# CARPenter: A Cellular Automata Based Resilient Pentavalent Stream Cipher

Rohit Lakra, Anita John, and Jimmy Jose(✉)

Department of Computer Science and Engineering,
National Institute of Technology Calicut, Kozhikode, India
{rohit_m160263cs,anita_p170007cs,jimmy}@nitc.ac.in

**Abstract.** Cellular Automata (CA) are a self reproducing model widely accepted for their applications in pattern recognition, VLSI design, error correcting codes, cryptography etc. They have also been widely accepted as good random number generators. The pseudorandom properties of 3- and 4-neighbourhood CA have been studied and they show that the neighbourhood radii has an impact on pseudorandomness. This motivated us to perform the exploration of 5-neighbourhood 1-dimensional CA for better cryptographic properties. We construct a class of linear and nonlinear rules for 5-neighbourhood CA and also propose a new stream cipher design using 5-neighbourhood CA inspired from the Grain cipher.

**Keywords:** Cellular Automata (CA) · 3-neighbourhood CA
5-neighbourhood CA · Cryptography · Stream Cipher

## 1 Introduction

In cryptography, the encryption techniques can be classified as symmetric key encryption and asymmetric key encryption. Symmetric key encryption encrypts plaintext into ciphertext using a common key shared between the sender and the receiver. This encryption can be done either on blocks of plaintext or one bit at a time. A block cipher encrypts a fixed size of $n$-bits block of data at a time. A stream cipher encrypts 1 bit or byte of data at a time. It normally uses a long stream of pseudorandom bits as the key. In order to implement a secure stream cipher, its pseudorandom generator should be unpredictable and the reuse of key should never happen. Stream ciphers are faster and have a lower hardware complexity than block ciphers. They are also appropriate when buffering is limited. The eSTREAM project [1] which was started as part of ECRYPT [2] aimed to promote the design of efficient stream ciphers. The finalists in eSTREAM were classified under two profiles namely, profile-1 and profile-2.

The ciphers in profile-1 were intended to give excellent throughput when implemented in software whereas the ciphers in profile-2 were intended to be efficient in terms of the physical resources required when implemented in hardware. Two widely studied ciphers Grain [3] and Trivium [4] belong to profile-2.


G. Mauri et al. (Eds.): ACRI 2018, LNCS 11115, pp. 352–363, 2018.
https://doi.org/10.1007/978-3-319-99813-8_32

Recent studies and research in the field of stream ciphers and CA have shown the use of CA as a better cryptographic primitive. Parallel transformations of stream cipher can be achieved using CA and this provides high throughput which is beneficial in the case of stream ciphers. Work done in [5–9] clearly discusses the cryptographic suitability of CA as stream ciphers. They also give some light to the fact that as the neighborhood of CA increases, the cryptographic properties of the cipher also increases if proper CA rules are employed but with the cost of time needed for doing the computation. FresCA [7] and Cavium [5] were the designs that applied CA in the eSTREAM finalists GRAIN and TRIVIUM respectively.

CA based stream ciphers CASTREAM [10] and FResCA were proposed in ACRI 2012 and ACRI 2016 respectively. Here, we propose CARPenter as a stream cipher based on 5-neighbourhood CA. This paper is organized as follows. Section 2 discusses the terminologies and basics of CA. Section 3 gives a literature survey on CA based stream ciphers. Section 4 discusses 5-neighbourhood CA and the linear and nonlinear rules associated with it. Description of the proposed stream cipher design is provided in Sect. 5. The cryptographic suitability of the new design is discussed in the last section.

## 2 Preliminaries

### 2.1 Cellular Automata

A cellular automaton is a collection of cells and each cell is capable of storing a value and a next-state computation function which is also called CA rule. Rules determine the behaviour of a cellular automata [11]. The state of each cell of a CA together at any instant $t$ defines the global state of the CA. The next state of the $i^{th}$ cell of a 3-neighbourhood CA at any instance $t$ is given by

$$S_i^{t+1} = f(S_{i-1}^t, S_i^t, S_{i+1}^t).$$

The next state of $i^{th}$ cell of a 5-neighbourhood CA is given by

$$S_i^{t+1} = f(S_{i-2}^t, S_{i-1}^t, S_i^t, S_{i+1}^t, S_{i+2}^t).$$

where $f$ is the next state function or rule, $S_i^{t+1}$ denotes the next state of the $i^{th}$ cell, $S_{i-2}^t$ is the current state of second left neighbour, $S_{i-1}^t$ is the current state of first left neighbour, $S_i^t$ is the current state of the cell to be updated, $S_{i+1}^t$ is the current state of first right neighbour, $S_{i+2}^t$ is the current state of second right neighbour. In general, the number of cells $n$ that participate in a CA cell update is given by $n = 2a+1$ where $a$ is the radius of the neighbourhood [11].

Cellular automata with null boundary is the one in which the left neighbour of the leftmost cell and the right neighbour of the rightmost cell are zero [12]. Hybrid CA is a cellular automata where more than one rule is involved in the generation of next state [12]. If a cellular automata of $n$ bits (where n is an integer) evolves $2^n - 1$ different states before getting back to the initial state, then it is called as maximum length CA.

There are 256 ($2^{2^3}$) and 4294967296 ($2^{2^5}$) such Boolean functions or rules possible for 3-neighbourhood CA and 5-neighbourhood CA respectively. Rules are named as decimal equivalent of the binary number that is formed by applying that rule to all $2^n$ possibilities of the neighbourhood of a $n$-neighbourhood CA. Last combination with all ones becomes the most significant bit and first combination with all zeros becomes the least significant bit.

### 2.2 Cryptographic Properties of Boolean Functions

Cryptographically suitable Boolean functions should satisfy certain properties. Some important cryptographic properties are discussed below. A detailed description of cryptographic properties can be found in [13]. Some basic definitions are provided to better understand some of the cryptographic properties.

*Affine Function:* A Boolean function which can be expressed as the XOR of some or all of its inputs and a Boolean constant is called Affine function.

*Hamming Weight:* The number of 1's in the truth table representation of a Boolean function is called its Hamming weight.

*Hamming Distance:* Hamming distance between two given functions is the Hamming weight of the XOR of the two functions.

**Balancedness.** The balanced Boolean functions have equal number of zeros and ones in their truth table. Balancedness should be satisfied by all the Boolean functions used in cryptographic applications. There is a statistical bias present in unbalanced Boolean functions which can be exploited by differential and linear cryptanalysis.

**Algebraic Degree.** Algebraic degree is the maximum number of variables present in an AND term among all the AND terms of a given Boolean function. Higher algebraic degree is necessary in order to have high linear complexity.

**Nonlinearity.** Nonlinearity of a Boolean function is given as the minimum Hamming distance of the given Boolean function to all the affine functions.

**Correlation Immunity.** A Boolean function is $k^{th}$ order correlation immune if the output of the given Boolean function is independent of atmost $k$ input variables.

**Resiliency.** A Boolean function which is both balanced and $k^{th}$ order correlation immune is called $k$-resilient. If a Boolean function is not $k$-resilient, then the output depends on at most $k$ input variables which can be exploited to recover the initial state of $k$ inputs.

## 3 Literature Survey on CA Based Stream Ciphers

CA have a natural tendency to resist fault attacks [9]. CASTREAM [10], CAR30 [6], CAvium [5] and FResCA [7] are some of the CA based stream ciphers. CASTREAM is a CA based stream cipher suitable for both hardware and software. It makes the nonlinearisation faster. In CASTREAM, each state bit is influenced by all key bits and IV bits after six iterations. CAR30 is a stream cipher based on CA Rule 30 and a maximum length linear hybrid CA with rule 90 and 150. It is efficient for both hardware and software and its generic design leads to its scaling up to any length of key and IV. This cipher is found to be faster than both Grain and Trivium. CAvium design is a modification of Trivium using CA which increases its strength against almost all the attacks against its reduced rounds. The design has faster startup as it has reduced the number of rounds from 1152 to 144 in the initialization phase and hence needs less clock cycles. It is more secure and faster than Trivium at the cost of more computations per iteration. FResCA (Fault Resistant Cellular Automata Based Stream Cipher) is a modification of Grain, another eSTREAM finalist. This is a 4-neighbourhood CA based Grain-like cipher whose initialization is 8 times faster than Grain since there are only 32 iterations in the initialization phase of FresCA whereas Grain has 256 iterations. FResCA eliminates fault attacks possible in Grain cipher and is also resistant to many other different attacks. Its cells are updated using linear and nonlinear rules and its output depends upon a nonlinear mixing function called NMix [14].

## 4 Five-Neighbourhood CA

In most of the applications, 1-dimensional 3-neighbourhood Cellular Automata are used. In [8], 4-neighbourhood nonlinear CA were studied and shown to provide good randomness and less correlation. In [11], Catell and Muzio have given a method to synthesize a 3-neighbourhood Linear Hybrid CA. Based on [11], Maiti and Roy Chowdhury in [15] have given an algorithm to synthesize 5-neighbourhood null boundary Linear Hybrid CA (LHCA) using two linear rules. The randomness and diffusion properties of 3-, 4- and 5-neighbourhood were studied and it was shown that the CA can be improved with increase in size of neighbourhood radius of the CA cell if appropriate CA rules are used. The diffusion rate of 5-neighbourhood CA is high and hence is found suitable for high speed application. The improvement comes at a cost of increased computation.

### 4.1 Five Neighborhood Linear Rules

Based on [15], we have found a 128-bit 5-neighbourhood Linear Hybrid CA rule vector. Out of the $2^{2^5}$ possible 5-neighbourhood rules, only $2^5 = 32$ rules are linear. Out of these 32 rules, only $2^3 = 8$ are of exactly 5-neighbourhood [15]. The combination of rule R0 and rule R1 given below gives the largest number of rule vectors (8) for 5-bit maximum length 5-neighbourhood CA [15]. Hence,

these two rules are considered for finding 128-bit 5-neighbourhood maximum length CA. These two rules are given as

$$\text{R0} : S_i^{t+1}=S_{i-2}^t \oplus S_{i-1}^t \oplus S_{i+1}^t \oplus S_{i+2}^t$$
$$\text{R1} : S_i^{t+1}=S_{i-2}^t \oplus S_{i-1}^t \oplus S_i^t \oplus S_{i+1}^t \oplus S_{i+2}^t$$

R0 and R1 are in resemblance to linear rules 90 and 150 respectively of the 3-neighbourhood CA. Rules 90 and 150 are used in [11] to synthesize a maximum length 3-neighbourhood hybrid CA. The state transition function of the $i^{th}$ cell of 5-neighbourhood CA using the rules R0 and R1 can be expressed as

$$S_i^{t+1} = S_{i-2}^t \oplus S_{i-1}^t \oplus d_i.S_i^t \oplus S_{i+1}^t \oplus S_{i+2}^t$$

where $d_i = 0$ if R0 is used and $d_i = 1$ if rule R1 is used [15].

An $n$-cell 5-neighbourhood CA can be represented as a combination of these two rules as an $n$-tuple $[d_1, d_2, ..., d_n]$ called as rule vector. A 5-neighbourhood CA is represented by a characteristic matrix over GF(2) and the characteristic matrix has a characteristic polynomial [15]. A characteristic polynomial is a degree $n$ polynomial, where $n$ is the length of rule vector of CA. A CA is maximum length if and only if its characteristic polynomial is primitive [16]. Theorem 1 [15] has been used to derive the characteristic polynomial of CA.

**Theorem 1:** Let $\triangle_n$ be the characteristic polynomial of a n-cell null boundary 5-Neighbourhood CA with rule vector $[d_1, d_2, \ldots, d_n]$. $\triangle_n$ satisfies the following relation

$$\triangle_n = (\mathrm{x} + d_n)\triangle_{n-1} + \triangle_{n-2} + (\mathrm{x} + d_{n-1})\triangle_{n-3} + \triangle_{n-4}, \mathrm{n} > 0$$

Initially $\triangle_{-3} = 0$, $\triangle_{-2} = 0$, $\triangle_{-1} = 0$, $\triangle_0 = 1$.

Theorem 1 provides an efficient algorithm to compute the characteristic polynomial of a CA. We found a 128-bit maximum length null boundary CA rule vector [0, 0, ..., 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0] and its primitive characteristic polynomial (CP) is

$$\begin{aligned}\text{CP} = {} & x^{128} + x^{127} + x^{125} + x^{122} + x^{120} + x^{119} + x^{117} + x^{115} + x^{113} + x^{112} + \\ & x^{111} + x^{110} + x^{106} + x^{104} + x^{103} + x^{94} + x^{90} + x^{89} + x^{88} + x^{87} + x^{85} + x^{84} + x^{83} + \\ & x^{82} + x^{79} + x^{78} + x^{76} + x^{75} + x^{72} + x^{71} + x^{69} + x^{67} + x^{65} + x^{64} + x^{62} + x^{58} + x^{57} + \\ & x^{56} + x^{53} + x^{51} + x^{49} + x^{48} + x^{44} + x^{43} + x^{42} + x^{39} + x^{37} + x^{36} + x^{35} + x^{34} + x^{30} + \\ & x^{26} + x^{25} + x^{24} + x^{23} + x^{21} + x^{20} + x^{19} + x^{18} + x^{15} + x^{14} + x^{10} + x^8 + x^4 + x^2 + x + 1.\end{aligned}$$

**Proof:**
Rule Vector:

$[d_1, d_2, \ldots, d_{118}, d_{119}, d_{120}, d_{121}, d_{122}, d_{123}, d_{124}, d_{125}, d_{126}, d_{127}, d_{128}]$ $= [0, 0, \ldots, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0]$

Derivation of the characteristic polynomial:-
Initially $\triangle_{-3}$=0, $\triangle_{-2}$=0, $\triangle_{-1}$=0, $\triangle_0 = 1$.

$$\begin{aligned}
\triangle_1 &= (x + d_1)\triangle_0 + \triangle_{-1} + (x + d_0)\triangle_{-2} + \triangle_{-3} \\
&= x \\
\triangle_2 &= (x + d_2)\triangle_1 + \triangle_0 + (x + d_1)\triangle_{-1} + \triangle_{-2} \\
&= x^2 + 1 \\
&\vdots \\
\triangle_{128} &= (x + d_{128})\triangle_{127} + \triangle_{126} + (x + d_{127})\triangle_{125} + \triangle_{124} \\
&= x^{128} + x^{127} + x^{125} + x^{122} + x^{120} + x^{119} + x^{117} + x^{115} + x^{113} + x^{112} + x^{111} + \\
&\quad x^{110} + x^{106} + x^{104} + x^{103} + x^{94} + x^{90} + x^{89} + x^{88} + x^{87} + x^{85} + x^{84} + x^{83} + x^{82} + \\
&\quad x^{79} + x^{78} + x^{76} + x^{75} + x^{72} + x^{71} + x^{69} + x^{67} + x^{65} + x^{64} + x^{62} + x^{58} + x^{57} + x^{56} + \\
&\quad x^{53} + x^{51} + x^{49} + x^{48} + x^{44} + x^{43} + x^{42} + x^{39} + x^{37} + x^{36} + x^{35} + x^{34} + x^{30} + \\
&\quad x^{26} + x^{25} + x^{24} + x^{23} + x^{21} + x^{20} + x^{19} + x^{18} + x^{15} + x^{14} + x^{10} + x^8 + x^4 + x^2 + x^1 + 1
\end{aligned}$$

$\triangle_{128}$ represents a characteristic polynomial (CP). Test for primitivity of obtained CP is done by using a primitive polynomial search program(ppsearch256) given in [17].

### 4.2 Five Neighborhood Nonlinear Rule

In [18], Leporati and Mariot have investigated bipermutive rules of a given radius and studied a set of 5-neighbourhood nonlinear rules for their cryptographic suitability. All the rules have been studied taking Rule 30 as the benchmark. Based on the test results obtained from NIST [19] and ENT [20] tests, the following two rules have been found out to be better [18].

$$\begin{aligned}
&\text{Rule } 1452976485 : S_i^{t+1} = (\neg S_{i-2}^t \cdot \neg S_i^t \cdot \neg S_{i+1}^t \cdot \neg S_{i+2}^t) + (\neg S_{i-2}^t \cdot \neg S_{i-1}^t \cdot S_{i+1}^t \cdot \neg S_{i+2}^t) \\
&+(\neg S_{i-2}^t \cdot S_i^t \cdot \neg S_{i+1}^t \cdot S_{i+2}^t) + (S_{i-2}^t \cdot \neg S_i^t \cdot \neg S_{i+1}^t \cdot S_{i+2}^t) + (S_{i-2}^t \cdot S_i^t \cdot \neg S_{i+1}^t \cdot \neg S_{i+2}^t) \\
&+(\neg S_{i-2}^t \cdot S_{i-1}^t \cdot S_{i+1}^t \cdot S_{i+2}^t) + (S_{i-2}^t \cdot \neg S_{i-1}^t \cdot S_{i+1}^t \cdot S_{i+2}^t) + (S_{i-2}^t \cdot S_{i-1}^t \cdot S_{i+1}^t \cdot \neg S_{i+2}^t)
\end{aligned}$$

$$\begin{aligned}
&\text{Rule } 1520018790 : S_i^{t+1} = (\neg S_{i-2}^t \cdot \neg S_{i-1}^t \cdot \neg S_{i+1}^t \cdot \neg S_{i+2}^t) + (\neg S_{i-2}^t \cdot \neg S_{i-1}^t \cdot S_{i+1}^t \cdot \neg S_{i+2}^t) \\
&+(\neg S_{i-2}^t \cdot S_{i-1}^t \cdot \neg S_i^t \cdot \neg S_{i+2}^t) + (S_{i-2}^t \cdot \neg S_{i-1}^t \cdot \neg S_{i+1}^t \cdot \neg S_{i+2}^t) + (\neg S_{i-2}^t \cdot S_{i-1}^t \cdot S_i^t \cdot S_{i+2}^t) \\
&+(S_{i-2}^t \cdot \neg S_{i-1}^t \cdot S_{i+1}^t \cdot S_{i+2}^t) + (S_{i-2}^t \cdot S_{i-1}^t \cdot \neg S_i^t \cdot S_{i+2}^t) + (S_{i-2}^t \cdot S_{i-1}^t \cdot S_i^t \cdot \neg S_{i+2}^t)
\end{aligned}$$

where '+' and '.' and ¬ represents OR, AND and NOT Boolean operations respectively.

## 5 Description of CARPenter - Cellular Automata Based Resilient Pentavalent Stream Cipher

Our cipher model is inspired by the design of Grain, one of the eSTREAM finalists and FResCA, a CA based version of Grain. The design consists of three blocks, namely linear block (L block), nonlinear block (NL block) of lengths 128 bits each and a nonlinear mixing block (NMix). Figure 1 shows initialization of the cipher and Fig. 2 shows the generation of keystream bits. Both linear and nonlinear block use 5-neighbourhood rules and together form the 256-bit state of the cipher. Output stream is produced by NMix block after performing nonlinear mixing.

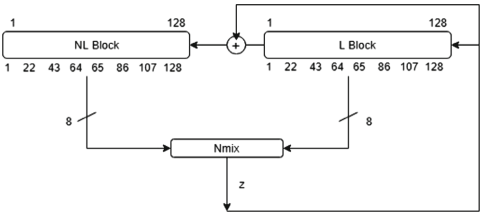


**Fig. 1.** Cipher initialization

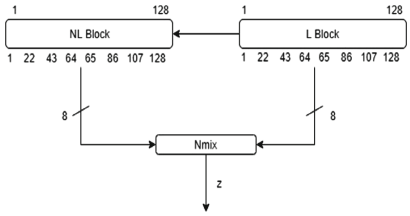


**Fig. 2.** Keystream generation

### 5.1 Nonlinear Block

Cells of nonlinear block will be updated using one of the 5-neighbourhood nonlinear rules (Rule 1452976485, Rule 1520018790) given in Sect. 4.2.

### 5.2 Linear Block

Cells of linear block are updated using a 5-neighbourhood Linear Hybrid CA rule vector which has been realized using two linear rules R0 and R1 discussed in Sect. 4.1. The cell positions 2, 8 and 10 use rule R1 and all the remaining 125 positions use rule R0 to realize the maximum length CA.

### 5.3 Nonlinear Mixing Block

NMix is a Boolean function which is nonlinear, balanced and reversible [14]. It is used as good key mixing function in block ciphers and also resists differential attacks. The NMix function is defined for two $n$-bits inputs. If input bit sets are $X = x_1, x_2, \ldots, x_{n-1}, x_n$ and $Y = y_1, y_2, \ldots, y_{n-1}, y_n$ and output bit set is $Z = z_1, z_2, \ldots, z_{n-1}, z_n$, then NMix for $i^{th}$ bit is defined as follows.

$$z_i = x_i \oplus y_i \oplus c_{i-1}$$
$$c_i = x_0.y_0 \oplus \cdots \oplus x_i.y_i \oplus x_{i-1}.x_i \oplus y_{i-1}.y_i$$

and $x_{-1} = y_{-1} = c_{-1} = 0, 0 \leq i \leq n-1$

Input to the NMix is eight bits each from both the linear and nonlinear blocks and the Most Significant Bit (MSB) of NMix is the output of the cipher. All the input bits are present in the computation of MSB of nonlinear mixing block which provides good diffusion.

### 5.4 Working of CARPenter

CARPenter is a Grain-like **C**ellular **A**utomata Based **R**esilient **Pent**avalent Stream Ciph**er**. The cipher has two phases, namely initialization phase and keystream generation phase. The initialization phase consists of 16 iterations

and the output is suppressed in this phase. Here, the number of iterations (16 iterations) is less when compared to Grain (256 iterations) and FresCA (32 iterations). The 128-bit key is loaded into the nonlinear block and the 128-bit IV is loaded into the linear block of the cipher. During this phase, the output is fed back to the linear and nonlinear blocks as shown in the Fig. 1. The output of the NMix function is XORed with the first bit in the linear block and this has dual role in nonlinear block. It acts as the second-right-neighbour of the $127^{th}$ bit and as both first- and second-right-neighbour of the $128^{th}$ bit in the nonlinear block. This is shown in Fig. 3. The output of NMix also acts as the second-right-

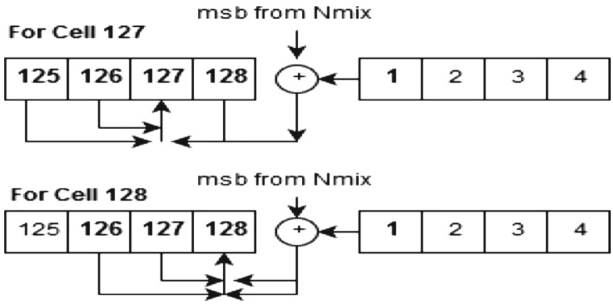


**Fig. 3.** Updation of cell 127 and cell 128 of nonlinear block

neighbour of the $127^{th}$ bit and as both first- and second-right-neighbours of the $128^{th}$ bit in the linear block. In each iteration, each bit in the nonlinear block changes its state according to the 5-neighbourhood nonlinear rule mentioned in Sect. 4.2. In the linear block, the state transition takes place according to the rules R0 and R1. We need to select taps in both linear and nonlinear blocks of the cipher. Taps are the bit positions that affect the output. Eight taps each are selected from both linear and nonlinear blocks so that the number of inputs to the NMix block are 16. The eight taps correspond to the bit positions 1, 22, 43, 64, 65, 86, 107, and 128 in both the blocks. In order to have influence of all the state bits in output in lesser number of iterations, the taps are positioned equally except the two middle ones. After initialization phase, the feed back lines are removed and the keystream bits are generated.

## 6 Security Analysis

### 6.1 NIST Statistical Test

National Institute of Standards and Technology (NIST) has developed a statistical test suite known as NIST-statistical test suite [19]. It is a package of 15 tests to test the randomness of pseudo-random binary sequence of arbitrary length. To test the randomness of CARPenter, a bit stream of length 0.1 billion bits has been generated and fed to the NIST test suite. Input bit stream is divided into

100 keystreams of 1 million bits each by the NIST test suite. All the tests passed with appropriate p-values as shown in Table 1.

**Table 1.** NIST test result

| | | Nonlinear rule - 1 | | Nonlinear rule - 2 | |
|---|---|---|---|---|---|
| SI.No | Test name | P-value | Status | P-value | Status |
| 1 | Frequency test | 0.955835 | Pass | 0.657933 | Pass |
| 2 | Block frequency test | 0.494392 | Pass | 0.289667 | Pass |
| 3 | Cumulative sums test | 0.595549 | Pass | 0.108791 | Pass |
| 4 | Runs test | 0.616305 | Pass | 0.955835 | Pass |
| 5 | Longest runs test | 0.171867 | Pass | 0.534146 | Pass |
| 6 | Rank test | 0.739918 | Pass | 0.191687 | Pass |
| 7 | FFT test | 0.153763 | Pass | 0.616305 | Pass |
| 8 | Non overlapping template test | 0.595549 | Pass | 0.289667 | Pass |
| 9 | Overlapping template test | 0.834308 | Pass | 0.595549 | Pass |
| 10 | Universal | 0.419021 | Pass | 0.334538 | Pass |
| 11 | Approximate entropy | 0.115387 | Pass | 0.419021 | Pass |
| 12 | Random excursions | 0.178278 | Pass | 0.026648 | Pass |
| 13 | Random excursions variant | 0.706149 | Pass | 0.723129 | Pass |
| 14 | Serial | 0.759756 | Pass | 0.319084 | Pass |
| 15 | Linear complexity | 0.994250 | Pass | 0.202268 | Pass |

### 6.2 Resiliency

The two bipermutive nonlinear rules used in the NL block of CARPenter are 2-resilient [18]. Since the rules are 2-resilient, they are both balanced and $2^{nd}$ order correlation immune.

### 6.3 Algebraic Attack

If the number of different input variables available in the output Boolean function is high, then the immunity against the algebraic attack will be high. The output function of CARPenter contains 16 and 68 different input variables in first and second iteration respectively and will increase with each iteration. After 16 iterations, at the time of keystream generation the output Boolean function will be affected by all the 256 bits of the cipher. So output Boolean function of the cipher will have high algebraic degree at the time of key generation and this fact will prevent the algebraic attack on CARPenter.

### 6.4 Linear Attack

Nonlinearity of output Boolean function in the first iteration is 32256 and will increase with each iteration. At the time of key generation phase, nonlinearity will be much higher.

### 6.5 Meier-Staffelbach Attack

Meier and Staffelbach attacked the Rule-30 based stream cipher designed by Wolfram in [21]. The state of the $i^{th}$ cell from time $t$ to $t+n$ (temporal sequence) is known to the attacker. This attack tries to guess the right half of initial state and then tries to generate the right adjacent neighbour of temporal sequence. Since there is a many-to-one mapping from the right side to the temporal sequence, a guessed right side value may give correct right adjacent sequence. Since there is a linear relation between the temporal sequence and the left half, the attack calculates the left half, by moving backward from $t+n$ to $t$. Then the calculated seed is used to generate the temporal sequence. Attack is successful if the generated temporal sequence matches with original temporal sequence.

This attack is not applicable to CARPenter. In order to compute the right adjacent neighbour of temporal sequence, knowledge of the state of left neighbour is required because of the use of 5-neighbourhood CA. Random value cannot be assigned to the left hand side of the temporal sequence because there is no many-to-one mapping from left hand side to the temporal sequence.

### 6.6 Time/Memory/Data Tradeoff Attack

If inner state of a stream cipher consists of $n$ bits, then $O(2^{n/2})$ is the complexity of this attack on stream cipher. Inner state of the CARPenter consists of 256 bits which makes it difficult to perform Time/Memory/Data/tradeoff attack.

### 6.7 Fault Attack

In this attack, a fault can be introduced at any bit position. The attacker has partial control over the timing and the position of the fault. She can observe the behaviour of the cipher by resetting the cipher and reintroducing the fault at different positions. Because of the use of CA in CARPenter, the fault tracking becomes impossible. In NL block, the fault will dissipate nonlinearly and any fault introduced in linear block will reach the nonlinear block in initialization phase itself making it difficult to track the fault.

## 7 Conclusion

We have proposed a Grain-like, 5-neighbourhood CA based stream cipher called CARPenter. The cipher exhibits very good cryptographic properties. The use of 2-resilient nonlinear rule makes our cipher resilient. Initialization phase of CARPenter is faster than Grain and FResCA. Generated keystream has good pseudorandomness and is strong against different attacks.

## References


1. The eSTREAM project. http://www.ecrypt.eu.org/stream/project.html. Accessed 12 May 2018
2. European network of excellence for cryptography. http://www.ecrypt.eu.org/. Accessed 12 May 2018
3. Hell, M., Johansson, T., Maximov, A., Meier, W.: A stream cipher proposal: grain-128. In: 2006 IEEE International Symposium on Information Theory, pp. 1614–1618, July 2006
4. De Canniere, C., Preneel, B.: Trivium specifications. In: eSTREAM, ECRYPT Stream Cipher Project (2006)
5. Karmakar, S., Mukhopadhyay, D., Roy Chowdhury, D.: Cavium - strengthening trivium stream cipher using cellular automata. J. Cell. Autom. **7**, 179–197 (2012)
6. Das, S., Roy Chowdhury, D.: CAR30: a new scalable stream cipher with rule 30. Cryptogr. Commun. **5**(2), 137–162 (2013)
7. Jose, J., Roy Chowdhury, D.: FResCA: a fault-resistant cellular automata based stream cipher. In: El Yacoubi, S., Was, J., Bandini, S. (eds.) ACRI 2016. LNCS, vol. 9863, pp. 24–33. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44365-2_3
8. Jose, J., Roy Chowdhury, D.: Investigating four neighbourhood cellular automata as better cryptographic primitives. J. Discrete Math. Sci. Cryptogr. **20**(8), 1675–1695 (2017)
9. Jose, J., Das, S., Roy Chowdhury, D.: Prevention of fault attacks in cellular automata based stream ciphers. J. Cell. Autom. **12**(1–2), 141–157 (2016)
10. Das, S., Roy Chowdhury, D.: *CASTREAM*: a new stream cipher suitable for both hardware and software. In: Sirakoulis, G.C., Bandini, S. (eds.) ACRI 2012. LNCS, vol. 7495, pp. 601–610. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33350-7_62
11. Cattell, K., Muzio, J.C.: Synthesis of one-dimensional linear hybrid cellular automata. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **15**(3), 325–335 (1996)
12. Chaudhuri, P.P., Roy Chowdhury, D., Nandi, S., Chattopadhyay, S.: Additive Cellular Automata Theory and Application, 1st edn. IEEE Computer Society Press, Washington, D.C. (1997)
13. Feng, D., Wu, C.-K.: Boolean Functions and Their Applications in Cryptography, 1st edn. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-48865-2
14. Bhaumik, J., Roy Chowdhury, D.: Nmix: an ideal candidate for key mixing. In: Proceedings of the International Conference on Security and Cryptography. SECRYPT 2009, 7–10 July 2009, Milan, Italy, pp. 285–288. INSTICC Press (2009). SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications
15. Maiti, S., Roy Chowdhury, D.: Study of five-neighborhood linear hybrid cellular automata and their synthesis. In: Giri, D., Mohapatra, R.N., Begehr, H., Obaidat, M.S. (eds.) ICMC 2017. CCIS, vol. 655, pp. 68–83. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-4642-1_7
16. McEliece, R.J.: Finite Fields for Computer Scientists and Engineers, 1st edn. Springer, Boston (1987). https://doi.org/10.1007/978-1-4613-1983-2
17. A primitive polynomial search program. http://notabs.org/primitivepolynomials/primitivepolynomials.htm. Accessed 12 May 2018

18. Leporati, A., Mariot, L.: Cryptographic properties of bipermutive cellular automata rules. J. Cell. Autom. **9**, 437–475 (2014)
19. Nist statistical test suite. https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software. Accessed 12 May 2018
20. ENT - a pseudorandom number sequence test program. http://www.fourmilab.ch/random/. Accessed 12 May 2018
21. Meier, W., Staffelbach, O.: Analysis of pseudo random sequences generated by cellular automata. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 186–199. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-46416-6_17