Novel and Efficient Cellular Automata based Symmetric Key Encryption Algorithm for Wireless Sensor Networks

K.J. Jegadish Kumar
Assistant Professor
SSN College of Engineering
Kalavakkam-603110 Chennai,

K. Chenna Kesava Reddy
Principal
Jyothismathi College of
Engineering and Technology
Shamirpet-500078, India

S. Salivahanan
Principal
SSN College of Engineering
Kalavakkam-603110
Chennai, India

ABSTRACT

Our proposed method L2D-CASKE, the lightweight 2-dimensional (2-D) Cellular Automata (CA) based symmetric key encryption algorithm is a 128 bit length that allows a key length of 128 bits. It is designed as a lightweight encryption algorithm. As being light weight, it can be easily applied on small devices such as wireless sensor motes, smart cards and other PDAs. An encryption algorithm having simple loop operations, based on iterative parameters is used in this work. A number of sequences and operations are used for this purpose. In this paper, the well designed algorithm was verified using MATLAB 7.7 (R2008) tool and the hardware synthesized on Field Programmable Gate Arrays (FPGAs). To achieve this, our algorithm uses loop design based on iterations and the same is realized on FPGA. Advantages of our algorithm can be listed in terms of its flexibility for a set of given constraints, less complexity involved in its performance. This is achieved through the usage of generic VHDL coding. Virtex-4 XC4VL25 -10ff668 is used as a test device in Xilinx 9.1 for realizing our algorithm.

Keywords: Cellular automata, Symmetric Encryption, FPGAs, VHDL.

1. INTRODUCTION

New era of computing and communication technologies has initiated handling of different types of handy devices that are widely used for portable applications. The portability and affordability factors of these devices are accomplished by imposing limitation on its storage, power backup, and computational requirements. Regardless of the type of device used, secured communication is a major requirement for every consumer using that device. Though many techniques exist in literature for information security, this paper considered the cryptographic method based on symmetric key. Implementation of the conventional algorithms requires lot of computational time requirement, memory, area and power. The block ciphers realized on the smart card technology [1] experiences side channel attacks that depend on time and power. The procedures required to protect against such attacks are costly and complex.

1.1 Cellular Automata

Cellular Automata (CA) is an organized lattice of cells and each cell have finite number of states, such as "TRUE" (T) or "FALSE" (F). The lattice dimensions can be of any finite value. Each cell within a collection of cells is called as hood. It is characterized relatively with respect to a particular cell. To start with at time t=0, a state is assigned to the cells. The new states of the cell depend on its own previous state and states of its neighborhood. The new states are assigned based on some predefined rule using mathematical calculations. In the next time stamp, the cell is considered to be in state “T” when the neighborhood cells are also in state “T”. If this condition is not satisfied then the cell is in “F” state. Each cell in the lattice is simultaneously updated in this way with the prescribed rule sets and mathematical functions [2].

Encryption, by theory requires highly complex actions such as permuting, flipping and altering data in such a way that it is undecipherable and provides complex relationship with the original text and the keys. This relationship should be non-linear so that decryption process is as tough as possible. The encryption process must be faster in time and cheaper in terms of the components involved [2]. CA provides a basic structure for highly parallel and complex operations upon which a basic encryption scheme can be built. CA based processor can be used to compute and alter data with high degree of linearity and complexity [3].

The message encryption is done by Pseudo Random Number Generators (PRNGs) using CA. The generation of new states in One-Dimensional (1-D) CA, can be considered as a sequence of random numbers [3]. Different security schemes have been proposed including symmetric key, hash functions and public key cryptography as observed by Sarkar, 2000. Further as stated by Wolfram [4, 5], Rule 30 promotes the use of large integers in the pseudo random number generation. Owing to this interesting chaotic property of the peculiar CA, Wolfram states that, this kind of CA is used as random number generator.

The structure of the paper is as follows: Section 2 describes studies of various symmetric key algorithms and its performances, Section 3 depicts our proposed algorithm in detail and Section 4 illustrates Matlab implementation and its result analysis to test the functionality of our algorithm. Then basic cryptanalysis tests and its results were shown to prove our security level. Subsequently, hardware implementations of our algorithm are presented. Finally, conclusion is expressed in Section 5.

2. RELATED WORK

Though AES [6] is a powerful algorithm in term of enhanced security, it is not suitable for portable devices. This is due to the large code size, memory space transmission delay and computational complexity. Conversely, block cipher ICEBERG [7] is designed specifically for handy devices. The proposed implementation is based on the specific consideration of reconfigurable hardware. Alternatively, the execution in software is inappropriate, because of its huge requirements in memory storage and speed of computation.

Speedy ciphers like DDP, abbreviation of Data Dependent Permutation are competent for high speed networks. To avoid the feeble keys recognized in the earlier DDP- based ciphering techniques, Switchable procedures are proposed in Cobra-H64 and Cobra- H128 [8]. Moreover, though these ciphers are suitable for high speed hardware architectures, they are identified to be area inefficient. The realization of modulo-232 arithmetic operations in software were described in Cobra-S128 [9]. Although DDP-based cipher is found to be efficient in the implementation of both hardware and software, they fail to suit in portable devices in terms of speed, area and power. Thus the use of optimum methods like parallel processing enhances the computational speed. They are also light weighted with respect to faster computation and memory space requirements.

A number of researchers made studies in the inherent property of parallel processing of CA for devising faster cipher routines[10,11]. Security is not guaranteed by the proposed method of Wolfram [11] without a larger key size. With the initial value, the repetitive application of the similar rule, can easily determine the secret key [12]. The method proposed by Blackburn et.al. outperformed the algorithm of Nandi et.al. owing to the affine property of CA [13-15]. Sen et.al (2002) proposed a CA cryptosystem (CAC) which combines both affine and non-affine transformations in order to achieve the non-linearity [16]. Seredynski et.al. (2004) proposed a reversible CA (RCA) cipher. This cipher is found to be suitable to address the criteria involved in avalanche property. However, excessive communication requirements act as a trade off problem for this method.

2.1 LCASE Algorithm Review

An one-Dimensional cellular automata - encryption algorithm called as Lightweight Cellular Automata-based Symmetric-key Encryption (LCASE) is described by Tripathy and Nandi (2009). They have used Rule 30 for encrypting plain-text with key. The fundamental design objective of LCASE is to effectively optimize the performance requirements of both software and hardware. However, the algorithm also has to address the conventional security constraints. Different issues considered for designing the proposed algorithm are:

- Defiant to attacks such as conventional cryptanalysis and timing analysis attacks.
- Simple and cost effective implementation in hardware
- Compatible to resource constrained devices
- High-speed and minimum code density on a wide range of platforms.

The implementation of cipher in either of hardware or software has to satisfy two important design considerations [2]. The algorithm should be fast and easy to implement. In order to address these constraints parallel processing methods are used. CA elements are found to be effective in the design process. Number of loops used in LCASE consists of two rounds namely, 1. Last-half (LH) 2. First-half (FH). The term ‘r’ refers to a complete round and ‘r+1’ round refers to FH. The selection of ‘r’ value is shown in the table 2.1.

Table 2.1 Selection of ‘r’ value

Bit-Keys	128	192	156
‘r’ value	12	14	16

This encryption is a reversible two way process for decryption. Each encryption round uses 1-D (3-neighbourhood) 32- bit periodic boundary CA. The basic element of a round uses Reversible Cellular Automata (RCA), Bit Permutation (BP), Non-autonomous Cellular Automata (NCA) and Reverse Substitution (RS). The rules used in LCASE for different operations are given in Table 2.2.

Table 2.2 Selection of rule for different operators

Operator	Rule
RCA (Second Order)	i. 30 ii. Skewed 30 iii. Skewed 45
NCA	218

In order to effectively obtain the non linearity property, suitable rules from the above table have to be chosen for the selected operators. In this cipher, RCA and NCA are the preferred operators over the conventional XOR, as the later is not suitable for non linear operations.

3. PROPOSED ALGORITHM

Our entire L2D-CASKE round comprises of simple XOR, Block Permutation, 1D RCA Key Generation, 2D RCA Margolus Neighborhood cell. The ‘r’ values are chosen from the table 2.1 and rule for the operator RCA used in our algorithm is obtained from Table 2.2. The RCA operator is used for 1-D CA based key generation.

.

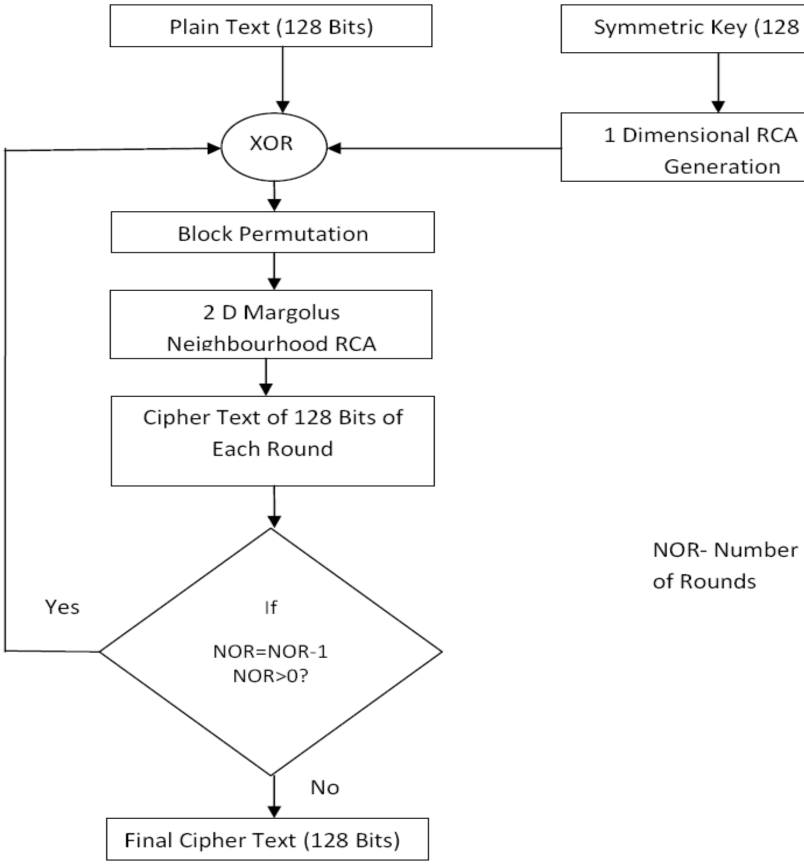

Plain Text (128 Bits)
Symmetric Key (128 Bits)
XOR
1 Dimensional RCA Key Generation
Block Permutation
2 D Margolus Neighbourhood RCA
Cipher Text of 128 Bits of Each Round
NOR- Number of Rounds
Yes
If
NOR=NOR-1
NOR>0?
No
Final Cipher Text (128 Bits)

Figure 1: Proposed Encryption Scheme

3.1 1D RCA Key Generation

RCA is the one in which the preceding pattern can be recovered from the given current pattern(s). Our proposed algorithm utilizes RCA (second order) [17]. The ‘i^{th}’ state of n bit pattern is determined by the clock cycles as shown in Table 3.1

Table 3.1 State determination

Clock cycles	‘i^{th}’ state of n bit pattern
t + 1	States of neighborhood pattern at ‘t’ & self pattern at (t − 1)
t − 1	patterns at t & (t + 1) clock cycle

Given below is an example of a 3-neighborhood second order RCA [2]:

$$x_i\,(t+1) \;=f\,(x_{i-1}\,(t),\, x_i\,(t),\, x_{i+1}\,(t)) \oplus x_i\,(t-1);$$

$$x_i\,(t-1) \;=f\,(x_{i-1}\,(t),\, x_i\,(t),\, x_{i+1}\,(t)) \oplus x_i\,(t+1).$$

Here, the states $x_i\,(t+1)$ and $x_i\,(t-1)$ are denoted respectively by the terms ξ_i and y_i . ξ is obtained based on two initial patterns of (Y, X) at time steps (t − 1) and t. Then, using two successive patterns (ξ, X), the initial pattern Y can be figured out. This operation is denoted as follows [2].

$$\xi \;=\; \text{RCA}\,(Y, X);\; Y \;=\; \text{RCA}\,(\xi, X).$$

Elementary CA rule 30 based second order periodic boundary 4-cell RCA Logic diagram is depicted in Figure 2.
The evolved pattern of such an RCA can be evaluated as [2]

$$x_i\,(t+1) = (x_{i-1}\,(t) \oplus (x_i\,(t) \vee x_{i+1}\,(t)) \oplus x_i\,(t-1)).$$

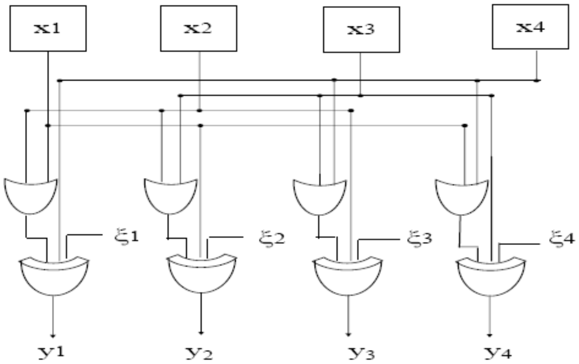


Figure 2: 4-cell Periodic Boundary RCA Logic Cell [2]

One can refer Tripathy and Nandi for further details about RCA logic configurations [2]. As an example states for the resultant configuration Y = RCA (X, K) using the rule 30 can be determined by the above equations.

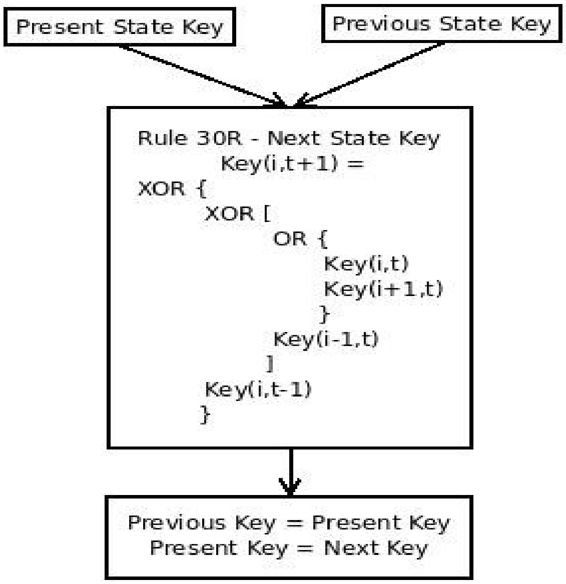


Figure 3: A Simple Key Generation Algorithm

3.2 Block Permutation

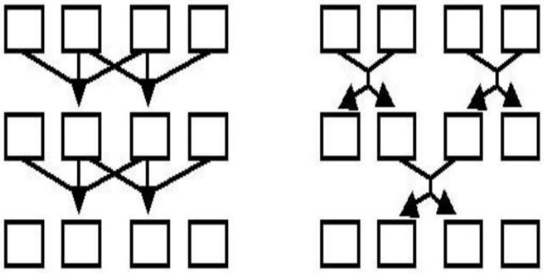
The Block Permutation (BP) operation permutes each 2x2 sub-block within a 4x4 block diagonally. The idea behind this permutation is to move the data at the boundaries at the centre of the block and vice versa. This process results in enhanced diffusion rate [18]. So it is difficult to perform differential cryptanalysis. Further, through simple means of wire crossings, one can achieve hard-wiring which makes the implementation of block permutation as an easy process. The block permutation operation is described in Figure 4. Increasing the redundancy content of the given plain-text is the principle idea of diffusion. The usage of Block Permutation operation in our proposed technique guarantees the diffusion condition.

A1	A2	B1	B2		D1	D2	C1	C2
A3	A4	B3	B4		D3	D4	C3	C4
C1	C2	D1	D2		B1	B2	A1	A2
C3	C4	D3	D4		B3	B4	A3	A4

Figure 4: Block Permutation Operation

3.3 2D RCA: Margolus Neighborhood

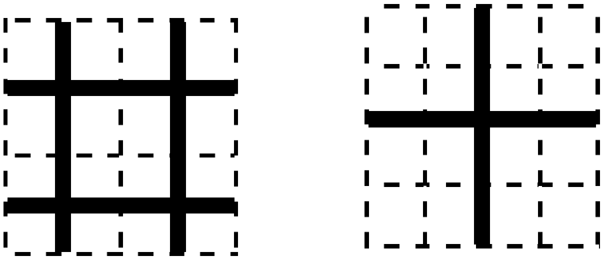
RCA in Two Dimensions: As we consider the plain-text to be of two-dimensions, we make use of 2D Reversible Cellular Automata (RCA). An approach called Partitioned Cellular Automata (PCA) is one more method to develop a reversible CA [19]. A familiar partitioning technique called Margolus neighborhood, is selected for its inherent invertible property. Need for this arises from the fact that conventional CAs are not reversible and difficult to perform the inversion process. As shown in Figure 3, the traditional Rule 30 evaluates the four likely patterns that stay alive for either state of the present cell. Evidently, there is a difficulty in determining the previous state of a cell with that of its present state. The major reason is that there is likelihood of losing information while evaluation of the states of previous cell. Figure 5(a) illustrates traditional 1-D CA [19] which consists of three inputs, one of them being the cell itself and other two being its neighbors. However the output is simply one cell. Comprising of same number of states, it is very uncertain that the one output cell can protect the entire useful content of the three preceding input cells. This proves that the state of the next three step cells depends on each state of the previous cells, thus distributing the information of the previous cells and thereby conserving the entire information allocated to the next state of the cells. However, the likelihood of this occurrence is not very high. Hence, we chose the Margolus neighborhood described in Figure 5(b).



(a) Conventional CA (b) Margolus Neighbourhood

Figure 5: Conventional CA Vs Margolus Neighbourhood [19]

It utilizes two cells of a block as inputs and gives out all the two cells of the same block as output. This process guarantees perpetuation of information. At some point, the exact use of same partitioning scheme repetitively leads to incapability of information to propagate through every partition. This demerit in Margolus neighborhood is overcome by alternating groups of odd and even cells every time stamp. Margolus neighborhoods are small enough devising totalistic rules renders little benefit. On the other hand, Margolus was interested in reversible CA [19-21], for which rules encompassing shifting, reflection, creation, annihilation and complementation abound in such small neighborhoods; the Billiard-Ball Machine was a result. In general, partitioning schemes defined in book Cellular Automata Machines (CAM) by Norman Margolus called it as Margolus neighborhood. This simple partitioning scheme contains a grid of cells split into set of four cells, as shown in Figure 6. A cellular automata rule is applied locally to these grids of cells. The repeated use of precisely identical partitioning method would cause the information unable to proliferate ahead of the restrictions of any individual partitions but the activeness on the whole likely to be unaffected. The functionality of the partitioning scheme is through grids that takes different spatial co-ordinates on every alternating time steps. The partitioning scheme for the alternating time steps is shown in Figures 6 (a) and (b). In this Margolus neighborhood partitioning scheme, the CA is partitioned as displaced boxes containing 2 cells on the sides of each boxes. Then the partitioning alters from one clock cycle to the next cycle so that element of one box is associated as an another element of an adjoining box on alternate clock cycles. In Figure 6(a), the box margins are drawn to represent the grouping of the even cells and the odd cells as in Figure 6(b), which signifies on the every other half of the cycle, the states of cells remains fixed and when the margin lines swings by one square in the crosswise direction such that the odd and even cells share boxes [20]. In Margolus neighborhood the contents of boxes are updated at once wholly as compared to the traditional 1-D CA neighborhood. The rule applied to all the four cell to alter the states based D Rules, generally called as Billiard Ball Machine rules [21].



(a) Odd Steps (b) Even Steps

Figure 6: Margolus Neighbourhood Partitioning [20]

In the Margolus neighbourhood, the entire matrix is divided into non-overlapping sub- blocks, each of 2x2 cells (bits), implying $2^{(2x2)}$=16 possible configurations for every sub-block. These 16 configurations are assigned with numbers ranging from 0 to 15 as shown in Figure 7.

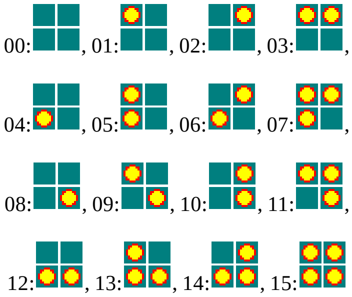


Figure 7: Sub-Block Configurations [21]

4. IMPLEMENTATION OF ALGORITHM AND RESULT ANALYSIS

Our objective of designing and implementing a 2D CA-based symmetric-key encryption algorithm has been fulfilled. MATLAB was used as the development and testing platform. The necessary criteria for a good cipher is that the relationship between the plain-text, the key and the cipher-text should be as involved and complex as possible, making it almost impossible to decipher the plain-text from the cipher-text without the exact key. The cipher-text on deciphering with the exact key should yield the plain-text without any error. Various tests were performed on our proposed algorithm to ensure that both the above criteria are satisfied. Our proposed algorithm was implemented straight forwardly by generic VHDL in Xilinx 9.1 tool to take advantage of the parallelism inherent in our design. MATLAB is only a sequential programming platform whereas VHDL facilitates simultaneous parallel operations. The HDL modules were also synthesized and the equivalent hardware circuitry of our algorithm was extracted. Our well-organized implementation accomplished efficient space, energy and speed constraints by realizing on the SPARTAN-3, xc3s1400.

4.1 Results in MATLAB

As discussed above, the preliminary testing was done using MATLAB. Plain texts of size 128-bits and key length of 128 bits were chosen and encrypted using the proposed cipher. The cipher text, the decrypted data and the difference between the plain text and the cipher-text were obtained corresponding to each plain-text. Also, for grayscale and colour images, Cipher Block Chaining (CBC) was used, where the plain texts are combined with the cipher text of the previous plain text before applying the cipher. Some of the results for plain-texts being a bit image, a grayscale image and a colour image followed in Figure 8.

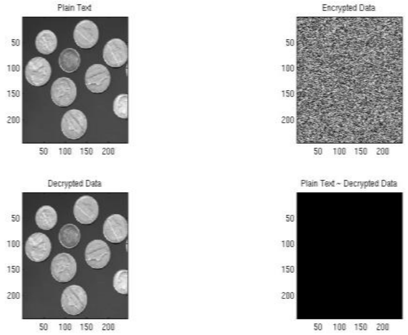


Figure 8: Encryption and Decryption of Grayscale Image

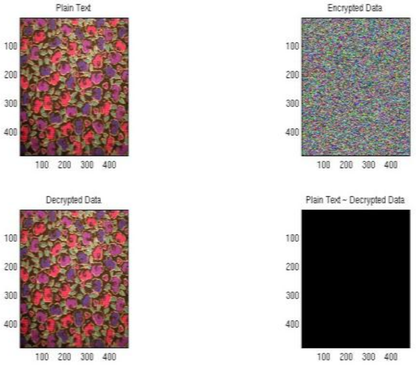


Figure 9: Encryption and Decryption of Color Image

4.2 Basic Cryptanalysis Results

With the aim of robustness assessment, rules are subjected to various tests. One of the attractive features of CAs is, non feasibility to judge each of its individual rule on a whole. The entire idea is that each element jointly leads to some momentous behavior and yet cannot recognize what each individual element is supposed to behave. Performance of all individual elements is too narrowly interwoven with states of the cells in its neighborhood. Achieving robustness to a rule is based on executing the complete CA on "numerous" problems, and each has its own known solution. And also necessary to verify that the given cell attains the correct "restraining" state every time. A good estimation of the fitness and robustness of a cell is evaluated as the ratio of number of exact final states to the total number of tests. Both "numerous" and "restraining" have to be given finite values that are adequate and realistic so as to end the simulation in a reasonable time. Some preliminary cryptanalysis was performed in MATLAB and its results are as follows.

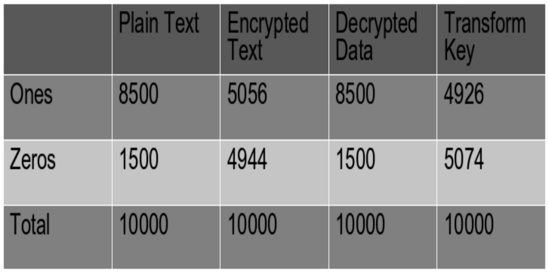
4.2.1 Basic Statistical Results

The results of some basic statistical analysis such as the distribution of ones and zeros, variation of ones in cipher-text with respect to the plain-text, etc. are summarized below.

4.2.1.1 Variation of Ones and Zeros - Balance Property

The number of ones and zeros (for randomly generated key) in the various images showed in Figure 8 are tabulated in Table 4.1

Table 4.1 Basic Statistical Results



	Plain Text	Encrypted Text	Decrypted Data	Transform Key
Ones	8500	5056	8500	4926
Zeros	1500	4944	1500	5074
Total	10000	10000	10000	10000

4.2.1.2 Variation of Ones

The following results, as shown in Figure 10 were obtained by variants in the number of ones in the plain-text between 0 and 128. It depicts the relationship between the bit distributions of the plain-text and the cipher-text for a 128-bit data. The graph shows minimal variations in the characteristics of the cipher-text throughout the large variations in the characteristics of the corresponding plain-text. This is a proof of the fact that the statistical properties of the cipher-text are pseudo-random and independent of the statistical properties of the plain-text.

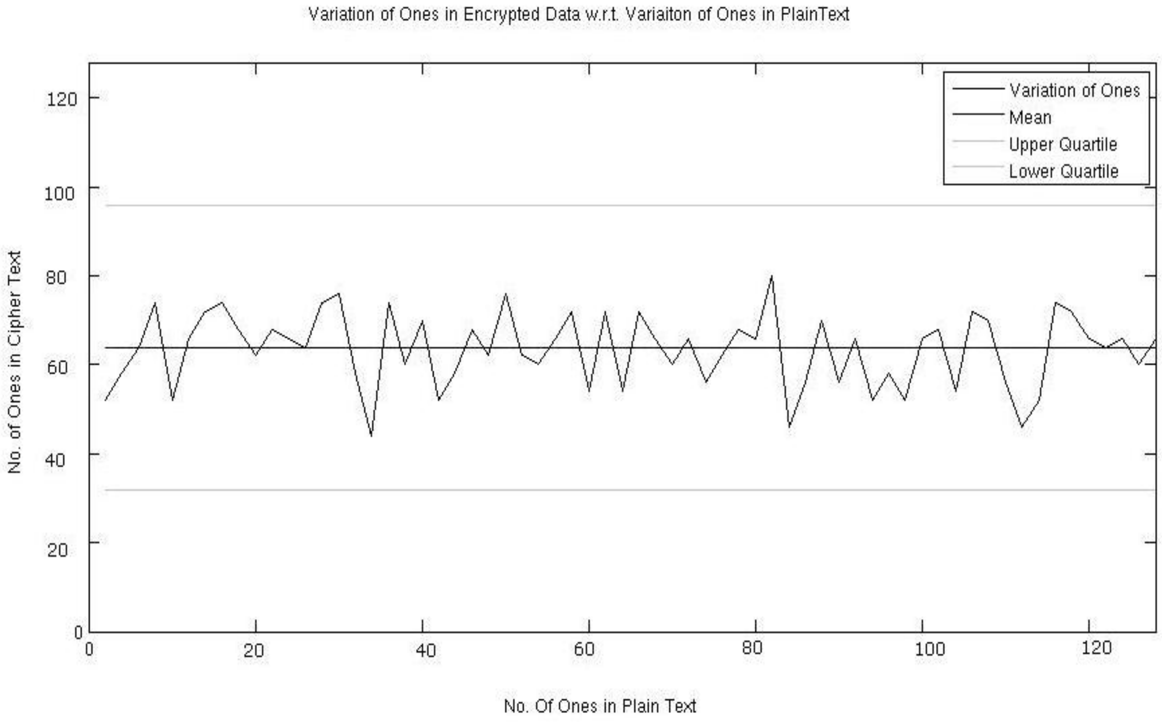


Figure 10: Variations of Ones in Encrypted Data

4.3 Hardware implementation

Our iterative loop architecture for L2D-CASKE consists of round function and key schedule. Resource consuming blocks are the 1D RCA key generation module, XOR, and 2D Margolus Neighborhood module; the block permutations are implemented by swapping the wires. FPGA based computation engines emerge out to be very much suitable for CA algorithms, which consists of a uniform structure composed of many Finite State Machines (FSMs), thus, harmonizing the innate design layout of the FPGA hardware. It is acclaimed that the entire CA pattern easily fits into the FPGA space and the resulting computing structure is simple. For CA computation, the whole pattern is embedded on to FPGA from the host machine and is run for the required number of generations. Finally, the resulting CA generation is uploaded back to the host system. The LUT count for various block ciphers are provided in the Table 4.2. Our proposed cipher uses 128-bit plain-text block as the other ciphers use the standard 128-bit plain-text block. Our efficient implementation achieves lower area expressed in terms of LUT and considerably higher throughputs on the target platform VIRTEX-4, XC4VL25-10ff668. Finally, power analysis are made on Xpower analyzer tool in Xilinx 9.1.

The implementation results prove that our security processor occupying 642 slices in terms of area, 169.22mW of total power consumption and throughputs of 2136 Mbits/sec at the maximum work frequency of 334 MHz is compatible to any Main processor of a sensor Node in Wireless sensor networks. Thus we have enhanced the performance of a sensor node by achieving maximum level security and efficiency regarding Area, Power and Throughputs.

Table 4.2 Comparison of Hardware Complexity in terms of 4-Input LUT Count and Power consumption

Block Cipher	Hardware Complexity in terms of LUTCounts , Throughput Mbits/sec	Total Power Consumption
LCASE[1]	582, NA	NA
ICEBERG[7]	704, NA	NA
AES[6]	3376,1563 Mbits/secs	NA
L2D-CASKE	642, 2136 Mbits/secs	169.22 mW

5. CONCLUSION

A two dimensional Cellular Automata based light-weight symmetric-key cryptosystem for hardware cost effective applications is proposed in this work. This encryption scheme meets all the basic requirements meted out by the AES. Also, this being light-weight, it can be easily applied on small devices such as sensor motes, smart cards, etc. The basic automata can be down scaled to work even on a basic 8-bit processor. The major advantage of our proposed scheme is the use of dynamic key schedule which minimizes the memory requirement on the smaller devices.

The scope for extension of this paper includes the use of cryptanalysis attacks such as differential cryptanalysis attacks, linear cryptanalysis attack and its variants, algebraic attacks, timing analysis attacks, etc., on our proposed algorithm. Apart from the above, field optimization can also be done in terms of better schedules such as non-autonomous cellular automata to mix the key and plain-text. On the whole our proposed cellular automata based block cipher can be used effectively to construct a highly effective encryption scheme for resource constrained devices like, Wireless Sensor Networks, RFIDs and other low powered portable devices.

6. REFERENCES

[1] P. Kocher, J.Jaffre and B.Jun, “Differential power analysis,” Crypto’99, LNCS 1666, pp. 398-412, Springer-Verlag, 1999.

[2] S Tripathy and S Nandi,"LCASE: Lightweight Cellular Automata-based Symmetric-key Encryption”, International Journal of Network Security, Vol.8, No.2 , Mar. 2009.

[3] Palash Sarkar, “A Brief History of Cellular automata”, Journal of ACM Computing Surveys (CSUR), Volume 32 Issue 1, March 2000.

[4] S. Wolfram, “Cryptography with Cellular Automata,” Crypto ’85, LNCS 218, pp. 429- 432, Springer-Verlag, 1986.

[5] S. Wolfram, “Random sequence generation by cellular automata,” Advances in Applied Maths, vol. 7, No. 2, pp. 123-169, 1986.

[6] J. Daemen, and V. Rijmen, Specification for the Advanced Encryption Standard (AES), Springer-Verlag, 2002.

[7] F. Standaert, G. Piret, G. Rouvroy, J. Quisquater, and J. Legat, “ICEBERG : An involutional cipher efficient for block encryption in reconfigurable hard- ware,” FSE ’04, LNCS 3017, pp. 279-299, Springer- Verlag, 2004.

[8] N. Sklavos, N. A. Moldovyan, and O. Koufopavlou, “High speed networking: Design and implementation of two new DDP-based ciphers,” Mobile Networks and Applications-MONET, Vol. 25, No. 1-2, pp. 219-231, Springer-Verlag, 2005.

[9] N. A. Moldovyan, P. A. Moldovyan, and D.H. Summerville, “On software implementation of fast DDP-based ciphers,” Internatiol Journal of Network Security, Vol. 4, No. 1, pp. 81-89, 2007.

[10] S. Nandi, B. K. Kar, and P. P. Chaudhuri, “Theory and application of cellular automata in cryptography,” IEEE Transaction on Computers, Vol. 43, No.12, pp. 1346-1357, 1994.

[11] S. Wolfram “Cryptography with Cellular Automata,” Crypto ’85, LNCS 218,pp.429-432,Springer-Verlag, 1986.

[12] C.K.Koc, and A.M.Apohan, “Inversion of cellular automata iteration,” IEE Proceedings of Computer and Digital Technique, Vol. 144, No. 5, pp. 279-284,1997.

[13] S. Nandi, B. K. Kar, and P. P. Chaudhuri, “Theory and application of cellular automata in cryptogra- phy,” IEEE Transaction on Computers, vol. 43, no.12, pp. 1346-1357, 1994.

[14] S. Blackburn, S. Murphy and K. Paterson, “Comments on theory and application of cellular automata in cryptography,” IEEE Transactions on Computers, Vol. 46, No. 5, pp. 637-638, 1997.

[15] F. Seredynski, K. Pienkosz, and P. Bouvry, “Reversible cellular automata based encryption,” NPC’04, LNCS 3222, pp. 411-418, Springer-Verlag, 2004.

[16] S.Sen, C.Shaw, D.R.Chowdhuri, N. Ganguly, and P. Pal Chaudhuri, “Cellular automata based cryptosystem

(CAC)”, ACRI-2002, LNCS 2513, pp. 303-314, Springer-Verlag, 2002.

[17] T. Toffoli and N. Margolus, “Invertible cellular automata: A review,” Physica D, vol. 45, pp. 229-253, (reprinted with correction as of Oct. 2001).

[18] Jérôme Durand-Lose “Representing Reversible Cellular Automata with Reversible Block Cellular Automata”, Discrete Mathematics and Theoretical Computer Science Proceedings AA (DM-CCG), 2001, pp 145–154.

[19] "Cellular Automata Rules Lexicon - Margolus Neighborhood." Free Software of mirek Wojtowicz. *http://www.mirekw.com/ca/rullex_marg.html*.

[20] “2D Reversible Cellular Automata" Lotus ArtificialLife.*http://www.alife.co.uk/ca/bbm/2d/*.

[21] Joaquin Cerda, Rafael Gadea, and Guillermo Paya, “Implementing a Margolus Neighborhood Cellular automat on FPGA”, IWANN 2003, LNCS 2687, pp. 121-128, Springer-verlag Berlin Heidelberg, 2003.