
Light Weight Cellular Automata Computations and Symmetric Key for Achieving Efficient Cryptography

¹Satyanarayana Reddy, ² M. Shiva Kumar

¹Mtech Scholar, Department of Electronics and Communication Engineering, Anurag Group of Institutions, JNTUH, Telangana, India.

²Assistant professor, Department of Electronics and Communication Engineering, Anurag Group of Institutions, Hyderabad, JNTUH, Telangana, India.

ABSTRACT

A novel cryptographic algorithm mainly depends on encryption and decryption using cellular automata. The algorithm has plaintext and key with 128bits and 12 rounds of operation, here we are extending size of plaintext and key to 256 bits with 16 rounds of operation, this increases the security. We also increased complexity by implementing different modules such as bit-permutation, inverse bit-permutation NRCA, RCA.

Keywords: RCA. NRCA, bit permutation, RBP.

INTRODUCTION

Cryptography is the art and science of keeping messages secure. The basic objective of network security and cryptographic algorithms is to communicate securely over an insecure medium. The security to the data is provided by the cryptographic algorithms. The word cryptography has come from a Greek word, which means secret writing. The message to be sent through an unreliable medium is known as plaintext, which is encrypted before sending over the medium. The encrypted message is known as cipher text, which is received at the other end of the medium and decrypted to get back the original plaintext message. Cryptographic algorithm is a mathematical function used for encryption and decryption. Cryptographic algorithms are broadly classified into three types: - i.e. symmetric algorithm, asymmetric algorithm and authentication. As per the symmetric algorithm, same key is used for encryption and decryption. In asymmetric algorithm, different keys are used for encryption and decryption. Authentication algorithms mean that the receiver should be sure about sender's identity. The cellular automata (CA) have been used since the forties of last century. It was used in many physical applications. The applications of Cellular Automata extended to fields such as biological models, image processing, language recognition, simulation, Computer architecture, cryptography etc. The Cellular Automata is also one of the modern methods used to generate binary pseudo-random a Sequences using registers. The concept of CA was initiated by J. Von Neumann and Stan Ulam in the early 1940's. He devised a CA in which each cell has a state space of 29 states, and showed that the devised CA can execute any computable operation. He studied the 1 dimensional rules of Cellular Automata. However, due to its complexity, in the 1970, the mathematician John Conway proposed his now famous game of life which received widespread interest among researchers. His research was based on 2D Cellular Automata rules. Stephen Wolfram studied in much detail and showed that a family of simple one-dimensional cellular

**Address for correspondence:*

esr.1506@gmail.com

automata rules (now famous Wolfram rules) and are capable of emulating complex behavior. The main concern of this paper is secret key systems. In such systems the encryption key and the decryption key are same (symmetric key). The encryption process is based on generation of pseudorandom bit sequences, and CAs can be effectively used for this purpose 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 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.

Cellular Automata has following inherent Properties

- Parallelism means that the individual cell updates are performed independently of each other. That is, we think of all of the updates being done at once
- Homogeneity means that each cell is updated according to the same rules.

In this paper a new algorithm for encryption and decryption is introduced. Here, we have mostly concentrated on securing the key, by using the concept of cellular automata. In cellular automata, we have different rules based on each rule we generate the random key for each round. By using these keys, we are performing encryption and decryption operation. In encryption, we have xormodule, bit permutation, non reversible cellular automata. In decryption, we have xormodule, reverse bit permutation, non reversible cellular automata.

The structure of the paper is as follows: Section 1 describes introduction, Section 2 describes key generation, Section 3 describes encryption and Section 4 describes decryption, Section 5 describes simulation waveforms, Section 6 describes conclusion, Section 7 describes references.

KEY GENERATION

The input key is 256 bit. It is divided into 2 blocks of 128 bit. Each block is given to RCA module. First block is further divided into sixteen 8 bit blocks then given to rule 30 then we xor k_1 with k_{16} to get the k_{16} of next block. Likewise k_{15} is xored with old k_{16} to get new k_{15} and so on. After completion of this operation we generate the random key for second round. As for first round the input key itself serves as the round key. In a similar manner, we apply rule 45, rule 86, rule 90, rule 105, rule 150, rule 165, rule 218, rule 30, rule 45, rule 86, rule 90, rule 105, rule 150, rule 165 for the further rounds. The same procedure follows for the second block as well.

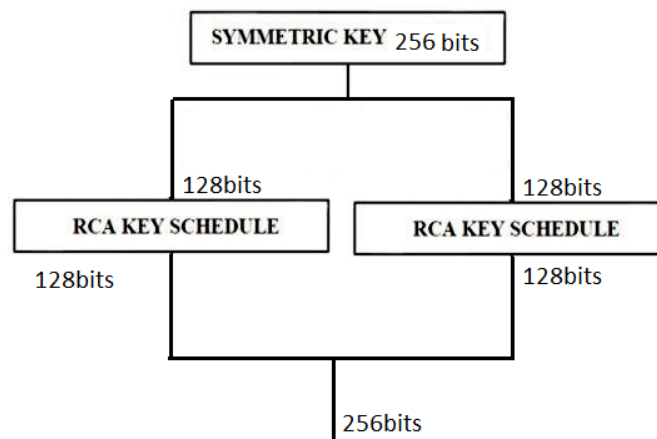


Fig1. 256 bit key schedule

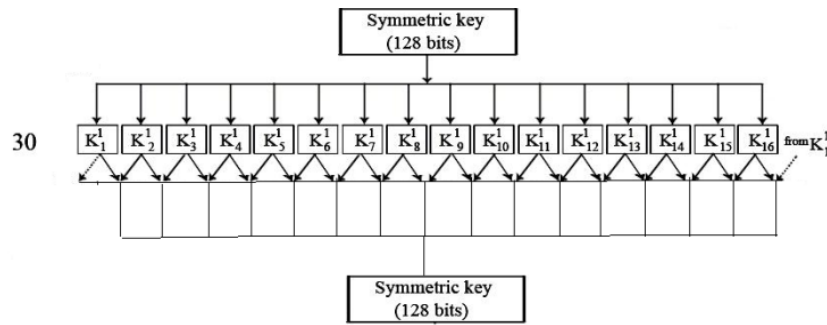


Fig2. Key generation round

ENCRYPTION

In encryption, we have three blocks i.e. xorblock, bit permutation, NRCA. The input to encryption is 256 bit plain text. This is divided into two 128 bit blocks. For the first block, we are passing 128 bit plaintext and 128 bit key, then the plaintext and key is further divided into sixteen 8 bit blocks. All these blocks of plaintext and key are given as input to xor block. Then we get sixteen 8 bit blocks as output from xor block. This output is clubbed such that we get 4 blocks of 32 bit. This serves as input to bit permutation block. In bit permutation the bits are arranged in different order based on the formula $(9 * I \text{ mod } 31) + 1$. The permuted output from bit permutation block is divided into sixteen 8 bit blocks and given to NRCA block along with input key. One of the inputs to the NRCA is the 8 bit input key. Each bit of this 8 bit input key is applied with one particular rule. Then the above output along with bit permutation output is xored. The same operation follows for 2nd block also. These operations are performed for 15 more rounds.

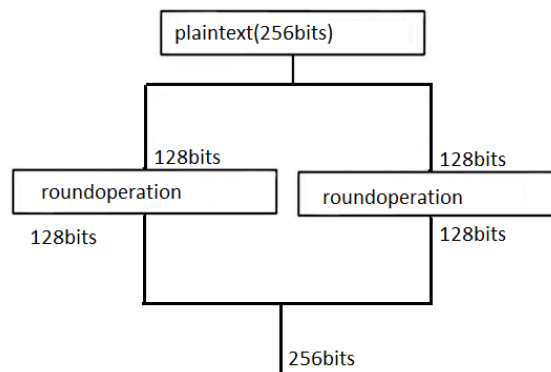


Fig3. 256 bit encryption

XOR Module

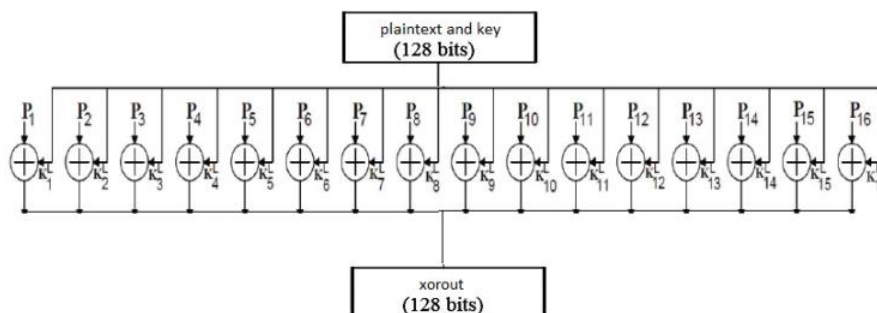


Fig4. XOR module

Bit Permutation

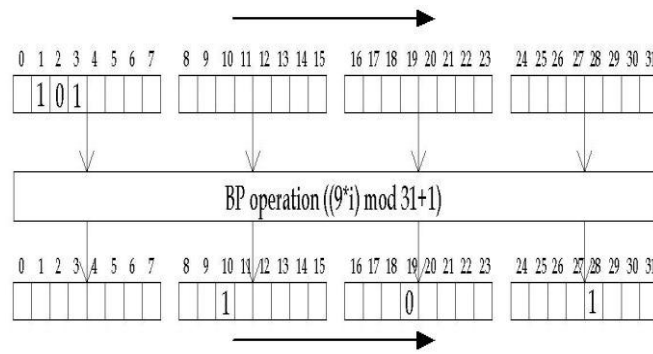


Fig5. 32bit BIT PERMUTATION

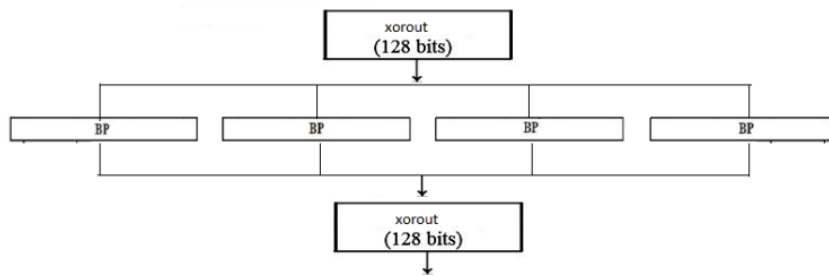


Fig6. 128bit BIT PERMUTATION

NRCA

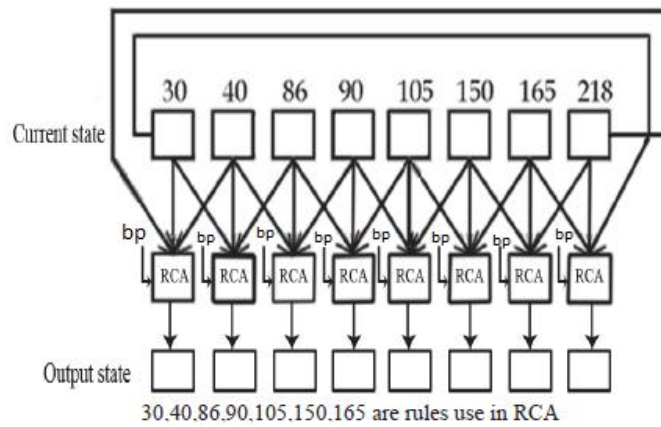


Fig7. 8 bit NRCA

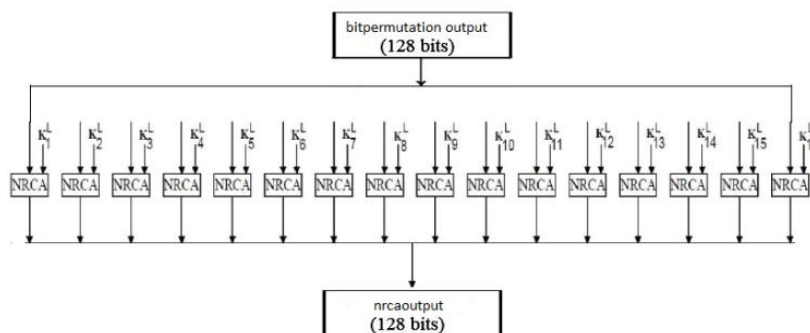


Fig8. 128 bit NRCA

DECRYPTION

In decryption, we have three blocks i.e. xor block, Reverse bit permutation, NRCA. The input to decryption is 256 bit cipher text. It is divided into two 128 bit blocks. For the first block, we are passing 128 bit cipher text and 128 bit key, then the cipher text and key is further divided into sixteen 8 bit blocks. All these blocks of cipher text and key are given as input to NRCA block. Then we get sixteen 8 bit blocks as output from xor block. This output is clubbed such that we get 4 blocks of 32 bit. This serves as input to reverse bit permutation block. In reverse bit permutation the bits are re organized into the original position. The output from Reverse bit permutation block is divided into sixteen 8 bit blocks and given to xor block along with input key. The same operation follows for 2nd block also. These operations are performed for 15 more rounds.

NRCA

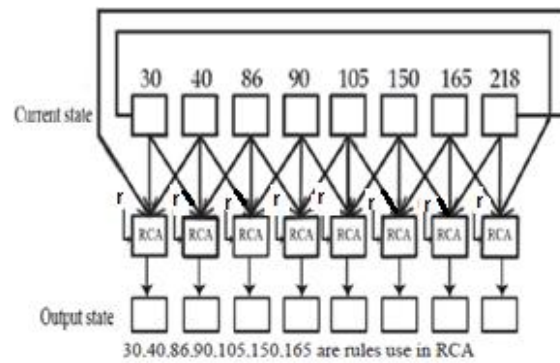


Fig9.8bit NRCA

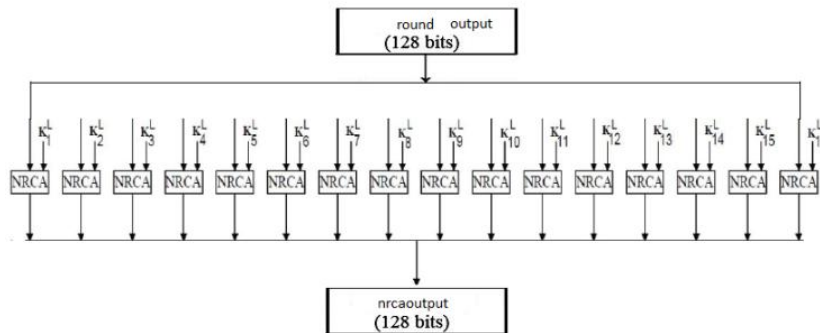


Fig10. 128bit NRCA

Reverse Bit Permutation

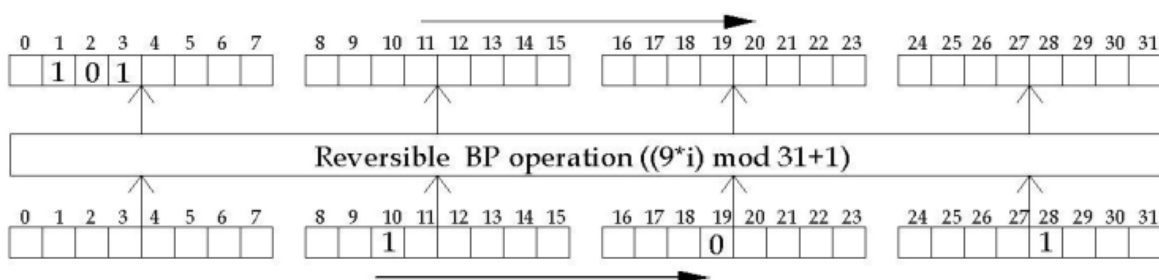


Fig11. 32bit REVERSE BIT PERMUTATION

CONCLUSION

A 256 bit cellular automata symmetric encryption and decryption is designed. The encryption has plaintext 256 bit and key 256 bit with 16 rounds of operation and key generation has 16 different keys are generated using 8 different transition rules for encryption and decryption. The decryption has ciphertext 256 bit and key 256 bit with 16 rounds of operation. The 256 bit encryption and decryption is simulated and synthesized using vhdl in Xilinx 13.2i

REFERENCE

- [1] B.Schneier, “Applied Cryptography,” Wiley, New York, 1996.
- [2] S Tripathy and S Nandi, CASE: “Lightweight Cellular Lightweight Cellular Automata-based Symmetric-key encryption.
- [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] Franciszek Serebinski, Pascal Bouvry, and Albert Y. Zomaya. “Cellular automata Computations and secret key cryptography,” Parallel Computing Journal, 30(5-6):753–766, 2004. F. Standaert, G. Piret, G. Rouvroy, J. Quisquater, and J. Legat, “ICEBERG: An involutonal cipher efficient for block encryption in reconfigurable hard- ware,” FSE ‘04, LNCS 3017, pp. 279-299, Springer-Verlag, 2004.
- [7] 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.
- [8] N. A. Moldovyan, P. A. Moldovyan, and D.H. Sum- merville, “On software implementation of fast DDP- based ciphers,” International Journal of Network Security, Vol. 4, No. 1, pp. 81-89, 2007.
- [9] T. Toffoli and N. Margolus, “Invertible cellular automata: A review,” Physica D, vol. 45, pp. 229-253, (reprinted with correction as of Oct. 2001).

AUTHOR’S BIOGRAPHY



Satyanarayana Reddy, received B.Tech (ECE) degree in 2012 from TRRCET, JNTUH & M.Tech (VLSI System Design) degree in 2015 from CVSR, JNTUH. His current research interests include the areas of very large scale integration (VLSI) high-speed digital communications, field-programmable gate arrays; synthesis of logic circuits and Analog and Digital vlsi circuits.



M. Shiva Kumar, received B.Tech (ECE) degree in 2005 from VIST, JNTUH & M.Tech (VLSI System Design) degree in 2013 from NCET, JNTUH. He is presently working as Assistant Professor in the Department of Electronics and Communication Engineering, Anurag Group of Institutions, Hyderabad. He is having 6 years of teaching & 3½ years of Telecom Industry experience. His current research interests include the areas of very large scale integration (VLSI) testing and fault-tolerant computing, and Antenna design.