Four Neighbourhood Cellular Automata as Better Cryptographic Primitives

Jimmy Jose and Dipanwita Roy Chowdhury

Crypto Research Laboratory,
Department of Computer Science and Engineering,
Indian Institute of Technology Kharagpur, India
{jimmy,drc}@cse.iitkgp.ernet.in

Abstract. Three-neighbourhood Cellular Automata (CA) are widely studied and accepted as suitable cryptographic primitive. Rule 30, a 3-neighbourhood CA rule, was proposed as an ideal candidate for cryptographic primitive by Wolfram. However, rule 30 was shown to be weak against Meier-Staffelbach attack [7]. The cryptographic properties like diffusion and randomness increase with increase in neighbourhood radius and thus opens the avenue of exploring the cryptographic properties of 4-neighbourhood CA. This work explores whether four-neighbourhood CA can be a better cryptographic primitive. We construct a class of cryptographically suitable 4-neighbourhood nonlinear CA rules that resembles rule 30. One 4-neighbourhood nonlinear CA from this selected class is shown to be resistant against Meier-Staffelbach attack on rule 30, justifying the applicability of 4-neighbourhood CA as better cryptographic primitives.

Key words: Cellular Automata, nonlinearity, CA rule 30

1 Introduction

Cellular Automata (CA) are characterised by speed in computation and can be modelled to offer high diffusion and randomness. Thus, CA can effectively be used as good cryptographic primitives. The simple and regular structure of CA is very well suited both for hardware and software implementation. Parallel transformations of CA allow high throughput as well as it provide resistance to correlation attacks and the resistance increases with increase in neighbourhood size. This resistance depends also on the choice of CA rules.

Three-neighbourhood CA have been extensively analysed for their suitability as cryptographic primitives. The 3-neighbourhood CA rule 30 was considered to produce good pseudorandom sequences [11]. Meier and Staffelbach [7] have shown that the construction does not provide the required security. All the 256 elementary 3-neighbourhood CA rules were analysed in [6] to find out rules which can generate pseudorandom sequences. It was found out that no 3-neighbourhood nonlinear balanced rule is correlation immune and hence the CA using these rules are susceptible to correlation attacks.

Literature mostly focus on 3-neighbourhood CA. Pseudorandomness, an important criterion for a cryptographic primitive, is also governed by the size of the neighbourhood. Correlation also reduces with increase in neighbourhood size. Thus 4-neighbourhood CA can provide good randomness and less correlation. However, only very few attempts [4, 5, 2] have been made towards the study of CA having neighbourhood size greater than 3. In [4], Lacharme et al. analysed all the 65536 elementary CA rules with four variables to find 200 nonlinear balanced functions which are 1-resilient. In [2] also, 1-resilient 5-neighbourhood elementary CA rules are analysed. Nonlinear and resilient rules are selected from 5-neighbourhood bipermutive CA rules in [5]. Wolfram proposed that rule 30 can be effectively used in the construction of stream ciphers [11]. Even though Meier and Staffelbach [7] have shown that the construction does not provide adequate security, still rule 30 has good cryptographic properties.

In this paper, we construct a class of 4-neighbourhood Cellular Automata where the rule structure functionally resemble the Boolean function of rule 30. We study the cryptographic properties of this set of CA rules like nonlinearity, balancedness, and correlation immunity and select one rule from the set with good cryptographic properties and show that it can resist Meier-Staffelbach attack on 3-neighbourhood nonlinear rule 30.

This paper is organised as follows. In Section 2, we discuss the construction of four neighbourhood CA rules resembling rule 30 of 3-neighbourhood CA. Section 3 analyses the cryptographic properties of these 4-neighbourhood CA rules. Section 4 shows the resistance by 4-neighbourhood nonlinear CA against Meier-Staffelbach attack. Section 5 concludes the work.

2 4-neighbourhood Nonlinear Cellular Automata

Wolfram developed Cellular Automata (CA) as a mathematical model for self-organising systems [9]. They are a group of cells each of which can be in one of k states. Normally k is 2, where the two states are represented by 0 and 1. The dimension of CA can also differ but single-dimensional CA are of particular interest. In each time step, all cells update their state. If this update is a function of itself and its immediate neighbours on either side, then the CA is known as a 3-neighbourhood CA.

$q_i(t+1) = f(q_{i-1}(t), q_i(t), q_{i+1}(t))$,
where $q_i(t)$ is the state of the i-th cell at time t, and f denotes the local transition function realised using combinational logic.

A 3-neighbourhood CA, where each cell can be in any one of the two states (represented as 0 and 1), can have 2^3 distinct neighbourhood configurations. There are 2^{2^3} distinct mappings from these neighbourhood configurations to next states. Each mapping is called a rule and these range from rule 0 to rule 255. Some 3-neighbourhood CA rules are
Rule30: $q_i(t+1) = q_{i-1}(t) \oplus (q_i(t) + q_{i+1}(t))$,
Rule 60: $q_i(t+1) = q_{i-1}(t) \oplus q_i(t)$,
Rule 90: $q_i(t+1) = q_{i-1}(t) \oplus q_{i+1}(t)$,

Rule 150: $q_i(t+1) = q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)$,
Rule 160: $q_i(t+1) = q_{i-1}(t).q_{i+1}(t)$,
Rule 250: $q_i(t+1) = q_{i-1}(t) + q_{i+1}(t)$,
where +, ., and $\oplus$ denote Boolean OR, AND, and XOR respectively.

If the cells in the CA depend on two left, itself, and one right cell for their update, then the CA is called left skewed 4-neighbourhood CA. Similarly, we can define right skewed 4-neighbourhood CA. They are shown in Fig. 1.

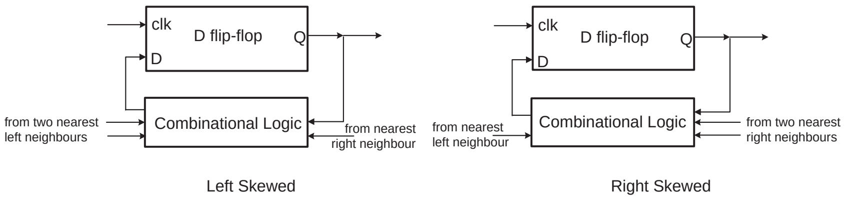
clk
D flip-flop
Q
D
Combinational Logic
from two nearest left neighbours
from nearest right neighbour
Left Skewed
clk
D flip-flop
Q
D
Combinational Logic
from nearest left neighbour
from two nearest right neighbours
Right Skewed

Fig. 1. Single Cell in Left Skewed and Right Skewed 4-neighbourhood CA

In case of 4-neighbourhood CA, there are 2^{2^4} distinct mappings and rules range from 0 to 65535. For a particular rule, all 4-neighbourhood configurations are listed from 1111 to 0000 and the resulting state of each configuration is also listed in the same order and is treated as the binary representation of the rule number.

A cross-section of a CA which uses two left skewed 4-neighbourhood CA rules is shown in Fig. 2. The enable input connected with the AND gates may be used to select one of the two rules. Logic 0 in the enable input selects rule 21930 while logic 1 selects rule 39270.

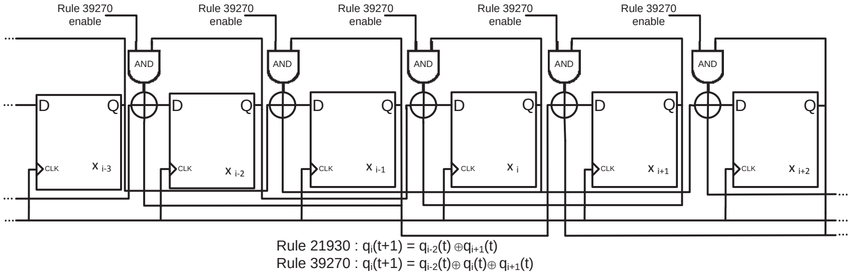

Fig. 2. 4-neighbourhood Linear Hybrid CA based on rules 21930, 39270 (left skewed)

2.1 Constructing 4-neighbourhood CA Rules Resembling Rule 30 of 3-neighbourhood CA

Rule 30 is probably the mostly studied and analysed 3-neighbourhood Cellular Automata rule and was considered as a good pseudorandom number generator [10, 3]. Wolfram proposed that rule 30 can be effectively used in the construction of stream ciphers [11]. Even though Meier and Staffelbach [7] have shown that the construction does not provide adequate security, still rule 30 has good cryptographic properties. Rule 30 performs an OR operation followed by an XOR operation. Interchanging the rule 30 operations results in rule 246.
Rule30: $q_i(t+1) = q_{i-1}(t) \oplus (q_i(t) + q_{i+1}(t))$
Rule 246: $q_i(t+1) = q_{i-1}(t) + (q_i(t) \oplus q_{i+1}(t))$
The number of cells in the CA participating in the computation of the new state increases with each iteration. In case of 3-neighbourhood CA, $2n+1$ cells participate in the computation, where n is the iteration number.

Table 1. Four-neighbourhood Nonlinear Rules

sl. no.	Rule No	Left Skewed Rule	sl. no.	Rule No	Left Skewed Rule
1	510	$q_{i-2} \oplus (q_{i-1} + q_i + q_{i+1})$	14	50070	$q_{i-1} \oplus q_i \oplus (q_{i-2} + q_{i+1})$
2	854	$(q_{i-2} + q_{i+1}) \oplus (q_{i-1} + q_i)$	15	51510	$q_{i-2} \oplus q_i \oplus (q_{i-1} + q_{i+1})$
3	1334	$(q_{i-2} + q_i) \oplus (q_{i-1} + q_{i+1})$	16	57630	$q_{i-2} \oplus q_{i-1} \oplus (q_i + q_{i+1})$
4	3870	$(q_{i-1} \oplus (q_{i-2} + q_i + q_{i+1})$	17	60350	$(q_{i-2} \oplus q_{i-1} \oplus q_i) + q_{i+1}$
5	4382	$(q_{i-2} + q_{i-1}) \oplus (q_i + q_{i+1})$	18	60894	$(q_{i-2} \oplus q_{i-1} \oplus q_{i+1}) + q_i$
6	13110	$q_i \oplus (q_{i-2} + q_{i-1} + q_{i+1})$	19	61438	$(q_i + q_{i+1}) + (q_{i-2} \oplus q_{i-1})$
7	21846	$q_{i+1} \oplus (q_{i-2} + q_{i-1} + q_i)$	20	63990	$(q_{i-2} \oplus q_i \oplus q_{i+1}) + q_{i-1}$
8	28662	$(q_{i-2} \oplus q_{i-1}) + (q_i \oplus q_{i+1})$	21	64510	$(q_{i-1} + q_{i+1}) + (q_{i-2} \oplus q_i)$
9	31710	$(q_{i-2} \oplus q_i) + (q_{i-1} \oplus q_{i+1})$	22	65022	$(q_{i-1} + q_i) + (q_{i-2} \oplus q_{i+1})$
10	32190	$(q_{i-2} \oplus q_{i+1}) + (q_{i-1} \oplus q_i)$	23	65430	$(q_{i-1} \oplus q_i \oplus q_{i+1}) + q_{i-2}$
11	39318	$q_i \oplus q_{i+1} \oplus (q_{i-2} + q_{i-1})$	24	65470	$(q_{i-2} + q_{i+1}) + (q_{i-1} \oplus q_i)$
12	42390	$q_{i-1} \oplus q_{i+1} \oplus (q_{i-2} + q_i)$	25	65502	$(q_{i-2} + q_i) + (q_{i-1} \oplus q_{i+1})$
13	43350	$q_{i-2} \oplus q_{i+1} \oplus (q_{i-1} + q_i)$	26	65526	$(q_{i-2} + q_{i-1}) + (q_i \oplus q_{i+1})$

Here, we consider 4-neighbourhood CA rules based on their resemblance with 3-neighbourhood CA rule 30 and the cryptographic properties are analysed to see their suitability as cryptoprimitive. Rule 30 contains one XOR and one OR. Similarly, the chosen 4-neighbourhood CA rules contain at least one XOR and OR and no other operators. Twenty-six 4-neighbourhood rules which resemble rule 30 of 3-neighbourhood CA are constructed and listed in table 1. As an example, rule 50070 contains one OR and two XORs. In this table, q_{i-2}, q_{i-1}, q_i, and q_{i+1} represent the two left-neighbours, self, and right-neighbour respectively of the left skewed 4-neighbourhood CA. Similar rules exist for right skewed CA.

The cryptographic properties of the 26 four-neighbourhood CA rules are studied in detail.

3 Cryptographic Properties of 4-neighbourhood CA

In this section, the cryptographic properties, namely nonlinearity, balancedness, and correlation immunity of 4-neighbourhood CA rules are explored. The number of cells in the CA participating in the computation of new state increases with each iteration. In 4-neighbourhood CA, $3n + 1$ cells participate in the computation in the n-th iteration. In the fourth iteration (i.e. $n = 4$), the computation depends on 13 cells and computation becomes unwieldy. So our experiment runs for three iterations where the cells involved in computation are 4, 7, and 10 respectively. A detailed description of cryptographic properties can be found in [1].

3.1 Analysis of 3-neighbourhood Nonlinear CA Rules

We analyse the 3-neighbourhood CA rules 30 and 246 for the first three iterations. The results are shown in table 2. It shows that rule 30 is both balanced and nonlinear. But it is not correlation immune. Rule 246 is neither balanced nor correlation immune but nonlinear. In fact, no nonlinear Boolean function of 3 variables is 1-resilient (balanced as well as first-order correlation immune) according to Siegenthaler bound [8].

Table 2. Cryptographic Properties of 3-neighbourhood Rules 30 and 246

sl. no.	Rule No	Nonlinearity			Balancedness			Correlation Immunity		
		1	2	3	1	2	3	1	2	3
1	30	2	4	36	True	True	True	0	0	0
2	246	2	6	22	False	False	False	0	0	0

3.2 Cryptographic Suitability of 4-neighbourhood Nonlinear CA

This section analyses the cryptographic properties of all the 26 synthesised 4-neighbourhood CA rules. The nonlinearity, balancedness, and correlation immunity corresponding to the chosen rules are listed in table 3.

A 4-neighbourhood CA having 32 cells with null boundary[1] is used for the experiment. The nonlinearity of the rules for first three iterations is computed. For example, in table 3, rule 510 has nonlinearity 2, 28, and 224 respectively in the first, second, and third iterations. If a rule is balanced in a specific iteration, then balancedness is represented with a *true* value. Otherwise, the value is *false*. For example, rule 51510 in table 3 has *true* value for balancedness in all the three iterations. The correlation immunity is measured and tabulated for each iteration. As an example, for rule 57630 in table 3, correlation immunity is measured as 1 for all the three iterations.

[1] A null boundary CA is a CA where the extreme cells in the boundaries are connected to the zero states

Table 3. Cryptographic Properties of rules in table 1

sl. no.	Rule No	Nonlinearity			Balancedness			Correlation Immunity		
		1	2	3	1	2	3	1	2	3
1	510	2	28	224	True	True	True	0	0	0
2	854	6	38	366	False	False	False	0	0	0
3	1334	6	30	412	False	False	False	0	0	0
4	3870	2	32	272	True	True	False	0	0	0
5	4382	6	42	412	False	False	False	0	0	0
6	13110	2	32	272	True	True	False	0	0	0
7	21846	2	28	224	True	True	True	0	0	0
8	28662	4	40	304	False	False	False	0	0	0
9	31710	4	40	392	False	False	True	0	0	1
10	32190	4	48	400	False	False	False	0	0	0
11	39318	4	32	368	True	True	True	1	1	1
12	42390	4	40	408	True	True	True	1	0	1
13	43350	4	48	384	True	True	True	1	2	1
14	50070	4	52	428	True	False	False	1	0	0
15	51510	4	40	408	True	True	True	1	0	1
16	57630	4	32	368	True	True	True	1	1	1
17	60350	4	16	60	False	False	False	0	0	0
18	60894	4	16	92	False	False	False	0	0	0
19	61438	2	2	2	False	False	False	0	0	0
20	63990	4	16	92	False	False	False	0	0	0
21	64510	2	3	5	False	False	False	0	0	0
22	65022	2	2	2	False	False	False	0	0	0
23	65430	4	16	60	False	False	False	0	0	0
24	65470	2	4	8	False	False	False	0	0	0
25	65502	2	3	5	False	False	False	0	0	0
26	65526	2	2	2	False	False	False	0	0	0

In table 3, eleven rules have nonlinearity greater than 350 in the third iteration. Seven rules qualify balancedness criterion for all the three iterations. Taking the characteristics nonlinearity, balancedness, and correlation immunity into consideration, rules 39318, 42390, 43350, 51510 and 57630 can be considered as good candidate rules.

4 Resistance Against Meier-Staffelbach Attack

Three neighbourhood CA rule 30 has been identified as a good cryptographic primitive. However, Meier and Staffelbach mounted attack against rule 30 based CA. In this attack [7], a portion of the temporal sequence (pseudo random sequence) is known. From the state values of the i-th cell - temporal sequence - for $n+1$ time steps from t to $t+n$, the attack tries to find the state value of cells at the t-th time step. Refer Fig. 3.

For a 3-neighbourhood CA, the values of the sites that can be computed from the initial site vector $q_{i-n}(t), ..., q_{i+n}(t)$ form a triangle as shown in Fig. 3

where $q_i(t)$ represents state of the i-th cell at time t. The column in bold face represents temporal sequence. If rule 30 is employed, a bit change propagates to the right with probability 1 and to the left roughly with a speed of $\frac{1}{4}$ in the triangle. Some changes in the right-hand initial sites $q_{i+1}(t), ..., q_{i+n}(t)$ of the triangle do not change the given portion of temporal sequence in column i or its right adjacent sequence in column $i+1$, i.e. there is a many-to-one mapping from the right-hand initial states to the temporal sequence or its right adjacent sequence. The attack is based on the principle that a random set of values for right-hand initial states may give correct right adjacent sequence even if the assigned right-hand initial state values were wrong. Knowledge of right adjacent sequence is equivalent to knowledge of the seed.

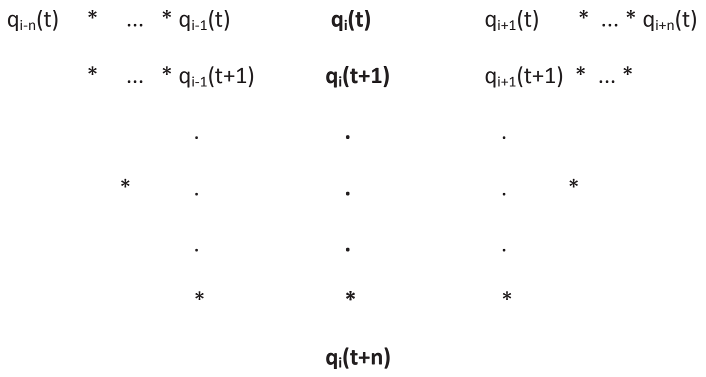

Fig. 3. Triangle determined by initial site vector $q_{i-n}(t), ..., q_{i+n}(t)$

Using the given temporal sequence in column i and the guessed values of $q_{i+1}(t), ..., q_{i+n}(t)$, the rule

$q_i(t+1) = q_{i-1}(t) \oplus (q_i(t) + q_{i+1}(t))$

completes the right hand side of the triangle. At this point, we know the values in columns i and $i+1$. To complete the left hand side of the triangle, we use the equation

$q_{i-1}(t) = q_i(t+1) \oplus (q_i(t) + q_{i+1}(t))$.

Finally, the values of $q_{i-n}(t), ..., q_{i-1}(t)$ are known. Now the computed seed is used to generate the temporal sequence. If the original temporal sequence and generated temporal sequence matches, we stop with success. Otherwise, the process is repeated by assigning another set of values to the right-hand initial states.

4.1 Inapplicability of Meier-Staffelbach Attack against 4-neighbourhood CA rules

In this section, we show that 4-neighbourhood CA resist the above attack. We take the rule 57630 from the class of 4-neighbourhood rules under study. The rule is defined as

$q_i(t+1) = q_{i-2}(t) \oplus q_{i-1}(t) \oplus (q_i(t) + q_{i+1}(t))$.

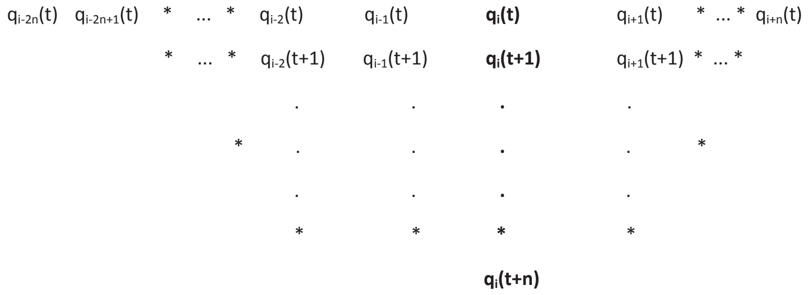
$q_{i-2n}(t)$ $q_{i-2n+1}(t)$ * ... * $q_{i-2}(t)$ $q_{i-1}(t)$ $q_i(t)$ $q_{i+1}(t)$ * ... * $q_{i+n}(t)$
* ... * $q_{i-2}(t+1)$ $q_{i-1}(t+1)$ $q_i(t+1)$ $q_{i+1}(t+1)$ * ... *
* *
* * * *
$q_i(t+n)$

Fig. 4. Triangle determined by initial site vector $q_{i-2n}(t), ..., q_{i+n}(t)$

Like 3-neighbourhood CA rule 30, we have many-to-one mapping from right-hand initial states to the temporal sequence or its right adjacent sequence when we use 4-neighbourhood CA rule 57630. For 4-neighbourhood CA, the initial site vector $q_{i-2n}(t), ..., q_{i-1}(t)$, $q_i(t), q_{i+1}(t), ..., q_{i+n}(t)$ determines a triangle as shown in Fig. 4. To find out right adjacent sequence in column $i+1$ of the triangle, we need left adjacent sequence represented by column $i-1$ of the triangle. Computation of left adjacent sequence needs the knowledge of left-hand initial states $q_{i-2n}(t), ..., q_{i-1}(t)$. So computation of right adjacent sequence requires left-hand initial states. Left-hand initial states can be computed with the knowledge of two columns adjacent to temporal sequence like $i+1$ and $i+2$ of the triangle by completing the left-hand side of the triangle. Arbitrary values cannot be assigned for left-hand initial states as is done for right-hand adjacent sequence in Meier-Staffelbach algorithm as there is no many-to-one mapping from left-hand initial states to the temporal sequence. The reason for the absence of many-to-one mapping is that a bit change in the triangle propagates to the right with probability 1.

Assume that we somehow managed to find out the right adjacent sequence (column $i+1$ of the triangle) from the given temporal sequence (column i of the triangle). Rule 57630 is rewritten to find $q_{i+1}(t+1)$ as

$q_{i+1}(t+1) = q_{i-1}(t) \oplus q_i(t) \oplus (q_{i+1}(t) + q_{i+2}(t))$.

Rearranging terms so that the term $q_{i-1}(t)$ is on the LHS,

$q_{i-1}(t) = q_{i+1}(t+1) \oplus q_i(t) \oplus (q_{i+1}(t) + q_{i+2}(t))$.

This equation is used to complete the left-hand side of the triangle in search of seed. The rule says that to find the values in cells at column $i-1$, we require the values in column $i+2$ also in addition to the values in columns i and $i+1$.

In 3-neighbourhood CA, searching for seed K_s is equivalent to computation of right adjacent sequence K_{r1} and is represented by the mapping

$F : \{K_s\} \rightarrow \{K_{r1}\}$.

In the case of left-skewed 4-neighbourhood rules, sequence to the right of right-adjacent sequence - denoted as K_{r2} - is also required in addition to K_{r1} and the mapping is

$F : \{K_s\} \rightarrow \{K_{r1}, K_{r2}\}$.

So in the case of 4-neighbourhood CA, the attack fails to compute right adjacent sequence K_{r1} as stated earlier even if the mapping above shows that it is not sufficient to find the seed.

5 Conclusion

In this paper, we have studied the cryptographic suitability of a class of four-neighbourhood nonlinear CA rules. We have shown the inapplicability of Meier-Staffelbach attack on rule 30 against four-neighbourhood CA by taking one rule from the class of rules under study.

References

1. Carlet, C.: Boolean functions for cryptography and error-correcting codes. In: Crama, Y., Hammer, P.L. (eds.) Boolean Models and Methods in Mathematics, Computer Science, and Engineering. Cambridge University Press, New York (2010)
2. Formenti, E., Imai, K., Martin, B., Yunés, J.B.: Advances on random sequence generation by uniform cellular automata. In: Computing with New Resources, pp. 56–70. LNCS (2014)
3. Gutowitz, H.: Cellular Automata: Theory and Experiment, A Bradford book, vol. 45. MIT Press (1991)
4. Lacharme, P., Martin, B., Solé, P., et al.: Pseudo-random sequences, boolean functions and cellular automata. Proceedings of BFCA pp. 80–95 (2008)
5. Leporati, A., Mariot, L.: 1-resiliency of bipermutive cellular automata rules. In: Automata. pp. 110–123 (2013)
6. Martin, B.: A walsh exploration of elementary ca rules. Cellular Automata Workshop pp. 25–30 (2006)
7. Meier, W., Staffelbach, O.: Analysis of pseudo random sequences generated by cellular automata. In: EUROCRYPT '91. pp. 186–199 (1991)
8. Siegenthaler, T.: Correlation-immunity of nonlinear combining functions for cryptographic applications. IEEE Transactions on Information Theory 30(5), 776–780 (1984)
9. Wolfram, S.: Statistical mechanics of cellular automata. Rev. Mod. Phys. 55, 601–644 (Jul 1983)
10. Wolfram, S.: Cryptography with cellular automata. In: CRYPTO '85. pp. 429–432 (1985)
11. Wolfram, S.: Random sequence generation by cellular automata. Advances in applied mathematics 7(2), 123–169 (1986)