NOCAS: A Nonlinear Cellular Automata Based Stream Cipher

Sandip Karmakar, Dipanwita Roy Chowdhury

AUTOMATA 2011, Santiago, Chile DMTCS proc. **AP**, 2012, 135–146

NOCAS : A Nonlinear Cellular Automata Based Stream Cipher

Sandip Karmakar[†] and Dipanwita Roy Chowdhury

Indian Institute of Technology, Kharagpur, WB, India

LFSR and NFSR are the basic building blocks in almost all the state of the art stream ciphers like Trivium and Grain-128. However, a number of attacks are mounted on these type of ciphers. Cellular Automata (CA) has recently been chosen as a suitable structure for crypto-primitives. In this work, a stream cipher is presented based on hybrid CA. The stream cipher takes 128 bit key and 128 bit initialization vector (IV) as input. It is designed to produce 2^{128} random keystream bits and initialization phase is made faster 4 times than that of Grain-128. We also analyze the cryptographic strength of this cipher. Finally, the proposed cipher is shown to be resistant against known existing attacks.

Keywords: Cellular Automata, Stream Cipher, NMix, Hybrid Nonlinear Cellular Automata

1 Introduction

The mass use of hand-held devices/PDA has popularized the use of stream ciphers. Stream ciphers are much less power consuming, requires small space for their operations and are faster in operation than other cryptographic algorithms. Generally, in stream ciphers a secret key and a public IV are input. Keystream bits are generated by the cipher per cycle of operation. The plain-text is XORed on the encryption side with the generated keystream to produce the cipher-text. Decryption is carried out by simply XORing the cipher-text with the keystream. The eStream project which started in year 2004 was an attempt to standardize stream ciphers. A large number of stream ciphers were submitted to this project. After a cryptanalysis phase ranging over 4 years, stream ciphers were filtered in 3 phases by their performance and security. At the final stage has Trivium [CP], Grain [HJM] and MICKEY [BD] which are hardware efficient and Rabbit [BVCZ], Salsa20/12 [Ber], HC-128 [Wu], SOSEMANUK [BBC$^+$] that are software based stream ciphers.

The eStream project categorized stream ciphers in two sections, hardware based and software based. Software based ciphers are expected to have optimized software performance, while hardware based ciphers are optimized for hardware. The submitted software based ciphers had a nonlinear filter function which combines LFSR (Linear Feedback Shift Register) and NFSR (Nonlinear Feedback Shift Register) bits. Trivium [CP] is reported as the fastest cipher providing hardware performance. Grain-128 [HJM] is

[†]Email: sandip1kk@gmail.com.

the next cipher in terms of hardware performance. It combines a LFSR and a NFSR bits by a nonlinear function. However, Grain-128 has been subjected to many attacks, like, dynamic cube attack [DS11], fault attacks [BCC$^+$09], [KC11]. [BCC$^+$09] breaks the cipher by inducing faults in the LFSR of Grain-128, while [KC11] breaks the cipher by injecting faults in the NFSR of the cipher. Our design of NOCAS follows the structure of Grain-128, it replaces the LFSR and NFSR by a maximum length CA and a hybrid nonlinear CA. The nonlinear filter function is replaced by NMix, a nonlinear key mixing function used for block ciphers. NOCAS is shown to be resistant against fault attack and initialization becomes 4 times faster than Grain.

Cellular Automata were studied as a good pseudorandom sequence generator. The main requirement of a stream cipher is good pseudorandom generation. Also parallel operations of CA, which may give high troughput to ciphers. Rule-30 based CA was studied by Wolfarm as a pseudorandom generator. But it was later cryptanalyzed by Miere and Stafflebach [MS91] mainly due to its correlation. This shows that only nonlinear CA needs to be operated to reduce its correlation. [KMC10] studied few hybrid CA structures for cryptographic applications. It is shown that those CA can provide good cryptographic characteristics. In this paper, we have chosen one such hybrid CA rule for nonlinear mixing of key bits and a maximum length CA for linear mixing and high period. The cipher takes 128 bit key and 128 bit initialization vector (IV). It initializes in 64 cycles. Bits from hybrid nonlinear CA and the maximum length linear CA are combined with a nonlinear filter function NMix to produce output bit. In the current paper, we show that NOCAS is expected to have high security and provides security against known attacks.

The paper is organized as follows. Following the introduction, we briefly discuss the basic definitions regarding cellular automata (CA) and give a brief specification of Grain-128 cipher in section 2. NOCAS is proposed in section 3. Security analysis of NOCAS is studied in section 4. The hardware implementations of NOCAS and Grain-128 are compared in section 5. Finally, the paper is concluded in section 6.

2 Preliminaries

In this section, we provide definitions relating CA and cryptographic properties. We also give a brief specification of the Grain-128 stream cipher.

2.1 Basics of Cellular Automata

A cellular automaton is a finite array of cells. Each cell is a finite state machine $C = (\{0,1\}, f)$ where, f is a mapping $f : \{0,1\}^n \to \{0,1\}$. The mapping f, called local transition function. n is the number of cells the local transition function depends on. On each iteration the CA each cell of the CA updates itself with respective f.

The number of neighbouring cells, f depends on, may be same or different on different directions of the automaton. f may be same or different for cells across the automaton. The array of cells may be multi-dimensional. A 1-dimensional CA, each of whose rule depends on left and right neighbour and the cell itself is called a 3-neighbourhood CA. Similarly, if each cell depends on 2 left and 2 right neighbours and itself only, it is called 5-neighbourhood CA. A CA whose cells depend on 1 left and 2 right neighbouring cells is called a 4-neighbourhood right skew CA. A left skewed 4-neighbourhood CA can be defined similarly.

The state of the i^{th} cell at time $(t+1)$ depends on states of $(i-1)^{th}$, i^{th} and $(i+1)^{th}$ cells at time t. So, the local transition function for a 3-neighbourhood CA cell can be expressed as follows:

Tab. 1: Truth table for $f = q_{i-1}(t) \oplus q_i(t)$

Input	Output
000	0
001	0
010	1
011	1
100	1
101	1
110	0
111	0

$$q_i(t+1) = f[q_i(t), q_{i+1}(t), q_{i-1}(t)]$$

where, f denotes the local transition function realized with a combinational logic, and is known as a rule of CA [CCNC]. The decimal value of the truth table of the local transition function is defined as the *rule number* of the cellular automaton. For example, consider, $f = q_{i-1}(t) \oplus q_i(t)$. Its truth table is shown in tab. 1. Since the decimal equivalent of the output 00111100 is 60, rule number of f is, 60. Other examples are:
Rule 30: $f = q_{i-1}(t) \oplus (q_{i+1}(t) + q_i(t))$, where + is the Boolean 'or' operator and $\oplus$ is the Boolean 'xor' operator.
Rule 90: $f = q_{i-1}(t) \oplus q_{i+1}(t)$.
Rule 150: $f = q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)$.

If the rule of all the cells are the same then it is called uniform cellular automata, otherwise, it is called hybrid cellular automata. A 4 cell linear hybrid cellular automata is shown in Fig. 1. This work employs both linear and nonlinear CA. We define linear and nonlinear cellular automata below, before proceeding further.

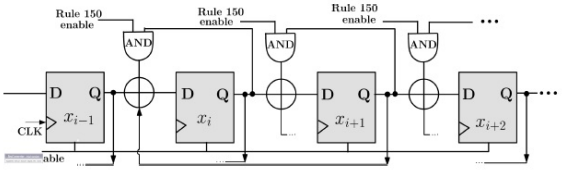


Fig. 1: A 4 Cell Linear Hybrid Cellular Automata based on Rules 90, 150

Definition 1 *Linear Cellular Automaton: A CA whose local transition function does not involve the '.' (Boolean and) operator in any of the cell is called the linear cellular automaton. For example, rule,* $f = q_{i-1}(t) \oplus q_{i+1}(t)$ *employed in each cell is a linear cellular automaton, where* $q_{i-1}(t)$ *and* $q_{i+1}(t)$ *denotes left and right neighbours of* i*-th cell at* t*-th instance of time.*

Definition 2 *Nonlinear Cellular Automaton: A CA whose local transition function is non-linear, i.e., involves at least one . operator, for at least one of the cells is a nonlinear cellular automaton. For*

example, rule, $f = q_{i-1}(t).q_{i+1}(t)$ employed in each cell is a nonlinear cellular automaton, where, $q_{i-1}(t)$ and $q_{i+1}(t)$ denotes left and right neighbours of the i^{th} cell at t^{th} instance of time.

2.2 Cryptographic Terms and Primitives

We next provide definitions of various terms and properties which Boolean functions should satisfy for cryptographic applications.

Definition 3 *Pseudorandom Sequence: An algorithmic sequence is pseudorandom if it cannot be distinguished from a truly random sequence by any efficient (polynomial time) probabilistic procedure or circuit.*

Definition 4 *Affine Function: A Boolean function which can be expressed as 'xor' ($\oplus$) of some or all of its input variables and a Boolean constant is an affine function.*

For example, $f(x_1, x_2) = x_1 \oplus x_2$ is an *affine function*, while the function, $f(x_1, x_2) = x_1 \oplus x_2 \oplus x_1.x_2$ is not an *affine function*, where, . is the Boolean 'and' operation and $\oplus$ is the Boolean 'xor' operation.

Definition 5 *Hamming Weight: Number of Boolean 1's in a Boolean function's truth table is called the Hamming weight of the function.*

Hamming weight of a function f is denoted as, $wt(f)$. For example, Hamming weight of $f(x_1, x_2) = x_1 \oplus x_2$ is, 2 and Hamming weight of $f(x_1, x_2) = x_1.x_2$ is 1.

Definition 6 *Balanced Boolean Function: If the Hamming weight of a Boolean function of n variables is 2^{n-1}, it is called a balanced Boolean function.*

Thus, $f(x_1, x_2) = x_1 \oplus x_2$ is balanced, while $f(x_1, x_2) = x_1.x_2$ is not balanced.

Definition 7 *Nonlinearity: Let, f be a Boolean function of variables, $x_1, x_2, \ldots x_n$ and A be the set of all affine functions in $x_1, x_2, \ldots x_n$. The minimum of the Hamming distances between f and the Boolean functions in A is the nonlinearity of f.*

Hence, nonlinearity of $f(x_1, x_2) = x_1.x_2$ is 1.

Definition 8 *Walsh Transform: Let $\bar{X} = (X_n, \ldots, X_1)$ and $\bar{\omega} = (\omega_1, \ldots \omega_n)$ both belong to $\{0,1\}^n$ and $\bar{X}.\bar{\omega} = X_n.\omega_1 \oplus \ldots X_1.\omega_n$. Let $f(\bar{X})$ be a Boolean function on n variables. Then the Walsh transform of $f(\bar{X})$ is a real valued function over $\{0,1\}^n$ that can be defined as $W_f(\bar{\omega}) = \Sigma_{\bar{X} \in \{0,1\}^n}(-1)^{f(\bar{X}) \oplus \bar{X}.\bar{\omega}}$. The Walsh transform is sometimes called the spectral distribution or simply the spectrum of a Boolean function.*

Definition 9 *Resiliency: A function $f(X_n \ldots X_1)$ is m-th order correlation immune (CI) iff its Walsh transform W_f satisfies $W_f(\bar{\omega}) = 0$; for $1 \leq wt(\bar{\omega}) \leq m$. Further, if f is balanced then $W_f(0) = 0$. Balanced m-th order correlation immune functions are called m-resilient functions. Thus, a function $f(X_n, \ldots, X_1)$ is m-resilient iff its Walsh transform W_f satisfies $W_f(\bar{\omega}) = 0$; for $0 \leq wt(\bar{\omega}) \leq m$.*

For example, resiliency of $f(x_1, x_2) = x_1 \oplus x_2$ is 1, but resiliency of $f(x_1, x_2) = x_1.x_2$ is 0.

Definition 10 *Algebraic Normal Form: Any Boolean function can be expressed as xor of conjunctions and a Boolean constant, True or False. This form of the Boolean function is called its Algebraic Normal Form (ANF).*

Every *Boolean function* can be expressed in *ANF*. As an example, $f(x_1, x_2, x_3) = x_1.x_2.x_3$ is in *ANF*, while $f(x_1, x_2, x_3) = (x_1 \oplus x_2).(x_2 \oplus x_3)$ is not in *ANF*. Its *ANF* representation is, $f(x_1, x_2, x_3) = x_1.x_2 \oplus x_1.x_3 \oplus x_2 \oplus x_2.x_3$.

Definition 11 *Algebraic Degree: The maximum number of literals in any conjunction of ANF of a Boolean function is called its degree. Ciphers expressible or conceivable as a Boolean function have algebraic degree which is the same as the degree of the ANF of the Boolean function.*

Thus, $f(x_1, x_2) = x_1 \oplus x_2 \oplus x_1.x_2$ has algebraic degree 2.

Next, we outline a test which has been developed to distinguish a given Boolean function from a truly random function.

2.3 d-Monomial Test

d-Monomial test is a statistical test for pseudorandomness proposed independently in [Saa] and [EJT]. It investigates the Boolean function representation of each output bit in terms of input bits. If a Boolean function of n Boolean variables is a good pseudorandom sequence generator, then it will have $\frac{1}{2}\binom{n}{d}$ d-degree monomials. A deviation will indicate non-randomness. For example, consider the function $f(x_1, x_2) = x_1 \oplus x_2$, it has 2, 1-degree monomials and 0, 2 degree monomial. It turns out that it has 1, 1-degree monomial more, hence it is expected to be non-pseudorandom. On the other hand $f(x_1, x_2) = x_1$ is expected to be a good pseudorandom generator.

In spite of its simplicity, this test gained huge appreciation in cryptography community. It proved to be a good tool in analyzing the degree of pseudorandomness of cryptographic systems. To the best of our knowledge, d-monomial test has not been applied to CA configurations previously. We explore different CA configurations under this test.

2.4 Specification of the Grain-128 Stream Cipher

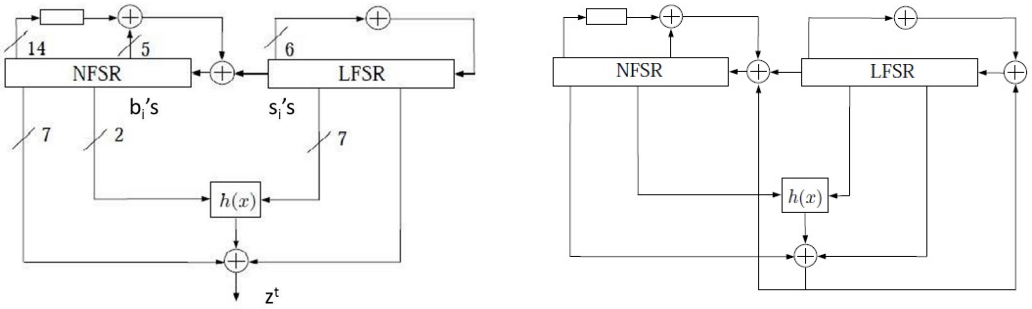
Grain-128 is a hardware based stream cipher enlisted in the final list of the eStream [est] project. We briefly describe the specification of the Grain-128 stream cipher here. A detailed description may be found in [HJM].

The Grain-128 stream cipher consists of three main building blocks, namely, an NFSR, an LFSR and an output function $h(x)$(Fig. 2(a)). The contents of the NFSR are denoted by $b_i, b_{i+1}, \ldots, b_{i+127}$ and the contents of the LFSR are denoted by, $s_i, s_{i+1}, \ldots, s_{i+127}$. The update function of the LFSR is given by,
$s_{i+128} = s_i + s_{i+7} + s_{i+38} + s_{i+70} + s_{i+81} + s_{i+96}$
The NFSR is updated by,

$$\begin{aligned} b_{i+128} &= s_i + b_i + b_{i+26} + b_{i+56} + b_{i+91} + b_{i+96} + b_{i+3}b_{i+67} + b_{i+11}b_{i+13} \\ &\quad + b_{i+17}b_{i+18} + + b_{i+27}b_{i+59} + b_{i+40}b_{i+48} + b_{i+61}b_{i+65} + b_{i+68}b_{i+84} \end{aligned}$$

The NFSR and the LFSR together represent the internal state of the cipher. A nonlinear filter function h is defined with 2 input bits from the NFSR and 7 input bits from the LFSR. The function h is defined by:
$h = b_{i+12}s_{i+8} + s_{i+13}s_{i+20} + b_{i+95}s_{i+42} + s_{i+60}s_{i+79} + b_{i+12}b_{i+95}s_{i+95}$.
The output function z^t is defined as,
$z^t = b_{t+2} + b_{t+15} + b_{t+36} + b_{t+45} + b_{t+64} + b_{t+73} + b_{t+89} + h + s_{t+93}$

An initialization phase is carried out before the cipher generates keystream bits. The 128 bit key, $k = (k_1, k_2, \ldots, k_{128})$ and the 96 bit initialization vector $IV = (IV_1, IV_2, \ldots, IV_{96})$ is loaded in the



(a) Operation of Grain-128 (b) Initialization of Grain-128

Fig. 2: The Grain-128 stream cipher

NFSR and the LFSR respectively as, $b_i = k_i, 1 \leq i \leq 128$ and $s_i = IV_i, 1 \leq i \leq 96$, rest of the LFSR bits, $(s_{97}, s_{98}, \ldots, s_{128})$ are loaded with 1. The cipher is run for 256 rounds without producing any keystream, during initialization the output function is fed back and xored with both the LFSR and the NFSR (Fig. 2(b)).

3 NOCAS: A CA Based Stream Cipher

In the previous section, we have seen structure of Grain-128. The cipher is simple in design consisting of only a LFSR and an NFSR. It is a lightweight cipher with fast startup and high throughput. Unfortunately, a number of attacks have been mounted on it [DS11], [BCC$^+$09], [KC11]. In [BCC$^+$09], faults are injected in the LFSR to deduce full secret key in only 22 faults, while [KC11] induces faults in the NFSR to get back the secret key in maximum 256 faults. In this section we present the specification of the cipher NOCAS (Hybrid **NO**nlinear **CA** based **S**tream Cipher), with by replacing the LFSR with a linear maximum length CA and the NFSR with a hybrid nonlinear CA.

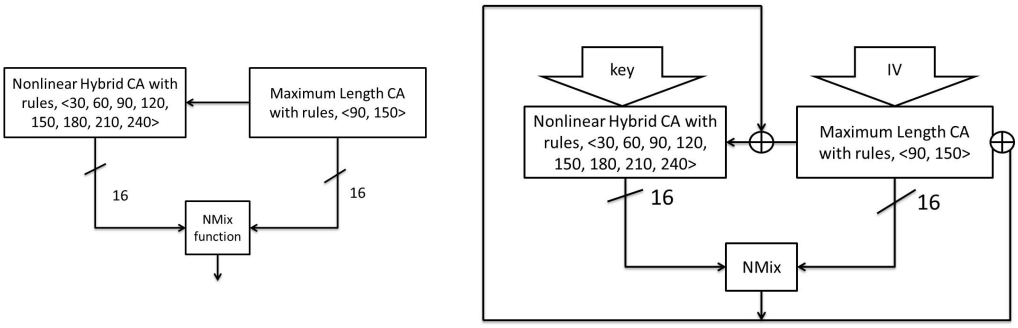
The building blocks of NOCAS are:

- A Hybrid Nonlinear CA of 128-bits with rules $< 30, 60, 90, 120, 150, 180, 210, 240 >$ repeated 16 times.
- A Linear Maximum Length CA of 128 bits with combinations of rules 90 and 150.
- The function NMix which is cryptographically suited nonlinear mixing function proposed in [BC09].

A block diagram of NOCAS is given in figure 3(a). Each of the building blocks are discussed in the following subsections.

3.1 Hybrid Nonlinear CA

In [KMC10], a number of cellular automata have been synthesized and their cryptographic properties have been studied. The authors have identified six hybrid nonlinear hybrid CAs (ruleset 1 to 6) which are cryptographically robust. Among these rulesets we choose ruleset 5 i.e., $30, 60, 90, 120, 150, 180, 210, 240$.



(a) Structure of NOCAS (b) Initialization of NOCAS

Fig. 3: The NOCAS Stream Cipher

The algebraic normal form of the rules used in ruleset 5 is shown in table 2. We briefly discuss the cryptographic properties of ruleset 5 CA next. In our design, we use null-boundary ruleset 5 CA. Ruleset 5 CA consists of cells operating on rules $30, 60, 90, 120, 150, 180, 210, 240$ spaced alternatively. The nonlinear register of NOCAS is of 128 bits, hence, 16 such hybrid CA cells are repeated in the design.

Tab. 2: ANF of 3-nbd Rules used in Ruleset 5

Rule #	ANF	Linear?
30	$(x_2.x_3) \oplus x_1 \oplus x_2 \oplus x_3$	No
60	$x_1 \oplus x_2$	Yes
90	$x_1 \oplus x_3$	Yes
120	$x_1 \oplus (x_2.x_3)$	No
150	$x_1 \oplus x_2 \oplus x_3$	Yes
180	$x_1 \oplus x_2 \oplus (x_2.x_3)$	No
210	$x_1 \oplus x_3 \oplus (x_2.x_3)$	No
240	x_1	Yes

In [KMC10] ruleset 5 is tested over three iterations for d-monomial test. We reproduce the result in table 3. Here, ruleset 5 is tested over three iterations for the cryptographic properties like, balancedness, nonlinearity, resiliency and algebraic degree (tab. 4).

It can be seen that over all the iterations, the CA generates balanced output and has a fast nonlinearity growth. Resiliency of the CA is constant, it has good algebraic degree which also increases fast with the iterations. Also, results of d-monomial test is satisfactory.

Tab. 3: d-Monomial Characteristics of Hybrid Ruleset 5 CA [KMC10]

	Number of n^{th} degree terms			
Rules	*1*	*2*	*3*	*4*
Ideal	*1,2,3*	*1,5,10*	*0,5,52*	*0,2,52*
Ruleset 5	3,2,4	1,3,5	0,2,6	0,0,3

Tab. 4: Cryptographic Properties of Ruleset 5

Iteration	**Balancedness**	**Nonlinearity**	**Resiliency**	**Degree**
1	Balanced	2	2	2
2	Balanced	8	2	3
3	Balanced	32	2	4

3.2 Maximum Length Linear CA

It is shown by researchers that $90, 150$ hybrid linear CA produces maximum length cycle for any CA length [CCNC]. In our design again we use null boundary 90, 150 hybrid CA, which is CA cells operating with rules 90 and 150 in such an arrangement so as to produce maximum length structure, and the end-cells are connected to nulls. It is known that such maximum length linear CA produces excellent pseudorandom sequences. The leftmost bit of this CA is fed to the rightmost bit position of hybrid CA in the structure of NOCAS. Clearly due to maximality of linear part and the design of NOCAS up to 2^{128} different states will be present NOCAS, which makes it possible to generate 2^{128} unique keystream bits.

3.3 NMix

NMix introduced in [BC09] is used to combine bits from hybrid CA and maximum length CA. The function possesses good cryptographic properties.

Definition 12 *For two n-bit inputs X and Y, the output Z given by NMix is defined as follows,*
$z_i = x_i \oplus y_i \oplus c_{i-1}$
$c_i = \oplus_{j=0}^{i} x_j y_j \oplus x_{i-1} x_i \oplus y_{i-1} y_i$
where, $0 \leq i \leq n-1, c_{-1} = 0, x_{-1} = 0, y_{-1} = 0$.

We use 16 bits each from nonlinear part and linear parts of NOCAS as input to NMix and take the MSB as its output. Due to this design all 16 input bits from nonlinear and linear parts are mixed fully in the output. The output function is clearly a 32 variable bent function having degree 2, hence, providing high nonlinearity. The 16 input bits from nonlinear and linear parts are chosen as bits, $1, 10, 19, 28, 37, 46, 55, 64, 65, 74, 83, 92, 101, 110, 119, 128$.

3.4 Initialization

Key and initialization vector (IV) are input to the nonlinear and linear parts of the cipher in 128 bits each. So that $n_i = k_i, 1 \leq i \leq 128$, where n_i is the nonlinear register and k_i is the i^{th} key bit, while, $l_i = v_i, 1 \leq i \leq 128$, where l_i is the linear register and v_i is the i^{th} IV bit. Once, key and IV are setup in respective registers, the cipher is clocked for 64 cycles without producing any keystream and the keystream is XORed with both the MSB of nonlinear and linear registers fig. 3(b).

4 Security Analysis of NOCAS

In this section, we present the security analysis of NOCAS. We will see that the employment of hybrid nonlinear CA provides resistance against popular existing attacks.

Tab. 5: Cryptographic Characteristics of NOCAS

Iteration	*Balancedness*	*Nonlinearity*	*Algebraic Degree*	*Resiliency*
1	Balanced	538	3	2
2	Balanced	1842	4	3
3	Balanced	5648	5	3
4	Balanced	12428	6	4

- *Linear Cryptanalysis*: Nonlinearity and resiliency are the most important requirements for a cryptographic system. Good nonlinearity characteristics indicate that the cipher is expected to be safe against linear cryptanalysis and also from algebraic attacks. Table 5 shows the nonlinearity of NOCAS with pass of iteration. In only 4 cycles of operation nonlinearity of NOCAS reaches 12428. It can be noted that the growth rate of nonlinearity is very steep. As complexity of linear cryptanalysis is directly related to nonlinearity, it can be claimed that NOCAS is resistant against linear cryptanalysis.
- *Correlation Attack:* Good nonlinearity characteristics does not imply correlation immunity, ie, good nonlinear ciphers can display correlations among key, plain-texts and cipher-texts, which is the basis of correlation attack. Also, balancedness is an important factor to prevent correlation attack. Table 5 illustrates the balancedness of the NOCAS output bit with iterations. All the output bit expressions are balanced in the initial 4 iterations. Hence, the cryptographic property balancedness holds good for NOCAS. Table 5 also tabulates the resiliency of NOCAS output bit with iterations. It reveals that higher resiliency is achieved by NOCAS at much lower number of iterations. Due to the faster growth of resiliency of output bit of NOCAS and its balancedness, it is expected to show resistance against correlation attacks.
- *Algebraic Attacks*: Algebraic cryptanalysis is dependent on the algebraic degree of a cipher. The increase of number of nonlinear terms of a cipher also increase the attack complexity. Table 5 shows the growth of algebraic degree of the output bit of NOCAS with iterations, while table 6 shows d-monomial characteristics of NOCAS with iterations, which shows almost exponential growth in nonlinear terms. It can be observed that in NOCAS the algebraic degree increases linearly. The growth in number of terms in the resultant Boolean expression and the number of different degree terms in the output equation are both high. Considering table 6 once again, note that, at iteration 4 only the number of nonlinear terms in the expression of the output bit is more than 400, which is more than double the number of nonlinear terms at iteration 3, it can be expected that any attempt to linearize the expression for algebraic attack will have to deal with exponential number of nonlinear terms with pass of iterations. Hence, algebraic attacks are not expected to yield good result against NOCAS. Ciphers having large algebraic degrees are resistant against linearization and algebraic attacks. So, NOCAS is expected to be resistant to these attacks both in reduced round version and the full key-IV setup version.

Tab. 6: d-monomial Test Result of NOCAS

Iteration	*Deg.-1*	*Deg.-2*	*Deg.-3*	*Deg.-4*	*Deg.-5*	*Deg.-6*
1	18	32	4	0	0	0
2	24	48	16	2	0	0
3	34	94	26	12	1	0
4	56	168	128	56	42	6

- *Scan-based Side Channel Attack*: Scan-chain based attack works because of the invertibility of the states of the cipher. The same will not be possible for NOCAS because of the presence of non-invertible CA rule 30. Though rule 30 is partially reversible, presence of linear and nonlinear rules in the CA configuration reduces the probability of the reversion exponentially with iterations. Hence, scan-based side channel attack will not be successful on NOCAS.

- *Cube Attack/AIDA attack*: Till date the most successful attacks on stream ciphers were cube attack and dynamic cube attack [DS11]. This attack exploits the fact that the distribution of the d-degree terms is far from ideal in d-monomial test. We tabulate in table 6 the d-monomial test values for the first 4 iterations of the output bit of NOCAS. This kind of distribution is expected to resist higher order differential attacks and distinguishers. The overall d-monomial characteristics of NOCAS is fairly good in view of the number of terms in middle degrees, presence of linear and highest degree terms. A large algebraic degree of a cipher will prevent the attack from practically being implemented. In case of NOCAS, the d-monomial test result is fairly good and the high algebraic degree growth rate is also an important factor in prevention of the attack on NOCAS. Hence, cube attack on NOCAS will not be successful on any reasonable number of rounds.

- *Fault Attack:* Fault attacks induce faults in the cipher registers and exploits the difference of faulty and fault-free cipher-text to deduce the secret key. In case of NOCAS, the design is such that it is difficult to produce linear or low-degree equations from faulty and fault-free cipher-texts. Hence, solving such a system is a hard problem. Therefore, fault attack is expected not to succeed against NOCAS.

5 Comparison of NOCAS with Grain-128

Both the ciphers, NOCAS and Grain-128 are synthesized on Xilinx 8.1 Vertex 4 FPGA. Table 7 compares the performances of NOCAS and Grain-128. The result shows that Grain-128 is hardware efficient than NOCAS while throughput is comparable. NOCAS achieves 4 times speedup in startup than Grain-128.

Tab. 7: Comparison of NOCAS and Grain-128

	No. of LUTs	**Throughput**	**Setup**
Grain-128	278	390 Mb/s	256 cycles
NOCAS	562	372 Mb/s	64 cycles

6 Conclusion

In the current paper, we have introduced a new stream cipher based on hybrid nonlinear CA called NOCAS. The design produces fast initialization in only 64 cycles. We have analyzed the cryptographic properties like balancedness, nonlinearity, resiliency and algebraic degree of NOCAS, which show it is a cryptographically robust cipher. The d-monomial test also produce fairly good result against NOCAS. Finally, we have shown that NOCAS is expected to resistant against popularly known existing attacks. It achieves 4 times speedup in initialization than Grain-128.

References

[BBC$^+$] Côme Berbain, Olivier Billet, Anne Canteaut, Nicolas Courtois, Henri Gilbert, Louis Goubin, Aline Gouget, Louis Granboulan, Cédric Lauradoux, Marine Minier, Thomas Pornin, and Herve Sibert. Sosemanuk, a fast software-oriented stream cipher. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

[BC09] Jaydeb Bhowmik and Dipanwita Roy Chowdhury. Nmix : An Ideal Candidate for Key Mixing. *SecCrypt 2009*, pages 285–288, 2009.

[BCC$^+$] Steve Babbage, Christophe De Canniere, Anne Canteaut, Carlos Cid, Henri Gilbert, Thomas Johansson, Matthew Parker, Bart Preneel, Vincent Rijmen, and Matthew Robshaw. The estream portfolio. *"http://www.ecrypt.eu.org/stream/portfolio.pdf"*.

[BCC$^+$09] Alexandre Berzati, Cecile Canovas, Guilhem Castagnos, Blandine Debraize, Louis Goubin, Aline Gouget, Pascal Paillier, and Stephanie Salgado. Fault analysis of grain-128. *Hardware-Oriented Security and Trust, IEEE International Workshop on*, 0:7–14, 2009.

[BD] Steve Babbage and Matthew Dodd. The stream cipher mickey 2.0. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

[Ber] Daniel J. Bernstein. Salsa20. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

[BVCZ] Martin Boesgaard, Mette Vesterager, Thomas Christensen, and Erik Zenner. The stream cipher rabbit. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

[CCNC] P. Pal Chaudhuri, D. Roy Chowdhury, S. Nandi, and S. Chattopadhyay. CA and Its Applications: A Brief Survey, Additive Cellular Automata - Theory and Applications vol.-1, pages 6-25. *eSTREAM, ECRYPT Stream Cipher Project*, 1997.

[CP] Christophe De Canniere and Bart Preneel. Trivium specifications. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

[DS11] Ita Dinur and Adi Shamir. Dynamic Cube Attack on Full Grain-128. *ePrint Cryptology Archive*, 2011.

[EJT] H. Englund, T. Johansson, and MS Turan. A Framework for Chosen IV Statistical Analysis of Stream Ciphers. *Progress in Cryptology - INDOCRYPT*, 2007:268–281.

[est] The estream project. *"http://www.ecrypt.eu.org/stream/"*.

[HJM] Martin Hell, Thomas Johansson, and Willi Meier. A stream cipher proposal: Grain-128. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

[KC11] Sandip Karmakar and Dipanwita Roy Chowdhury. Fault Analysis of Grain-128 by Targeting NFSR. *AfricaCrypt 2011*, 2011.

[KMC10] Sandip Karmakar, Debdeep Mukhopadhyay, and Dipanwita Roy Chowdhury. d-monomial Tests on Cellular Automata for Cryptographic Design. *ACRI 2010*, 2010.

[MS91] Meier and Staffelbach. Analysis of Pseudo Random Sequences Generated by Cellular Automata. *"EUROCRYPT: Advances in Cryptology: Proceedings of EUROCRYPT"*, 1991.

[Saa] Markku-Juhani O. Saarinen. Chosen IV Statistical Attacks on eStream Stream Ciphers. *http://www.ecrypt.eu.org/stream*.

[Wola] Wolfram. Cryptography with Cellular Automata. *CRYPTO: Proceedings of Crypto*, 1985.

[Wolb] S. Wolfram. Random Sequence Generation by Cellular Automata. *Advances in Applied Mathematics, vol.-7, pages 123-169*.

[Wu] Hongjun Wu. Stream cipher hc-128. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.