AN EFFICIENT IMPLEMENTATION OF LRCA SCHEME FOR ENCRYPTION/DECRYPTION

V MNSSVKR GUPTA[1], K.V.S. MURTHY[2], DR.A. YESU BABU[3], R SHIVA SHANKAR[4]

[1,2,4] Department of CSE, S.R.K.R Engg. College, Affiliated to Andhra University, Bhimavaram, W.G.District, A.P. INDIA
[3]Professor & HOD, Department of CSE, Sir C R Reddy College Of Engineering, Eluru, AP, INDIA

Abstract— Network has become a significant way to transmit information because of the arrival of information era and the rapid development of Internet. Various multimedia digital products (such as text, images, videos, sound data, etc.) spread on the net. How to protect the benefit of the investors and legal rights owners is becoming an upcoming problem to solve. In this couple of years, the data encryption/decryption (modern cryptography) technique develops rapidly, which can provide a new effective approach to deal with this problem. In this paper a strong time efficient cryptosystem is proposed. A novel approach in cellular automata is used in which the plain text is arranged into layers of binary digital planes and then encrypted based on the rule set of Automata. This scheme exhibits strength by inheriting the naive properties of Cellular Automata, unpredictability, homogeneity, parallelism and sensitivity to the initial conditions. The proposed scheme is analyzed for time efficiency and observed to possess better confusion and diffusion properties when compared with Advanced Encryption Standard (AES). This scheme has advantage, that it has variable key size and block size; depending on the size of the plain text chosen. Simulation results show that the proposed system is on par with AES.

Index Terms—**Cellular Automata (CA); AES; LRCA; Encryption; Decryption.**

I. INTRODUCTION

As the technology is rapidly advancing day by day sharing of information over the internet is experiencing an explosive growth, which in turn is also posing new threats and vulnerabilities. From the past decades to nowadays, cryptography is part of our everyday lives, being found in such commonplaces as gas meters, cash payment systems, vehicle alarms, maritime charts, TV signal scramblers, the internet, and so forth.

The evolution of cryptography has been paralleled by the evolution of cryptanalysis — of the "breaking" of codes and ciphers. In 1917, the breaking of the Zimmerman telegram was instrumental in bringing the United States into World War I. The course World War II was dramatically altered by the cryptanalysts. The proliferation of computers and communications systems in the 1960s brought with-it a demand from the private sector for means to protect information in digital form and to provide security services. In 1970's Feistel at IBM introduces the new secret key algorithm called DES (Data Encryption Standard) is the most well-known cryptographic mechanism in history. Later from 1975 to 1985 public-key algorithms like Diffieand Hellman, Rivest, Shamir, and Adleman known as RSA then after in 1991 Digital signatures are introduced.

Over the past two decades Cryptographic techniques have become essential part of any secure digital communication. Information security is a protection of the interests of those relying information and the information systems and communications that deliver the information from harm resulting from failures of availability, confidentiality, and integrity.

A. Cryptography:

It is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication. Cryptography is not the only means of providing information security. Cryptography is used to hide information and gives information about the prevention and detection of cheating and other malicious activities. **Cryptanalysis** is the science, skill and practice of attempting to, or succeeding in violating the security of a cryptosystem, typically by trying to find the cryptographic key from matched plain text-cipher text pairs, or less commonly of trying to recover plain text from cipher text without knowing the cryptographic key. Cryptanalyst is one who practices cryptanalysis. The study of **cryptography** and **cryptanalysis** and all related subjects is known as Cryptology. In cryptographic terminology the actual/original information is known as plain text **or clear text**. Cipher text is the protected form of plain text and the process of converting plain text into cipher text using a cryptographic algorithm and a cryptographic key is known as **Encryption** and the process of converting cipher text into plain text using a cryptographic algorithm and a cryptographic key is known as the **Decryption**. When cipher text is correctly decrypted the result is recovered plain text. A mathematical function used to change plain text into cipher text (encryption) or vice versa (decryption). The function normally needs an extra parameter called the cryptographic key is called as Cryptographic Algorithm. A Cryptographic Keyes a collection of one or more numerical parameters without which it is supposed to be hard for an entity to recover certain information which has been encrypted. Key also means the index under which parameters are stored in a database.

B. Cryptographic goals

- Confidentiality is a service used to keep the content of information from all but those authorized to have it.

Secrecy is a term synonymous with confidentiality and privacy.

- Data Integrity is a service which addresses the unauthorized alteration of data. It does this by detecting data manipulation by unauthorized entities.
- Authentication is a service related to identification. Two parties entering into a communication should identify each other. Information delivered over a channel should be authenticated. Cryptography is usually subdivided into two major classes: entity authentication and data origin authentication.
- Non-repudiation This is a service which prevents an entity from denying previous commitments or actions, for example having received a message, or having signed a document as a last resort it is used to resolve disputes
- Confidentiality, integrity and availability (CIA) are maintained. These three concepts are at the core of almost every security program—if not by name, at least in practice. This is shown in Fig 1. They are most commonly described as a tri-angular view of security, with each side directly related to the other two.

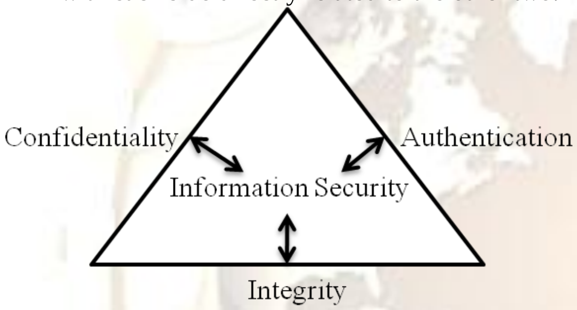

Fig 1 CIA relationship

C. *Criteria to evaluate primitives*

Any cryptographic scheme should have the following properties as shown in Fig 2:

- Level of security. This is usually difficult to quantify. Often it is given in terms of the number of operations required (using the best methods currently known) to defeat the intended objective. Typically the level of security is defined by an upper bound on the amount of work necessary to defeat the objective.
- Functionality. Primitives will need to be combined to meet various information security objectives. These primitives are most effective for a given objective will be determined by the basic properties of the primitives.
- Performance. This refers to the efficiency of a primitive in a particular mode of operation. (For example, an encryption algorithm may be rated by the number of bits per second which it can encrypt.)
- Ease of implementation. This refers to the difficulty of realizing the primitive in a practical instantiation. This might include the complexity of implementing the primitive in either a software or hardware environment.
- Methods of Operation. Primitives, when applied in various ways and with various inputs, will typically exhibit different characteristics; thus, one primitive could provide very different functionality depending on its mode of operation or usage.

D. *Cellular Automata*

Cellular Automata is a discrete model that consists of grids of cells in which each cell can exist in finite number of states. Every cell can change its state based on the states of neighboring cells by following a prescribed rule. Cellular Automata with its inherent properties like Parallelism, Homogeneity, and Unpredictability, as well as it being easily implementable in both software and hardware systems, has become an important tool to develop cryptographic methods.

Physical view

A Cellular Automaton (CA) is an infinite, regular lattice of simple finite state machines that change their states synchronously, according to a local update rule that specifies the new state of each cell based on the old states of its neighbors.

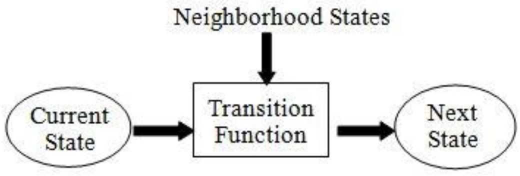

Fig 2 Updating current cell states

Cellular automata are a collection of cells that each adapts one of a finite number of states. Single cells change in states by following a local rule that depends on the environment of the cell.

Properties of Cellular Automata

- CA is a discrete simulation method; hence Space and Time are defined in discrete steps.
- CA is built up from cells, that are lined up in a string for one-dimensional automata arranged in a two or higher dimensional lattice
- The number of states of each cell is finite and discrete, all cells are identical
- The future state of each cell depends only of the current state of the cell and the states of the cells in the neighborhood
- The development of each cell is defined by rules

Dimensions of CA

A Cellular Automata is collection of cells; the shape of the cell not only square but also triangular and hexagonal and CA may be in one dimensional (1D), two dimensional (2D), three dimensional (3D) and many more as shown in Fig 3.

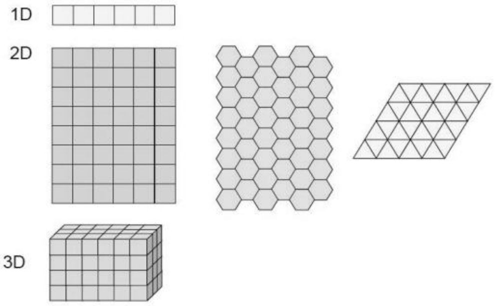

Fig 3 Dimensions of CA

E. *State of the cells in CA*

i. *Elementary Cellular Automata*

The state of all cells at time‘t’ is called configuration of CA at time ‘t’ and is denoted CA^t, the next state of the CA is denoted by CA^{t+1}. Considering the convenience to be treated by computer, many researchers only considered every cell only has two states {0, 1}. The state of a cell at the next time step is determined by the transition function along with current state of the cell and states of surrounding neighborhood cells. This phenomenon is represented as follows:

$$CA_i^{t+1} = f(C_{i-r}^t, C_{i-r+1}^t, \dots, C_{i-1}^t, C_i^t, C_{i+1}^t, \dots, C_{i+r}^t)$$

Where C_i^tmeans i^{th}cell at time't', C_i^{t+1}means i^{th} cell at time't+1', r is the neighborhood radius. Take radius as one (r=1 i.e. 3-neighborhood) with one dimensional cellular automata then the next state of CA is represented as

$$CA_i^{t+1} = f(C_{i-1}^t, C_i^t, C_{i+1}^t)$$

There are two classes of cellular automata one is Uniform Cellular Automata and Non Uniform Cellular Automata. If all the cells obey the same rule then it is uniform cellular automata. Otherwise it is non-uniform cellular automata. Several types of boundary conditions can be considered, a CA with periodic boundary has the extreme cells are adjacent to each other.

ii. Reversible Cellular Automata

By applying a rule to each cell c_i of the configuration CA_i^ta new configuration CA_i^{t+1} is obtained. This transformation can also be defined by a global transition function, which as an input takes configuration $CA^t$and results in a successive configuration CA^{t+1}. A CA is reversible if and only if the global transition function is one-to-one and hence every configuration not only has one successor but also has one predecessor.

Successor:

$$CA_i^{t+1} = f(C_{i-r}^t, C_{i-r+1}^t, \dots, C_{i-1}^t, C_i^t, C_{i+1}^t, \dots, C_{i+r}^t)$$

Predecessor:

$$CA_i^{t-1} = g(C_{i-r}^t, C_{i-r+1}^t, \dots, C_{i-1}^t, C_i^t, C_{i+1}^t, \dots, C_{i+r}^t)$$

Here f is the transition rule for moving forward and g is the transition rule for moving backward. As an example considers rules 15 and 85 (as shown in Table I and Table II respectively) are in the following way:

TABLE I Reversible Rule 15

Rule 15	111	110	101	100	011	010	001	000
	0	0	0	0	1	1	1	1
CA^t	1	0	1	1	1	0	0	0
CA^{t+1}	1	0	1	0	0	0	1	1
CA^{t+2}	0	0	1	0	1	1	1	0

TABLE II Reversible Rule 85

Rule 85	111	110	101	100	011	010	001	000
	0	1	0	1	0	1	0	1
CA^t	0	0	1	0	1	1	1	0
CA^{t+1}	1	0	1	0	0	0	1	1
CA^{t+2}	1	0	1	1	1	0	0	0

From the Table I and Table II, a CA moved to *n* time steps (iteration) by using one rule, another counterpart rule can be applied till the same number of time steps to obtain the original configuration of the CA. For example, Original CA moved to next state by performing 2 iterations using the rule 15, after that by applying rule 85 on the new state of CA the original CA is reconstructed, hence the rules 15 and 85 are reversible to each other up to any number of iterations.

II. LITERATURE SURVEY

Cryptography has been a part of our everyday lives for some time now, here there is a need for more efficient and secure encryption algorithms; it makes sense to consider alternatetechniques. Cellular automata as a medium for encryption are an attractiveidea in theory because most CA can be implemented on very fast hardwarehence a CA-based scheme may have the potential to encrypt and decryptmessages faster than existing techniques.

Most investigations into CA-based cryptosystems have been aimed at traditional secret key systems. There appear to be very few CA based public-key cryptosystems in the literature; one is the Finite Automata Public-Key Cryptosystem, although it uses non- homogeneous CA. Kari outlines an idea for a public-key cryptosystem based on reversible cellular automata, and poses the question of how to implement the key generation algorithm. The objective of public-key cryptosystem based on reversible cellular automata (RCA) is to design an RCA that is very hard to invert without some secret knowledge. That way, the RCA can be published and its inverse can be kept as the private key.

A CA is said to be an RCA if for every current configuration of the CA there is exactly one past configuration (pre-image). If one thinks of a CA as a function mapping configurations to configurations, reversibility implies that this function is objective.For one dimensional CA there are known algorithms for deciding whether a rule is reversible or irreversible.For CA of two or more dimensions it has been proved that the reversibility is undesirable for arbitrary rules. For finite CAs that is not reversible, there exist patterns for which there are no previous states. These patterns are called Garden of Eden patterns. In other words, no pattern exists which will develop into a Garden of Eden pattern.

A. Multi-layered Cellular Automata

Recently the concepts of Multi-layered Cellular Automata (MCA) are studied by Ramin Ayanzadeh et al [1]. The concept of multi-layer cellular automata (as shown in Fig 4) and a novel neighborhood structure is introduced from that a new approach for generating normal random numbers is proposed. First layer consists of binary cellular automata which are responsible for activating and inactivating cells in next layers. A cellular automaton with integer values is used for these layers. Interaction between layers of represented cellular automata leads to a dynamic and complex behavior of proposed model. Multi-layer cellular automata generate better normal random numbers.

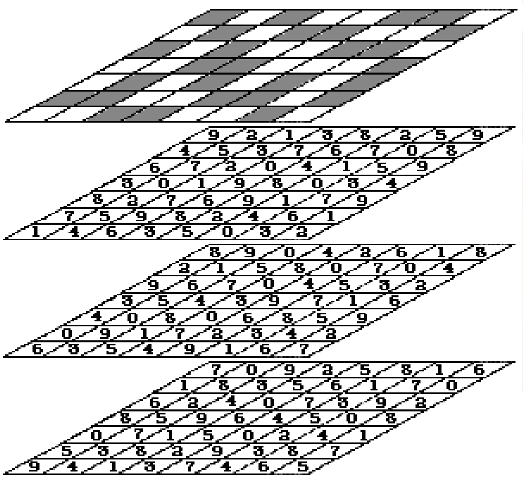

Fig 4 Multi-layered CA for generating normal pseudo random numbers

B. PRNS generation using Cellular programming

ThePRNSsgeneration depends upon the rules applied on the cellular automata, if good rules are applied good quality ofPRNS is generated, to identify good rules from the set of available rules for that here an evolutionary approach called cellular programming, in that the behavior of each rule is studied and thePRNSs generated by that rule is tested, if the rule performance is good and it passed all the NIST standard then those rules are taken applied on the CA to generate thePRNS, thisPRNS acts as secret key in encryption process. Cellular Programming is an evolutionary computation technique introduced to discover rules for non-uniform CAs. Fig 5 shows a CP system implemented to discover rules. The system consists of a population of N rules and each rule is assigned to a single cell of CAs. After initiating states of each cell, i.e. setting an initial configuration, the CAs start to evolve according to assigned rules during a predefined number of time steps. Each cell produces a stream of bits –PRNS. After stopping evolving CAs all PRNSs are evaluated. The entropy E_h is used to evaluate the statistical quality of each PRNS. To calculate a value of the entropy each PRNS is divided into subsequences of a size h. In all experiments the value h = 4 was used. Let k be the number of values which can take each element of a sequence (in our case of binary values of all elements $k = 2$) and k^h a number of possible states of each sequence ($k^h = 16$). E_h can be calculated in the following way:

$$E_h = -\sum_{j=1}^{k^h} p_{h_j} \log_2 p_{h_j}$$

Where p_{h_j}, is a measured probability of occurrence of a sequenceh_j in PRNS. The entropy achieves its maximal value $E_h = h$ when the probabilities of the k^h possible sequences of the length h are equal to $1/_{k^h}$. It is worth to mention that the entropy is only one of possible statistical measures of PRNSs. It will be used as a fitness function of CP. To decide about final statistical quality of PRNSs and a suitability of discovered rules for cryptography purposes some additional tests must be conducted[17].

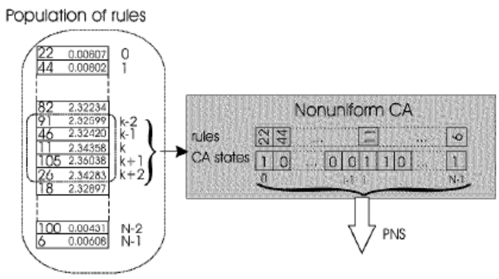

Fig 5 CP environment for evolution of rules of non-uniform CAs

C. Secret key Generation using CA

FranciszekSeredynskiet al. [8] described the process of generating thePRNS, which makes the strong encryption keys. During the generation ofPRNSs they have been taken one dimensional (1D), non-uniform CA. The quality ofPRNSs highly depends on a set of applied CA rules; those rules are identified using CP (see Section2.2). As the result of collective behavior of discovered set of CA rules very high qualityPRNSs are generated. The quality ofPRNSs outperforms the quality of known one dimensional CA-basedPRNS generators used in the secret key cryptography. The extended set of CA rules which was found makes the cryptography system much more resistant on breaking a cryptography key. The same process is reviewed by Tomassini& Sipper [9] instead of using 1D they considered 2D, various neighborhood structures are used, by this process also the generatedPRNSs are very strong and resistant against cryptographic attacks, later this has been extended to multi-dimensional cellular automata.

D. Block Encryption using CA

i. Block encryption using Elementary CA

PetreAnghelescu etal. [5] presented an encryption system implemented on a structure of Hybrid Additive Cellular Automata (HACA) used for securing data sent over the internet. They used the combination of Hybrid CA and Additive CA. Along with that they used the chaotic rules for providing good security.Samir Kumar Bandyopadhyay et al. [3] used CA in DES and AES.

ii. Block encryption using RCA

XIA Xuewen et al. [4], traditional reversible cellular automata (RCA) is fit for cryptography for its rules being an affine function, i.e. one reversible CA's rule can be applied in encryption process while another counterpart rule can be applied in decryption process. In order to improve the complexity of CA's dynamics, which is a crux in cryptography, traditional CA model is replaced by multi-granularity cellular automata (MGCA). Based on MGCA and RCA a cryptography algorithm that is proposed which called MGRCA. In MGRCA, cells have different granularity and can adjust their granularity dynamically by "split-recombination" behavior during the process of encryption and decryption. A multi-granularity cellular automata (MGCA) is presented aimed to improve the dynamic complexity of CA. In MGCA, each and every iteration consists of two steps; one is state-change and other is split-recombination. The former step is similar to the iteration of traditional CA and the latter step demonstrates the split of bulky-granularity CA and the recombination of fine-granularity CA.

It is undesirable if a given two-dimensional CA is reversible. This is true even when restricted to CA using the von Neumann neighborhood.

III. THEORETICAL ANALYSIS

A. Existing System

In the year 2009 XIA Xuewen et al. [4] has proposed data encryption using multi-granularity reversible cellular automata, in that only non-uniform cellular automata is considered. Later in the year 2010RaminAyanzadehet al. [1] has proposed multi-layer cellular automata and a novel neighborhood structure. According to these concepts, a scheme for generating normal random numbers is proposed. First layer consists of binary cellular automata which are responsible for activating and inactivating cells in next layers. A cellular automaton with integer values is used for these layers. Interaction between layers of represented cellular automata leads to a dynamic and complex behavior of proposed model. Main idea of this model is based on central limit theorem to generate normal random numbers.

B. Proposed System

Proposed system deals with the Layered Cellular Automata (LCA) is considered, where in automata can be viewed as a system, that consists of layers, and each layer is consists of rows of 1D cellular automata. The proposed system named as Layered Reversible Cellular Automata (LRCA), which is the combination of LCA and RCA, LRCA is a block encryption technique (large block size) with symmetric key. The text is converted into binary form and arranged in layers later it is iterated to construct the cipher text. This arrangement of cipher text gives the good confusion and diffusion to augment the security. In the proposed system a Layered Cellular Automata is considered where in the automata can be viewed as a system that consists of layers.The text is converted into binary form and arranged in layers (as shown in Fig 6) where each row is considered as a 1D CA with periodic boundary with radius equal to unity. 1D rule are used for encryption on each layer.

C. Requirements Elicitation

The requirements elicitation is the first process in requirements development. The process of gathering the requirements is called as Requirements Elicitation. The term elicitation is used in books and research to raise the fact that good requirements cannot just be collected from the customer, as would be indicated by the name requirements gathering. Requirements elicitation is non-trivial because you can never be sure you get all requirements from the user and customer by just asking them what the system should do. Requirements elicitation practices include interviews, questionnaires, user observation, workshops, brain storming, use cases, role-playing and prototyping. Requirements elicitation is a part of the requirements engineering process, usually followed by analysis and specification of the requirements.

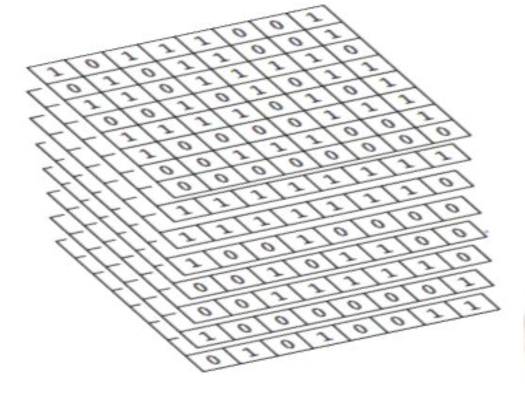

Fig 6 Layered Cellular Automata

D. Module Description

This system mainly divided into three modules as per the functionality. Fig 7 gives the diagrammatic representation of LRCA, which mainly contains the three modules, those are as follows:

1. Ruleset Generation
2. Encryption Process
3. Decryption Process

i. *Ruleset Generation*

Procedure for Ruleset generation

Take some set of rules which are reversible, both the sender and receiver shall agree on those rules that are to be used along with indexes created for the rules for the particular encryption. They shall also agree on the index and size in the key to be used to generate the number of iterations along with shift indicators. A random series of indexes is generated to identify the particular rule that is to be used on each cell of a layer. A shift on the sequence of rules is applied from row to row by treating the rows on the all the layers sequentially.

Algorithm for Ruleset Generation

The following steps give the proposed scheme for the ruleset generation.

[1]. Select the reversible rules

In this system 3-neighborhood structure is used, while this structure is considered there exist2^{2^3} $(2^8 = 256)$rules. From those rules there are only 6 reversible rules. By taking the key size into consideration, the following sets of reversible rules are used.In each pair, first rule used for encryption and second rules used for decryption.

[2]. Index the rules for both encryption and decryption

Index the rules from zero, and then separate the encryption and decryption rules separately. (Table 3.4-1)

[3]. Generate Random series

Generate a series of random numbers consists of only index numbers (i.e.0 to 3)

[4]. Generate encryption and decryption ruleset

Map the encryption rules to the above random numbers then collect that mapped series as encryption ruleset. To generate the decryption ruleset map the decryption rules to the random series.

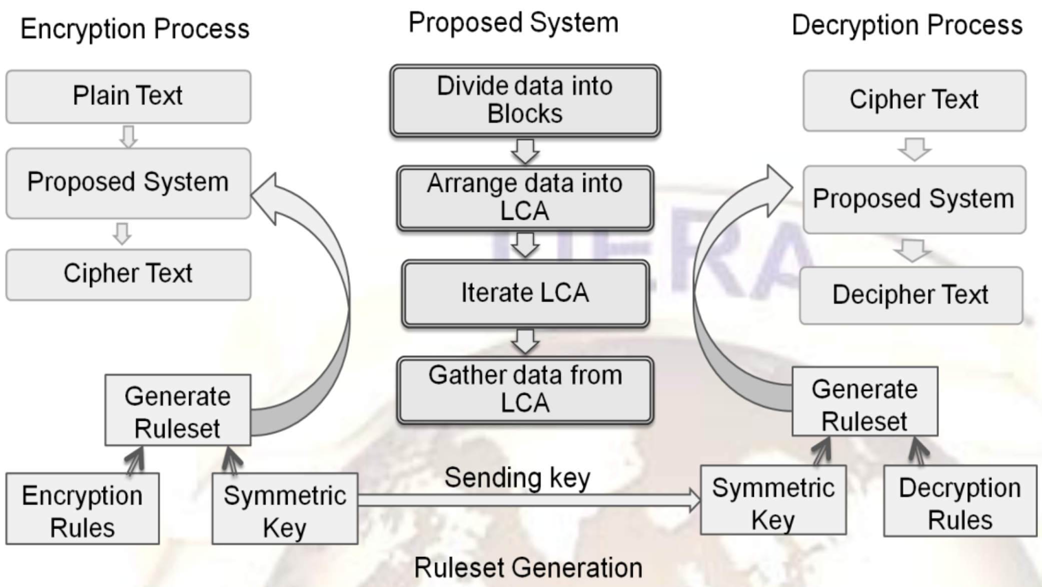

Fig 7 Encryption and Decryption process

[5]. Shifting both encryption and decryption ruleset

Shifting of ruleset can be done when moving from one layer to another layer which gives the good confusion and diffusion and also gives the unpredictability to an attacker. Pseudo Code for Ruleset generation for encryption and decryption is shown in Table III.

Example for Ruleset generation

Step 1 : Reversible ruleset: (170,240), (85, 15), (51, 51), (204,204)

Step 2 : Rule set for encryption and decryption process

Index	Encryption Ruleset	Decryption Ruleset
0	170	240
1	85	15
2	204	204
3	51	51

Step 3 : Random Series: 3 1 0 2 1 2 0 3..... 0 1

Step 4 : Encryption Ruleset: 51 85 170 204 85 204 170 51 ….. 170 85

Decryption Ruleset: 51 15 240 204 15 204 240 51 ….. 240 15

Step 5 : Encryption Ruleset: 170 204 85 204 170 51 ….. 170 8551 85

Decryption Ruleset: 240 204 15 204 240 51 ….. 240 1551 15

ii. *Encryption Process*

Procedure for Encryption

Divide the plain text into blocks of size 4096 characters (ASCII values for each character is considered so that there in total 4096*8 bits), (padding bits are added whenever needed) and the text is converted into binary sequence and the bits are arranged into 8 layers where each layer consists of 64*64 bits. Arrange the first bits of all the characters in the first layer and second bits of all characters in the second layer and continuing this process arrange the eight bit of all the characters in the eighth layer. Then apply a CA rule set on each cell of each layer and iterate each layer up to some predefined number of iterations, then each layer produce the cipher bits from each cell. Again the cipher bits are converted into text by converting them into ASCII values to produce the ciphertext. Pseudo Code for encryption process is shown in Table III.

Algorithm for Encryption Process

The steps involved in the encryption process are as follows:

[1]. Divide the plain text into blocks

Take the data in the plain text and divide into blocks of size 4096 bytes, take each block perform the following steps.

[2]. Take each block of plain text then repeat the steps from 3 to 5

[3]. Take each character's binary sequence arrange that into layers

Take the binary sequence of each character in a block then place first bit of all characters binary sequence in the first layer, second bit of all characters in the binary sequence in the second layer, like that last bit (8th bit) of all characters in last layer (8th layer).(Table IV)

[4]. Apply the encryption ruleset on LRCA

Apply the encryption ruleset on each layer to move the layered reversible CA to its next state depends upon the number of iterations. The number of iterations is calculated based on the selection of index values convert the each selected index values into binary combine the binary sequence of those index values to get the actual number of iterations. (Fig 8)

[5]. Gather all layers in LRCA to form the cipher text

Take first binary bit in all the layers to form the binary sequence convert that into ASCII which gives the first character in ciphertext, Take second binary bit in all the layers to form the binary sequence convert that into ASCII which gives the second character in ciphertext, like that Take last binary bit in all the layers to form the binary sequence convert that into ASCII which gives the last character in ciphertext. (Table V) (Fig 9)

Example for Encryption Process

Step 1 : Plain text: Cellular Automata provides parallelism
Blocks: ||Cellular|| Automat||a provid||esparal||lelism00

Step 2 : Plain text: Cellular

Step 3 :

TABLE IV ASCII and binary sequence of one block of plain text

Character	ASCII value	Binary Sequence
C	67	01000011
e	101	01100101
l	108	01101100
l	108	01101100
u	117	01110101
l	108	01101100
a	97	01100001
r	114	01110010

Step 4 :

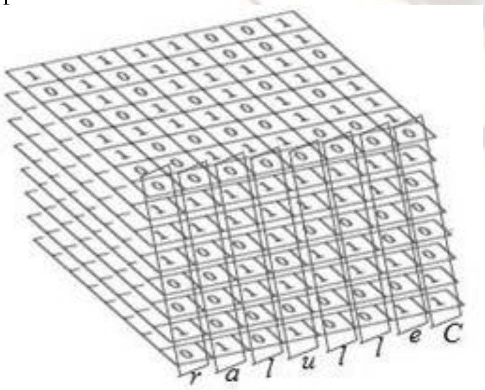

Fig 8 Arrangement of plain text into layers

Step 5 :

TABLE V Binary sequence and ASCII representation of cipher text

Binary Sequence	ASCII value	Character
00100100	36	$
01110001	113	q
01110111	119	w
00110100	52	4
00100110	38	&
01100100	100	d
00101000	40	(
00111000	56	8

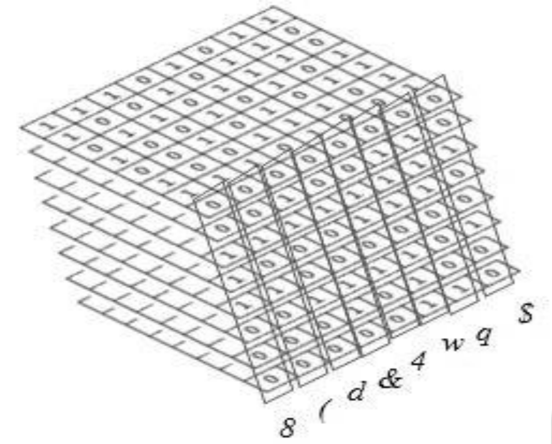

Fig 9 Gathering the layers to form ciphertext

iii. *Decryption Process*

Procedure for Decryption

The cipher text is converted back to binary form and the sequence of rules used in encryption is generated from the key the corresponding reversible rules are used in the same manner that of encryption on each cell for a predefined number of iterations, and the corresponding plain text is extracted from the binary sequence.

Algorithm for Decryption Process

The steps involved in the decryption process are as follows:

[1]. Divide the cipher text into blocks

Take the data in the cipher text and divide into blocks of size 4096 bytes, take each block perform the following steps.

[2]. Take each block of cipher text then repeat the steps from 3 to 5

[3]. Take each character's binary sequence arrange that into layers

Take the binary sequence of each character in a block then place first bit of all characters binary sequence in the first layer, second bit of all characters in the binary sequence in the second layer, like that last bit (8th bit) of all characters in last layer (8th layer).(Table 3.4-4)

[4]. Apply the decryption ruleset on LRCA

Apply the decryption ruleset on each layer to move the layered reversible CA to its next state depends upon the number of iterations. The number of iterations is calculated based on the selection of index values convert the each selected index values into binary combine the binary sequence of those index values to get the actual number of iterations. (Fig 10)

[5]. Gather all layers in LRCA to form the decipher text

Take first binary bit in all the layers to form the binary sequence convert that into ASCII which gives the first character in decipher text, Take second binary bit in all the layers to form the binary sequence convert that into ASCII which gives the second character in decipher text, like that Take last binary bit in all the layers to form the binary sequence convert that into ASCII which gives the last character in decipher text. (Table VI) (Fig 11)

Example for Decryption Process

Step 1 : Ciphertext: $qw4&d(8@ dzryisi(37;q&`dntxp *)0vagrw/-
Blocks: ||$qw4&d(8||@ dzryis||i(37;q&`||dntxp *)||0vagrw/-

Step 2 : Ciphertext: $qw4&d(8

Step 3 :

TABLE VII ASCII and binary sequence of one block of ciphertext

Character	ASCII value	Binary Sequence
$	36	00100100
q	113	01110001
w	119	01110111
4	52	00110100
&	38	00100110
d	100	01100100
(	40	00101000
8	56	00111000

Step 4 :

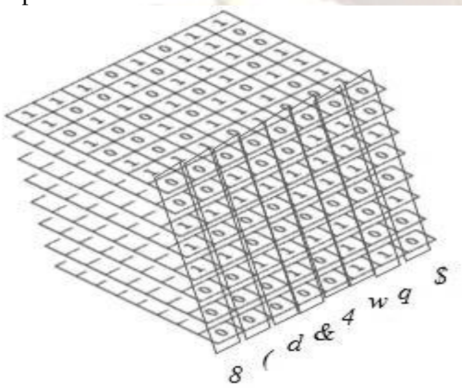

Fig 10 Arrangement of cipher text into layers

Step 5 :

TABLE VIII Binary sequence of cipher text into ASCII

Binary Sequence	*ASCII value*	*Character*
01000011	*67*	*C*
01100101	*101*	*e*
01101100	*108*	*l*
01101100	*108*	*l*
01110101	*117*	*u*
01101100	*108*	*l*
01100001	*97*	*a*
01110010	*114*	*r*

Note: Examples for encryption and decryption simulated to block length plain text as 8bytes but actual implementation worked out with 4096 bytes (i.e. layer size is 64×64).

Key Size: If only four rules are used then a random 128 bit key is generated in which subsequent 2 bits are used to identify indexes. If six rules are used then a random 192 bit key is generated in which subsequent 3 bits are used to identify indexes. In this LRCA a 128 bit key is used. It is shown in Table IX

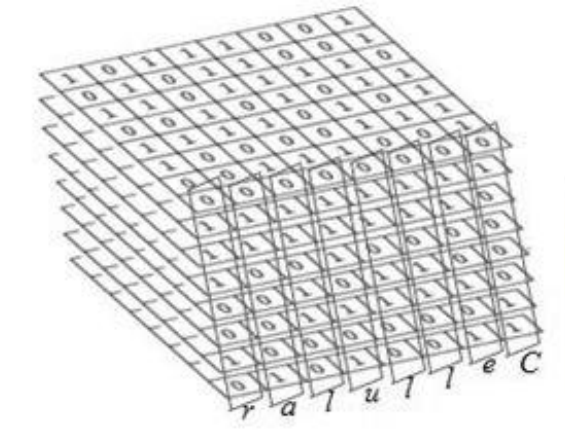

Fig 11 Gathering the layers to form plain text

TABLE IX Key size

Size of the layer	*No. of Rules*	*Key size*
32×32	*4*	*64*
32×32	*6*	*96*
64×64	*4*	*128*
64×64	*6*	*192*

Pre-shared information

Both the sender and receiver shall agree on the set of rules that are to be used along with indexes created for the rules for the particular encryption. They shall also agree on the index and size in the key to be used to generate the number of iterations along with shift indicators. A random series of indexes is generated to identify the particular rule that is to be used on each cell of a layer. A shift on the sequence of rules is applied from row to row by treating the rows on the all the layers sequentially.

IV. SYSTEM TESTING

Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding, Testing presents an interesting anomaly for the software engineer.

A. *Unit Testing*

Unit testing focuses verification effort on the smallest unit of software design that is the module. Using procedural design description as a guide, important control paths are tested to uncover errors within the boundaries of the module. The unit test is normally white box testing oriented and the step can be conducted in parallel for multiple modules.

During the testing of proposed system, each and every small unit is tested by giving the different inputs and different outputs. Unit testing has been done for both encryption and decryption processes. Unit testing for encryption process is as follows and the same process is repeated for the decryption also.

Table X Unit Testing

Unit Name	Input	Output	Test result
nuca(int rules[])	CA rule and CA initial condition	Gives new state of CA	pass
genruleset_enc()	Random numbers	Encryption ruleset	pass
arrange2d(char data[])	Plain text	Arrange data into LCA	pass
gather_8()	Data from LCA	Cipher text	pass

Table

B. *Integration Testing*

Integration testing is a systematic technique for constructing the program structure, while conducting test to uncover errors associated with the interface. The objective is to take unit tested methods and build a program structure that has been dictated by design. After the testing of each and every module individually, interface each module with other modules, building the program structure for encryption and decryption is done. Then the integration testing is performed on individuals.

Table XI Integration Testing

Program Name	Input	Output	Test result
encryption(String plaintextfile, String ciphertextfile)	Plain text file	Cipher text file	pass
Decryption(String ciphertextfile, String deciphertextfile)	Cipher text file	Deciphered text file	pass

C. *Test cases*

A test case in software engineering is a set of conditions or variables under which a tester will determine whether system is working correctly or not. The mechanism for determining whether a system has passed or failed such a test is known as a test oracle. A test case is usually a single step, or occasionally a sequence of steps, to test the correct behavior/functionalities, features of an application. Additional information that may be included mainly (It will shown in Table XII,XIII, XIV):

- Test case ID
- Test case objective
- Test case description
- Test case result

Table XII Test case 1

Test case ID	TC 01 (Test whether the selected file is text file or not)
Test case Objective	Only accept the text files.
Test case Description	To satisfy the objective of test case, filters are enabled while browsing the files. Then automatically it shows only the text files that available in the selected directory.
Test case Result	Accepts only text files. (Pass)

Table XIII Test case 2

Test case ID	TC 02 (Plain text given to the LCA)
Test case Objective	Ciphertext bits are produced

Test case Description	Plain text given to the LCA, then the text data is converted to ASCII and that ASCII values are arranged in layers then iterations applied on LCA, bits in the layers are gathered then convert them into ASCII, finally ciphertext is generated
Test case Result	Ciphertext is produced correctly. (Pass)

Table XIV Test case 3	
Test case ID	TC 03 (Cipher text and key is selected)
Test case Objective	Correct plain text is produced
Test case Description	Correct cipher text and key is selected then exact plain text generated because the key automatically generates the correctly, then plain text is reconstructed.
Test case Result	Produced plain text is correct. (Pass)

V. RESULT ANALYSIS

A. Brute Force Attack

Brute force attack or exhaustive key search is a strategy that can in theory be used against any encrypted databy an attacker who is unable to take advantage of any weakness in an encryption system that would otherwise make his/her task easier. It involves systematically checking all possible keys until the correct key is found. In the worst case, this would involve traversing the entire search space.

The key length used in the encryption determines the practical feasibility of performing a brute force attack, with longer keys exponentially more difficult to crack than shorter ones. Brute force attacks can be made less effective by obfuscating the data to be encoded, something that makes it more difficult for an attacker to recognize when he/she has cracked the code. One of the measures of the strength of an encryption system is how long it would theoretically take an attacker to mount a successful brute force attack against it.

Case 1: If only pre-shared information is known, there are 2^{128} *number of possible combination of keys*

Case 2: Neither pre-shared information nor the key is known then to guess the key there are $4! * 2^{128} * 120 * 2^{8}$ *possibilities exists.*

Confusion: Confusion refers to making the relationship between the key and cipher text as complex as possible and it is observed that the number of bits changed in the key results the number of bits changed in cipher text.

Comparison of confusion property with AES: Fig 12 gives the confusion levels of proposed algorithm are compared with AES algorithm by taking plain text of size 4k with key size 128 bit. It is observed that confusion levels obtained by the proposed algorithm are almost same as that of AES with marginal betterment.

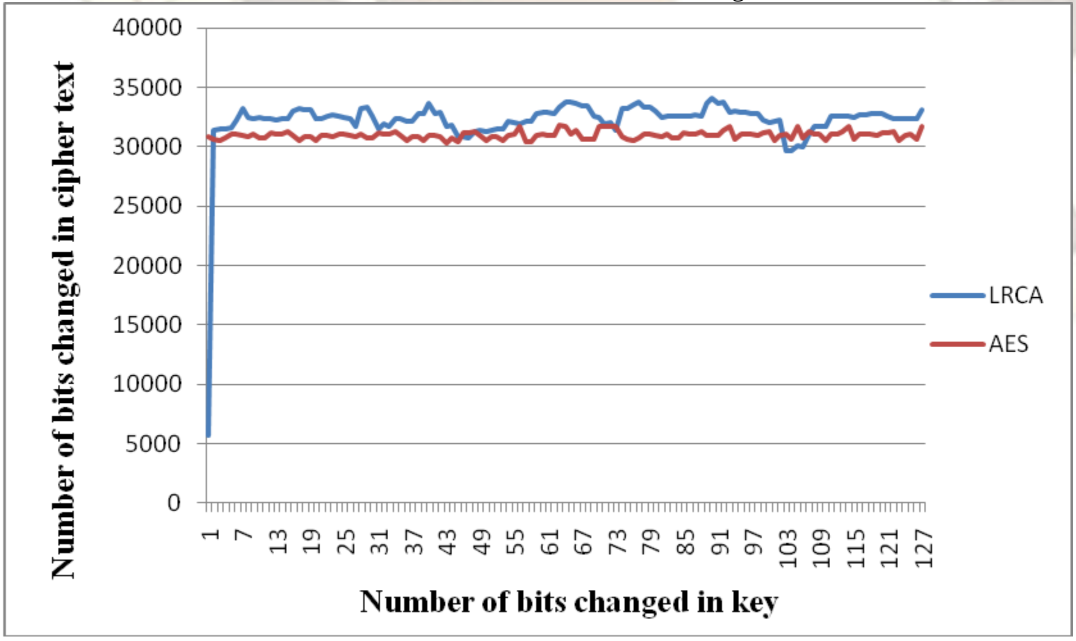

Figure 12 Confusion property of LRCA and AES

Confusion property of LRCA with different keys :Fig 13 gives the confusion property of proposed algorithm, which is tested with various keys applied on same plain text gives nearly 50% to 55% change in ciphertext.

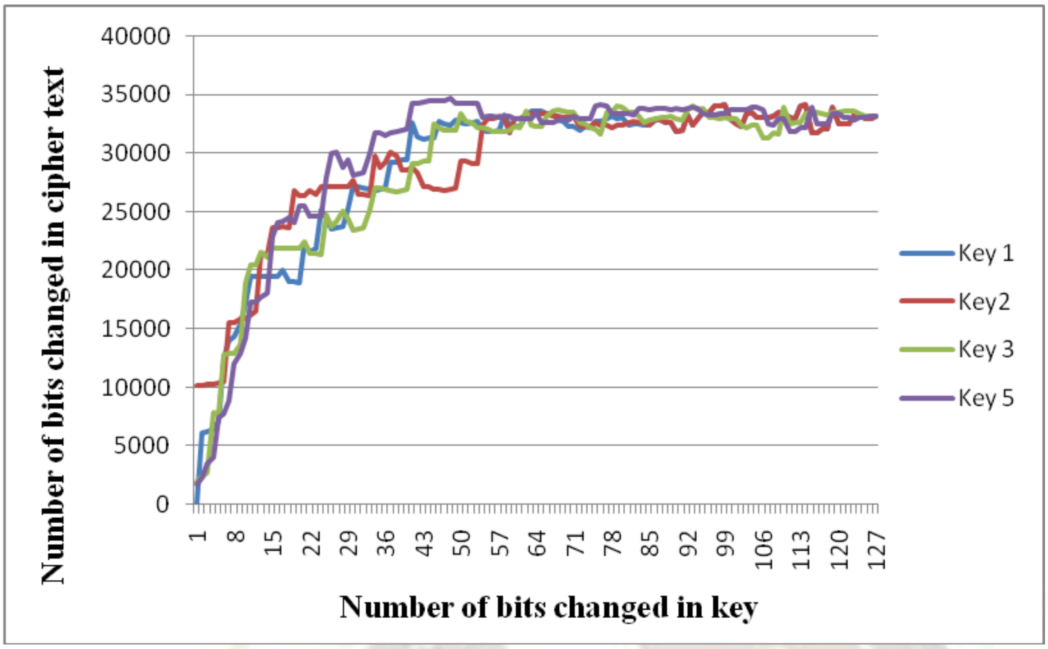

Figure 13 Confusion properties of LRCA different keys and same plain text

Diffusion: Diffusion refers to making the relationship between the plain text and cipher text as complex as possible and it is observed as the number of bits changed in the plain text results the number of bits changed in cipher text.
Comparison of diffusion property with AES: Fig 14 gives the diffusion levels of proposed algorithm are compared with AES algorithm by taking plain text of size 55bytes with key size 128 bit. It is observed that diffusion levels obtained by the proposed algorithm are better as compared with AES.

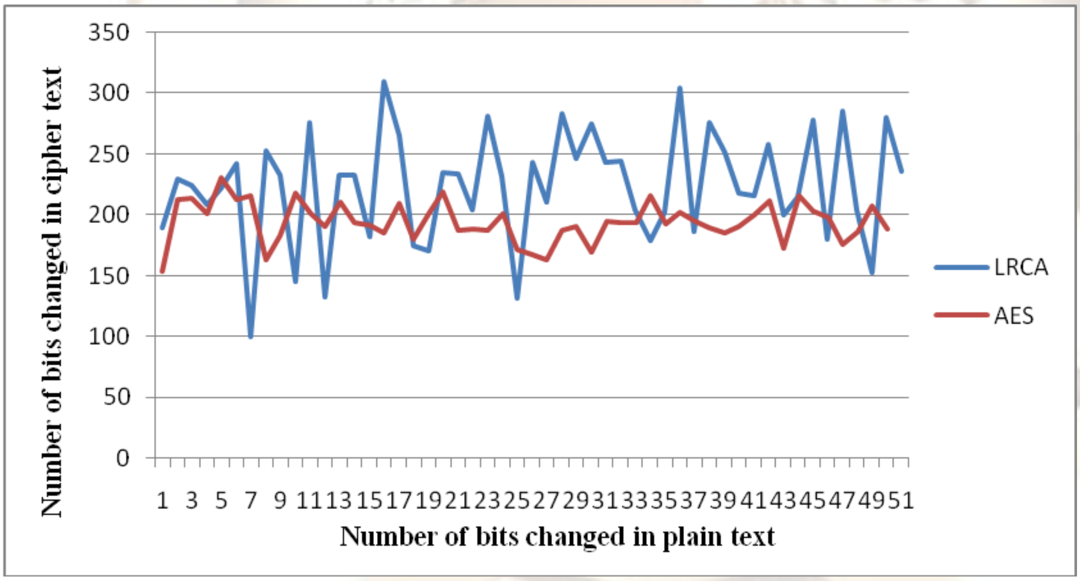

Figure 14 Diffusion property of LRCA and AES
Diffusion property of LRCA with different plain texts: Fig 15 gives the diffusion property of proposed algorithm, which is tested with same key applied on various plain texts gives nearly 75 to 80% change in ciphertext.

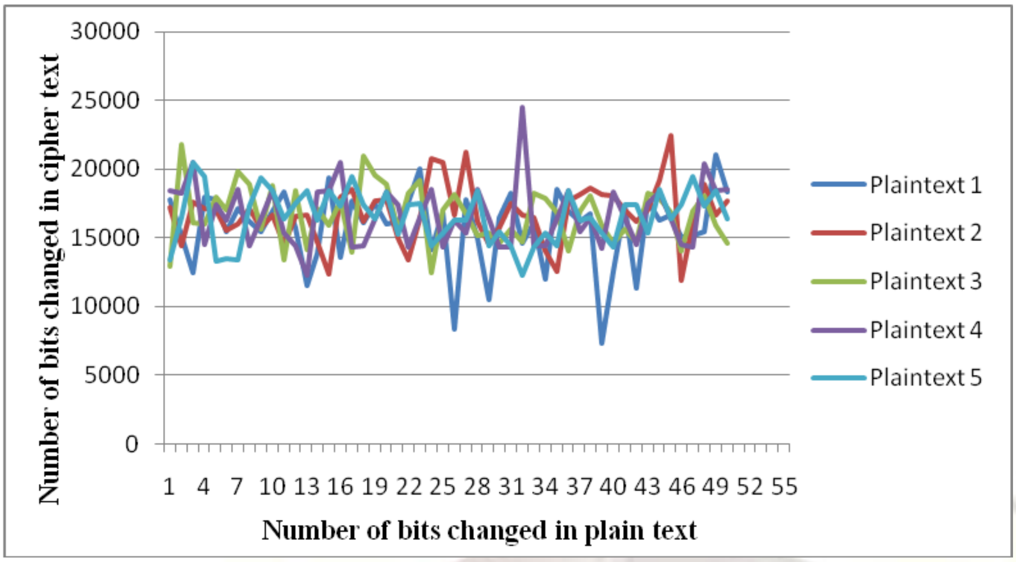

Figure 15 Diffusion property of LRCA with same key and different plain text

Time analysis of LRCA and AES: The encryption and decryption processes are taken together and it is observed that the time taken by LRCA is less than AES when same plain text and key used in both algorithms varying the size of the plain text. Fig 16 and Table XV gives the comparison of proposed algorithm and AES with respect to time.

Table XV Timing analysis between LRCA and AES

S. No	Plain text (KB)	LRCA (ms)	AES (ms)	Reduction (ms)
1	4	402	406	4
2	6	416	422	6
3	8	447	454	7
4	16	578	582	5
5	24	601	608	7
6	32	766	772	6

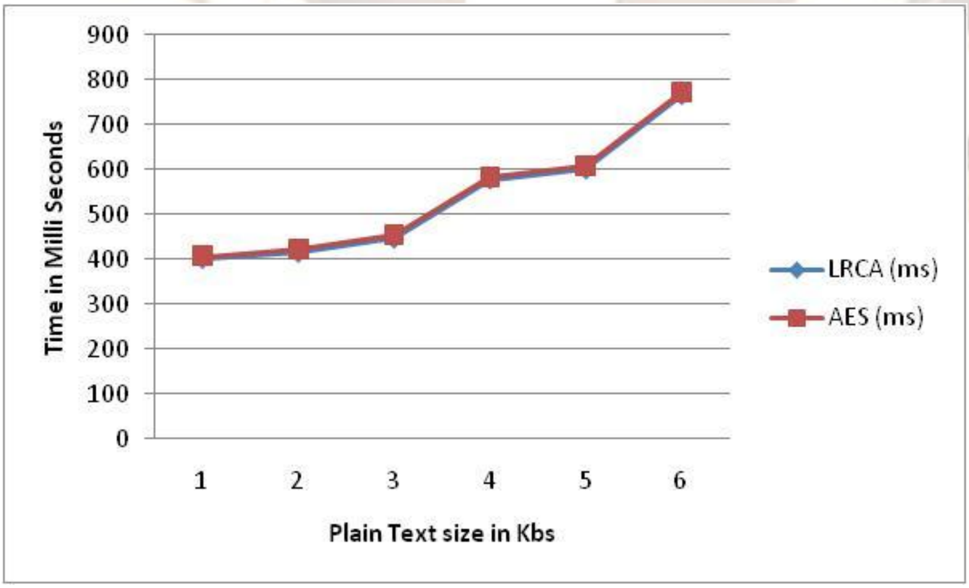

Figure 0-1 Comparison of LRCA and AES with respect to time

VI. CONCLUSION AND FUTURE WORK

Owing to the progress in information technologies and the growth of the Internet, vast amounts of data such as text and images have been digitized for easy storage, processing and transmission over the Internet. To prevent the transmitted data from being tampered with or grabbed from the Internet, many approaches have been proposed over the past decade. In this paper we proposed LRCA encryption/decryption, this method is compared with AES for various parameters like confusion, diffusion and time gives the better results. The proposed system is developed and analysis is done along with AES on various sizes of plain text and results are evaluated. AES shows that proposed system exhibits good confusion and diffusion properties and is time efficient than AES.

Future Work

The Proposed approach in principle invokes possibility of defining transformation functions based on the neighbors of different layers. This may lead to analysis of a new class of CA and is much of theoretical interest.

REFERENCES

[1]. Ramin Ayanzadeh, Yaghoub Moghaddas, Saeid Setayeshi, HadiGheibyand KavehHassani,“Multi-Layer Cellular Automata for Generating Normal Random Numbers”, ICEE, pp.43-48, 2010.

[2]. F. Maleki, A. Bijari, A. Mohades and M. E. Shiri,“Rule Discovery for Pseudorandom Number Generator Based on Cellular Automata”, IEEE, pp.739-744,2010.

[3]. Samir Kumar Bandyopadhyay and Somaditya Roy, “Cryptosystem for Information Security”, International Journal on Computer Science and Engineering, pp.1419-1422, 2010.

[4]. XIA Xuewen, LI Yuanxiang, XIA Zhuliang and WANG Rong, “Data Encryption Based on Multi-Granularity Reversible Cellular Automata”, International Conference on Computational Intelligence and Security, pp.192 -196, 2009.

[5]. Petre Anghelescu, Silviu Ionitaand Emil Sofron, “Block Encryption Using Hybrid Additive Cellular Automata”, Seventh International Conference on Hybrid Intelligent Systems, pp. 132- 137, 2007.

[6]. M. Seredinsky and P. Bouvry, “Block encryption using reversible cellular automata,” LNCS, pp. 785–792, 2004.

[7]. T. G. Mattos and J. G. Moreira, “Universality Classes of Chaotic Cellular Automata”, pp. 448-451, 2004.

[8]. Franciszek Seredynski, Pascal Bouvry and Albert Y. Zomaya, “Secret Key Cryptography with Cellular Automata”, International Parallel and Distributed Processing Symposium, pp.900-906, 2003.

[9]. M. Tomassini and M. Sipper, “On the Generation of High-Quality Random Numbers by Two-Dimensional Cellular Automata”, IEEE Transactions on Computers, pp. 1140-l151, 2000.

[10]. M. Tomassini and M. Perrenoud, “Stream Ciphers with One- and Two-Dimensional Cellular Automata”, LNCS, pp. 722-733, 2000.

[11]. S. Nandi and Pal Chaudhuri, “Analysis of Periodic and Intermediate Boundary Cellular Automata”, IEEE Transactions on computers, pp. 1-12, 1996.

[12]. S. Nandi, B. K. Kar, and P. Pal Chaudhuri, “Theory and Applications of Cellular Automata”, IEEE Transactions on Computers, pp.1346-1357, 1994.

[13]. Wentian Li and Norman Packard, “The Structure of the Elementary Cellular Automata Rule Space”, Complex Systems, pp.281-297, 1990

[14]. S.Wolfram, “Cryptography with Cellular Automata”, LNCS, pp. 429-432, 1986.

[15]. William Stallings, “Cryptography and Network security: Principles and Practices”, Prentice Hall Inc., pp. 30-50, 1999.

[16]. John Gordon, “Introduction to Cryptography”, Concept Labs, pp.1-52, 1998.

[17]. Moshe Sipper, “Evolution of Parallel Cellular Machines the Cellular Programming Approach”, Springer, pp. 106-159, 1997.

[18]. Paul C. van Oorschot Alfred J. Menezes and Scott A. Vanstone, “Handbook of Applied Cryptography”, CRC Press, pp.47-63, 1996.

[19]. Adam Clarridge and Kai Salomaa, “ A Cryptosystem Based on the Composition of Reversible Cellular Automata”, pp. 26-42,1995.