A New Distinguisher on Grain v1 for 106 rounds

Santanu Sarkar

Department of Mathematics,
Indian Institute of Technology,
Sardar Patel Road, Chennai 600036, India.
sarkar.santanu.bir@gmail.com

Abstract. In Asiacrypt 2010, Knellwolf, Meier and Naya-Plasencia proposed distinguishing attacks on Grain v1 when (i) Key Scheduling process is reduced to 97 rounds using 2^{27} chosen IVs and (ii) Key Scheduling process is reduced to 104 rounds using 2^{35} chosen IVs. Using similar idea, Banik obtained a new distinguisher for 105 rounds. In this paper, we show similar approach can work for 106 rounds. We present a new distinguisher on Grain v1 for 106 rounds with success probability 63%.

Keywords: Differential Cryptanalysis, Distinguisher, Grain v1, Stream Cipher

1 Introduction

The Grain v1 is a well-known hardware-efficient, synchronous and bit oriented stream cipher. Designed in 2005 by Hell, Johansson and Meier [17], it has been widely studied for nearly a decade mostly because of its simplistic structure and selection in the eStream hardware profile (profile 2) portfolio [13]. In order to prevent the correlation attacks [6] on Grain v0, the modified versions Grain v1 [17] was proposed after incorporating certain changes. Grain 128 and Grain 128a are inspired from Grain v1, and use a similar structure.

Küçük et al. [8] proposed related key-IV attack on Grain v1. They observed that for any (K, IV) pair, there exist related (K', IV') pair with probability 0.25 that generates 1-bit shifted keystream. Bjørstad [7] showed that Grain v1 has a low resistance to BWS sampling. Other cryptanalytic results related to this cipher have been presented in [14, 15, 19, 24, 26, 27].

In [9], an attack on nonlinear filter generators with linear resynchronization and filter function with few inputs is presented. To avoid such attacks, the initialization of stream ciphers should be designed carefully. The common design paradigm (including the Grain family) of stream ciphers is as follows. The key K and initialization vector IV are loaded into the state along with some padding bits. Next, state update function is applied to the internal state iteratively for a number of rounds without producing any output (key-stream). Hence, the number of rounds is important for both security and efficiency of the cipher, since increasing the number of rounds will slow down the cipher, but at the same time likely to increase the security. Hence, finding the minimal number of rounds that would ensure the conjectured security level is a critical task, and studying the ciphers in its reduced variant (i.e., treating as if the key-streams are avaiable just after the key & IV are loaded to the register).

Trivium [18], another candidate in the hardware profile of eStream, has been cryptanlysed for reduced round by many researchers. Englund et al. [14] showed statistical weaknesses on Trivium for 736 rounds. Aumasson et al. [1] were able to build a distinguisher on Trivium after 790 round. Independently Knellwolf et al. [21] built a distinguisher up to 806 rounds.

Grain v1 is studied extensively for reduced round. In [2], a non-randomness for 81 round has been reported. In [20], Knellwolf et al. proposed a distinguisher for 97 rounds and 104 rounds. However results of [20] were based on experiments only. Later, Banik [3] proved a theoretical result for 97 rounds. Recently a distinguisher for 105 round has been proposed in [4]. These attacks on Grain v1 are known as *Conditional Differential Cryptanalysis* (CDC), which was first introduced by Ben-Aroya and Biham [5] for block cipher cryptanalysis. It studies the output frequency of derivatives of output bit on specifically chosen IV.

However, in recent terminology, CDC on stream cipher can be described as dynamic cube attack. Cube attacks, introduced by Dinur and Shamir [11], have been used in cryptanalysis. Although cube attack works [10, 12] successfully on Grain 128, its performance on Grain v1 is not that effective. Using CDC, Knellwolf et al., in their Asiacrypt 2010 paper [20] obtained a practical distinguisher on Grain 128 for 215 rounds. Higher order conditional differential attacks on Trivium and Grain 128 have been studied in [22]. CDC has been applied successfully in [23] on Grain 128a. In this paper, we show that one can attack Grain v1 up to 106 rounds using CDC method.

The paper is organized as follows. In Section 2, we describe the design of Grain v1. We present our experimental results in Section 3. Section 4 gives a new distinguisher on Grain v1 up to 106 rounds. Conclusion is presented in Section 5.

2 Brief Description of Grain v1

Grain v1 has 80 bit key K and 64 bit initialization vector IV. The structure of the Grain v1 is depicted in Fig. 1. The state consists an 80-bit LFSR and an 80-bit NFSR. The update function of the LFSR is given by: $y_{t+80} = f(Y_t)$, where $Y_t = [y_t, y_{t+1}, \ldots, y_{t+79}]$ is an 80-bit vector that denotes the LFSR state at the t^{th} clock interval and f is a linear function on the LFSR state bits obtained from a primitive polynomial in $GF(2)$ of degree 80. The NFSR state is updated as $x_{t+80} = y_t \oplus g(X_t)$. Here, $X_t = [x_t, x_{t+1}, \ldots, x_{t+79}]$ is an 80-bit vector that denotes the NFSR state at the t^{th} clock interval and g is a non-linear function of the NFSR state bits.

The output keystream is produced by combining the LFSR and NFSR bits as $z_t = h'(X_t, Y_t) = \bigoplus_{a \in A} x_{t+a} \oplus h(X_t, Y_t)$, where A is some fixed subset of $\{0, 1, 2, \ldots, n-1\}$. Below we present the detailed description.

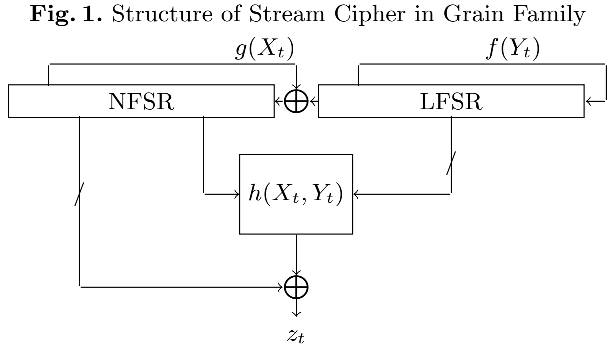
As stated, the key-stream generation of Grain v1 consists of three phases. In the first phase, the key & IV bits are loaded to the state register in the *Key Loading Algorithm* routine; then the state bits are updated during the *Key Scheduling Algorithm* routine; and next the *Pesudo-Random Generation Algorithm* routine produces the key-streams. These routines are described as follows.

Key Loading Algorithm (KLA) The key (80-bits) is loaded in the NFSR and the IV(64-bits) is loaded in the 0^{th} to the 63^{th} bits of the LFSR. The remaining 64^{th} to 79^{th} bits of the LFSR are loaded with 1.

Key Scheduling Algorithm (KSA) After the KLA, for the first 160 clocks, the keystream produced at the output point of the function h' is XOR-ed to both the LFSR and NFSR update functions. So during the first 160 clock intervals, the LFSR and the NFSR bits are updated as $y_{t+80} = z_t \oplus f(Y_t)$, $x_{t+80} = y_t \oplus z_t \oplus g(X_t)$.

Pseudo-Random keystream Generation Algorithm (PRGA) After the completion of the KSA, z_t is no longer XOR-ed to the LFSR and the NFSR but it is used as the Pseudo-Random keystream bit. Hence in this phase, the LFSR and NFSR are updated as $y_{t+80} = f(Y_t), x_{t+80} = y_t \oplus g(X_t)$.

Fig. 1. Structure of Stream Cipher in Grain Family



The LFSR update rule is given by $y_{t+80} = y_{t+62} \oplus y_{t+51} \oplus y_{t+38} \oplus y_{t+23} \oplus y_{t+13} \oplus y_t$. The NFSR state is updated as $x_{t+80} = y_t \oplus g(x_{t+63}, x_{t+62}, x_{t+60}, x_{t+52}, x_{t+45}, x_{t+37}, x_{t+33}, x_{t+28}, x_{t+21}, x_{t+15}, x_{t+14}, x_{t+9}, x_t)$, where,

$g(x_{t+63}, x_{t+62}, x_{t+60}, x_{t+52}, x_{t+45}, x_{t+37}, x_{t+33}, x_{t+28}, x_{t+21}, x_{t+15}, x_{t+14},$
$x_{t+9}, x_t)$
$= x_{t+62} \oplus x_{t+60} \oplus x_{t+52} \oplus x_{t+45} \oplus x_{t+37} \oplus x_{t+33} \oplus x_{t+28}$
$\oplus x_{t+21} \oplus x_{t+14} \oplus x_{t+9} \oplus x_t \oplus x_{t+63}x_{t+60} \oplus x_{t+37}x_{t+33} \oplus x_{t+15}x_{t+9}$
$\oplus x_{t+60}x_{t+52}x_{t+45} \oplus x_{t+33}x_{t+28}x_{t+21} \oplus x_{t+63}x_{t+45}x_{t+28}x_{t+9}+$
$x_{t+60}x_{t+52}x_{t+37}x_{t+33} \oplus x_{t+63}x_{t+60}x_{t+21}x_{t+15}$
$\oplus x_{t+63}x_{t+60}x_{t+52}x_{t+45}x_{t+37} \oplus x_{t+33}x_{t+28}x_{t+21}x_{t+15}x_{t+9}$
$\oplus x_{t+52}x_{t+45}x_{t+37}x_{t+33}x_{t+28}x_{t+21}.$

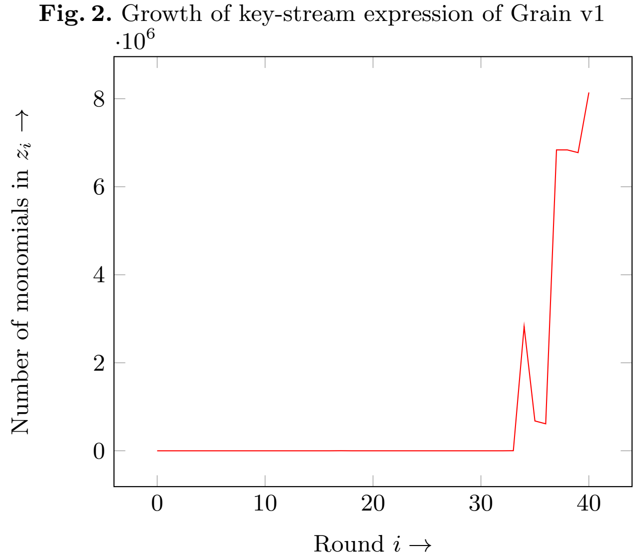
The key-stream is produced by combining the LFSR and NFSR bits as:

$$z_t = \bigoplus_{a \in A} x_{t+a} \oplus h(y_{t+3}, y_{t+25}, y_{t+46}, y_{t+64}, x_{t+63}),$$

where, $A = \{1, 2, 4, 10, 31, 43, 56\}$ and $h(s_0, s_1, s_2, s_3, s_4) = s_1 \oplus s_4 \oplus s_0 s_3 \oplus s_2 s_3 \oplus s_3 s_4 \oplus s_0 s_1 s_2 \oplus s_0 s_2 s_3 \oplus s_0 s_2 s_4 \oplus s_1 s_2 s_4 \oplus s_2 s_3 s_4$.

3 Biases Beyond 105 Rounds of KSA

Fig. 2. Growth of key-stream expression of Grain v1



As evident from the description, the NFSR update function used in Grain v1 is of degree 6. So symbolic expressions (treating the key & IV as symbolic variables and then doing the state update operation) of Grain v1 grow very fast. In Fig. 2, we show the number of monomials in key-stream expression of Grain v1 over some initial rounds.

As mentioned, Knellwolf et al. [20] observed a new distinguisher on Grain v1. Now we briefly explain how one can interpret the idea of [20] as a dynamic cube attack. Recall from Section 2 that Grain v1 contains 80-bit key $k_0, \ldots, k_{79}$ and 64-bit IV $v_0, \ldots, v_{63}$. Grain v1 is initially loaded with $X_0 = [k_0, \ldots, k_{79}]$ and $Y_0 = [v_0, \ldots, v_{63}, \overbrace{1, \ldots, 1}^{16}]$ (here X_0 corresponds to NFSR and Y_0 corresponds to LFSR).

Next start with NFSR $X_0' = [k_0, \ldots, k_{79}]$ but different LFSR $Y_0' = [v_0, \ldots, 1 \oplus v37, v_{63}, \overbrace{1, \ldots, 1}^{16}]$. That is, in cube attack terminologies, v_{37} is chosen as cube. Thus two states S_0 and S_0' initialized by (X_0, Y_0) and (X_0', Y_0') are different only at one position. Suppose z_i and z_i' are the key stream bits for S_0 and S_0' respectively at i-th round of KSA. They observed experimentally that if $z_{12} = z_{12}', z_{34} = z_{34}', z_{40} = z_{40}'$ in KSA and KSA is reduced to 97 rounds, the first output bit in PRGA will be same with probability more than 0.5. In ACISP 2014, Banik [3] gave the theoretical justification for this result.

Recently, Banik [4] showed a distinguishing attack for 105 round. Instead of 37-th bit of IV, he chose 61-bit of IV for the differential. In his work, it is considered the equality of $z_{15} = z_{15}', z_{36} = z_{36}', z_{39} = z_{39}'$ and $z_{42} = z_{42}'$ in KSA.

In this paper, we experiment for all single IV differential. Thus we have a total of 64 differentials. For any such differential, in the initial rounds of KSA, it is highly likely that $z_i = z_i'$ is satisfied. We load symbolically with $X_0 = [k_0, \ldots, k_{79}]$ in NFSR and $Y_0 = [v_0, \ldots, v_{63}, \overbrace{1, \ldots, 1}^{16}]$ in Sage [25]. Next we run KSA for few rounds, and find z_i as a polynomial of $k_0, \ldots, k_{79}, v_0, \ldots, v_{63}$. For each v_j, we identify first four rounds where coefficient of v_j in z_i is not constant for $0 \leq j \leq 63$. We identify these rounds using Algorithm 1. In step 3 of the algorithm, $\mathcal{I}_A$ corresponds to the ideal generated by a set of polynomials in A.

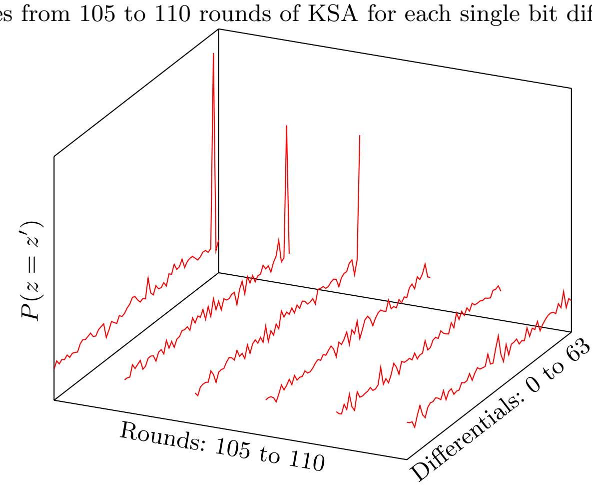
Input: v_j, z_i and an empty array A
Output: An array A

1 $i = 0$;
2 **while** $\left(\textit{Coefficient } c_{ij} \textit{ of } v_j \textit{ in } z_i \textit{ is nonconstant } \& \ |A| < 4 \right)$ **do**
3   **if** $c_{i,j} \notin \mathcal{I}_A$ **then**
4   Include $c_{i,j}$ in A ;
    end
5     $i = i + 1$;
end

Algorithm 1 Generating polynomial equations in KSA

Conditions for each differential are presented in Appendix A. We find the probability of the equality of the first output keystream bits for each KSA round 105 to 128. Our probability is taken over 2^{30} random key-IV.

Fig. 3. Basies from 105 to 110 rounds of KSA for each single bit differential on IV



Our experimental values have been presented in Fig.3 for rounds 105 to 110. Here x axis corresponds to the rounds of KSA, y corresponds to each differential and z corresponds the equality of output keystream bits. From the Fig. 3, it is clear we may get distinguisher using the similar idea of [20] for 106 and 107 rounds. In fact, we observe

$$P\big(z_{105} = z'_{105} \mid z_{15} = z'_{15} \ \&\ z_{36} = z'_{36} \ \&\ z_{39} = z'_{39} \ \&\ z_{42} = z'_{42}\big) = 0.500365,$$

$$P\big(z_{106} = z'_{106} \mid z_{16} = z'_{16} \ \&\ z_{34} = z'_{34} \ \&\ z_{37} = z'_{37} \ \&\ z_{40} = z'_{40}\big) = 0.500245,$$

$$P\big(z_{107} = z'_{107} \mid z_{17} = z'_{17} \ \&\ z_{35} = z'_{35} \ \&\ z_{38} = z'_{38} \ \&\ z_{41} = z'_{41}\big) = 0.500246,$$

when differentials are given on v_{61}, v_{62} and v_{63} respectively.

After 107 rounds, all curves become almost flat. Thus it seems beyond 107 rounds, it might not be possible to attack Grain v1 using single differentiable.

4 New result on Grain v1: Distinguisher upto 106 rounds

Grain v1 is first intialised with $X_0 = [k_0, \ldots, k_{79}]$ and $Y_0 = [v_0, \ldots, v_{63}, \overbrace{1, \ldots, 1}^{16}]$. Here X_0 corresponds to NFSR and Y_0 corresponds to LFSR.

Now choose v_{62} as cube. Hence start with NFSR $X'_0 = [k_0, \ldots, k_{79}]$ but different LFSR $Y'_0 = [v_0, \ldots, 1 \oplus v_{62}, v_{63}, \overbrace{1, \ldots, 1}^{16}]$.

Thus two states S_0 and S'_0 initialized by (X_0, Y_0) and (X'_0, Y'_0) different only at one position. But when more and more KSA rounds are completed, more and more positions of the states will differ. The idea is to delay the diffusion of the differential for as many KSA rounds as possible, by imposing many algebraic conditions over key and IV. We find algebraic expressions using Sage [25]. The conditions may be classified in to two types:

- **Type 1:** Conditions only on IV
- **Type 2:** Conditions on both Key and IV.

Let z_t and z'_t be the bit produced in the t-th KSA round when states are loaded by (X_0, Y_0) and (X'_0, Y'_0). Recall for r-th reduced version of Grain v1, all bits z_i, z'_i are unknown to the attacker for $i < r$. But giving Type 1 and Type 2 conditions, attacker can guarantee that $z_i \oplus z'_i = 0$ for few initial rounds. The attack idea is as follows:

1. For $i = 0, \ldots, 15$, it is not difficult to show that $z_i = z'_i$. Hence we do not need any condition to make $z_i \oplus z'_i = 0$ for $0 \leq i \leq 15$.
2. When $i = 16$, $z_i \oplus z'_i$ is polynomial degree 2 over Key and IV. Now we set $v_{19} = v_{41} = 1, v_{46} = 0$ and $v_0 = k_1 \oplus k_2 \oplus k_4 \oplus k_{10} \oplus k_{31} \oplus k_{43} \oplus k_{56} \oplus v_3 \oplus v_{13} \oplus v_{23} \oplus v_{25} \oplus v_{38} \oplus v_{51}$. Then $z_{16} = z'_{16}$. Thus we have three Type 1 conditions $v_{19} = v_{41} = 1, v_{46} = 0$ and one Type 2 condition $C_1 : v_0 = k_1 \oplus k_2 \oplus k_4 \oplus k_{10} \oplus k_{31} \oplus k_{43} \oplus k_{56} \oplus v_3 \oplus v_{13} \oplus v_{23} \oplus v_{25} \oplus v_{38} \oplus v_{51}$.
3. For $i = 17, \ldots, 26$, z_i will be always equal to z'_i.
4. When $i = 27$, z_{27} will be always different from z'_{27}. So by imposing any conditions, we can not make $z_{27} \oplus z'_{27} = 0$.
5. z_i will be always equal to z'_i for $i = 28, \ldots, 33$.
6. When $i = 34$, $z_{34} \oplus z'_{34}$ will be an algebraic expression on Key and IV. However if attacker sets 17 Type 1 conditions $v_2 = v_{15} \oplus v_{18} \oplus v_{25} \oplus v_{31} \oplus v_{40} \oplus v_{53} \oplus v_{56} \oplus v_{59}, v_{63} = 0, v_{14} = v_{24} \oplus v_{39} \oplus v_{52}, v_{13} = v_{23} \oplus v_{38} \oplus v_{51}, v_{17} = v_{42}, v_{43} = 0, v_{47} = 0, v_{38} = 0, v_4 = 0, v_1 = 0, v_5 = 0, v_{20} = 0, v_{21} = 0, v_{26} = 0, v_{27} = 0, v_{37} = 0, v_{48} = 0$ and one Type 2 condition
$$C_2 : v_{59} = f_1(K),$$
where $f_1(K)$ is a polynomial over Key of degree 16 and 9108 monomials, $z_{34} = z'_{34}$.
7. We have $z_i = z'_i$ for $i = 35, 36$.
8. When $i = 37$, again $z_{37} \oplus z'_{37}$ will be an algebraic expression on Key and IV. Now attacker sets 7 Type 1 conditions $v_{15} = v_{18} \oplus v_{25} \oplus v_{31} \oplus v_{53} \oplus v_{55} \oplus v_{56} \oplus v_{59}, v_{16} = v_{54}, v_{49} = 1, v_{28} = 0, v_6 = 0, v_{50} = 0, v_{23} = v_{45}$
and two Type 2 conditions
$$C_3 : v_3 = k_4 \oplus k_5 \oplus k_7 \oplus k_{13} \oplus k_{34} \oplus k_{46} \oplus k_{59} \oplus k_{66}$$
$$C_4 : v_7 = v_{29} \oplus f_2(K),$$

where $f_2(K)$ is a polynomial over Key of degree 15 and 1535 monomials. Then we have $z_{37} = z'_{37}$.

9. We have $z_i = z'_i$ for $i = 38, 39$.
10. If we set 7 Type 1 conditions $v_{58} = v_7, v_{57} = v_{44} \oplus v_{29}, v_{51} = 0, v_{52} = 0, v_{10} = 0, v_{32} = 0, v_{53} = 0$ and 2 Type 2 conditions

$$\begin{aligned} &C_5 : v_9 = k_7 \oplus k_8 \oplus k_{10} \oplus k_{16} \oplus k_{37} \oplus k_{49} \oplus k_{62} \oplus v_{31} \\ &C_6 : v_8 = f_3(K), \end{aligned}$$

where $f_3(K)$ is a polynomial over Key of degree 15 and 1572 monomials, $z_{40} = z'_{40}$.

Thus we have a total of 34 Type 1 conditions and 6 Type 2 conditions $C_1, \ldots, C_6$. We can rewrite the Type 2 conditions as

$$\begin{aligned} C_1 : v_0 &= K_1 \oplus v_3 \oplus v_{13} \oplus v_{23} \oplus v_{25} \oplus v_{38} \oplus v_{51}, \\ C_2 : v_{59} &= K_2, \\ C_3 : v_3 &= K_3, \\ C_4 : v_7 &= K_4 \oplus v_{29}, \\ C_5 : v_9 &= K_5 \oplus v_{31}, \\ C_6 : v_8 &= K_6, \end{aligned}$$

where K_is are function of Key only for $1 \leq i \leq 6$. Hence for fixed Key, K_is are fixed.

Now since attacker does not know the values $K_1, \ldots, K_6$, he has to consider all combinations. Let $U = [K_1, K_2, K_3, K_4, K_5, K_6]$. Then for each $U \in \{0, 1\}^6$, attacker chooses such that

$$\begin{Bmatrix} v_{19} = v_{41} = 1, v_{46} = 0, v_{63} = 0, v_{14} = v_{24} \oplus v_{39} \oplus v_{52}, \\ v_{13} = v_{23} \oplus v_{38} \oplus v_{51}, v_{17} = v_{42}, v_{43} = 0, v_{47} = 0, v_{38} = 0, \\ v_4 = 0, v_1 = 0, v_5 = 0, v_{20} = 0, v_{21} = 0, v_{26} = 0, v_{27} = 0, \\ v_{37} = 0, v_{48} = 0, v_{49} = 1, v_{28} = 0, v_6 = 0, v_{50} = 0, \\ v_{23} = v_{45}, v_{51} = 0, v_{52} = 0, v_{10} = 0, v_{32} = 0, v_{53} = 0, \\ v_0 = K_1 \oplus v_3 \oplus v_{13} \oplus v_{23} \oplus v_{25} \oplus v_{38} \oplus v_{51} \\ v_{59} = K_2, v_3 = K_3, v_7 = K_4 \oplus v_{29}, v_9 = K_5 \oplus v_{31}, v_8 = K_6 \end{Bmatrix}$$

Hence for the correct choice of $K_1, \ldots, K_6$, we have $z_{16} = z'_{16}, z_{34} = z'_{34}, z_{37} = z'_{37}$ and $z_{40} = z'_{40}$.

Note that due to Type 1 conditions, IV space is reduced to $\{0, 1\}^{64-34} = \{0, 1\}^{30}$. Corresponding to 6 Type 2 conditions, attacker divides this space into $2^6 = 64$ partitions. Here free IV variables are: $v_{11}, v_{12}, v_{18}, v_{22}, v_{24}, v_{25}, v_{29}, v_{30}, v_{31}, v_{33}, v_{34}, v_{35}, v_{36}, v_{39}, v_{40}, v_{42}, v_{44}, v_{45}, v_{54}, v_{55}, v_{56}, v_{60}, v_{61}$.

Since there are 6 expressions on the unknown key, the attacker chooses all 64 options. Among these 64 options, one must be correct. For each option, attacker takes the dynamic variables $v_0, v_{59}, v_3, v_7, v_9, v_8$ accordingly. So for fixed key, we have 64 values corresponds to the probability $P(z_{106} = z'_{106})$ for each Type 2 condition. Since we have only 23 free IV, approach of [20] will not work here directly. We use the idea as follows.

We consider only those probabilities for which $P(z_{106} = z'_{106}) > 0.5$, and we add all such probabilities. Let the sum of these probabilities be S. For the random case, this sum will be

$$S_R = 64 \times \frac{1}{\sqrt{2\pi}\sigma} \int_{Np}^{N} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \left(\frac{x}{N} - p \right) dx, \tag{1}$$

where N is the size of sample space, $\mu = \frac{N}{2}, \sigma^2 = \frac{N}{4}$ and $p = 0.5$. For $N = 2^{23}$, value of S_R will be 0.0044.

From our experiment with 1000 random keys, we observe that for 63% situations, the sum in Equation (1) for Grain v1 is greater than 0.0044 when we are using all 23 free IV variables. Thus we can distinguish Grain v1 from random source up to 106 rounds with success probability 0.63.

We try similar idea for 107 rounds. But the algebraic expressions for 107 rounds are much more complicated. Hence getting constraints on Key and IV i.e, Type 1 and Type 2 conditions would be very difficult for this case.

5 Conclusion

In this paper, we have first presented experimental results for all single bit differential on IV. From these experiments, it seems that one may find a distinguisher on Grain v1 for 106 and 107 rounds. Then we have presented our result Grain v1 for 106 rounds. We have shown that it is possible to divide the search space into 64 partitions so that for one partition of IV values the differential of key stream bits at certain positions will be zero. Experiments show that one can distinguish Grain v1 for 106 rounds with 63% success probability.

From our experiments, it seems one may attack Grain v1 up to 107 rounds. However, in this case the conditions are much more complicated. We leave this as an open problem.

References

1. J.-P. Aumasson, I. Dinur, W. Meier and A. Shamir. Cube testers and key recovery attacks on reduced-round MD6 and Trivium. In FSE 2009, LNCS, Vol. 5665, pp. 1–22, 2009.
2. J. P. Aumasson, I. Dinur, L. Henzen, W. Meier and A. Shamir. Efficient FPGA Implementations of High-Dimensional Cube Testers on the Stream Cipher Grain-128.

In SHARCS - Special-purpose Hardware for Attacking Cryptographic Systems, 2009.
3. S. Banik. Some Insights into Differential Cryptanalysis of Grain v1. In ACISP 2014, LNCS, Vol. 8544, pp. 34–49, 2014.
4. S. Banik. A Dynamic Cube Attack on 105 round Grain v1. IACR Cryptology ePrint Archive 2014: 652. Available at `http://eprint.iacr.org/2014/652`.
5. I. Ben-Aroya and E. Biham. Differtial Cryptanalysis of Lucifer. In Crypto 1993, LNCS, Vol. 773, pp. 187–199, 1993.
6. C. Berbain, H. Gilbert and A. Maximov. Cryptanalysis of Grain. In FSE 2006, LNCS, Vol. 4047, pp. 15–29, 2006.
7. T. E. Bjørstad. Cryptanalysis of Grain using Time/Memory/Data tradeoffs (v1.0 / 2008-02-25). Available at `http://www.ecrypt.eu.org/stream`.
8. C. De Cannière, O. Küçük and B. Preneel. Analysis of Grain's Initialization Algorithm. In AFRICACRYPT 2008, LNCS, Vol. 5023, pp. 276–289, 2008.
9. J. Daemen, R. Govaerts and J. Vandewalle. Resynchronization weaknesses in synchronous stream ciphers. In EUROCRYPT 1993. LNCS, vol. 765, pp. 159–167, 1993.
10. I. Dinur, T. Güneysu, C. Paar, A. Shamir and R. Zimmermann. An Experimentally Verified Attack on Full Grain-128 Using Dedicated Reconfigurable Hardware. In Asiacrypt 2011, LNCS, Vol. 7073, pp. 327–343, 2011.
11. I. Dinur and A. Shamir. Cube Attacks on Tweakable Black Box Polynomials. In EUROCRYPT 2009, LNCS, Vol. 5479, pp. 278-299, 2009.
12. I. Dinur and A. Shamir. Breaking Grain-128 with Dynamic Cube Attacks. In FSE 2011, LNCS, Vol. 6733, pp. 167–187, 2011.
13. The ECRYPT Stream Cipher Project. eSTREAM Portfolio of Stream Ciphers. Revised on September 8, 2008.
14. H. Englund, T. Johansson and M. S. Turan. A framework for chosen IV statistical analysis of stream ciphers. In INDOCRYPT 2007, LNCS, Vol. 4859, pp. 268–281, 2007.
15. S. Fischer, S. Khazaei and W. Meier. Chosen IV statistical analysis for key recovery attacks on stream ciphers. In AFRICACRYPT 2008, LNCS, Vol. 5023, pp. 236–245, 2008.
16. H. Fredricksen. A survey of full length nonlinear shift register cycle algorithms, SIAM Rev., 24 (1982), pp. 195–221, 1982.
17. M. Hell, T. Johansson and W. Meier. Grain - A Stream Cipher for Constrained Environments. ECRYPT Stream Cipher Project Report 2005/001, 2005. Available at `http://www.ecrypt.eu.org/stream`.
18. C. De Cannière and B. Preneel. Trivium. Available at `http://www.ecrypt.eu.org/stream/p3ciphers/trivium/trivium_p3.pdf`.
19. S. Khazaei, M. Hassanzadeh and M. Kiaei. Distinguishing Attack on Grain. ECRYPT Stream Cipher Project Report 2005/071, 2005. Available at `http://www.ecrypt.eu.org/stream`
20. S. Knellwolf, W. Meier and M. Naya-Plasencia. Conditional Differential Cryptanalysis of NLFSR-based Cryptosystems. In ASIACRYPT 2010, LNCS, Vol. 6477, pp. 130–145, 2010.
21. S. Knellwolf, W. Meier and M. Naya-Plasencia. Conditional differential cryptanalysis of Trivium and Katan. In SAC 2011, LNCS, Vol. 7118, pp. 200–212, 2011.
22. S. Knellwolf and W. Meier. High order differential attacks on stream ciphers. In Cryptography and Communications, Vol. 4(3-4), pp. 203–215, 2012.
23. M. Lehmann and W. Meier. Conditional Differential Cryptanalysis of Grain-128a. In CANS 2012, LNCS, Vol. 7712, pp. 1–11, 2012.

24. Y. Lee, K. Jeong, J. Sung and S. Hong. Related-Key Chosen IV Attacks on Grain-v1 and Grain-128. In ACISP 2008, LNCS, Vol. 5107, pp. 321–335, 2008.
25. W. Stein. Sage Mathematics Software. Free Software Foundation, Inc., 2009. Available at `http://www.sagemath.org`. (Open source project initiated by W. Stein and contributed by many).
26. P. Stankovski. Greedy Distinguishers and Nonrandomness Detectors. In INDOCRYPT 2010, LNCS, Vol. 6498, pp. 210–226, 2010.
27. H. Zhang and X. Wang. Cryptanalysis of Stream Cipher Grain Family. IACR Cryptology ePrint Archive 2009: 109. Available at `http://eprint.iacr.org/2009/109`.

Appendix A: Condition on key-stream for Different Locations

Shaded conditions for 37 and 61 are previously explored by others [20, 4]. In this paper, we consider the conditions for 62.

Table 1. Different KSA round numbers for different IV locations.

Location	Rounds	Location	Rounds	Location	Rounds	Location	Rounds
0	16 17 34 35	16	13 33 35 36	32	7 29 35 41	48	2 23 41 42
1	17 18 35 36	17	14 34 36 37	33	8 30 36 42	49	3 24 42 43
2	19 34 35 36	18	15 34 35 37	34	9 31 37 43	50	4 25 43 44
3	0 20 35 36	19	16 35 36 38	35	10 32 38 44	51	5 16 26 34
4	1 21 36 37	20	17 36 37 39	36	11 33 39 45	52	6 17 27 35
5	2 22 37 38	21	18 37 38 40	37	12 34 40 46	53	7 28 34 35
6	3 23 38 39	22	19 38 39 41	38	13 16 34 35	54	8 29 35 36
7	4 24 39 40	23	16 20 34 39	39	14 17 35 36	55	9 30 36 37
8	5 25 40 41	24	17 21 35 40	40	15 34 35 36	56	10 31 34 37
9	6 26 41 42	25	0 22 34 35	41	16 34 35 36	57	11 32 35 38
10	7 27 42 43	26	1 23 35 36	42	17 35 36 37	58	12 33 36 39
11	8 28 43 44	27	2 24 36 37	43	18 36 37 38	59	13 34 37 40
12	9 29 44 45	28	3 25 37 38	44	19 37 38 39	60	14 35 38 41
13	10 16 30 34	29	4 26 38 39	45	20 38 39 40	61	15 36 39 42
14	11 17 31 35	30	5 27 39 40	46	0 21 39 40	62	16 34 37 40
15	12 32 34 35	31	6 28 34 40	47	1 22 40 41	63	17 35 38 41