ANALYSIS OF ELEMENTARY CELLULAR AUTOMATA CHAOTIC RULES BEHAVIOR

K. Salman

Middle Tennessee State University, Murfreesboro, Tennessee, U.S.A.

ABSTRACT

We present detailed and in depth analysis of Elementary Cellular Automata (ECA) with periodic cylindrical configuration. The focus is to determine whether Cellular Automata (CA) is suitable for the generation of pseudo random number sequences (PRNs) of cryptographic strength. Additionally, we identify the rules that are most suitable for such applications. It is found that only two sub-clusters of the chaotic rule space are actually capable of producing viable PRNs. Furthermore, these two sub-clusters consist of two majorly non-linear rules. Each sub-cluster of rules is derived from a cluster leader rule by reflection or negation or the combined two transformations. It is shown that the members of each sub-cluster share the same dynamical behavior. Results of testing the ECA running under these rules for comprehensively large number of lattice lengths using the Diehard Test suite have shown that apart from some anomaly, the whole output sequence can be potentially utilized for cryptographic strength pseudo random sequence generation with sufficiently large number of p-values pass rates.

KEYWORDS

Elementary Cellular Automata, Pseudo Random Number Sequences, Diehard Test suite, Chaotic Rules, p-values.

1. INTRODUCTION

The subject of security of data from a transmitting agent to a receiving agent is well researched in both academia and professional research entities and may fall under encryption/decryption schemes. The basic idea is to find a means of delivering a message from one side to another side using public transmission media such that the message when interrupted during the transmission interval display zero information to the non-intended receiver. Obviously, the message must undergo some transformation in order to hide the message in a noise like background. Of course a persistent or interesting third party may resort to extremely powerful means such as super computers and good algorithms, depending on the importance of the content of the message, to try to recover the message from the distorting means used. Many excellent textbooks have been written to explain the various techniques used to encrypt and decrypt the message, such as [1]. One common and established scheme used in the art is the synchronous stream cipher, the basic structure of which is depicted in figure 1. The message to be transmitted by the ***sender*** is usually referred to as *Plaintext* denoted by $T$is mixed with the output R (*Random Sequence*) of the random number generator (**RNG**) by the exclusive-or primitive represented by the operation symbol $\oplus$ to yield the resultant C which is the encrypted message and usually referred to as *ciphertext*, thus: $C = T \oplus R$. The mixing operation, usually referred to as *encryption*, will make the ciphertext look like a random sequence and therefore an unintended third party (the Eavesdropper) should not be able to use it in order to extract the original message T. However, the intended ***Receiver*** can easily recover the original message by applying the same mixing

operation on the *ciphertext* C using the exact replica of the random number generator R, thus $T = C \oplus R$. This operation is usually referred to as *decryption*. It should be noted that the mixing operation, the exclusive-or, is a linear operation by means of which the recovery of the original message is facilitated. In order for the two operations *encryption* and *decryption* to work properly the random number generators at both sides should obviously be identical and use the exact same seed. However, when the random number generator used is extracted from a natural noise source then no matter what seed is used the two random sequences generated at both sides will not be the same by virtue of the action of the random nature of the source. Therefore natural sources based random number generators are precluded from utilization in stream ciphers. Mathematical based algorithms can thus present themselves as a viable alternative. It should be easy to realize that any algorithm based on mathematical means cannot be random, rather *pseudo random* and the viability needs to be ascertained.

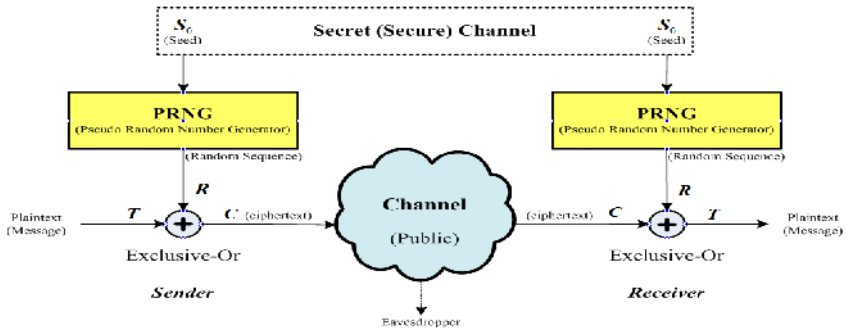


Figure 1. Typical Synchronous Stream Cipher.

In order for the Pseudo Random Number Generator (PRNG) to be strong it must produce a sequence that has at least the following salient attributes which are intrinsic in the natural RNGs:

- Uniformity of the output digits
- Long cycle length.
- No repeated patterns can be detected in the sequence.
- The sequence is easy to generate but hard to reproduce or best described by the so called the next bit unpredictability, i.e. the next bit cannot be predicted or extrapolated even with the *a priori* knowledge of the whole sequence.
- The algorithm must be simple and both operationally and computationally efficient.

Several algorithms were used in the past to generate PRNs, notably Linear Feedback Shift Registers (LFSRs). These registers proved to be simple and efficient and were used quite extensively until an algorithm developed by [2], showed that the linear complexity of such registers can be reduced drastically to just two span lengths of the register and rendered such devices unsuitable for encryption/decryption of the synchronous stream ciphers. The breakdown of these registers is in principle due to the linear primitive used exclusively in the feedback path. Non-linear Feedback Shift Registers, on the other hand, could prove neither efficient nor viable for such applications. Recently, however, cellular automata have been utilized to generate strong PRNs, [3]. To the knowledge of the author, in spite of the wealth of available research on cellular automata and the different configurations published, no thorough analysis of the strength of

periodic and uniform ECA for production of pseudo random number sequences seem to exist in the open literature. This paper is intended to address this issue at the outset.

The focus of this paper is to test the quality of the randomness of the output of uniform Elementary Cellular Automata to ascertain the strength of such novel means in utilization for pseudo random sequence generation of cryptographic strength. Such strength is gauged in this paper by the passing of the widely accredited state of the art Diehard Test Suite. The paper is arranged as follows: section 2 will describe the theory of elementary cellular automata, followed by the dynamics of ECA balanced chaotic rule equivalence in section 3, description of the Diehard test suite use in testing the output data string in section 4 while section 5 gives an account of the test data results with explanations, and finally the conclusions and suggestions for further research.

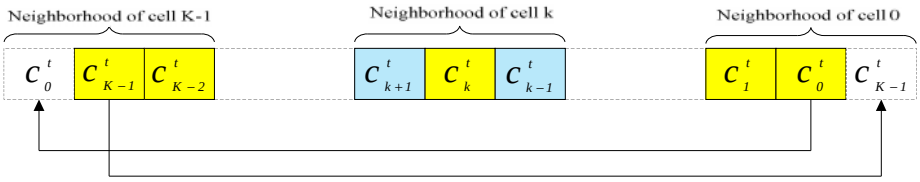
2. THEORY OF ELEMENTARY CELLULAR AUTOMATA (ECA)

We start by defining our ECA by a nonuple mathematical structure $[K, \Sigma, T, \eta(c_k^t), \Phi, \phi_x, \Omega, S_0, \varphi]$, where:

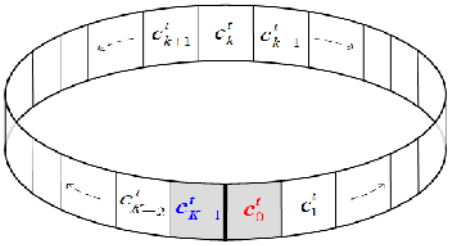
- $K \in \mathbb{N}$ is the ECA span length and is made up of interconnected identical memory cells c_k^t that are indexed spatially by $k \in K$ and temporally by $t \in \mathbb{Z}$.
- Σ is a finite set of states so that each memory cell c_k^t at location k and time step t can occupy one element of the finite set.
- $T \in \mathbb{Z}$ is the total time evolution of the ECA.
- $\eta(c_k^t) \in K$ is the neighbourhood of the memory cell c_k^t where it expands equally and spatially left and right at equal radius $r \in \eta$ in which case it can be stated that $\eta = 2r + 1$.
- Φ is the global rule of the ECA which is defined by the mapping $\Phi: \Sigma^K \rightarrow \Sigma^K$.
- $\phi_x \in \Re$ is the local transition rule defined by the mapping $\phi_x: \Sigma^\eta \rightarrow \Sigma$ where $\Re$ is the ECA rule space and $x \in 2^{\Sigma^\eta}$.
- Ω is the global ECA state space defined by $\Omega = \Sigma^K$.
- $S_0 = \{c_k^{t=0}\}^K$ is the initial seed of the automaton at time step $t = 0$.
- φ is the output function that works on the output string of the data generated by the ECA to form the resultant pseudo random string output used in the stream cipher.

The ECA under consideration is one dimensional (1-D) implying that it consists of a linear finite lattice of interconnected and identical cells of length $K, K \in \mathbf{N}$. Each cell denoted mathematically by c_k^t is interconnected with one cell to the right hand side denoted by c_{k-1}^t and another cell to the left hand side denoted by c_{k+1}^t. The alphabet is defined to be {0,1} over Φ_2, and the neighbourhood id defined by $\eta = 2r + 1 = 3$ where $r = 1$ that follows immediately from the definition of binary elementary cellular automata. The interconnection is symmetrically valid for $1 \leq k \leq K - 2$ leaving the two extreme end cells c_{K-1}^t and c_0^t to be made adjacently interconnected as depicted in figure 2(a). The whole ECA is made effectively circular and the

shaded extreme end cells coloured blue and red are made adjacent as indicated by the thick black demarcation line separating them in figure 2(b). The evolution of this circular (periodic) automaton will eventually form cylindrical ECA automata.



(a)



(b)

Figure 2- Illustration of periodic (circular) ECA configuration.

The next state of a cell away from the two boundaries can be represented by $c_k^{t+1} = \phi_x(c_{k+1}^t, c_k^t, c_{k=1}^t)$ under rule ϕ_x for $1 \le k \le K-2$. It follows that the next state of the extreme end cells, the left hand cell and right hand cell or the boundary cells under the same rule are logically represented by $c_{K-1}^{t+1} = \phi_x(c_0^t, c_{K-1}^t, c_{K-2}^t)$ and $c_0^{t+1} = \phi_x(c_1^t, c_0^t, c_{K-1}^t)$, respectively. The rule numbering used in this paper is implementation of the rule numbering scheme suggested by Wolfram, [4], and is now widely used by those skilled in the art. The rule number is quoted in the decimal number system and can be defined by $x = \sum_{i=0}^{2^3-1}(2^i \square m_i)$, where $m_i \square \Phi_2$ denote the minterms (as they are usually referred to in switching circuit theory) indexed by $i \square 2^\eta$ or $i \square 2^3$ which represents the row number in decimal of the truth table, as depicted in table 1. Since the mapping is $\{0,1\}^3 \rightarrow \{0,1\}$ it follows that each ECA rule will have a truth table the size of which consists of 2^η rows or 2^3 unique minters m_i. It can be seen that the size of the ECA rule space $\Re$ will therefore be 2^{2^η}or $2^{2^3} = 256$ unique rules. They are $\phi_0, \phi_1, \ldots, \phi_{30}, \ldots, \phi_{255}$, as depicted in figure 3. Each rule is uniquely determined by these minterms $\{m_j\}$, which can either be a form of a rule table:

$$\{m_7, m_6, m_5, m_4, m_3, m_2, m_1, m_0\}$$

Or a rule number $x = \sum_{i=0}^{2^3-1}(2^i \square m_i)$, where $m_i \square \Phi_2$ that gives the decimal representation of the above string:

$$(m_7 m_6 m_5 m_4 m_3 m_2 m_1 m_0)_2$$

		Neighborhood			ECA Rules ϕ_x					
		c_{k-1}^t	c_k^t	c_{k+1}^t	ϕ_0	ϕ_1		ϕ_{30}		ϕ_{255}
minterms	m_0	0	0	0	0	1	---	0	---	1
	m_1	0	0	1	0	0	---	1	---	1
	m_2	0	1	0	0	0	---	1	---	1
	m_3	0	1	1	0	0	---	1	---	1
	m_4	1	0	0	0	0	---	1	---	1
	m_5	1	0	1	0	0	---	0	---	1
	m_6	1	1	0	0	0	---	0	---	1
	m_7	1	1	1	0	0	---	0	---	1

Figure 3- ECA rule space table.

Table 1- ECA Rule Minterms Table.

$c_{k+1}^t c_k^t c_{k-1}^t$							
111	110	101	100	011	010	001	000
m_7	m_6	m_5	m_4	m_3	m_2	m_1	m_0

In order to realize the hardware of the ECA running under rule ϕ_x the logical expression of this rule must be derived in minimum form. One common approach is by means of Karnaugh mapping as illustrated in figure 4 for the two rules ϕ_{30} and ϕ_{45}.

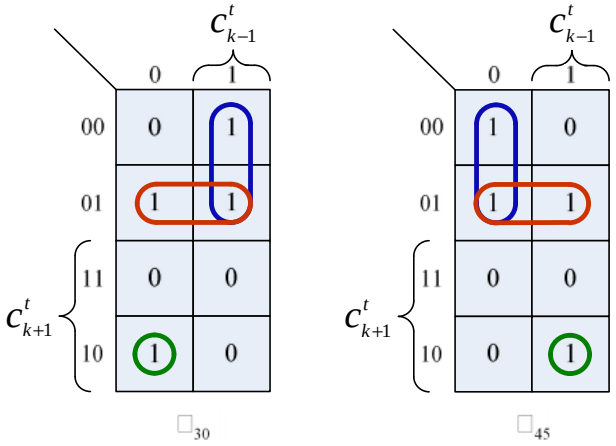


Figure 4- Karnaugh-Maps of the two chaotic rules ϕ_{30} and ϕ_{45}.

The next state minimized logical expression can be derived from the K-Map shown in figure 4 as $c_k^{t+1} = c_{k+1}^t \oplus (c_k^t + c_{k-1}^t)$ for rule ϕ_{30} and as $c_k^{t+1} = c_{k+1}^t \oplus (c_k^t + \overline{c_{k-1}^t})$ for rule ϕ_{45}. An example of the realization of this expression for rule ϕ_{45} is illustrated in figure 5.

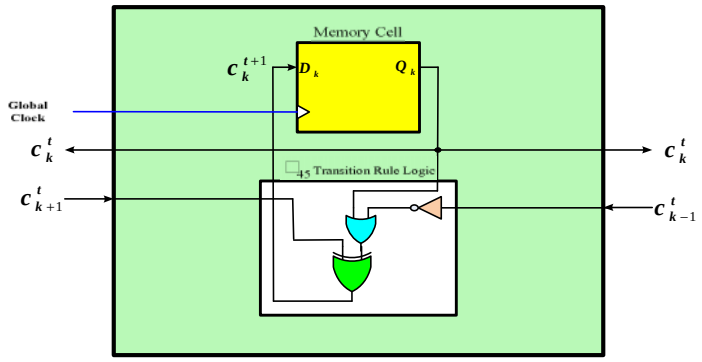


Figure 5- Φ_{45} logic circuit realization with memory cell at spatial index k.

The evolution of the periodic ECA in a single time step is depicted in figure 6 while figure 7 illustrates the evolution of the periodic ECA running under Φ_{45} with an arbitrary starting seed for seven contiguous time steps. The random nature of the ECA output at the seventh time step is apparent, which is not the case with all the ECA rule space.

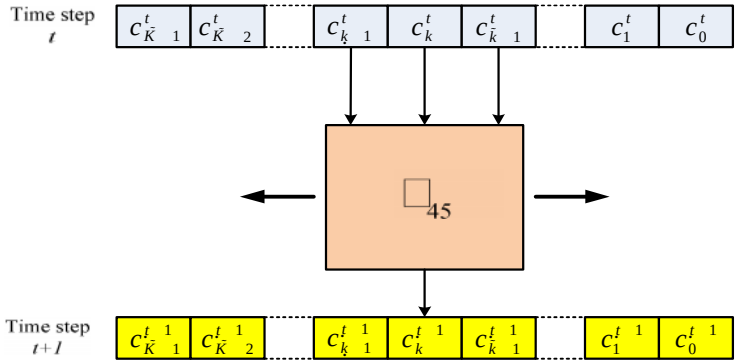


Figure 6- One time step evolution illustration using Φ_{45} .

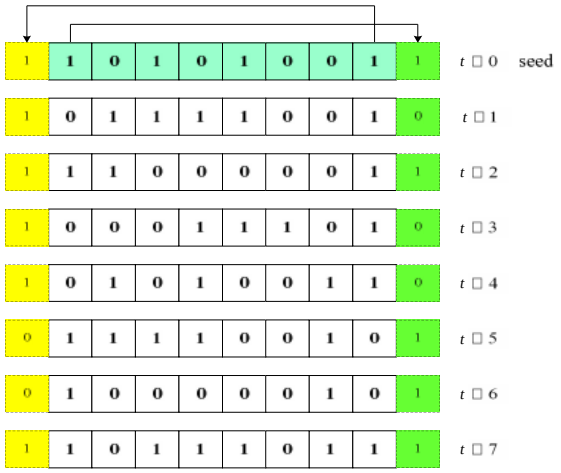


Figure 7- seven time-steps evolution of uniform periodic ECA with 8-cell span length using Φ_{45} .

The ECA rule space has been conjectured by [4] to be classified into four different classes:

- Class I: Evolution leads to homogeneous fixed points.
- Class II: Evolution leads to periodic configurations.
- Class III Evolution leads to chaotic, aperiodic patterns.
- Class IV Evolution produces persistent, complex patterns of localized structures.

Although this classification was based primarily on the phenomenological nature of the spatiotemporal patterns of the ECA dynamics, more in depth studies, such as that attributed to [5], inferred almost the same classification. The only class that can be useful for possible strong PRN generation is class III when the ECA is run uniformly under one local rule. However, in order for the output binary sequence to be useful it has to have at least some common attributes as listed in the previous section. The first attribute, Uniformity of the output digits, can only be satisfied when the local rule used has balanced minters. This means that the number of asserted minterms must equal to the number of unasserted minterms. It can be seen that the sub-space of the ECA rule space that satisfy this condition reduces to $\frac{2^{\eta}}{\left(\frac{2^{\eta}}{2}\right)!\left(2^{\eta}-\left(\frac{2^{\eta}}{2}\right)\right)!} = 70$ unique rules, to be denoted by $\varsigma \subset \Re$. However, this sub-space, albeit satisfies the first randomness attribute as outlined above does not contain purely chaotic rules of class III but rather other rules that belong to the other three classes. It has been shown here that only some of these rules, exactly 16 rules and denoted by $\zeta \subset \varsigma, \varsigma \subset \Re$, are actually chaotic, belong to class III and may prove to be useful for PRNs generation. These rules are:

$$\phi_{30}, \phi_{45}, \phi_{60}, \phi_{75}, \phi_{86}, \phi_{89}, \phi_{90}, \phi_{101}, \phi_{102}, \phi_{105}, \phi_{135}, \phi_{149}, \phi_{150}, \phi_{153}, \phi_{165}, \phi_{195}$$

It can be seen that this sub-space consists of two groups, one group is dominated by non-linear primitives in their logical expressions and will be denoted by $\zeta_{non-Linear} \subset \zeta$ and consist of all the constituent rules in the two cluster leaders ϕ_{30} and ϕ_{45}. The second half of the chaotic rules sub-space relies only on linear primitives in their logical expressions, and is denoted by $\zeta_{Linear} \subset \zeta$:

$$\phi_{60}, \phi_{90}, \phi_{102}, \phi_{105}, \phi_{150}, \phi_{153}, \phi_{165}, \phi_{195}$$

The following table 2 provides the Karnaugh Map based minimized logical expressions of the 16 rules based on the neighbourhood of the three cells, c_{k+1}^{t}, c_{k}^{t}, and c_{k-1}^{t}. It can be seen that eight rules are non-linear while the other eight rules are linear. The different colours associate the clusters of the rules. The rules marked yellow belong to the cluster of ϕ_{30}, those marked with green belong to ϕ_{45}, and these two clusters are the non-linear rules. The rules marked orange are the cluster of rule ϕ_{60}, while the blue colour show the two rules ϕ_{90} and ϕ_{165}. The last two linear rules ϕ_{105} and $\phi_{150}$are coloured pink. It can be seen that the cluster rules of ϕ_{90} and ϕ_{105}consist of just two rules.

Table 2-Chaotic Rules sub-space $\varsigma \subset \Re$ Logic Expressions

Rule Cluster	Minterms	Next State logical expression	Rule Dynamics
ϕ_{30}	ϕ_{30} = {0,0,0,1,1,1,1,0}	$c_k^{t+1} = c_{k+1}^t \oplus (c_k^t + c_{k-1}^t)$	Non-Linear $\zeta_{non-Linear} \subset \zeta$
	ϕ_{86} = {0,1,0,1,0,1,1,0}	$c_k^{t+1} = c_{k-1}^t \oplus (c_{k+1}^t + c_k^t)$	
	ϕ_{135} = {1,0,0,0,0,1,1,1}	$c_k^{t+1} = c_{k+1}^t \oplus (\overline{c_k^t} + \overline{c_{k-1}^t})$	
	ϕ_{149} = {1,0,0,1,0,1,0,1}	$c_k^{t+1} = c_{k-1}^t \oplus (\overline{c_{k+1}^t} + \overline{c_k^t})$	
ϕ_{45}	ϕ_{45} = {0,0,1,0,1,1,0,1}	$c_k^{t+1} = c_{k+1}^t \oplus (c_k^t + \overline{c_{k-1}^t})$	
	ϕ_{75} = {0,1,0,0,1,0,1,1}	$c_k^{t+1} = c_{k+1}^t \oplus (\overline{c_k^t} + c_{k-1}^t)$	
	ϕ_{89} = {0,1,0,1,1,0,0,1}	$c_k^{t+1} = c_{k-1}^t \oplus (c_{k+1}^t + \overline{c_k^t})$	
	ϕ_{101} = {0,1,1,0,0,1,0,1}	$c_k^{t+1} = c_{k-1}^t \oplus (\overline{c_{k+1}^t} + c_k^t)$	
ϕ_{60}	ϕ_{60} = {0,0,1,1,1,1,0,0}	$c_k^{t+1} = c_{k+1}^t \oplus c_k^t$	Linear $\zeta_{Linear} \subset \zeta$
	ϕ_{102} = {0,1,1,0,0,1,1,0}	$c_k^{t+1} = c_k^t \oplus c_{k-1}^t$	
	ϕ_{153} = {1,0,0,1,1,0,0,1}	$c_k^{t+1} = c_k^t \overline{\oplus} c_{k-1}^t$	
	ϕ_{195} = {1,1,0,0,0,0,1,1}	$c_k^{t+1} = c_{k+1}^t \overline{\oplus} c_k^t$	
ϕ_{90}	ϕ_{90} = {0,1,0,1,1,0,1,0}	$c_k^{t+1} = c_{k+1}^t \oplus c_{k-1}^t$	
	ϕ_{165} = {0,0,0,1,1,1,1,0}	$c_k^{t+1} = c_{k+1}^t \overline{\oplus} c_{k-1}^t$	
ϕ_{105}	ϕ_{105} = {0,1,1,0,1,0,0,1}	$c_k^{t+1} = \overline{c_{k+1}^t \oplus c_k^t \oplus c_{k-1}^t}$	
	ϕ_{150} = {1,0,0,1,0,1,1,0}	$c_k^{t+1} = c_{k+1}^t \oplus c_k^t \oplus c_{k-1}^t$	

Based on previous work, [6], the sixteen chaotic rules were shown to belong to clusters of rules formed by the transformations, *complementation*, *negation* and *reflection* on any member rule of the cluster as depicted in figure 8.

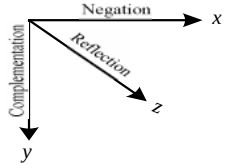


Figure 8- Basic transformations in Rule equivalence.

Suppose that the minterms of a generic rule ϕ_x are:

$$\{m_7, m_6, m_5, m_4, m_3, m_2, m_1, m_0\}$$

Then by reflection transformation of the minterms, $\phi_x \xrightarrow{reflection} \phi_{x_R}$ we obtain the rule ϕ_{x_R} with the minterms

$$\{m_7, m_3, m_5, m_1, m_6, m_2, m_4, m_0\}$$

according to the scheme depicted in figure 9:

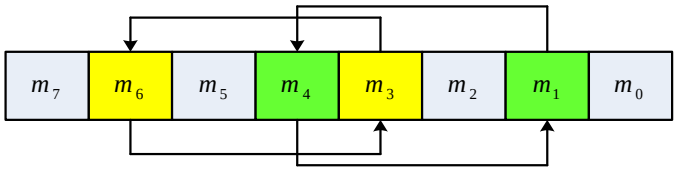


Figure 9-ECA Rule Reflection Scheme

Note that rule *reflection* transformation undergoes the exchange or swap of the unsymmetrical minterms, thus: $(m_{001} \rightarrow m_{100})_2$ or $(m_1 \rightarrow m_4)_{10}$ and $(m_{100} \rightarrow m_{001})_2$ or $(m_4 \rightarrow m_1)_{10}$ as well as $(m_{011} \rightarrow m_{110})_2$ or $(m_3 \rightarrow m_6)_{10}$ and $(m_{110} \rightarrow m_{110})_2$ or $(m_6 \rightarrow m_3)_{10}$ for the generic rule ϕ_x. Similarly, rule *negation* transformation $\phi_x \xrightarrow{negation} \phi_{x_N}$ results in forming ϕ_{x_N} with the minterms:

$$\{\overline{m}_0, \overline{m}_1, \overline{m}_2 \overline{m}_3, \overline{m}_4, \overline{m}_5, \overline{m}_6, \overline{m}_7\}$$

The two transformations *reflection* followed by *negation* $\phi_x \xrightarrow{reflection\ and\ negation} \phi_{x_{RN}}$ or the reverse order $\phi_x \xrightarrow{negationre\ and\ flection} \phi_{x_{NR}}$ results in forming $\phi_{x_{RN}}$ with the minterms:

$$\{\overline{m}_0, \overline{m}_4, \overline{m}_2 \overline{m}_6, \overline{m}_1, \overline{m}_5, \overline{m}_3, \overline{m}_7\}$$

These transformations are illustrated in table 3 for the generic rule ϕ_x generating rules ϕ_{x_R}, ϕ_{x_N}, and $\phi_{x_{RN}}$ noting that $\phi_{x_{RN}}$ and $\phi_{x_{NR}}$ are two representations of the same rule. Figure 10 shows the formation of the rule cluster of generic rule ϕ_x hypercube. Taking a specific example $\phi_{30} \xrightarrow{reflection} \phi_{86}$ is represented by the minterms $(00011110)_2$ change under rule *reflection* transformation to the following minterms $(01010110)_2$ which represents ϕ_{86}. Similarly, $\phi_{30} \xrightarrow{negation} \phi_{135}$ is represented by changing the minterms $(00011110)_2$ under rule *negation* transformation to the following minterms $(10000111)_2$ and for ϕ_{149} changing $(00011110)_2$ into $(10010101)_2$ under rule *negation* transformation followed by rule *reflection* transformation or in reverse order, as illustrated in table 4.

Table 3- Generic Rule equivalence minterms modification

rules	minterms							
	m_7	m_6	m_5	m_4	m_3	m_2	m_1	m_0
ϕ_{x_R}	m_7	m_3	m_5	m_1	m_6	m_2	m_4	m_0
ϕ_{x_N}	$\overline{m}_0$	$\overline{m}_1$	$\overline{m}_2$	$\overline{m}_3$	$\overline{m}_4$	$\overline{m}_5$	$\overline{m}_6$	$\overline{m}_7$
$\phi_{x_{RN}}$	$\overline{m}_0$	$\overline{m}_4$	$\overline{m}_2$	$\overline{m}_6$	$\overline{m}_1$	$\overline{m}_5$	$\overline{m}_3$	$\overline{m}_7$

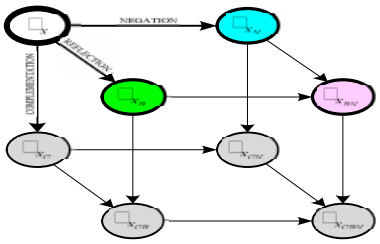


Figure 10- Illustration of Rule equivalence Hypercube of the generic rule cluster ϕ_{30}

Table 4- ϕ_{30} equivalence transformations minterms modification

rules	minterms							
	0	**0**	**0**	**1**	**1**	**1**	**1**	**0**
ϕ_{86}	0	1	0	1	0	1	1	0
ϕ_{135}	1	0	0	0	0	1	1	1
ϕ_{149}	1	0	0	1	0	1	0	1

The cluster rules that are useful for PRN sequence generation are those of rules ϕ_{30}, ϕ_{45}, ϕ_{60} and ϕ_{90} as well as the two rules ϕ_{105} and ϕ_{150}. However, the rules in the lower plane of the hypercubes of rules ϕ_{30} and ϕ_{45} as illustrated in figure 11 that are formed by the *complementation* transformation have been found to be inconsequential for the purposes of PRN sequence generation except for the *complementation* transformation of rule ϕ_{105}. Each of these two hypercubes of rules ϕ_{30}, and ϕ_{45} consists of eight rules while the hypercube of ϕ_{60}consists of four rules and the last two rules ϕ_{90} and ϕ_{105} consists of just two rules. By virtue of the structure of the minterms of rule ϕ_{60} only two transformations and their succession can produce different rules. The *complementation* transformation of rules ϕ_{60}, ϕ_{195}, ϕ_{102}, ϕ_{153} wrap around themselves. The last two rules ϕ_{90} and ϕ_{105} produce different rules each with just one transformation. The reduced number of rules in the rule cluster is referred to as *collapsed clusters*, [6].

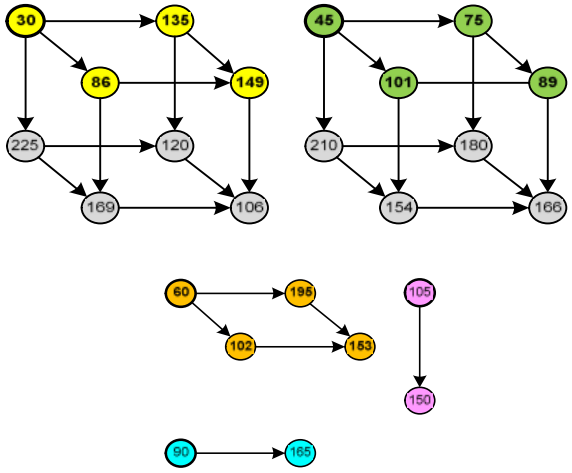


Figure 11- Hypercube representation of balanced chaotic rule clusters.

3. DYNAMICS OF ECA BALANCED CHAOTIC RULE EQUIVALENCE

The dynamics of the 16 balanced chaotic rules identified in the previous section will be observed by the phenomenological appearance of the spatio-temporal images of relatively short automaton span length and limited time evolution. For example table 5 displays the spatio-temporal images of the ECA running under the four chaotic rules of rule cluster ϕ_{30} for a span length $K = 61$ and time-steps evolution $T = 200$ for the same seed that consists of a single one on the extreme right hand cell and all the other cells are set at zero. In these images a black cell represents the state of the cell at "1" and a white cell when the cell state is "0". The similarity of the dynamical behavior is clear from inspection of the images. The mirror *reflection* transformation of ϕ_{30} is clear by inspecting the images of the two rules ϕ_{30} and ϕ_{86} or by inspecting the images of the two rules ϕ_{135} and ϕ_{149}. Likewise the *negation* transformation of rule ϕ_{30} is also apparent when inspecting the image of the two rules ϕ_{30} and ϕ_{135} or by inspecting the images of the two rules ϕ_{86} and ϕ_{149}. The image of rule ϕ_{149} is obtained by either *reflection* transformation followed by *negation* transformation or the reverse order on the image of rule ϕ_{30}. It can be seen that the appearance of the self similar fractals are the same in the images of all the four rules. It can be said that the four rules behave dynamically in the same manner starting from the same seed. Although the seed consists of a single one on the extreme right hand side the evolution of less than one hundred time steps yields a fairly complex image. It is also observable that the spreading of the patterns takes place at maximum speed which is usually referred to as the *speed of light* that corresponds to the $\pm 45^{o}$ diagonal spreading. Therefore starting from a single cell the whole lattice span $K = 61$ of the ECA fills up after $K/2$ time-steps. These four rules exhibit good speed of spreading amongst the 16 chaotic rules. Although the self similar or fractals of the cluster rule ϕ_{45} are different from the triangular like shapes of the self similar or fractals of cluster rule ϕ_{30} but the complexity of the dynamics are not less random. The three transformations on ϕ_{45} also behave similarly to the previous rule cluster ϕ_{30}. The images shown in table 6 illustrate the role of the transformations on ϕ_{45}. The rule equivalence of rule cluster ϕ_{60} for the same seed and number of evolution time steps likewise demonstrate the similarity of the dynamical behavior

of the rules ϕ_{60}, ϕ_{102}, ϕ_{195} and ϕ_{153}. However, the dynamics are not as random as exhibited by the rule equivalence of cluster rules ϕ_{30} and ϕ_{45}. Symmetrical behavior is quite prevalent in the dynamics, a feature that is obviously undesirable for pseudo random sequence generation. The same argument may apply to the rule equivalence of ϕ_{90} and ϕ_{105}. The symmetrical nature in the dynamics of rule ϕ_{90} and its reflection transformation ϕ_{165} highlight the possible linear relationship in the mathematical expressions of the two rules. The same also applies to rule ϕ_{105} and its complement transformation equivalent rule ϕ_{150}.

Table 5- Spatio-Temporal images of chaotic rule equivalence of ϕ_{30}

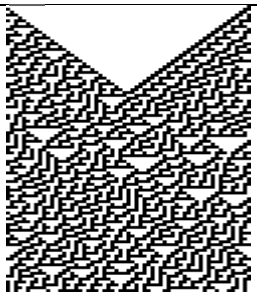
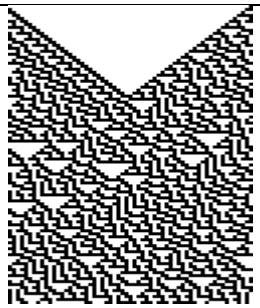
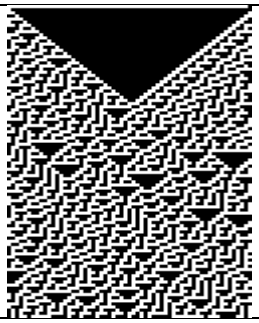
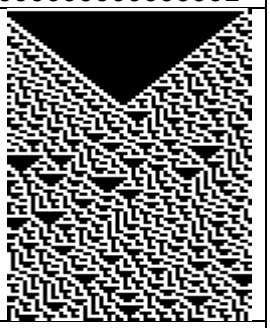
Chaotic Rule Equivalence of rule cluster ϕ_{30}			
K = 61, T = 100			
S^0 = 0000000000000000000000000000000000000000000000000000000000000001			
			
ϕ_{30}	ϕ_{86}	ϕ_{135}	ϕ_{149}

Table 6- Spatio-Temporal images of chaotic rule equivalence of ϕ_{45}

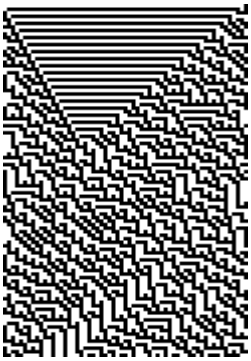
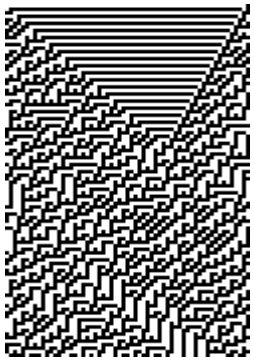
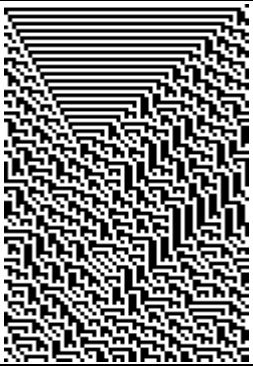
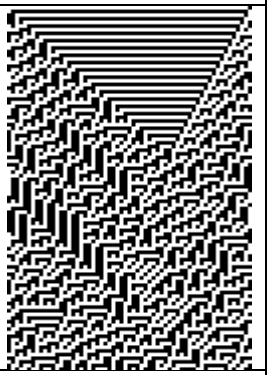
Chaotic Rule Equivalence of rule cluster ϕ_{45}			
K = 61, T = 100			
S^0 = 0000000000000000000000000000000000000000000000000000000000000001			
			
ϕ_{45}	ϕ_{101}	ϕ_{75}	ϕ_{89}

Table 7- Spatio-Temporal images of chaotic rule equivalence of ϕ_{60}

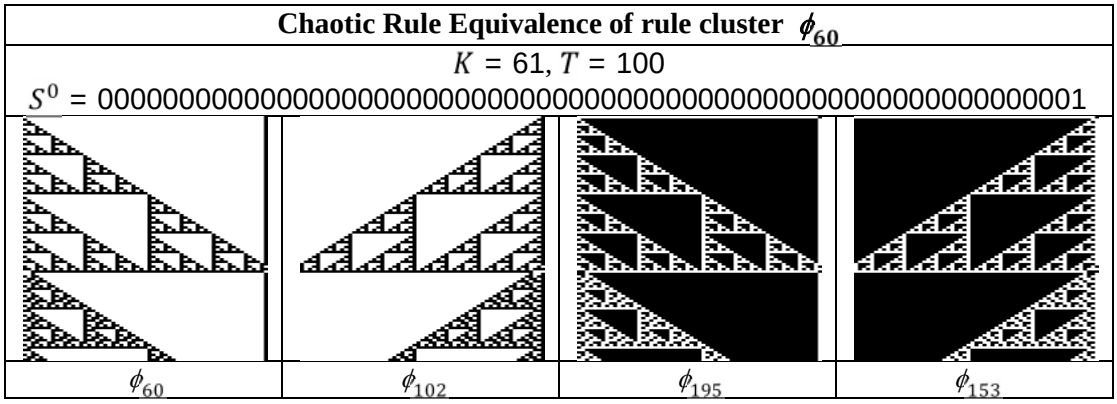
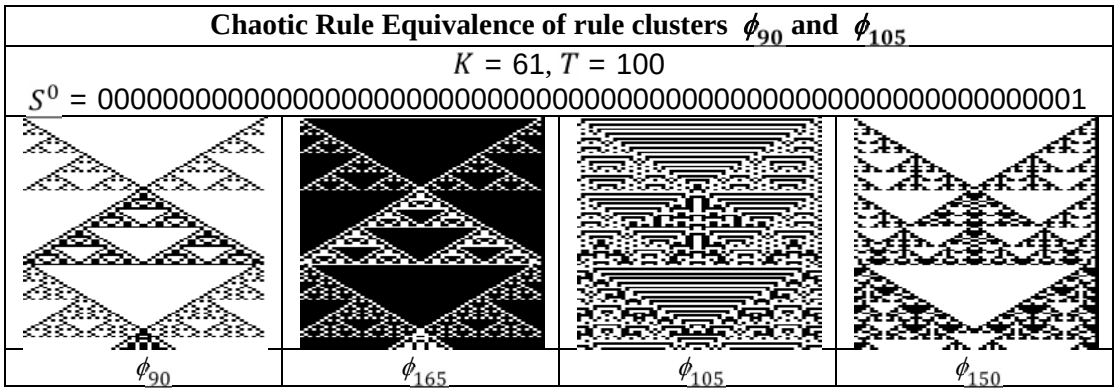


Table 8- Spatio-Temporal images of chaotic rule equivalence of ϕ_{90} and ϕ_{105}



4. DIEHARD TEST SUITE

As stated previously, the intended use of the ECA in this paper is for utilization as key generator in a synchronous stream cipher. The key of the stream cipher is obtained from the seed originally fed to the cellular automaton denoted by S^0. The ECA then evolves with time and produce the output stream that consists of the concatenation of $\{S^t\}^{t=T}$ conditioned by the output function φ. Let the output key generated by the ECA represented by $\Delta = \varphi(\{S^t\}^{t=T})$, then for this to be a viable key for a cryptographic strength stream cipher we must be able to assess the strength of the quality of its randomness. Fortunately, the Diehard test suite [?], has been available to use for free and has been regarded as the most stringent test suite as of the writing of this paper. The most recent version of this test suite consists of 22 individual tests as follows:

1. The BIRTHDAY SPACINGS TEST (9+**1**)
2. The tough BIRTHDAY SPACINGS TEST (1)
3. The GCD TEST-(Big Crush)
4. The GORILLA TEST -(Big Crush)
5. The OVERLAPPING 5-PERMUTATION TEST-(Big Crush)
6. The BINARY RANK TEST for 31x31 matrices (1)
7. The BINARY RANK TEST for 32x32 matrices (1)
8. The BINARY RANK TEST for 6x8 matrices (25+**1**)
9. The BITSTREAM TEST (20)

10. The OPSO TEST means Overlapping-Pairs-Sparse-Occupancy (23)
11. The OQSO TEST means Overlapping-Quadruples-Sparse-Occupancy (28)
12. The DNA TEST (31)
13. The COUNT-THE-1's TEST on a stream of bytes (1)
14. The COUNT-THE-1's TEST for specific bytes (25)
15. The PARKING LOT TEST (10+**1**)
16. The MINIMUM DISTANCE TEST (10+**1**)
17. The 3-D SPHERES TEST (20+**1**)
18. The SQUEEZE TEST (1)
19. The OVERLAPPING SUMS TEST (10+**1**)
20. The UP-DOWN RUNS TEST (3)
21. The CRAPS TEST (2)
22. the CRAPS TEST 2 with different dice (2)

The tests vary in the number of *p*-values they yield, as described above. The Diehard test suite requires that the output stream to be tested must be converted to a binary form that is suitable for the test suite. It also requires that the size of the unconverted data must exceed 80 mega bits for 19 of the 22 tests, hence the binary file produced must be greater or equal to 10 Mbytes. The other three tests, 3, 4, and 5 require much larger data size, namely exceeding 2.7 Giga bits. These three tests are dubbed as *Big Crush* [22], a name borrowed to indicate the difficulty in passing the test. The other 19 tests likewise can be dubbed as S*mall Crush*. When Diehard runs the small crush it produces 230 *p*-values. Some of the *p*-values are produced by the Kolmogorov-Smirnov test, as indicated by a plus sign in the above list. There is also an overall Kolmogorov-Smirnov test *p*-value which can be considered as an indication of Pass/Fail criteria, [7-9]. This paper reports the findings of running the Small Crush test while the Big Crush test will be the subject matter for future publication.

5. ECA RULE SPACE DYNAMICS ANALYSIS

Following the discussion of section II above, the number of rules that need to be tested is reduced to the chaotic rules sub-space $\zeta\subset\varsigma,\varsigma\subset$ that consists of the 16 chaotic rules:

$$\phi_{30},\phi_{45},\phi_{60},\phi_{75},\phi_{86},\phi_{89},\phi_{90},\phi_{101},\phi_{102},\phi_{105},\phi_{135},\phi_{149},\phi_{150},\phi_{153},\phi_{165},\phi_{195}$$

These rules belong to the chaotic rules of class III where the λ parameter =0.5, [5]. All the other rules that belong to the other three classes stand no chance in the testing simply because either the global dynamics collapse to a very small attraction cycle (named so since any transient of states reaching this cycle will be trapped in this cycle for eternity), this case applies to the two classes I and II, or the global dynamics are rich with repeated patterns, as is the case with class IV. This paper is concerned with testing the whole data generated by the ECA running under one chaotic rule in contrast to the approach adopted by [3], and explores the ability of the data collected to be utilized as strong cryptographic data. Since the Diehard test suite for good reasons is the test suite of choice, it is thus required to collect contiguous concatenated output sequences of size exceeding 80 Mbits without modifications or decimation. The output function φ is constructed to output the data stream as illustrated in figure 12.

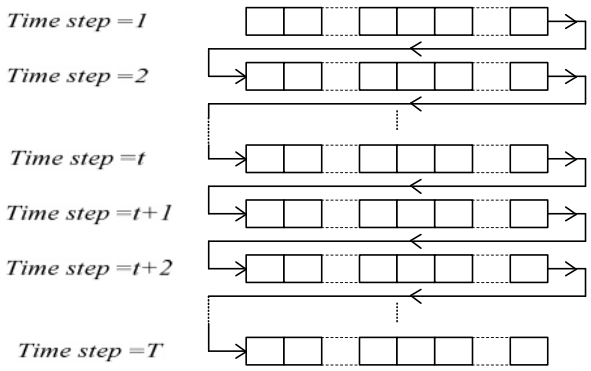


Figure 12 - Concatenation of the ECA output data stream by the output function φ.

It should be helpful if one could find an easy approach to qualify a sequence of pseudo random numbers for testing prior to applying a suite of tests. It can be stated that one of the easiest tests is the uniform distribution of 0's and 1's in the binary sequence under test which is attributed to [10]. However, before attempting to find this distribution the length of the sequence or at least its lower bound must be determined. Hence the data in the sequence must not repeat implying that the period of the pseudo random sequence must be longer than the length of the sequence in use. The LFSR can meet this requirement quite easily at the cost of low linear complexity. In comparison the ECA can provide superior linear complexity but the length of the period is much less than that provided by the LFSR. Hence the need to accommodate long periods is essential when utilizing ECA in pseudo random number generation. The period P of the LFSR in its maximum sequence mode is computed linearly and is represented by $P_{LFSR} = 2^L$-1, where $L \in N$ is the span length of the LFSR. On the other hand the maximum period P_{ECA} attainable from ECA is not an easy task to compute particularly when non-linear chaotic rules are used and does not seem to follow any analytically discernible trend. One clear fact is that for the same span length $P_{LFSR} > P_{ECA}$. Information about the periods of the ECA amongst a host of other features during its time evolution are obtainable from study of the global dynamics of the ECA for arbitrary local transition rule and span length. For example, figure 13 depicts the global dynamics and gives the two state diagrams of the ECA running uniformly in a periodic boundary configurations under ϕ_{30} for $K = 5$. Two cycles of attraction can be identified the maximum is of size 5-states and the minimum of size 1-state. The minimum attraction cycle is only reachable by the state $31_{10} = 11111_2$, while the maximum attraction cycle is reachable by the five states 3_{10}, 6_{10}, 12_{10}, 17_{10}, and 24_{10}. All these six states, coloured green, are usually referred to as Garden Of Eden (GOE) states signifying that they are unreachable which means that they have no pre-images or predecessors. Additionally, the minimum attraction cycle has a transient trajectory of just one state; it is the GOD state 31_{10}, which is typical with ϕ_{30}, while the maximum attraction cycle can be reached by any state in any one of the five transient branches. One can easily infer that the maximum number of states to be traversed before the maximum attraction cycle begins to repeat is 5+5=10 states while the minimum attraction cycle requires just one state before the ECA is locked in the attraction cycle. Therefore the best scenario for this case of the ECA running uniformly under rule ϕ_{30} for $K = 5$ is to limit the seeds to a repertoire of five GOE states excluding the GOE state 31_{10}. In comparison, the LFSR of span length 5 in a maximum cycle feedback configuration can be seeded with any state from the state space of 2^5-1, i.e. all the span length permutations excluding the all 0's state, and runs for 2^5-1 = 31states before repeating.

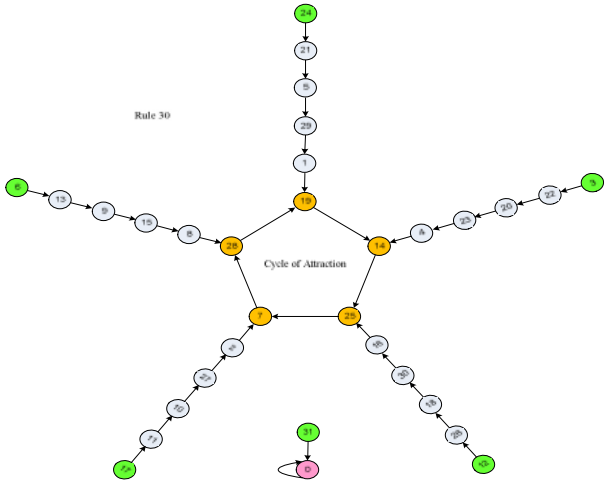


Figure 13- State Diagram of a periodic ECA running uniformly under rule ϕ_{30} for $K = 5$.

We can arguably state that we are interested in the major trajectory from a GOE state, let its length be denoted by α, usually referred to as a transient, to the major attraction cycle, let the length of this cycle be denoted by β, as well as the minimum trajectory from a GOE state to the minimum attraction cycle. Obviously, the best scenario here is to form a seed repertoire that consists of a collection of GOE states that has the major transient to the major attraction cycle. Let the total number of states traversed before the major cycle repeats be denoted by ζ , giving $\zeta = \alpha + \beta$, then for viable cryptographic applications the output string of data generated by the ECA, denoted byΔ, must not exceed ζ, i.e. $\zeta \geq \Delta$. For large data requirements the ECA must run with a sufficiently large span length K, such that the above lower bounds can be satisfied. Let G represent the total number of the Garden of Eden states available for an arbitrary chaotic rule ϕ_x, then $g \subset G$ represents the proper subset of the Garden of Eden states that lead to the largest attraction cycle or cycles because it excludes those Garden of Eden states that transit to shorter cycle or cycles including such states as all 0's or all 1's for both even and odd span lengths and the two states 1010… or 0101… for the even span lengths when using ϕ_{30}. This subset should be the set available for use as seeds in any encryption/decryption setting that make use of the whole ECA output to the stream cipher. This subset of states will be able to guarantee the orbit of the longest transient evolution but at the cost of limited number of seeds, a case that can be considered restrictive and inefficient for the utilization of cellular automata in encryption/decryption applications.

The advantage of the LFSR cannot be overemphasized with the exception of computational complexity where the ECA inherently gains the upper hand compared to the LFSR. All the necessary information, such as the primitive polynomial and the feedback taps about the LFSR can be derived from a collection of $2 \cdot K$ contiguous bits from the output data stream, [**], the ECA minimum complexity on the other hand asymptotically approaches $O(\xi \cdot 2^K \cdot T)$, where ξ represents the cardinal of the rule sub-space applicable to the periodic configuration. Several authors have researched the global dynamics of the ECA rule space, [4,6,11], most prominently was that due to [6] that culminated in the publication of the atlas of global dynamics of cellular automat. The main goal of all these attempts is to give a detailed account of the state transition graphs or diagrams or as it is referred to by [6], the field of basin of attraction of the ECA rule

space. The information that can be extracted from such data is of particular pertinence when trying to study the viability of cellular automata in generating cryptographic pseudo random number sequences. The results published in the above referenced atlas, however, gave details of the state transition diagrams for $K \leq 15$ possibly due to the prohibitive computational expense in the attempt beyond $K = 15$ because the ECA has to run for the entire state space 2^K, but of course larger sizes of K can lead to better understanding of the behavior of the ECA and can also justify the viability of the use of ECA in pseudo random number generation. Excerpts of the data from the atlas of global dynamics [6] for ϕ_{30} is tabulated and presented in the following table 9. It can be seen that the periods of the maximum attraction cycles are inferior to the maximum cycle lengths of the LFSR and do not follow a linear trend. In addition the LFSR can be seeded with all states 2^K except the all 0's state while the ECA in the uniform periodic setting should be seeded with those states that lead to the maximum attraction cycle in order to maximize the period of the output sequence. Such conditions add severe restrictions to the collection of the seeds available for the applications sought in this paper. For example the total number of states available for seeding a uniform and periodic ECA running under ϕ_{30} when $K = 11$ is 1551 which is the addition of the maximum attraction cycle length and all the transients leading to the cycle as compared to 2047 for the LFSR for the same span length. Therefore there are two main factors that influence the required data size; the cellular automaton span length $K \in N$ and the number of steps of the evolution time $T \in Z$. Let Δ denotes the output data size, then it is required that the ECA under test should provide a data size of $\Delta \geq K \times T$. It can be seen that the number of time steps T follows the assignment of the span length K. The ECA selected chaotic rule space was subjected to run on cellular automaton span length range $27 \leq K \leq 1503$. The minimum span length $K = 27$ was selected based on the possible cycle length of the ECA matching the maximum cycle length of the Linear Feedback Shift Register, albeit the ECA maximum cycle length cannot reach the maximum cycle length of the same span length of the LFSR as will be explained shortly. For $\Delta = 80$ Mbits the span length must satisfy $K \geq log\Delta/log2$. As explained previously, the ECA is a synchronous sequential circuit that evolves according to the local transition rule $\phi_x, x \in 2^{2^\eta}$ therefore for each ϕ_x a group of state transition diagrams must exist that covers all the states 2^K for a span length $K \in N$. It can be seen that a detailed compilation of such state diagrams for the range of K used in this paper is presently computationally too exhaustive.

The ECA was run for each span length in the range $27 \leq K \leq 1503$ for each of the 16 chaotic rules mentioned above. Each output of size ≈ 80 M bits was tested individually by the Diehard test suite to ensure the inclusion of all the 19 individual tests mentioned above. The results are shown in the figures 14-19 and the following findings can be derived:

1. Cluster ϕ_{30} seems to outperform the other 12 rules.
2. Cluster ϕ_{45} is the second best.
3. Cluster ϕ_{60}, Rules 90 and 165 and Rules 105 and 150 seem to perform worst with low passing rates compared to the two non-linear cluster rules 30 and 45.

The results of the four rules in cluster ϕ_{30} seem to be almost identical and the same apply to the other clusters therefore it should suffice to show the results of the leading rules in the clusters. In depth study of the results of ϕ_{30} shows that the range of results undergoes many regions. The first region covers seed length range $27 \leq K \leq 73$ where the ECA failed to pass all the *p*-values. However at $K = 73$ it passed 228 p-values and failed one *p*-value as well as the KS overall p-

value. In our criterion this is equivalent to failing the tests. In this way the first seed where the ECA passed all the 230 *p*-values happened at $K = 109$. If one considers the suitability of the ECA for pseudo random generation based on passing all the *p*-values produced by the Diehard test suite then the region just mentioned $27 \leq K \leq 108$ can be considered unsuitable for pseudo random number generation when running uniformly under ϕ_{30} in the periodic configuration. One important observation is the frequent dip in the number of *p*-values passed at 32-bit intervals. Starting from $K = 30$ with 32 number of p-values passed with a neighbour span lengths passing more *p*-values, a clear dip in the number of *p*-values passed takes place at $K = 62$ where the *p*-values passed by the neighbour span lengths are 123 and 92. It repeats after another 32-bit increment, i.e. at $K = 94$ and $K = 126$ and so on. The number of *p*-values passed increase after the dip at $K = 287$with the number of *p*-values passed equal 180, the next dip takes place at $K = 319$ with 210 number of p-values passed while at 2-bit neighbouring seeds the ECA fails around just 2 *p*-values. Similar patterns but more condensed also observable with rule cluster 45. In this cluster the dip happens more frequently and at span length differences of 4, 8, 16 and 32bits. The increase of the rate of failures is quite apparent. The intervals of failures with the other rules also take place at similar span length differences as with rule cluster 45. There is no clear explanation for the reason why such failures happen at the differences in span lengths described above. However, since the numbers are powers of two and that the Diehard test suite is in fact based on testing binary numbers of 32-bit wide, it may have some sensitivity to the results obtained. Further research is obviously warranted for this problem. By inspecting the results of the rest of the chaotic rules it seems that none of the other rules behave better than ϕ_{30}. From the above results one can state that the data presented in table 9 may lead to the contention that running the ECA with a random seed may probabilistically result in the failure of the output data stream in passing the test suite. This should be clear from the results presented in this paper.

Table 9. State Diagrams Data for ϕ_{30} for span length range $3 \leq K \leq 14$.

Rule 30						**LFSR** (span length L)
ECA span length K	#of attractors producing		Maximum Attractor period	Total # of states in the maximum attractor state diagrams	Total # of GOE states	Maximum cycle length $2^L - 1$
	Less than maximum period	Maximum period				
3	-	1	-	-	3	7
4	3	1	8	12	5	15
5	1	1	5	30	6	31
6	-	3	-	-	12	63
7	1	8	63	77	10	127
8	4	1	40	224	33	255
9	1	2	171	414	57	511
10	4	2	15	420	56	1023
11	2	1	154	1551	136	2047
12	8	4	102	975	91	4095
13	4	1	832	2600	964	8191
14	17	1	1428	13818	1478	16383

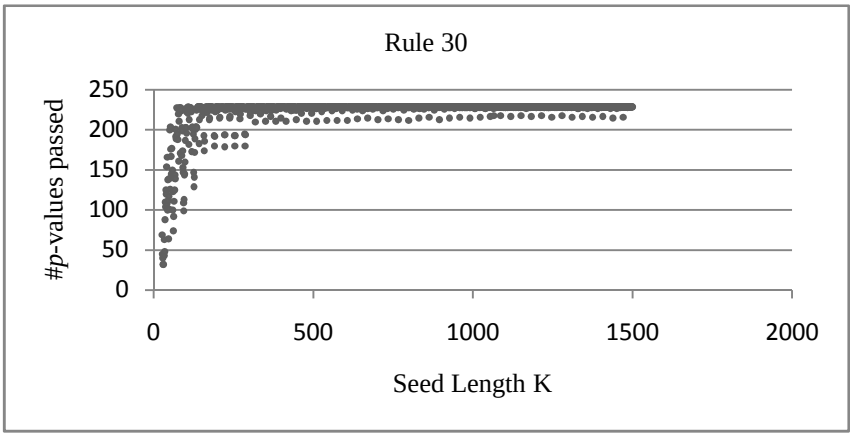


Figure 14 – Diehard Results of ECA running uniformly under ϕ_{30} for span length range $27 \leq K \leq 1503$

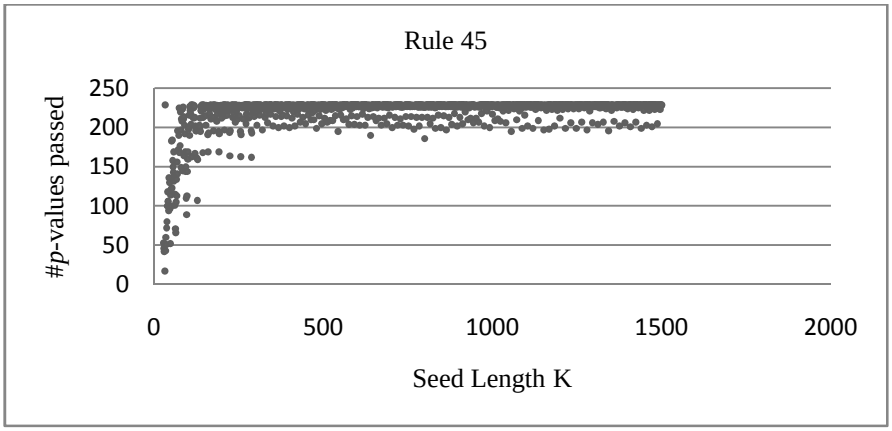


Figure 15 – Diehard Results of ECA running uniformly under ϕ_{45} for span length range $27 \leq K \leq 1503$

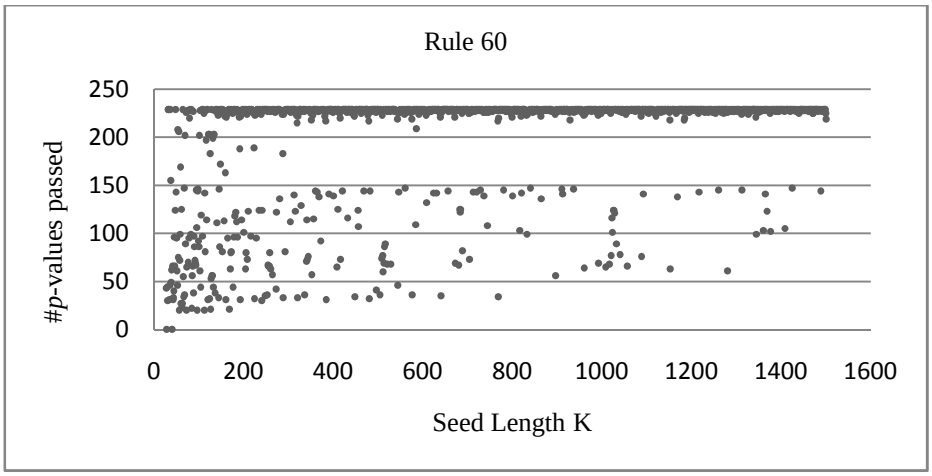


Figure 16 – Diehard Results of ECA running uniformly under ϕ_{60} for span length range $27 \leq K \leq 1503$

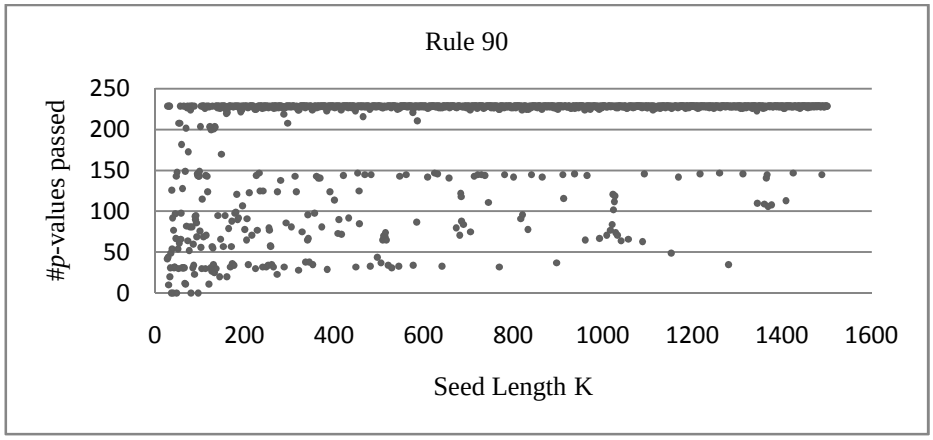


Figure 17 – Diehard Results of ECA running uniformly under ϕ_{90} for span length range $27 \leq K \leq 1503$

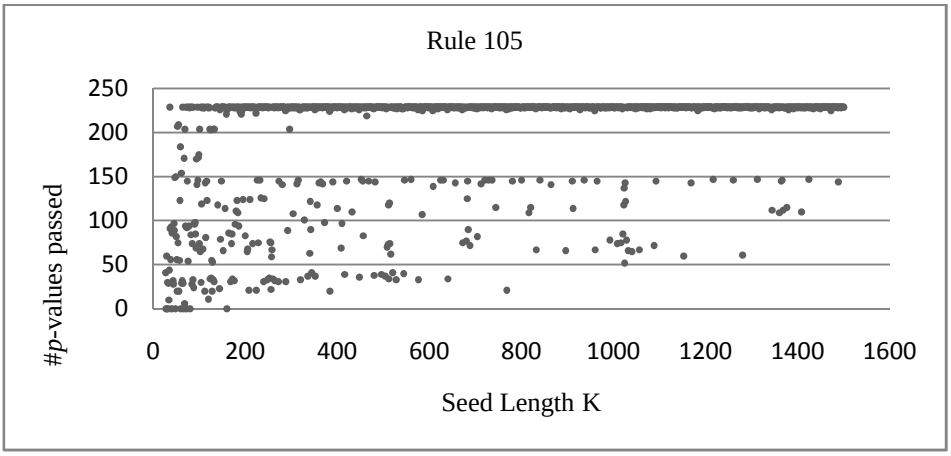


Figure 18 – Diehard Results of ECA running uniformly under ϕ_{105} for span length range $27 \leq K \leq 1503$

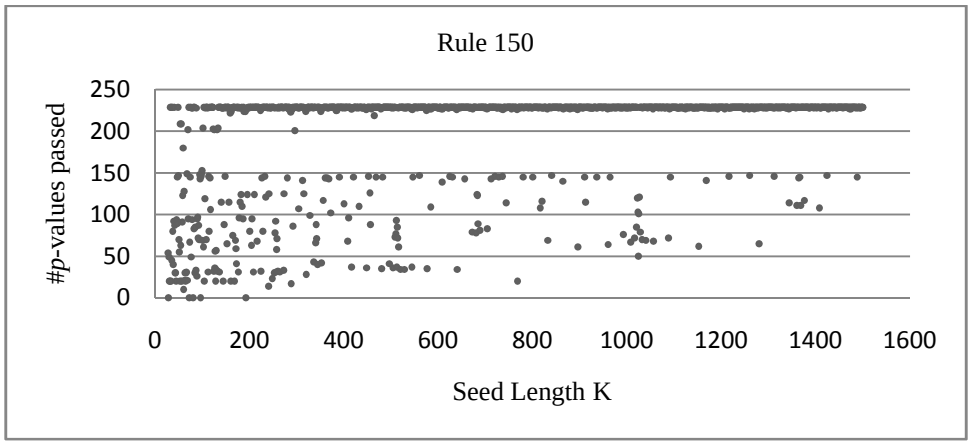


Figure 19 – Diehard Results of ECA running uniformly under ϕ_{150} for span length range $27 \leq K \leq 1503$

6. CONCLUSIONS

It has been shown that if passing all the *p*-values generated by the Diehard test suite is considered a good criterion to qualify a sequence of pseudo random numbers as being cryptographically strong then only a sub-space $\zeta \subset \Re$ of the ECA rule space $\Re$ can be considered useful for this application. The ECA under test is designed to run uniformly on one local transition rule in a periodic boundary configuration. The ECA was run for a wide range of span lengths $27 \leq K \leq 1503$. A random seed was used for each span length and the same seed was used by all the 16 rules in the sub-space $\zeta \subset \Re$. In the case that the ECA fails any of the 230 *p*-values by the Diehard test suite, 9 more random seeds are generated and the ECA is re-run on all these seeds to create a new average of the *p*-values passed and then recorded. The results evidently show that cluster ϕ_{30} outperforms all the other rules followed by cluster $_{45}$. Although the linear rules sub-space $\zeta_{Linear} \subset \zeta$ can pass all the Diehard 230 *p*-values, it is clear that in addition to the fact that the passing rate is visibly lower than the results obtained from the rules of the non-linear sub-space $\zeta_{non-Linear} \subset \zeta$ the dependencies of these rules on the linear primitives operations makes them inherently amenable to cracking and render them weak in cryptographic applications. The results also show that all the rules exhibit some periodic patterns of seed length differences of 32, 16, 8 and 4, or in terms of powers of two. $\{2\}_2^5$. Since the common denominator is a power of two up to power of 5, it has little to do with the span lengths since the failures take place at odd numbers of the span length K as well as even numbers in addition to prime numbers. One of the possible causes of this trend may be attributed to the tests in the Diehard test suite itself. The Diehard deals with numbers of 32-bit wide and therefore some sensitivity to this number may propagate to the results and possible inclusion of divisible numbers as well. In conclusion it can be deduced for recommendation that when the ECA is intended for use to generate pseudo random numbers of high or cryptographic quality based on the criterion stated above, namely that the whole data output is used as key number generator for the synchronous stream and that the output must pass all the 230 *p*-values generated by the Diehard test suite, then the best choice would be cluster rule 30, i.e. any one of the four rules $\phi_{30}, \phi_{86}, \phi_{135}, \phi_{149}$, running with a repertoire of seeds collected as discussed in the body of this paper. Any seed used outside this carefully compiled repertoire would leave the output data to the probabilistic failure. A recommendation for future work would be to construct an efficient algorithm that allows the computation in polynomial time of the state diagrams of the useful rules for the range of span lengths $K \epsilon N$. The results will facilitate the compilation of a reliable repertoire of a set of seeds to be used for the applications in question.

REFERENCES

[1] Menezes, J. Alfred, van Oorschot, C. Paul, & Vanstone A. Scott (1996), "Handbook of Applied Cryptography", Editors: Kenneth H. Rosen, CRC Press, ISBN: 0-8493-8523-7, Fifth Printing Edition, 816 pages.

[2] Berlekamp, E.R. (1968), "Algebraic Coding Theory", McGraw-Hill, New York, chapter 7.

[3] Wolfram, Stephen (1986), "Random Sequence Generation by Cellular Automata", Advances In Applied Mathematics 7, pp 123-169.

[4] Wolfram, S. (2002), "A New Kind of Science", Champaign, IL: Wolfram Media, ISBN: 1579550088.

[5] Langton, Chris G. (1990), "Computation at the edge of chaos: Phase transitions and emergent computation", Physica D, 42:12-37.

[6] Wuensche Andrew, and Mike Lesser (1992), "The Global Dynamics of Cellular Automata", Reference Volume I, Addison Wesley Publishing Company, ISBN: 0-201-55740-1.

[7] K. Salman (2013), "Elementary Cellular Automata (ECA) Research platform", Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Software Engineering (JSSE), Volume 3, Issue 6, June Edition, 2013, pp 1-15.

[8] K. Salman (2013), "Feedback Shift Registers As Cellular Automata Boundary Conditions", First International Conference on Computational Science and Engineering (CSE-2013), pp 11-21, http://airccj.org/CSCP/vol3/csit3302.pdf

[9] K. Salman (August 2013), "Analysis Of Elementary Cellular Automata Boundary Conditions", International Journal of Computer Science & Information Technology (IJCSIT) Vol. 5, No 4, pp. 35-51

[10] S.W. Golomb (1986), "Shift Register Sequences", Aegean Park Press; Revised edition, ISBN-10: 0894120484.

[11] Edward Jack Powley, (2009), "Global Properties of Cellular Automata", Ph.D., Department of Computer Science, University of York.