Circuits and Systems, 2016, 7, 222-230
Published Online April 2016 in SciRes. http://www.scirp.org/journal/cs
http://dx.doi.org/10.4236/cs.2016.74021

Scientific Research Publishing

KAMAR: A Lightweight Feistel Block Cipher Using Cellular Automata

Jegadish Kumar Kailairajan Jeyaprakash, Joseph Gladwin Sekar*, Kamaraj Villayutham

Department of Electronics and Communication Engineering, Sri Sivasubramaniya Nadar College of Engineering, Chennai, India
Email: *josephs@ssn.edu.in

Received 4 March 2016; accepted 17 April 2016; published 20 April 2016

Abstract

Wireless Multimedia Sensor Network (WMSN) is an advancement of Wireless Sensor Network (WSN) that encapsulates WSN with multimedia information like image and video. The primary factors considered in the design and deployment of WSN are low power consumption, high speed and memory requirements. Security is indeed a major concern, in any communication system. Consequently, design of compact and high speed WMSN with cryptography algorithm for security, without compromising on sensor node performance is a challenge and this paper proposes a new lightweight symmetric key encryption algorithm based on 1 D cellular automata theory. Simulations are performed using MatLab and synthesized using Xilinx ISE. The proposed approach supports both software and hardware implementation and provides better performance compared to other existing algorithms in terms of number of slices, throughput and other hardware utilization.

Keywords

Cryptography, WMSN, Cellular Automata, Reversible Cellular Automata, KAMAR, Feistel Block Cipher, Key Scheduling Algorithm, FPGA

1. Introduction

Security is a critical factor in every communication system in this world; may it be a simple short distance communication or communication between large servers that deal with a large volume of data. One of the recent trends of WSNs is transfer of multimedia information like video, still images, audio etc., through self-organized networks. Such networks are called as Wireless Multimedia Sensor Networks (WMSNs) [1] [2]. Conventional wireless sensor nodes are limited by processor speed to handle multimedia information. Hence, the challenge is

*Corresponding author.

How to cite this paper: Jeyaprakash, J.K.K., Sekar, J.G. and Villayutham, K. (2016) KAMAR: A Lightweight Feistel Block Cipher Using Cellular Automata. *Circuits and Systems*, **7**, 222-230. http://dx.doi.org/10.4236/cs.2016.74021

to design a sensor node prototype with low power consumption and high speed processors O (GHz). Some of such high speed and low power processor hardware specifications available in literature [3] are listed below in **Table 1**.

In this paper, a new fast symmetric key encryption named as KAMAR (the name of the algorithm derived from KAmaraj and K. J. J. kuMAR) based on the cellular automata theory, suitable for WMSN applications is proposed. The objective therefore is to design and develop an encryption algorithm that occupy less slices/area, consumes minimum power and offers high throughput.

A cellular automaton is finite or infinite dimension grid of cells arranged regularly and can have finite number of states. In information theory, the cell states lie in Galois field, GF (2). For each cell, states depend on the "states of group of cells" (including the cell itself) and it is called as neighbourhood. An initial state (time $t=0$) is selected by assigning a state for each cell [4]-[7]. A new generation is created according to some mathematical fixed rule called local update rule function or simply rule function. This local update rule function is applied to the whole grid simultaneously and the new state of each cell in terms of the current state of the cell and the states of the cells in its neighbourhood is determined [8]-[10].

The simplest form of CA is one dimensional with two possible states. A cell and its two neighbours form a neighbourhood of 3 cells, so there are $2^3=8$ possible configurations for a neighbourhood. A rule is a deciding function for each configuration, for the cell to have state 1 or a 0 in the next generation. Therefore, the total number of possible rules is $2^8=256$ [11]. These 256 CAs invented by wolfram [6] are generally referred as Wolfram code, who numbered each rule from 0 to 255. The rule 30, 45 and rule 110 CAs are predominantly attractive in applications of cryptography [9] [10]. These rules are of attractive because they generate composite and random patterns. Rule 30 is universally used as random number generator in Wolfram's program Mathematic for cryptographic applications [6].

Reversible Cellular Automata (RCA) is defined as the high order CA in which the future $\left(C_x^{t+1}\right)$ states of the grid of cells $\left(C_x\right)$ are calculated using the present $\left(C_x^t\right)$ and past $\left(C_x^{t-1}\right)$ configuration of the cells. Generally, second order CA is used to construct local transition rule function. A second order local transition rule function is defined as in Equation (1)

$$C_i^{t+1}=f\left(C_x^t,C_x^{t-1}\right) \tag{1}$$

RCA rule functions RCA rule-30 and RCA rule-45 are used to construct S-Box and key scheduling algorithm for KAMAR block cipher.

In general, the RCA rule-30 function is defined as in Equations (2) and (3)

$$x_i\left(t+1\right)=\left[x_i\left(t\right)\vee x_{i+1}\left(t\right)\oplus x_{i-1}\left(t\right)\right]\oplus x_i\left(t-1\right) \tag{2}$$

$$x_i\left(t-1\right)=\left[x_i\left(t\right)\vee x_{i+1}\left(t\right)\oplus x_{i-1}\left(t\right)\right]\oplus x_i\left(t+1\right) \tag{3}$$

And the evaluated function for RCA rule-45 is given in Equations (4) and (5)

Table 1. List of MHz to GHz processors for WSN.

Processor	Processor Speed	Bits	RAM	Power Consumption (mW)
Processor	Processor Speed	Bits	RAM	Power Consumption (mW)
TIMSP430 F2419	8 MHz	16	12 KB	8
Freescale MPC8313	50 MHz	8/32	128 Kb	8
ARM-OKI ML674K	33 MHz	16/32	512 Kb	145
Freescale-MPC8313	333 MHz	32	GB External	520
IMote 2.0	400 MHz	32	32 MB	574
Intel PXA255	400 MHz	32	64 MB	620
ADVANTICYS	1.8 GHz	32	2 GB	1

$$x_i(t+1) = \left[x_i(t) \vee \overline{x}_{i+1}(t) \oplus x_{i-1}(t) \right] \oplus x_i(t-1) \quad (4)$$

$$x_i(t-1) = \left[x_i(t) \vee (\overline{x}_{i+1}(t) \oplus x_{i-1}(t) \right] \oplus x_i(t+1) \quad (5)$$

In the above equations, the states $x_i(t+1), x_i(t), x_i(t+1)$ denote next state, current state and previous state of cells respectively. The paper is organized as follows: Section 2 discusses related works; Section 3 illustrates the design specification and procedure of algorithm; Section 4 discusses the security analysis of the proposed KAMAR algorithm; Section 5 discusses the algorithm performances in hardware implementation.

2. Related Works

Seung-Jo Han *et al*. (1996) discussed Data Encryption Standard (DES) called the Improved-DES. The Improved-DES is stronger than the DES against differential cryptanalysis for cryptographic security. The authors proposed the improved DES by dividing one data block (96 bits) into 3 sub-blocks of 32 bits and then perform different f functions on each of the 3 sub-blocks, and then increase the S1-S8 of the S-boxes to S1-S16, satisfying the Strict Avalanche Criterion (SAC) and the correlation coefficient. Finally the key length is increased to 112 bits. The analysis showed that the Unicity Distance (UD) in the Improved-DES is increased than the DES's UD [12]. However, the cost of hardware implementation is extensively large. Blowfish *et al*. (2008) describes that the Blowfish cryptosystem is a very fast encryption algorithm consisting of two different modules, a sub-key & S-box generation phase, and an encryption phase. The author presents the encryption scheme and demonstrates how fast the encryption scheme is, as compared to the sub key and S-Box generation. The secrecy of the encryption routine is explained by using several test files of different types, as well as study of the security with respect to the number of rounds [13].

Ren Fang *et al.* (2009) described KASUMI is a block cipher with the Feistel network. The small and efficient hardware of the KASUMI block cipher is the core of the 3GPP confidentiality algorithm-f8, and the 3GPP integrity algorithm-f9. In designing the hardware, they focused on optimizing the implementation of FO/FI functions that are the major components of KASUMI. They proposed three methods for this optimization: using a loop-structure in the implementation to reduce the number of the FO/FI function, realizing S7 and S9-boxes in combinational logic and optimization of extended key's generation [14].

P. Israsena *et al.* (2006) proposed and efficient implementation of algorithm for persistent, ubiquitous applications employing RFID devices, low-cost and secure RFIDs tags. The ICs for such systems have stringent requirements in terms of cost related to area and power consumption, creation conventional encryption unsuitable. The author discusses the potential of employing the TEA algorithm for medium secure systems. It is found that using the implementation style wished-for, TEA based encryption hardware can be made to meet the necessities. The potential usage of low-cost secure RFID for applications such as secure device tracking is also discussed [15]. The author discusses the potential of employing the operationally-lean XTEA-based hash function core for encryption/authentication in medium secure systems. It is found that with the parallel implementation style proposed, the core to be compact and faster [16].

3. Algorithmic Description

In cryptography, a Feistel cipher is a symmetric structure used in the construction of block ciphers, it is also commonly known as a Feistel network. A large proportion of block ciphers like DES, Blowfish, Kasumi, TEA, XTEA, Camellia uses this scheme. The Feistel structure has the advantage that encryption and decryption operations are very similar, even identical in some cases, requiring only a reversal of the key schedule. Also, the interesting feature of the Feistel based block cipher is that the round functions are iterated functions and works for half of the input bits. Thus reduces the computational time of the algorithm. Therefore, the size of the code or circuitry required to implement such a cipher is nearly halved.

The proposed algorithm structure is Feistel network. The algorithm takes plaintext of 128-bit and key input of 128-bit, 192-bit and 256-bit. The fundamental nature of the feistel cipher is that a single round offers inadequate security but that multiple rounds offer increased security. Based on the statistical analysis of the proposed algorithm simulated in matlab, the number of rounds "n" is calculated using the empirical formula as given below in Equation (6).

$$n = \left\lfloor \frac{(K_l + b_l)}{b_l/8} \right\rfloor \tag{6}$$

where K_l is the key length and b_l is the block length. The number of rounds n_r for 128-bit block length and 128-bit, 192-bit, 256-bit key length are evaluated as 16, 20, and 32 respectively based on statistical analysis. The basic statistical analysis is evaluating correlation factor between the each iterative round output and its input. It is found that after certain number of iterations the correlation factor remained almost constant around 0.03 approx.

3.1. Encryption/Decryption Function

Let F be the round function and $K_0, K_1, K_2, \cdots, K_{n-1}$ be the sub-keys for the rounds $0,1,2,3,\cdots,n-1$ respectively.

The complete operation of KAMAR is described below as:

1) The plaintext block is divided into two equal halves, (L_0, R_0)

2) For each round, compute

$$L_{i+1} = K_i \oplus S_i$$

$$R_{i+1} = (\neg L_i) \oplus [K_i \oplus S_i]$$

where $\neg$: represents inversion operation

$$S_i = RCA30(H_i)$$

$$H_i = Circleftshift(R_i, 9)$$

Then the ciphertextis (R_{n+1}, L_{n+1}).

3) Decryption of a ciphertext (R_{n+1}, L_{n+1}) is accomplished in reverse direction by computing for $i = n; n-1, \cdots, 0$ and $H_i = Circrightshift(R_i, 9)$.

Then (L_0, R_0) is the plaintext again.

3.2. Single Round Function

The single round function is described in the **Figure 1**. The round function takes right half (64-bit) as an input from the 128-bit plaintext or ciphertext. This 64-bit is circular left shifted by 9 bits and then divided into 8 words of *8*-bit each.

The internal function of RCA rule-30 based S-Box is shown in **Figure 2**. The right hand side of the **Figure 1** presents the key scheduling round function and encryption/decryption algorithm at the left hand side. The 8 bit output of each S-box is the function of preloaded value and the input bits. The value $(6363636363636363)_H$ is used as the preloaded value in the RCA rule-30 S-Box function. The outputs of all the eight S-box are concatenated and XOR-ed with 64-bit sub-key K_i.

Let S-Box input set $S_{in} = \{S_{i1}, S_{i2}, \cdots, S_{i8}\}$ and the corresponding S-Box output set $S_{out} = \{S_{o1}, S_{o2}, \cdots, S_{o8}\}$ and the function RCA30: S

3.3. Key Scheduling Algorithm

The key scheduling algorithm takes 128-bit, 192-bit or 256-bit as input and generates *n* 64-bit sub-keys for each round of encryption with respect to the key length.

The complete operation of key scheduling algorithm to generate key space is described below as:

1) The initial key block is divided into two equal halves, (KL_0, KR_0)

2) For each round $i = 0,1,\cdots,n-1,$ compute

$$KL_{i+1} = RCA45(HR_i, C)$$

$$KR_{i+1} = RCA45(HL_i, C)$$

where *C* is 64-bit constant of hex value $(6363636363636363)_H$

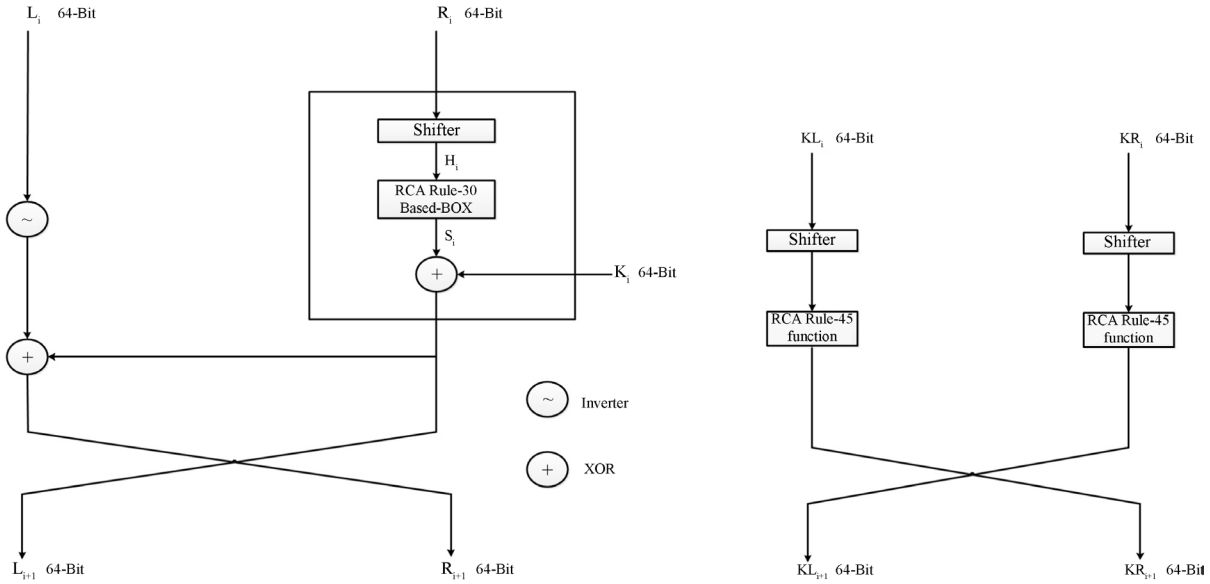

Figure 1. Encryptions/decryption round function and key scheduling of KAMAR.

$$HR_i = Circleftshift\left(KR_i, 9\right)$$

$$HL_i = Circrightshift\left(KL_i, 9\right)$$

Then, in general the key space pair generated is $\left(KR_{n+1}, KL_{n+1}\right)$.

Random sub-keys are generated using the key scheduling algorithm as shown in **Figure 2**. The initial pre-loaded HEX value $\left(6363636363636363\right)_H$ denoted as constant is initialized to the left and right half of the RCA-45 function in sub-key generation round. Here, the 128 bit input key is divided into two half namely KL and KR, where KL and KR are the left and right half of the input key. Both KL and KR is circular shifted by 9 bits before it is given as an input to the RCA-45 function. The output of the key scheduling round function generates Sub-key space pair as $\left(K_i, K_{i+1}\right)$. The complete set of key space is defined as $K = \left\{\left(K_0, K_1\right), \left(K_2, K_3\right), \cdots, \left(K_{n-2}, K_{n-1}\right)\right\}$.

4. Security Analysis

4.1. Differential Cryptanalysis

Differential cryptanalysis attack is one of the widely known attacks against the block cipher. Biham and Shamir introduced cryptanalysis against DES block cipher. In this cryptanalysis, the difference propagation from the plaintext to cipher-text is exploited. These difference propagations are assigned with probabilities to the possible keys and used to determine the most probable one.

A cipher is said to be resistance against this type of cryptanalysis only if the maximum differential probability is small. In this proposed algorithm, to measure this probability, 8-bit RCA cell-based S-Boxes (8-bit input to 8-bit output) are considered and 16 S-boxes arranged parallel to each other. The differential probability of KAMAR is calculated by using the following theorem described in [9].

Theorem 1: -If P_d is the maximum differential probability of all S-Boxes and *D* be the minimum number of active S-Boxes, then the maximum differential characteristic probability *P* is bounded by P_d^D [9].

Hence, searching worst case assumption from the complete entries made in the distribution table for the RCA rule 30 based S-Box operations and its differential characteristic probability is 2^{-2}. Also, the difference on the input bit affects different bytes at end of each round process. Therefore, 7 active S-boxes exist after a single round of KAMAR. Using theorem 1, 12 round KAMAR will have differential probability of

$$P \le 2^{-2\times 7\times 12} = 2^{-168}$$

Thus, KAMAR is effectively resistant to differential cryptanalysis attack.

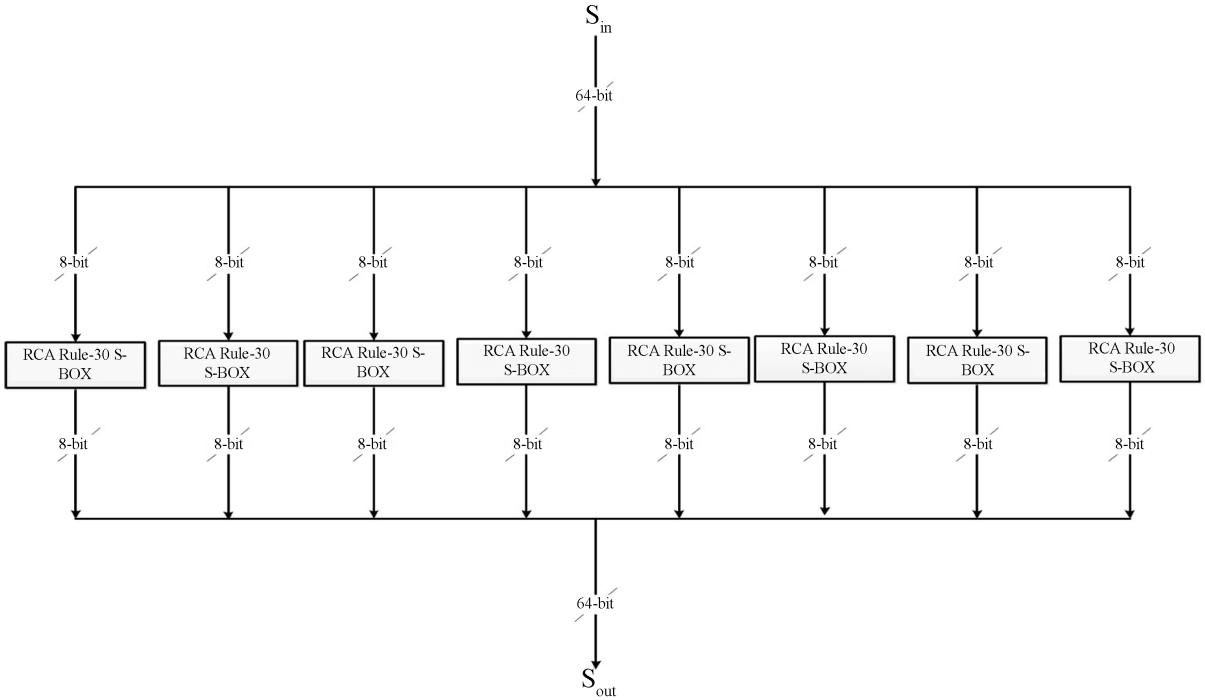

Figure 2. Internal structure of RCA rule-30 based S-BOX.

4.2. Linear Cryptanalysis

KAMAR is designed to provide security by focusing on minimizing the largest bias and finding different structures to increase the number of active S-Boxes in the round function. As discussed earlier, the 16 S-Boxes are arranged in parallel. The maximum probability of $\frac{11}{16}$ defines the magnitude of the correlation between the output of a linear expression and the nonlinear function. The linear probability bias ε is the difference between the 1/2 and the probability of a linear expression.

That is,

$$\left|\frac{1}{2}-\frac{3}{4}\right| = 2^{-2}.$$

Thus, for each round in the KAMAR, the minimum number of active S-Box is observed to be 7 and, therefore, the correlation probability of one round KAMAR is $2^{-2\times 7} = 2^{-14}$. Hence, the proposed algorithm is resistant to linear cryptanalysis.

5. Hardware Implementation and Synthesis Results

In the hardware implementation (FPGA), the HDL module of the proposed KAMAR algorithm is synthesized, and its equivalent hardware circuitry of the algorithm is extracted. The well-organized FPGA implementation results were extracted after place and route with the ISE 12.1i tool from Xilinx on a VIRTEX-4, XC4VL25-10 ff668 platform with speed grade of 12. The Mentor Graphics Modelsim SE PLUS 6.0c is used to post map simulate and verify the hardware module functionality.

The iterative loop architecture for KAMAR shown in **Figure 3** consists of a round function on the right hand side and a key scheduling function on the left hand side. In this design, resource consuming blocks are the 1-D RCA Rule-45 based key generation module, XOR, RCA Rule-30 based S-Box and Shifter. According to the specifications, the key scheduling arrangements consists of two multiplexers that allows switching the right and left half of the round key in the algorithm using appropriate control *Switch*. The *roundHalfKey* control to the multiplexer provides the round function with the right part of the round key for the first round execution and the left part of the round key for the second round function. The *select* control to the multiplexer selects the initial/

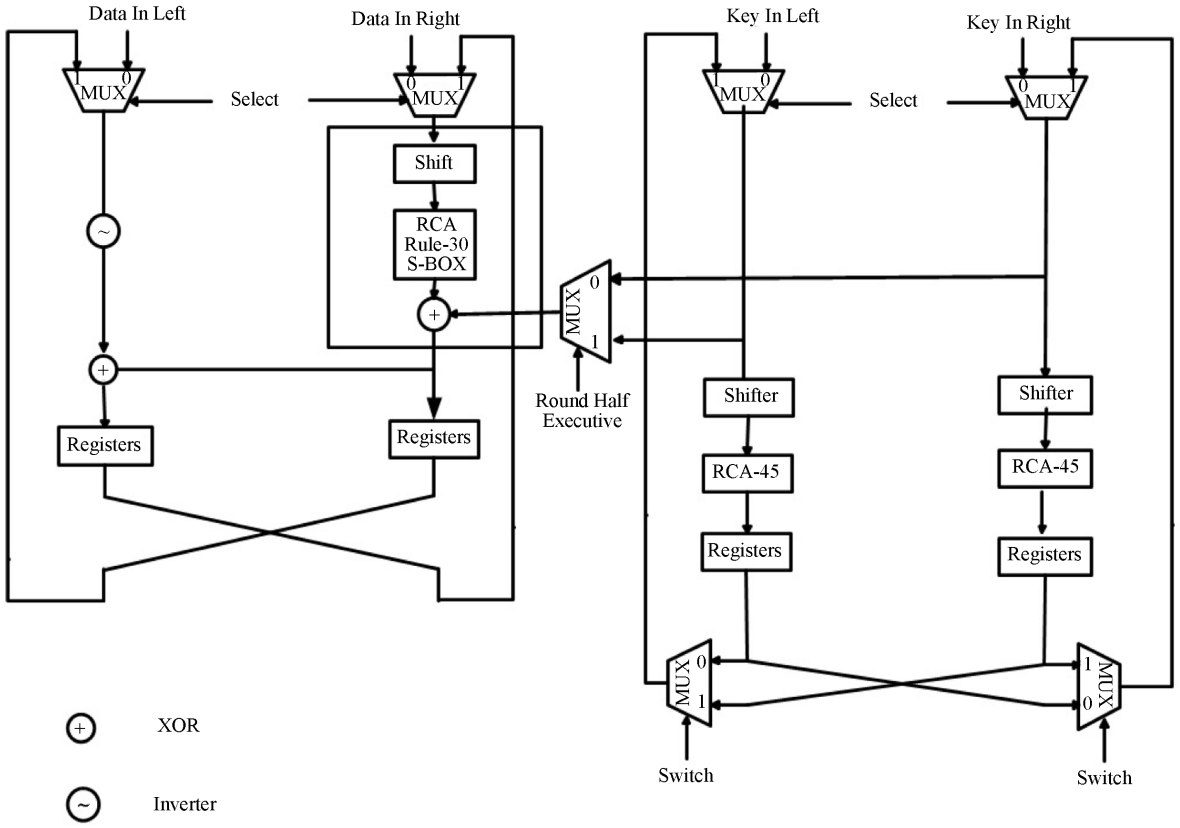

Figure 3. Loop architecture of KAMAR.

feedback data block and key to the round function. The loop is executed repeatedly until final round execution of encryption/decryption is completed.

For comparative discussions, a few implementations results of block ciphersL2DCASKE [10], SEA [19] [20], AES [17] [18] [21], DESXL [22], PRESENT [23], XTEA [24] are presented in **Table 2**. The symbol # denotes the block ciphers that are implemented straightforwardly in to the same FPGA device to aid precise comparative analysis. The summary of the results presents the area requirements (in slices), the work frequency, and the throughput of block cipher implementation. It is observed that the work frequency of KAMAR is higher compared to other cipher implementation. The bit/slice is a normalized parameter to denote the area efficiency of the various block cipher implementation. From the results, KAMAR is signified as area efficient block cipher with acceptable throughput.

6. Conclusions

This paper presented a new Symmetric key feistel block cipher based on reversible CA. The encryption algorithm is based on a particular class of reversible CA. One dimensional CA using radius 1 rule is used. The CA rule 30 and 45 S-box function with 8-bit input and 8-bit output is designed to operate over 128 bit data. The same operation in reverse order is performed during decryption. Due to a huge key space a brute-force attack appears practically impossible. The algorithm can be easily extended by using larger block size. Because of the parallel nature of CA, this algorithm can be implemented on a massively parallel platform and ensures high encryption/decryption speed. An optimized and Synthesizable VHDL code is developed for the FPGA implementation of KAMAR and other block ciphers.

KAMAR executes one round per six clock cycle, computes the round and the key round in parallel and supports both encryption and decryption at minimal cost. Compared to other recent block ciphers, KAMAR offers very less area utilization and nominal increase in throughput. Consequently, this algorithm can be considered as an interesting alternative for Wireless Multimedia Sensor Networks. Scope of further research includes low power ASIC implementations as well as further cryptanalysis and security evaluations.

Table 2. Comparison of Implementation results with other block ciphers.

Algorithm	FPGA Device	Structure	Data Block/Key length	**Nr**	# of slices	Freq (MHz)	Throughput (Mbits/sec)	Thr./Area Mbits/sec/slice	Bit/Slice
# KAMAR (Proposed Algorithm)	XC4VLX25	Feistel	128/128	16	312	550	2200	7.05	0.410
AES [17]	XCV100E	Non-Feistel	128/128	10	1125	161	215	0.19	0.114
AES [18]	XCV3200E	Non-Feistel	128/128	10	1769	167	2085	1.18	0.072
SEA [19]	XC4VLX25	Feistel	126/126	117	438	241	260	0.59	0.288
SEA [20]	XC4VLX25	Feistel	126/126	117	360	189	203	0.56	0.350
#L2DCASKE	XC4VLX25	Non-Feistel	128/128	12	336	334	2673	7.95	0.381
#AES	XC4VLX25	Non-Feistel	128/128	10	606	214	2743	4.52	0.211
#DESXL	XC4VLX25	Feistel	64/56	16	323	260	1039	3.21	0.198
#PRESENT	XC4VLX25	Feistel	64/80	31	266	436	901	3.38	0.241
#XTEA	XC4VLX25	Feistel	64/128	32	285	210	421	1.47	0.224

# denotes personal implementation.

References

[1] Misra, S., Reisslein, M. and Xue, G.L. (2008) A Survey of Multimedia Streaming in Wireless Sensor Networks. *IEEE Communications Surveys and Tutorials*, **10**, 18-39. http://dx.doi.org/10.1109/SURV.2008.080404

[2] Gures, E. and Akan, O.B. (2005) Multimedia Communication in Wireless Sensor Networks. *Proceedings Annals of Telecommunications*, **60**, 799-827.

[3] Augusto, M., Claudionor, M.M., Diogenes, N.C. and Silva, C.D. (2003) Survey on Wireless Sensor Network Devices. *Proceedings of IEEE Emerging Technologies and Factory Automation*, **1**, 537-544.

[4] Sarkar, P. (2000) A Brief History of Cellular Automata. *Journal of ACM Computing Surveys* (*CSUR*), **32**, 80-107. http://dx.doi.org/10.1145/349194.349202

[5] Nandi, S., Kar, B.K. and Chaudhuri, P.P. (1994) Theory and Application of Cellular Automata in Cryptography. *IEEE Transaction on Computers*, **43**, 1346-1357. http://dx.doi.org/10.1109/12.338094

[6] Wolfram, S. (1986) Cryptography with Cellular Automata. *Crypto*'85, *LNCS* 218, Springer-Verlag, 429-432. http://dx.doi.org/10.1007/3-540-39799-x_32

[7] Koc, C.K. and Apohan, A.M. (1997) Inversion of Cellular Automata Iteration. *IEE Proceedings of Computer and Digital Technique*, **144**, 279-284. http://dx.doi.org/10.1049/ip-cdt:19971518

[8] Blackburn, S., Murphy, S., Paterson, K., *et al.* (1997) Comments on Theory and Application of Cellular Automata in Cryptography. *IEEE Transactions on Computers*, **46**, 637-639. http://dx.doi.org/10.1109/12.589245

[9] Tripathy, S. and Nandi, S. (2009) LCASE: Lightweight Cellular Automata-based Symmetric-key Encryption. *International Journal of Network Security*, **8**, 243-252.

[10] Jegadish Kumar, K.J., Chenna Kesava Reddy, K. and Salivahanan, S. (2011) Novel and Efficient Cellular Automata based Symmetric Key Encryption Algorithm for Wireless Sensor Networks. *International Journal of Computer Applications* (*IJCA*), **13**, 30-37. http://dx.doi.org/10.5120/1767-2424

[11] Sen, S., Shaw, C., Chowdhuri, D.R., Ganguly, N. and Pal Chaudhuri, P. (2002) Cellular Automata Based Cryptosystem (*CAC*). *Information and Communications Security*, Springer-Verlag, 303-314. http://dx.doi.org/10.1007/3-540-36159-6_26

[12] Han, S.J., Oh, H.-S. and Park, J. (1996) The Improved Data Encryption Standard (DES) Algorithm. *Proceedings of IEEE 4th International Symposium on Spread Spectrum Techniques and Applications*, Volume 3, Mainz, 22-25 September 1996, 1310-1314.

[13] Meyers, R.K. and Desoky, A.H. (2008) An Implementation of the Blowfish Cryptosystem. *IEEE International Symposium on Signal Processing and Information Technology*, Sarajevo, 16-19 December 2008, 346-351.

[14] Fang, R., Ying-Jian, Y. and Xiao-Bing, F. (2009) A Small and Efficient Hardware Implementation of the KASUMI. *International Conference on Information Engineering*, Taiyuan, 10-11 July 2009, 377-380.

[15] Israsena, P. (2006) Securing Ubiquitous and Low-Cost RFID Using Tiny Encryption Algorithm. *Proceedings of the* 1*st International Symposium on Wireless Pervasive Computing*, Phuket, 16-18 January 2006. http://dx.doi.org/10.1109/iswpc.2006.1613621

[16] Israsena, P. (2006) On XTEA-Based Encryption/Authentication Core for Wireless Pervasive Communication. *International Symposium on Communications and Information Technologies*, Bangkok, 18 October-20 September 2006, 59-62. http://dx.doi.org/10.1109/iscit.2006.339887

[17] Pramstaller, N., Mangard, S., Dominikus, S. and Wolkerstorfer, J. (2004) Efficient AES Implementations on ASICs and FPGAs. *Proceedings ofthe Fourth Workshop on the Advanced Encryption Standard* (*AES*)*—State of the Crypto Analysis*, Volume 3373 of LNCS, Springer-Verlag, 98-112.

[18] Standaert, F.X., Rouvroy, G., Quisquater, J.J. and Legat, J.D. (2003) Efficient Implementation of Rijndael Encryption in Reconfigurable Hardware: Improvements and Design Tradeoffs. *Proceedings of Cryptography Hardware Embedded devices* (*CHES*), Volume 2779 of the series Lecture Notes in Computer Science, 334-350. http://dx.doi.org/10.1007/978-3-540-45238-6_27

[19] Mace, F., Standaert, F.X. and Quisquater, J.J. (2008) FPGA Implementation(s) of a Scalable Encryption Algorithm. *IEEE Transactions on VLSI Systems*, **16**, 212-216. http://dx.doi.org/10.1109/TVLSI.2007.904139

[20] Jegadish Kumar, K.J., Salivahanan, S. and Chenna Kesava Reddy, K. (2010) Implementation of Low Power Scalable Encryption Algorithm. *International Journal of Computer Applications*, **11**, 14-18.

[21] Standaert, F.X., Rouvroy, G., Quisquater, J.J. and Legat, J.D. (2003) A Methodology to Implement Block Cipher in Reconfigurable Hardware and Its Application to Fast and Compact AES RIJNDAEL. 11*th ACM International Symposium on Field-Programmable Gate Arrays* (*FPGA*'03), 216-224.

[22] Leander, G., Paar, C., Poschmann, A. and Schramm, K. (2007) New Lightweight DES Variants. *Proceedings of FSE*'07, Volume 4593 of LNCS, Springer-Verlag, 196-220. http://dx.doi.org/10.1007/978-3-540-74619-5_13

[23] Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y. and Vikkelsoe, C. (2007) PRESENT: An Ultra-Lightweight Block Cipher. *Proceedings of CHES*'07, Volume 4727 of LNCS, Springer-Verlag, 450-466. http://dx.doi.org/10.1007/978-3-540-74735-2_31

[24] Wheeler, D. and Needham, R. (1997) TEA Extensions. Technical Report, Computer Laboratory, University of Cambridge, Cambridge.