed algorithm results in an efficient reduction of the FPGA performance of LP-Virtex 6 compared to Virtex-7. The FPGA performance of the implementation of the proposed algorithm resulted in improvement by 21.27% in LUT (look up table), 77% in FFs (flip flops) and 53.2% in Slices in comparison with implementation of QTL (quantitative trait locus) algorithm.

A new chaos-based post-processing algorithm (Teh et al., 2019) that provides entropy amplification and eliminates statistical bias so that performance is improved. This method employs intrinsic features of chaos like confusion and diffusion and avalanche effect (hypersensitivity to changes in input) so as to accomplish these goals. The entropy bits that generates the Random bits perturb the conditions of the hyperchaotic system (CCML) is iterated to guarantee maximum diffusion. To measure the effectiveness two types of audio based TRNG are implemented the results in comparison with the state of the art literature shows that the TRNGs have better forward and backward security, with no statistical defects.

A circuit for 32-bit random number generation (Devi et al.,) at 125MHz frequency implemented on a highly efficient Artix-7 FPGA Board. The NIST test and Diehard test were performed to pass these randomness tests. The diehard test with 15 tests resulted in values ranging 0.0052–0.925 thus proving it to be a secured random number.

3. THE PROPOSED SIMPLE ENCRYPTION SCHEME

The main objective of the proposed simple encryption scheme is to encrypt plaintext data using simpler, easier and a secure encryption method. The scheme should have an encryption and decryption technique where a good amount of diffusion and confusion is generated between the input and the output. In this research gray scale images are taken as plaintext the operation is done on byte.

The simple encryption scheme belongs to SPARX family of ciphers, that is based ARX method (Bache et al., 2017; Satheesh and Kiran, 2019) comprises of two phases. In the first phase a novel key generating scheme is used to generate the input keys for encryption. In the second phase comprises of encryption where in the Addition-modulo/ Multiplication Modulo is performed, Next Shift operation or Rotation by n/2 bits is performed the output of this step is XORed with the next key K2, so as to complete the encryption step.

The proposed schemes are makes use of the arithmetic operations like Modular-Addition and Modular-Multiplication operations, the source of non-linearity is infused by these operations when compared to those cryptographic algorithms that has SPN based structure that uses Substitution BOX and Permutation as a source of nonlinearity, the XOR and Rotate operation infuses more confusion and diffusion that makes the input output relation more complex. The two reasons for preferring the above operations over the SPN (substitution-permutation

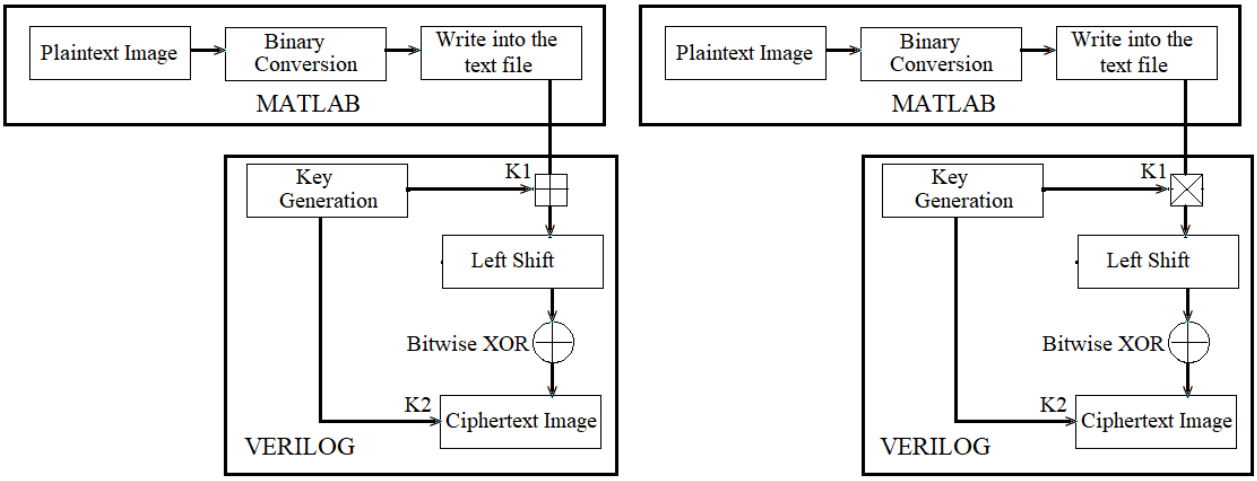

Fig. 1. The proposed lightweight encryption scheme

networks) are, one elimination of look-up tables for the S-BOX based structure. Secondly it minimizes the number of operations or rounds when compared to the lightweight ciphers like PRESENT, SIMON (Abed et al., 2019) that has many rounds. Thus the area and the power dissipation is reduced and increasing the speed for encryption. Fig. 1 shows the proposed lightweight simple encryption scheme.

3.1 Encryption Process of the Proposed Lightweight Simple Encryption Method

The encryption process comprises of two steps. In the first step of the encryption process is the key generation (random number generation). In this research a novel key generation method is designed. Two keys K1 and K2 are used for encryption of one byte of data. The second step is the encryption process where the plaintext adds/multiplies the plaintext with key K1 and then modulo operation is done. Next left shift by 5 bits and lastly the output of the previous step is EXORed with key K2 thus making the encryption more secured.

Initially the plaintext image is converted to binary and then saved in a text file using MATLAB code. Next the encryption stages are implemented in Verilog and comprise of the following operations.

3.1.1 Addition-Modulo Operation

In this step the input plaintext (image pixel value) of 8-bits and the key K1 are taken and addition modulo n is done, if Pa is the input plaintext of 8-bits then key K1 while n is a prime number, thus the resulting output of this stage is Ca and is given by

$$Ca = (Pa \oplus K1) mod\ n$$

3.1.2 Multiplication Modulo Operation

In this step the input plaintext (image pixel value) of 8-bits and the key K1 are taken and addition modulo n is done, if Pa is the input plaintext of 8-bits then key K1 while n is a prime number, thus the resulting output of this stage is Ca and is given by

$$Ca = (Pa \times K1) mod\ n$$

3.1.3 Left Shift

In this stage, the output of the previous stage Ca is left shift by n/2 bits.

3.1.4 EXOR with Key K2

The last step of the encryption process is the EXOR operation. The output of left shift operation is XORed with key K2 to result in the cipher-text.

3.2 Key Generation Process

The key generation technique is the most vital part of any encryption and decryption process (Justin et al., 2016). The security of the entire data depends on the key generation scheme, the security of the data is said to be lost, if the intruder gets to know the key. Thus the design of the key generation technique has to be implemented in such a way, that it shall be hard to deduce key K, known the initial approximate of the key K'.

The simple encryption scheme employs a novel key generation scheme that uses Permutation, addition-modulo, Shift, XOR and Bit-shuffling operations (Poojari et al., 2020), the shift, permute and bit-shuffling operation infuse more diffusion while addition-modulo operations infuses nonlinearity thus infusing the confusion for the attacker. Thus the proposed cipher scheme is computationally secure and its optimized design with low power and low area makes it a suitable candidate for IoT. The proposed novel key scheduling scheme is shown in Fig. 2.

The key generation process takes an initial seed of 64-bit (can be extended to 128 bits) after an initial permutation then addtion-modulo operation is done (for performing modulo n operation n = 257 is selected since it is a prime number), next the 8-bit result is left shift by 5 bits and

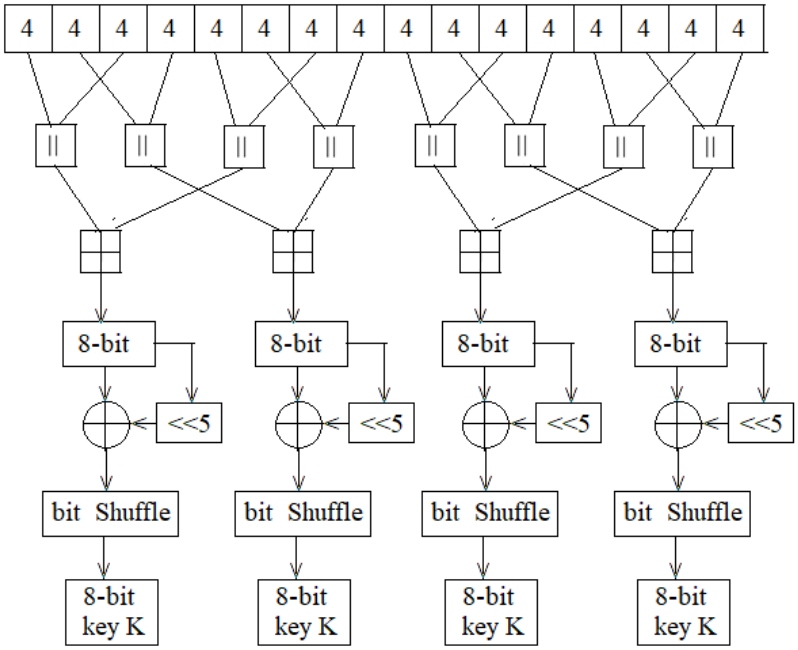

Fig. 2. The key generation process for the proposed simple encryption scheme

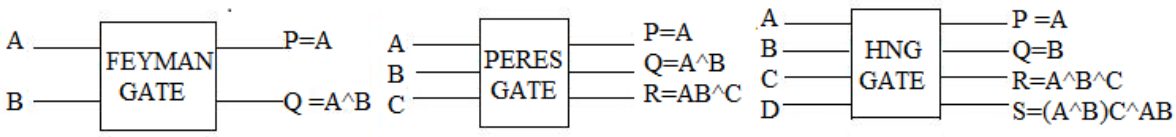

Fig. 3. Feynman, Peres and HNG gate

XORED with the previous result and in the last step bit-shuffling is implemented so as to get the 8-bit key which is then fed as an input keys K1, K2 to the encryption process.

3.3 Reversible Logic and Vedic Mathematics based Simple Encryption Scheme

The implementation of simple encryption scheme consists of operations: Addition-modulo, Left-Shift and EXOR operations for ARX encryption scheme and Multiplication-modulo, Left-Shift and EXOR operations for MRX scheme. Vedic-mathematics and reversible-logic is used to implement the Addition-modulo which consists of two steps. In the first step the adders are implemented using reversible logic and then in the second step the modular algorithm is designed using modified Montgomery algorithm. The multiplication-modulo is designed using reversible logic and Vedic-mathematics. The multipliers are implemented using Vedic-mathematics. The adders, multipliers and modulo algorithm are the basic modules of the processor architecture implementation.

Reversible-logic is a new computational technique that has an ability to minimize any loss in information. Any reversible logic should satisfy following two conditions

- Mapping of input pattern to specific output pattern.
- Equal number of inputs and outputs (1:1 mapping).

For implementing the simple encryption scheme three reversible Logic gates Feynman Gate, Peres Gate and HNG gate are used as seen in Fig. 3.

The hierarchical design structure is used to implement the proposed encryption scheme. The Leaf cells are half-adder and one-bit reversible adder for designing n-bit adder the design is implemented using Peres Gate. A control signal is used by the reversible adder that adds when

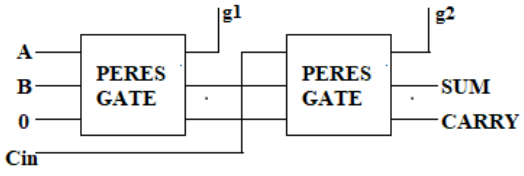

Fig. 4. Full-adder using Peres gate

the control signal is assigned 1 and subtracts when the control signal is assigned 0. Thus it can implement addition/subtraction operation. Since in this encryption scheme only addition operation is performed thus the gate can be reduced from four to two. While the prototype consists of four gates the modified adder consists of only two gates. A 2 × 2 R L Feynman gate can also be used as an inverter by assigning the other input to 1 and thus the quantum cost of this RL gate is one. A 3 × 3 R L Peres-gate

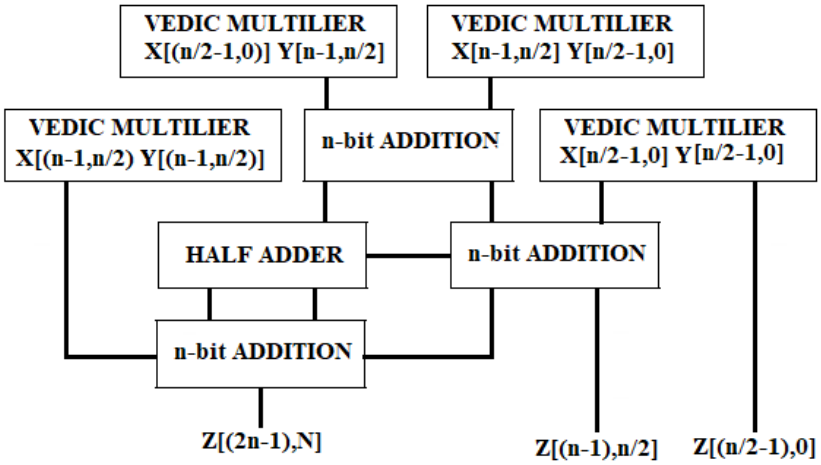

Fig. 5. Multiplication using Vedic-mathematics and reversible logic

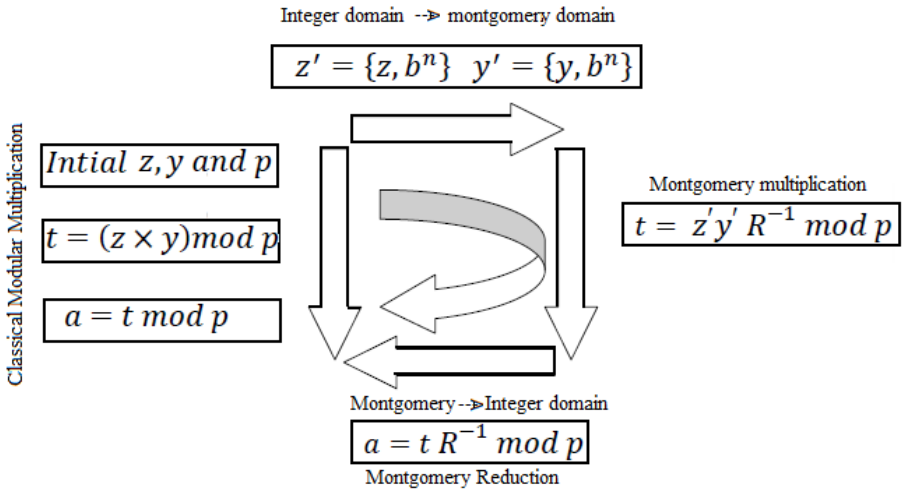

Fig. 6. Classical modular algorithm versus Montgomery modular algorithm

has a quantum cost of four. The garbage outputs are g1 and g2. The full-adder designed using two reversible gates is shown in Fig. 4

Similarly, n-bit adder can be designed by lopping the one-bit adder. A 32-bit adder is designed using the Peres gate for the simple encryption scheme. Fig. 5 describes the Vedic multiplier implementation. First 2 × 2 Vedic multiplier is implemented and this forms the leaf module for designing an N × N multiplication. For implementing a 4 × 4 multiplication, both the multiplicand and multiplier will be split into two parts of 2-bits each. The partial products are obtained by performing a 2 × 2 multiplication on the parts. Then a 8-bit product is obtained to from these partial products.

The implementation of modular algorithm is performed by Modified Montgomery modular algorithm. Considering the security and speed as the aspects for the design of the cipher scheme and other hardware implementation parameters like timing, area and power the design is implemented. Some cryptographic algorithms perform the exchanging of keys by implementation of modular arithmetic using addition, multiplication, exponentiation, thus increase in area and timing. These issues could be avoided by implementing modified Montgomery modular algorithm. Large range of input values like 512 bits or so can be implemented using the Montgomery algorithm.

The classical/Euclidean modular algorithm versus the Montgomery Modular algorithm is shown in Fig 6.

The algorithm implements *zy mod p.* with z, y and p each of 16-bits each (four hexadecimal digits) are the inputs to the algorithm. Assume a value, R is the value in Montgomery domain should be such that $R > p$, for easier understanding and reduce the complexity the calculations are implemented using hexadecimal values. Thus $R = b^k$ is taken, where b is the base and k is the width of the input. Thus $R = 16^4$. Thus by adjoining 4 hexadecimal zeros the values are converted to Montgomery domain and then the number is divided by p. This reduces the structure by two multipliers. Then multiply the inputs using Montgomery multiplication algorithm. Montgomery Multiplication algorithm implementation steps are shown in Fig 7.

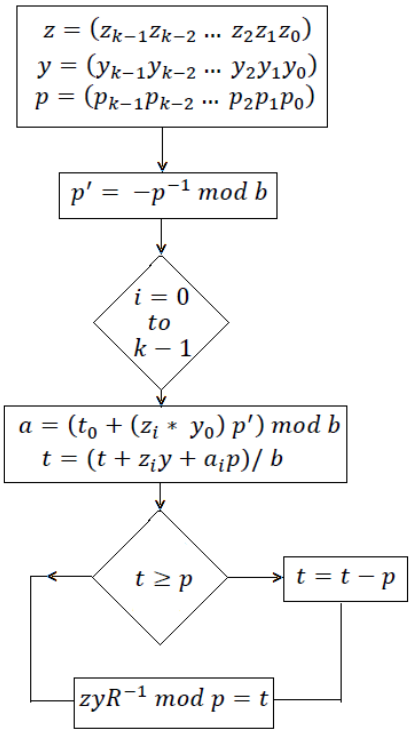

Fig. 7. Montgomery multiplication

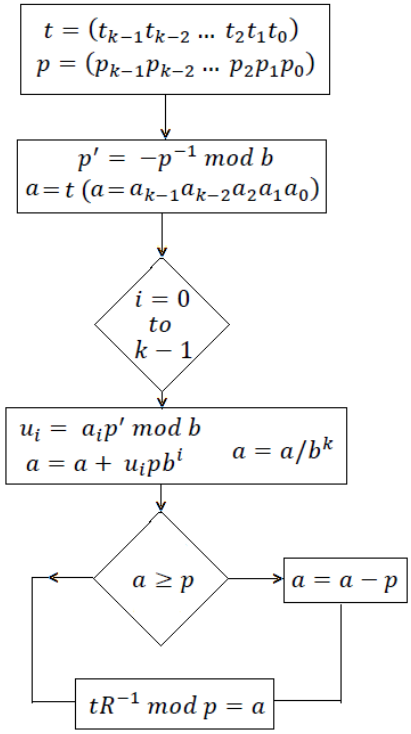

Fig. 8. Montgomery reduction

The next step of the implementation is Montgomery reduction where the product is returned to the integer domain and lastly inverse algorithm to deduce inverse of p is implemented.

4. IMPLEMENTATION RESULTS AND ANALYSIS

The simple encryption scheme is implemented in software using MATLAB Tool and Hardware using Xilinx Verilog platform. The 256 × 256 gray-scale image is taken as plaintext input.

4.1 MATLAB Implementation

The evaluation parameters for software implementation like Visual Analysis, Histogram, correlation coefficient and Entropy are performed on taking image as plaintext.

4.1.1 Visual Testing

Fig. 9 (a), (b) and (c) shows image encryption for the proposed simple encryption scheme implemented in MATLAB, it can be inferred that the encrypted images do not give any hint about the original image making it difficult to the attackers.

4.1.2 Histogram

From the histogram of original input image has a nonlinear distribution of pixel values with different peaks and crests in the histogram while the cipher text image has a uniform distribution of pixel values as shown in Fig. 9 (a), (b) and (c), thus shows making the cipher text secure, thus it is a vital requirement, that the plaintext-images and cipher-text images are not statistically identical thus preventing the leak of information to the intruders.

4.1.3 Entropy

The Entropy of an image is a measure of a quantity used to characterize the sum of information that an encryption algorithm must encode. And is calculates as

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

where pi represents the probability of occuring pixel i. The value that is said to be absolute randomness is said to have an ideal entropy value is 8. Entropy values for three different images are calculated (shown in Table 1) and is compared with Usman et al. (2017). If the entropy values are in the range 7.9 < E < 8, then it is said that, the image have a higher degree of randomness thus ensuring a better security of the cipher-text image.

4.1.4 Correlation Coefficient

The strength of the any encryption scheme is evaluated by the correlation. Two neighboring pixels are selected randomly and the correlation between them is calculated. The correlation coefficient between any original plaintext image and the cipher-text image is computed by

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

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

Where p and q represent the image pixel values.

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

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

The correlation coefficient for three cipher-text images is calculated and shown in Table 2. The calculated values nearer to zero thus infers that the plaintext image and cipher text image pixel values are not strongly related. Thus

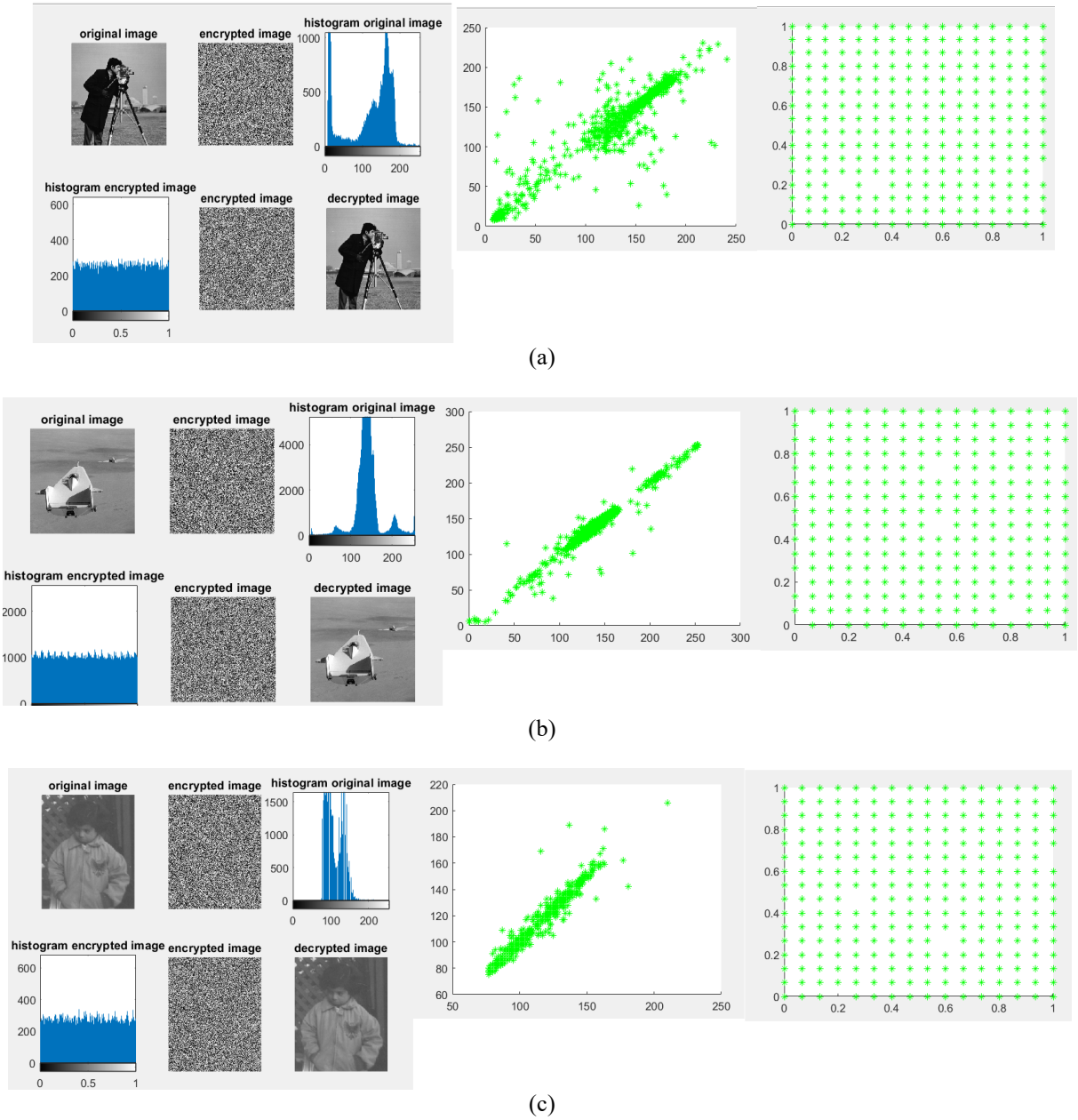

Fig. 9. Visual, Histogram and correlation-coefficient of gray-scale images. (a) Cameraman (b) Lifting bod and (c) pout images

Table 1. Comparitive results for entropy of cipher image

Encryption scheme	Cameraman	Lena	Panda
Proposed ARX encryption	7.9944	7.9947	7.9938
Proposed MRX encryption	7.9887	7.9894	7.9938
Usman et al. (2017)	7.9973	7.9973	7.9971

deducing that proposed encryption scheme is secure. The correlation between the plaintext-image and cipher-text image is observed in a graphs of Fig. 9 (a) (b) and (c) shows that the correlation of plaintext images is concentrated at the diagonal of the x-y axis while the correlation of the cipher text images is uniformly distributed all over.

4.2 FPGA Implementation

The FPGA implementation is done writing a Verilog code using Xilinx Vivado Tool. The parameters like Path delay, Slice LUTS, Slice Registers and IOB's and total on chip power and Area (Dynamic + Static) in terms of Watts are evaluated.

The RTL schematic of simple encryption schemes using ARX and MRX operations is shown in Fig. 10 and Fig. 11 respectively.

Virtex-7 Xc7vx330t, Artix-7 7a100tcsg324 devices are used to evaluate the FPGA performances for the proposed encryption scheme. From Table 3 it can be seen that Artix-7 FPGA device has better performance compared to Virtex-6 or Virtex-7 devices. The proposed SES ARX/MRX has been compared state-of the art works, both with conventional ciphers like the AES (Priyanka et al., 2016) and the light-weight ciphers like hybrid Algorithm(Acholli and Ningappa, 2019) ARX (Satheesh and Kiran, 2019) and HIGHT (Sruthi et al., 2016).

Table 2. Results for correlation coefficient of the cipher image

Encryption scheme	Lena	Cameraman	Panda
Proposed ARX encryption	-0.0022	0.0398	-0.0058
Proposed MRX encryption	-0.0023	-0.0250	-0.0138
Usman et al. (2017)	0.0012	0.0012	0.0022

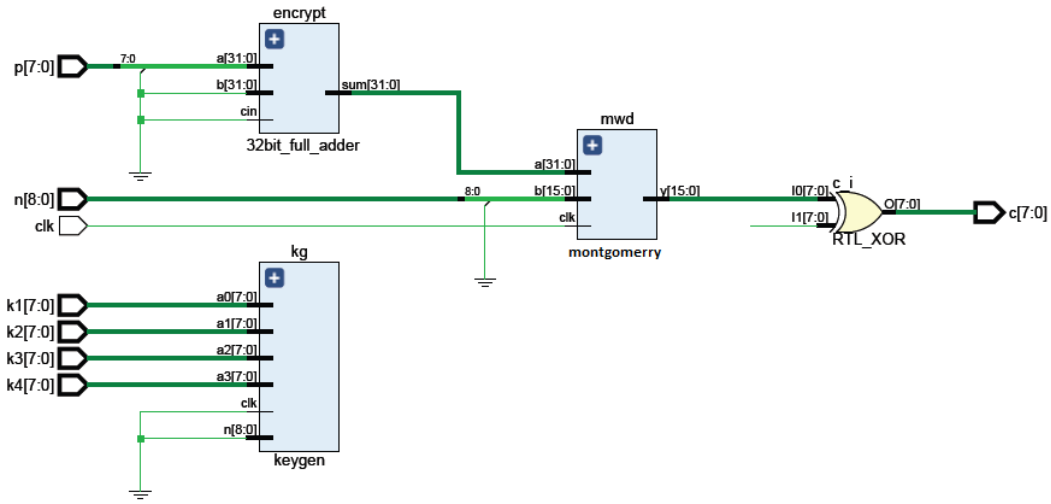

Fig. 10. RTL schematic of simple encryption scheme using ARX operations

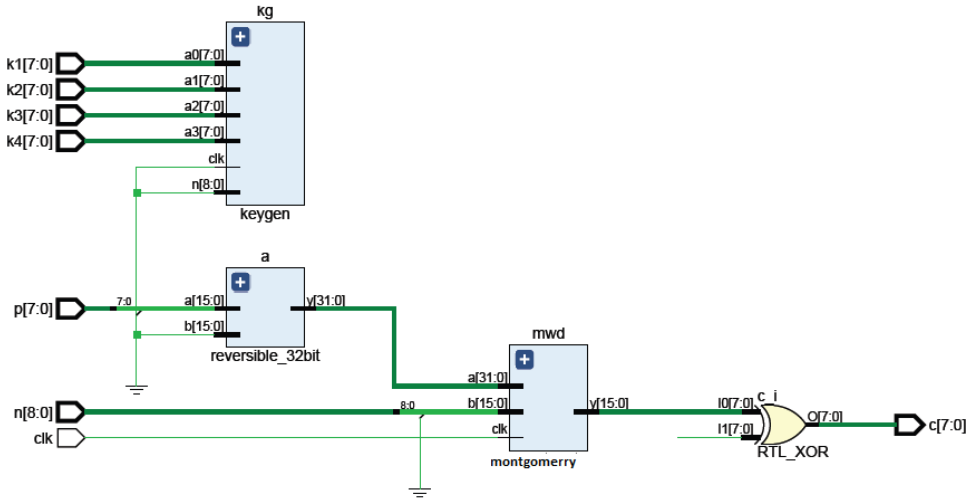

Fig. 11. RTL schematic of simple encryption scheme using MRX operations

Table 3. Implementation results of the proposed simple encryption scheme

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	(Acholli and Ningappa, 2019)HCA	37/204000	18/408000	16/51000	--	121.4
Spartan-3E-1600E	(Priyanka et al., 2016) AES	2255/29504	6/250	1661/14752	0.441	--
Spartan 6	(Satheesh and Kiran, 2019) ARX	604/9312	204/232	346/4656	--	--
Spartan 6	(Sruthi et al., 2016)HIGHT	2689/27288	1/296	2409/54576	0.607	8.34

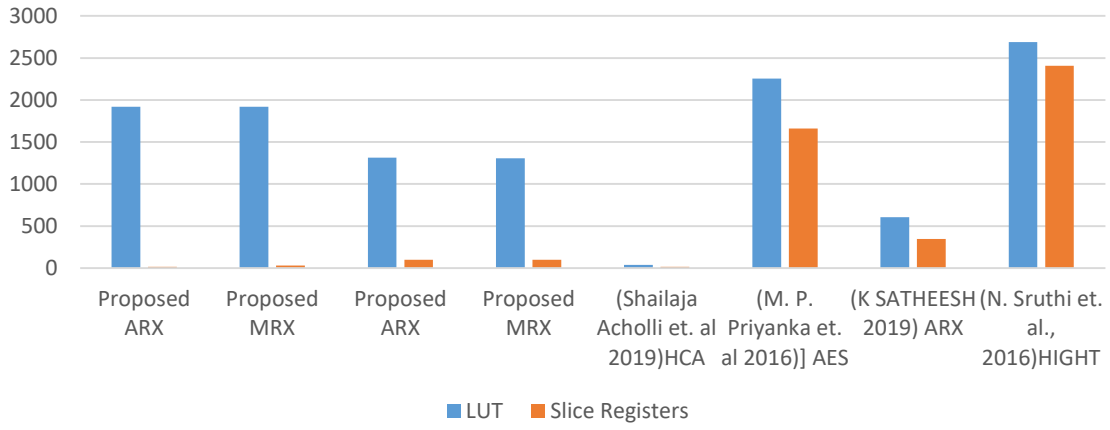

Fig. 12. LUT slice registers comparison chart for various ciphers

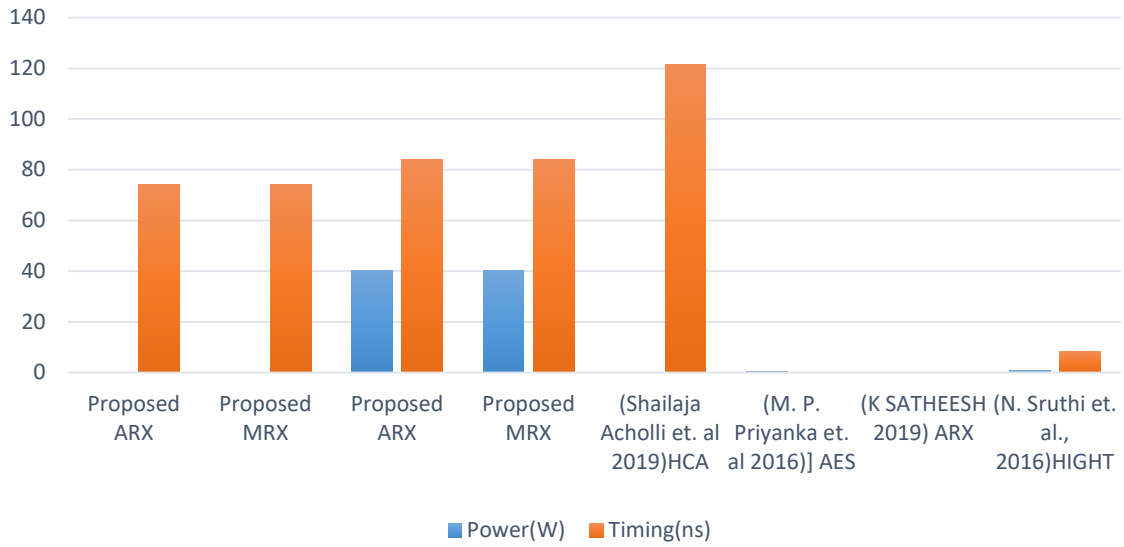

Fig. 13. Power and timing comparison chart for various ciphers

It is observed that the proposed simple encryption scheme ARX/MRX implemented using reversible logic and Vedic mathematics have better performance parameters. Fig. 12 and Fig. 13 shows the comparison charts of the various ciphers for different performance parameters.

5. CONCLUSION

In this paper an image encryption using the proposed simple encryption scheme implementing ARX/MRX operations is performed. The binary conversion of the plaintext image was made by using MATLAB version 2018a and stored in a text file and the evaluation parameters like histogram, entropy and correlation coefficients were evaluated and analysed, from the results it is found that the proposed encryption schemes achieved security. The stored text file was given as input to Verilog and performance parameters like area, power and timing reports are obtained by using Xilinx-Vivado Tool resulted in better performance compared to the state of the art implementations.

The future work, different other adders and multipliers or modular algorithms can be implemented using different logic styles and also with new for key-generation techniques can be applied to improve security, performance and efficiency.

ACKNOWLEDGMENT

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.

REFERENCES

Abed, S., Jaffal, R., Mohd, B.J., Alshayeji, M. 2019. FPGA modeling and optimization of a SIMON lightweight block cipher. Sensors, 19, 913.

Acholli, S., Ningappa, K.G. 2019. VLSI implementation of hybrid cryptography algorithm using LFSR key. International Journal of Intelligent Engineering and Systems, 12, 10–19.

Bache, F., Schneider, T., Moradi, A., Giineysu, T. 2017. SPARX-A side-channel protected processor for ARX-based cryptography, Design, Automation & Test in Europe Conference & Exhibition (DATE), Lausanne, 990–995.

Dehnavi, S.M., Rishakani, A.M., Shamsabad, M.M., Maimani, H., Pasha, E. 2016. Cryptographic properties of addition modulo 2n. IACR Cryptology ePrint Archive 181.

Devi, D.I., Chithra, S., Sethumadhavan, M. 2019. Hardware random number generator using FPGA, Journal of Cyber Security and Mobility, 8, 409–418. doi: https://doi.org/10.13052/jcsm2245-1439.841

Justin, R., Mathew, B.K., Abe, S. 2016. 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, 1155–1162.

Katagi, M., Moriai, S. 2008. Lightweight cryptography for the internet of things; Sony corporation, 7–10. http://dx.doi.org/10.1016/j.istr.2012.10.005

Khanam, R., Rahman, A., Pushpam, May, 2017. Review on reversible logic circuits and its application, 2017 International Conference on Computing, Communication and Automation (ICCCA2017), 5–6.

Kumar, V.G.K., Rai, S.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, 755. Springer, Singapore. https://doi.org/10.1007/978-981-13-1951-8_59.

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

McKay, K.A., Bassham, M., Turan, M.S., Mouha, N. 2016. DRAFT NISTIR 8114 report on lightweight cryptography, National Institute of Standards and Technology Internal Report 8114.

Mohd, B.J., Hayajneh, T., Vasilakos, A.V. 2015. A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues. Journal of Network Computer Applications. 58, 73–93.

Patel, S.T. Mistry, N.H. 2015. A survey: lightweight cryptography in WSN, in International Conference on Communication Networks (ICCN). IEEE.

Poojari, A., Nagesh, H.R., Kumar, K.V.G., Rai, S.C. 2020. A novel key scheduling algorithm for lightweight cryptographic applications, International Journal of Advanced Trends in Computer Science and Engineering, 9, https://doi.org/10.30534/ijatcse/2020/96912020

Priyanka, M.P., Prasad, E.L., Reddy, A.R. 2016. FPGA implementation of image encryption and decryption using AES 128-bit core, 2016 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, 1–5.

Rana, S., Hossain, S., Shoun, H.I., Kashem, M.A. 2018. An effective lightweight cryptographic algorithm to secure resource-constrained devices, International Journal of Advanced Computer Science and Applications (IJACSA) 9. http://dx.doi.org/10.14569/IJACSA.2018.091137

Sruthi, N., Nandakumar, R., Rajkumar, P. 2016. Design and characterization of HIGHT cryptocore, 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, 205-209. https://doi.org/10.1109/SCOPES.2016.7955798.

Teh, J.S., Teng, W., Samsudin, A., Chen, J. 2020. A post-processing method for true random number generators

based on hyperchaos with applications in audio-based generators. Front. Comput. Sci. 14, 146405. https://doi.org/10.1007/s11704-019-9120-2.

Usman, M., Ahmed, I., Aslam, M.I., Khan, S., Shah, U.A. 2017. SIT: A lightweight encryption algorithm for secure internet of things. International Journal of Advanced Computer Science and Applications(ijacsa), 8, http://dx.doi.org/10.14569/IJACSA.2017.080151.

Vergos, H.T., Efstathiou, C., Nikolos, D. 2002. Diminished-one modulo 2n + 1 adder design, IEEE Transactions on Computers, 51, 1389–1399.

Vishwakarma, P.P., Tripathy, A.K., Vemuru, S. 2020. Designing a cryptosystem for data at rest encryption in mobile payments. International Journal of Applied Science and Engineering, 17, 373–382. https://doi.org/10.6703/IJASE.202012_17(4).373

Wang, Z., Jullien, G.A., Miller, W.C. An algorithm for multiplication modulo (2^N-1), ASILOMAR '95 Proceedings of the 29th Asilomar Conference on Signals, Systems and Computers (2-Volume Set), 956.

Weber, M., Boban, M. 2016. Security challenges of the internet of things, 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 638–643, doi: 10.1109/MIPRO.2016.7522219.

Zhang, X. 2011. Reversible data hiding in encrypted image, in IEEE Signal Processing Letters, 18, 255–258, doi: 10.1109/LSP.2011.2114651.

Zhang, X. 2012. Separable reversible data hiding in encrypted image, in IEEE Transactions on Information Forensics and Security, 7, 826–832, doi: 10.1109/TIFS.2011.2176120.

Zimmermann, R. Apr. 1999. Efficient VLSI implementation of modulo 2 n ±1 addition and multiplication, Proc. 14th IEEE Symposium on Computer Arithmetic, 158–167.