# A Dynamic Cube Attack on 105 round Grain v1

Subhadeep Banik

DTU Compute, Lyngby, Denmark.
subb@dtu.dk

**Abstract.** As far as the Differential Cryptanalysis of reduced round Grain v1 is concerned, the best results were those published by Knellwolf et al. in Asiacrypt 2011. In an extended version of the paper, it was shown that it was possible to retrieve **(i)** 5 expressions in the Secret Key bits for a variant of Grain v1 that employs 97 rounds (in place of 160) in its Key Scheduling process using $2^{27}$ chosen IVs and **(ii)** 1 expression in Secret Key bits for a variant that employs 104 rounds in its Key Scheduling using $2^{35}$ chosen IVs. However, the second attack on 104 rounds, had a success probability of around 50%, which is to say that the attack worked for only around one half of the Secret Keys.
In this paper we propose a dynamic cube attack on 105 round Grain v1, that has a success probability of 100%, and thus we report an improvement of 8 rounds over the previous best attack on Grain v1 that attacks the entire Keyspace. We take the help of the tool $\Delta\mathsf{Grain}_{\mathsf{KSA}}$, proposed by Banik at ACISP 2014, to track the differential trails induced in the internal state of Grain v1 by any difference in the IV bits, and we prove that a suitably introduced difference in the IV leads to a distinguisher for the output bit produced in the $105^{th}$ round. This, in turn, helps determine the values of 6 expressions in the Secret Key bits.

**Keywords:** eStream, Differential Cryptanalysis, Dynamic Cube Attack, Grain v1, Stream Cipher.

## 1 Introduction

The Grain v1 stream cipher, designed by Hell, Johansson and Meier in 2005 [18], is in the hardware profile of the eStream portfolio [1]. It is a synchronous bit oriented cipher designed so as to minimize hardware complexity. After two attacks [7, 22] on the initial design of the cipher were published, the modified version Grain v1 [18] was proposed. Later, the designers came up with a second version of Grain, i.e., Grain-128 [19] that uses a 128 bit Key. Thereafter, the cipher Grain-128a [2] was designed for the dual purpose of message authentication alongside message encryption. For detailed cryptanalytic results related to this family, the reader may refer to [4–6,8–10,15,17,20,25] and the references therein.

Cube attacks were first introduced by Dinur and Shamir in [13] and have been used extensively to attack reduced round variants of the Grain family. Although the attack paradigm can be used to cryptanalyze any symmetric key cryptosystem with manipulatable public variables, we shall focus on stream ciphers in this paper, in which the IV plays the role of the public variable. A cube attack on a stream cipher proceeds in the following manner (for a detailed description of cube attacks, kindly refer to [13, 14]).

**A.** The attacker chooses any non-empty subset $C$ of the IV bit variables, commonly called cube variables. All the IV bits outside of $C$ assigned to some constant, usually zeros.
**B.** The attacker enumerates $2^{|C|}$ IVs by assigning the cube variables to all possible values. He then obtains a segment of keystream bits produced by some fixed Secret Key $K$ and each of the $2^{|C|}$ IVs enumerated above.
**C.** If the attacker is able to determine that the sum of all the corresponding keystream bits produced by the $2^{|C|}$ IVs leads to a linear equation on the Secret Key bits of $K$, then he effectively obtains one linear equation on the Secret Key variables. If not, he discards the cube set $C$, and starts step **A** with another randomly generated cube set.

Ideally, the attacker would continue this process until he obtains sufficient number of linear equations to solve for the Secret Key, but in many practical cube attacks published so far [11–13, 16] the attacker is only able to determine a fraction of the the Secret Key bits. This is because when the attacker starts with a random cube set $C$, it very rarely leads to a linear equation on the Secret Key bits. As a result, the attacker needs to test a lot of random cube sets before he is able to arrive at one in which the cube sum is a linear equation on the Secret Key variables. This process is usually quite time consuming and can even take weeks to complete [13]. Very recently, a Moebius-transform based approach was adopted in [16], that was able to find 12 such cube sets (that lead to linear equations in Secret Key bits) for a version of Trivium reduced to 799 initialization rounds, in just about 2 hours.

On the other hand, Dynamic Cube Attacks, are a class of cube attacks that aim to establish if by stipulating some algebraic relation between the Secret Key and IV variables, one is able to observe some testable non-random property in the cube sum. In this type of attack, some of the IV variables outside the cube set are chosen as dynamic cube variables. Each dynamic cube variable $\nu$ is related to one or more expressions in the Secret Key bits. The attacker first has to guess the values of each of these expressions correctly to compute the value of $\nu$. Thereafter he performs the cube sum, and tests it for some non-random property. The attack parameters should be designed in a manner so that, if the values of the Secret Key expressions are guessed correctly, then the attacker would be able to detect some non-randomness in the cube sum, and not otherwise. This would therefore enable the attacker to determine the values of each of those Secret Key expressions. So, if there are $e$ number of expressions to be guessed, the cube sum needs to be computed $2^e$ times, once for each of the guesses.

In [11,12], dynamic cube attacks have been used to successfully cryptanalyze reduced-round variants as well as full Grain 128. In [21], cube distinguishers were used to distinguish a variant of Grain-128a, that employs 189 out of the 256 rounds in the Key Scheduling process. However, due to the relative complex nature of the component functions used in the design of Grain v1, there have not been many advances in this direction against it. The best published work on Grain v1 is by Knellwolf et al [24], an extended version of which appeared in [23, Chapter 3.4]. The attack, which can be best described as a dynamic cube attack over a single-dimensional cube (i.e., the cardinality of the cube set is one), achieves the following objectives:

**a)** It retrieves 5 expressions in the Secret Key bits for a variant of Grain v1 that employs 97 rounds (in place of 160) in its Key Scheduling process using $2^{27}$ chosen IVs.

**b)** It retrieves 1 expression in Secret Key bits for a variant that employs 104 rounds in its Key Scheduling using $2^{35}$ chosen IVs. However, as reported in [23, Chapter 3.4], this attack has a success probability of around 50%, which is to say that the attack works for only around one half of the Secret Keys.

The values of these Secret Key expressions are deduced by observing certain non-randomness in the keystream bits generated by the chosen IVs. More specifically, the authors could enumerate a set of IVs for which, the sum of the output bits over the single dimensional cube were biased towards 0. Very recently, some work has been done in [3], towards proving the theoretical correctness of these attacks. In this work a tool called $\Delta\mathsf{Grain}_{\mathsf{KSA}}$ was proposed to track the differential trails introduced in the internal state of Grain v1 by any difference in the IV bits. Using the tool, the theoretical correctness of the work presented in [24] was proven.

### 1.1 Contribution and Organization of the paper

In this work, we make use of the tool $\Delta\mathsf{Grain}_{\mathsf{KSA}}$ to further improve the work presented in [24]. We first outline a heuristic algorithm that enables us to determine a suitable single dimensional cube (i.e, the $61^{st}$ IV bit). We show that a differential introduced via the $61^{st}$ IV bit in Grain v1, leads to a distinguisher for the keystream bit produced in the $105^{th}$ round in Grain v1, if certain algebraic

conditions involving the Secret Key and the IV bits are satisfied. This, in turn, leads to the deduction of 6 expressions in the Secret key bits. This amounts to an improvement of 8 rounds over the previous best attack on Grain v1 that works for any Key in the Keyspace.

In Section 2, we give the complete mathematical description of Grain v1. In Section 3, we give a brief description of the tool $\Delta\mathsf{Grain}_{\mathsf{KSA}}$ as reported in [3]. In Section 4, we will outline how the above tool can be used to attack on Grain v1, and thereafter enumerate the algebraic details of the attack. Section 5 concludes the paper.

## 2 Description of Grain v1

The exact structure of the Grain family is explained in Figure 1. It consists of an $n$-bit LFSR and an $n$-bit NFSR. Certain bits of both the shift registers are taken as inputs to a combining Boolean function, whence the key-stream is produced. The update function of the LFSR is given by the equation $y_{t+n} = f(Y_t)$, where $Y_t = [y_t, y_{t+1}, \ldots, y_{t+n-1}]$ is an $n$-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 $n$. The NFSR state is updated as $x_{t+n} = y_t \oplus g(X_t)$. Here, $X_t = [x_t, x_{t+1}, \ldots, x_{t+n-1}]$ is an $n$-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 key-stream 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 $\{0, 1, 2, \ldots, n-1\}$.

Grain v1 uses an $n = 80$-bit key $K$, and an $m = 64$-bit initialization vector $IV$. The key is loaded in the NFSR and the IV is loaded in the $0^{th}$ to the $(63)^{rd}$ bits of the LFSR. The remaining bits of the LFSR are loaded with the all one pad `0x ffff`. Hence at this stage, the $2n$ bit initial state is of the form $K||IV||P$.

**Key Scheduling Algorithm (KSA)** For the first $2n$ clocks, the key-stream bit produced by the cipher is XOR-ed to both the LFSR and NFSR update functions, i.e., during the first $2n$ clock intervals, the LFSR and the NFSR bits are updated as $y_{t+n} = z_t \oplus f(Y_t)$, $x_{t+n} = y_t \oplus z_t \oplus g(X_t)$.
**Pseudo-Random key-stream Generation Algorithm (PRGA)** After the completion of the KSA, $z_t$ is used as the Pseudo-Random key-stream bit. It is no longer XOR-ed to the LFSR and the NFSR. Therefore during this phase, the LFSR and NFSR are updated as $y_{t+n} = f(Y_t), x_{t+n} = y_t \oplus g(X_t)$.

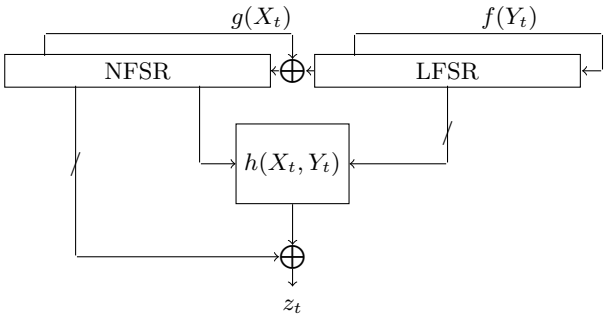


**Fig. 1.** Structure of Stream Cipher in Grain Family

### 2.1 Mathematical description of Grain v1

Grain v1 consists of an 80 bit LFSR and an 80 bit NFSR. It uses an 80-bit Key and a 64-bit IV, and a 16-bit pad $P =$ `0x ffff`. Certain bits of both the shift registers are taken as inputs to a combining

Boolean function, whence the key-stream is produced. The update function of the LFSR is given by the equation

$$y_{t+80} = y_{t+62} \oplus y_{t+51} \oplus y_{t+38} \oplus y_{t+23} \oplus y_{t+13} \oplus y_t \stackrel{\Delta}{=} f(Y_t).$$

The NFSR state is updated as follows

$$\begin{aligned} x_{t+80} = y_t \oplus g(&x_{t+63}, x_{t+62}, x_{t+60}, x_{t+52}, x_{t+45}, x_{t+37}, x_{t+33}, x_{t+28}, x_{t+21}, x_{t+15} \\ &x_{t+14}, x_{t+9}, x_t), \end{aligned}$$

where $g(x_{t+63}, x_{t+62}, \ldots, x_t)$

$$\begin{aligned} \stackrel{\Delta}{=} \quad g(X_t) \; = & x_{t+62} \oplus x_{t+60} \oplus x_{t+52} \oplus x_{t+45} \oplus x_{t+37} \oplus x_{t+33} \oplus x_{t+28} \oplus x_{t+21} \oplus x_{t+14} \oplus x_{t+9} \oplus x_t \oplus \\ & x_{t+63}x_{t+60} \oplus x_{t+37}x_{t+33} \oplus x_{t+15}x_{t+9} \oplus x_{t+60}x_{t+52}x_{t+45} \oplus x_{t+33}x_{t+28}x_{t+21} \oplus \\ & x_{t+63}x_{t+45}x_{t+28}x_{t+9} \oplus x_{t+60}x_{t+52}x_{t+37}x_{t+33} \oplus x_{t+63}x_{t+60}x_{t+21}x_{t+15} \oplus \\ & x_{t+63}x_{t+60}x_{t+52}x_{t+45}x_{t+37} \oplus x_{t+33}x_{t+28}x_{t+21}x_{t+15}x_{t+9} \oplus \\ & x_{t+52}x_{t+45}x_{t+37}x_{t+33}x_{t+28}x_{t+21}. \end{aligned}$$

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

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

where $A = \{1, 2, 4, 10, 31, 43, 56\}$ and $h(s_0, s_1, s_2, s_3, s_4) = s_1 \oplus s_4 \oplus s_0s_3 \oplus s_2s_3 \oplus s_3s_4 \oplus s_0s_1s_2 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_1s_2s_4 \oplus s_2s_3s_4$.

## 3 The tool $\Delta\mathsf{Grain}_{\mathsf{KSA}}$

### 3.1 Generalized Grain cipher

For completeness of this paper, we will discuss in brief the tool $\Delta\mathsf{Grain}_{\mathsf{KSA}}$ which was discussed in [3]. The primary purpose of such a tool is to track the differential trails in the internal state of the cipher during the KSA phase, which is introduced due a difference in the IV bits. First a generalized Grain stream cipher is defined which covers the descriptions of Grain v1, Grain-128 and Grain-128a as well. Any cipher in the Grain family consists of an $n$-bit LFSR and an $n$-bit NFSR ($n = 80, 128, 128$ for Grain v1, Grain-128 and Grain-128a respectively). The update function of the LFSR is given by the equation

$$y_{t+n} = f(Y_t) = y_t \oplus y_{t+f_1} \oplus y_{t+f_2} \oplus \cdots \oplus y_{t+f_a},$$

where $Y_t = [y_t, y_{t+1}, \ldots, y_{t+n-1}]$ is an $n$-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 $n$. The NFSR state is updated as

$$\begin{aligned} x_{t+n} &= y_t \oplus g(X_t) = y_t \oplus g(x_t, x_{t+g_1}, x_{t+g_2}, \ldots, x_{t+g_b}) \\ &= y_t \oplus x_t \oplus x_{t+g_1} \oplus \cdots \oplus x_{t+g_{b_0}} \oplus g'(x_{t+g_{b_0+1}}, x_{t+g_{b_0+2}}, \ldots, x_{t+g_b}) \end{aligned}$$

Here, $X_t = [x_t, x_{t+1}, \ldots, x_{t+n-1}]$ is an $n$-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 in which the NFSR locations $0, g_1, g_2, \ldots, g_{b_0}$ only contribute linearly. The output key-stream is produced by combining the LFSR and NFSR bits as

$$\begin{aligned} z_t = x_{t+l_1} \oplus x_{t+l_2} \oplus \cdots \oplus x_{t+l_c} \oplus y_{t+i_1} \oplus y_{t+i_2} \oplus \cdots \oplus y_{t+i_d} \oplus \\ h(y_{t+h_1}, y_{t+h_2}, \ldots, y_{t+h_e}, x_{t+j_1}, x_{t+j_2}, \ldots, x_{t+j_w}). \end{aligned}$$

Here $h$ is another non-linear combining Boolean function. Note that Grain v1, Grain-128 and Grain-128a are particular instances of the generalized Grain cipher.

### 3.2 $\Delta\mathsf{Grain}_{\mathsf{KSA}}$

Let $S_0 = [X_0||Y_0] \in \{0,1\}^{2n}$ be the initial state of the generalized Grain KSA and $S_0^{\phi} = [X_0^{\phi}||Y_0^{\phi}]$ be the initial state which differs from $S_0$ in some LFSR location $\phi \in [0, m-1]$, where $m$ is the length of the IV ($m = 64, 96, 96$ for Grain v1, Grain-128 and Grain-128a respectively).

The tool ascertains how the corresponding internal states in the $t^{th}$ round $S_t$ and $S_t^{\phi}$ differs from each other, for some integer $t > 0$. In the original definition of $\Delta\mathsf{Grain}_{\mathsf{KSA}}$ described in [3], the engine takes as input the difference location $\phi \in [0, m-1]$ and the value $r$ of the number of rounds, and returns the following:

**(i)** a set of $r$ integer arrays $\chi_t$, for $0 \leq t < r$, each of length $c + d$,
**(ii)** a set of $r$ integer arrays $\Upsilon_t$, for $0 \leq t < r$, each of length $e + w$ and
**(iii)** an integer array $\Delta Z$ of length $r$.

We will modify the definition of the routine so that the engine additionally returns three other sets of integer arrays given by (the definition of these arrays will be given shortly)

**(iv)** a set of $r$ integer arrays $\mathcal{F}_t$, for $0 \leq t < r$, each of length $a + 1$,
**(v)** a set of $r$ integer arrays $\mathcal{G}_{lin,t}$, for $0 \leq t < r$, each of length $b_0 + 1$ and
**(vi)** a set of $r$ integer arrays $\mathcal{G}_{nlin,t}$, for $0 \leq t < r$, each of length $b - b_0$

Note that as already defined in the description of generalized Grain, $d, c$ are the number of LFSR, NFSR bits which are linearly added to the output function $h$. And $e, w$ are the number of LFSR, NFSR bits that are input to the function $h$, $a + 1$ is the number of LFSR bits input to the function $f$, $b_0 + 1$ is the number of NFSR bits in the linear part of the function $g$ and $b + 1$ is the total number of inputs of $g$.

A generalized differential engine $\Delta_{\phi}$-$\mathsf{Grain}_{\mathsf{KSA}}$ with an $n$-cell LFSR $\Delta L$ and an $n$-cell NFSR $\Delta N$ is defined. All the elements of $\Delta L$ and $\Delta N$ are integers. The $t^{th}$ round state of $\Delta L$ is denoted as $\Delta L_t = [u_t, u_{t+1}, \ldots, u_{t+n-1}]$ and that of $\Delta N$ is denoted as $\Delta N_t = [v_t, v_{t+1}, \ldots, v_{t+n-1}]$. Initially all the elements of $\Delta N, \Delta L$ are set to 0, with the only exception that – the cell numbered $\phi$ of $\Delta L$ is set to 1. The initial states $\Delta N_0, \Delta L_0$ are indicative of the difference between $S_0$ and $S_0^{\phi}$ and the $t^{th}$ states $\Delta N_t, \Delta L_t$ are indicative of the difference between $S_t$ and $S_t^{\phi}$.

The update functions of the registers $L, N$ are defined in the following manner. The function $\mathsf{lin} : \cup_{i=1}^{\infty} \mathbb{Z}_+^i \rightarrow \{0, 1, 2\}$ is defined as follows. (where $\mathbb{Z}_+$ is the set of non negative integers)

$$\mathsf{lin}(q_1, q_2, \ldots, q_i) = \begin{cases} q_1 + q_2 + \cdots + q_i \bmod 2 & \text{if } \max(q_1, q_2, \ldots, q_i) \leq 1, \\ 2, & \text{otherwise.} \end{cases}$$

Define the vectors $\mathcal{F}_t, \mathcal{G}_{lin,t}, \mathcal{G}_{nlin,t}$ as follows: $\mathcal{F}_t = [u_t, u_{t+f_1}, \ldots, u_{t+f_a}]$, $\mathcal{G}_{lin,t} = [v_t, v_{t+g_1}, \cdots, v_{t+g_{b_0}}]$, and

$$\mathcal{G}_{nlin,t} = [v_{t+g_{b_0+1}}, v_{t+g_{b_0+2}}, \ldots, v_{t+g_b}]$$

The intermediate variables $\ell_t, r_t, \Omega_t$ are defined as $\ell_t = \mathsf{lin}\ (\mathcal{F}_t)$, $r_t = \mathsf{lin}\ (u_t, \mathcal{G}_{lin,t})$, $\Omega_t = 2 \cdot \mathsf{OR}(\mathcal{G}_{nlin,t})$. Here $\mathsf{OR}$ is a map from $\cup_{i=1}^{\infty} \mathbb{Z}_+^i \rightarrow \{0, 1\}$ which roughly represents the logical 'or' operation and is defined as

$$\mathsf{OR}(q_0, q_1, \ldots, q_i) = \begin{cases} 0, \text{ if } q_0 = q_1 = q_2 = \cdots = q_i = 0, \\ 1, \text{ otherwise.} \end{cases}$$

Define the following vectors:

$$\chi_t = [v_{t+l_1}, v_{t+l_2}, \ldots, v_{t+l_c}, u_{t+i_1}, u_{t+i_2}, \ldots, u_{t+i_d}],\ \Upsilon_t = [u_{t+h_1}, u_{t+h_2}, \ldots, u_{t+h_e}, v_{t+j_1}, v_{t+j_2}, \ldots, v_{t+j_w}].$$

Note that $\chi_t(\Upsilon_t)$ is the set of cells in $\Delta_\phi$-$\mathsf{Grain}_{\mathsf{KSA}}$ which corresponds to the bits which are linearly added to the output function $h$ (input to $h$) in the $t^{th}$ KSA stage of the actual cipher. The $t^{th}$ key-stream element $\pi_t$ produced by this engine is given as (the term exception will be explained shortly)

$$\pi_t = \begin{cases} 1 & \text{if } \chi_t, \Upsilon_t \text{ throws up an exception} \\ \mathsf{lin}\left(\mathsf{lin}(\chi_t), 2 \cdot \mathsf{OR}(\Upsilon_t)\right) & \text{otherwise.} \end{cases}$$

Now $\Delta L$ updates itself as $u_{t+n} = \mathsf{lin}(\ell_t, \pi_t)$. And similarly, $\Delta N$ updates itself as $v_{t+n} = \mathsf{lin}(r_t, \Omega_t, \pi_t)$. It has been argued in [3], that the values in the registers $L_t, N_t$ in $\Delta\mathsf{Grain}_{\mathsf{KSA}}$ represent the differences in the corresponding algebraic systems $S_t$ and $S_t^\phi$, which are the $t^{th}$ round internal states of the Generalized Grain initialized by two IVs that differ in the $\phi^{th}$ bit. Similarly the output element $\pi_t$ represents the difference between the output bits $z_t$ and $z_t^\phi$ produced in the $t^{th}$ rounds by $S_t$ and $S_t^\phi$ respectively. In particular if $u_t$ (resp. $v_t, \pi_t$) is equal to

$\rightarrow 0$, the difference between $y_t$ and $y_t^\phi$ (resp. $x_t, x_t^\phi$ and $z_t, z_t^\phi$) is always 0.
$\rightarrow 1$, the difference between $y_t$ and $y_t^\phi$ (resp. $x_t, x_t^\phi$ and $z_t, z_t^\phi$) is always 1.
$\rightarrow 2$, the difference between $y_t$ and $y_t^\phi$ (resp. $x_t, x_t^\phi$ and $z_t, z_t^\phi$) is probabilistically either 0 or 1 and the exact value would depend on the exact value of the initial vector $S_0$ and actual update functions.

Before we proceed to the actual attack, we make a note that the definition of $\pi_t$ has the term exception in it. This term is necessary as sometimes due to the nature of the function $h$ used in the actual implementation of any version of Grain, $\pi_t$ might fail to capture the difference between $z_t, z_t^\phi$. For example, in Grain v1, for $\phi = 37$, and $t = 30$, the values of $\chi_t$ and $\Upsilon_t$ are as follows:

$$t = 30: \quad \chi_t = \mathbf{0}, \quad \Upsilon_t = [u_{t+3} = 0, u_{t+25} = 0, u_{t+46} = 0, u_{t+64} = 1, v_{t+63} = 0]$$

Here $\mathbf{0}$ is the all zero vector.This implies that if we introduce an IV differential at location 37 then at KSA round 30, all state bits in $S_{30}$ and $S_{30}^{37}$ involved in the computation of their respective keystream bits are equal except the bits $y_{t+64}$ and $y_{t+64}^{37}$, which are deterministically unequal, i.e., $y_{94} = 1 \oplus y_{94}^{37}$ always holds. Then, it follows that

$$\begin{aligned} z_{30} \oplus z_{30}^{37} &= h(y_{33}, y_{55}, y_{76}, y_{94}, x_{93}) \oplus h(y_{33}, y_{55}, y_{76}, 1 \oplus y_{94}, x_{93}) \\ &= y_{33}y_{76} \oplus y_{33} \oplus y_{76}x_{93} \oplus y_{76} \oplus x_{93} = 1. \end{aligned}$$

The above follows because $y_{76}$ is initialized to 1 as it is a part of the `0x ffff` padding that is used in Grain v1. Thus, $z_{30}$ and $z_{30}^{37}$ are deterministically unequal. But had we not checked for this, during the calculation of $\pi_t$, the value of $\pi_{30}$ would be computed as 2. An event of this type is termed an exception, and it occurs due to the nature of the function $h$ used in the design of Grain v1. To prevent a situation like this one must always check if for some $t$, the values of $\chi_t$ and $\Upsilon_t$ throw up an exception. If it does, the value 1 is assigned to $\pi_t$. The algorithm is presented formally in Algorithm 1.

### 3.3 Using $\Delta_\phi$-$\mathsf{Grain}_{\mathsf{KSA}}$ to model Knellwolf's attack [23, Chapter 3.4]

The basic philosophy of the attack in [23, Chapter 3.4] is as follows. The attacker introduces a difference in the internal states of two initializations of the Grain v1 cipher via the $37^{th}$ IV bit. Thereafter, by imposing several algebraic conditions of **Type 1** and 5 conditions of **Type 2** between the Secret Key and the IV variables the attacker prevents the propagation of this difference into the NFSR at KSA rounds $t = 12, 34, 40$.

**Type 1** conditions are of the form $F_1(IV) = 0$ which involve only the IV bits.
**Type 2** conditions are of the form $F_2(K, IV) = 0$ which involve both the Key and IV bits.

**Input**: $\phi$: An LFSR location $\in [0, m-1]$, an integer $r(>0)$;

**Output**: An integer array $\Delta Z$ of $r$ elements;

**Output**: Two integer arrays $\chi_t, \Upsilon_t$ for $0 \leq t < r$ ;

---

$[u_0, u_1, \ldots, u_{n-1}] \leftarrow \mathbf{0}, [v_0, v_1, \ldots, v_{n-1}] \leftarrow \mathbf{0}$, $t \leftarrow 0$, $u_\phi = 1$;

**while** $t < r$ **do**

  $\Upsilon_t \leftarrow [u_{t+h_1}, u_{t+h_2}, \ldots, u_{t+h_e}, v_{t+j_1}, v_{t+j_2}, \ldots, v_{t+j_w}]$ ;

  $\chi_t \leftarrow [v_{t+l_1}, v_{t+l_2}, \ldots, v_{t+l_c}, u_{t+i_1}, u_{t+i_2}, \ldots, u_{t+i_d}]$;

  $\ell_t \leftarrow \mathsf{lin}(u_t, u_{t+f_1}, u_{t+f_2}, \ldots, u_{t+f_a})$;

  $r_t \leftarrow \mathsf{lin}(u_t, v_t, v_{t+g_1}, \cdots, v_{t+g_{b_0}})$;

  $\Omega_t \leftarrow 2 \cdot \mathsf{OR}(v_{t+g_{b_0+1}}, v_{t+g_{b_0+2}}, \ldots, v_{t+g_b})$;

  **if** $\chi_t, \Upsilon_t$ *throws up an* exception **then**

    $\pi_t \leftarrow 1$

  **end**

  **else**

    $\pi_t \leftarrow \mathsf{lin}\,(\mathsf{lin}(\chi_t), 2 \cdot \mathsf{OR}(\Upsilon_t))$

  **end**

  $u_{t+n} \leftarrow \mathsf{lin}(\pi_t, \ell_t)$, $v_{t+n} \leftarrow \mathsf{lin}(\pi_t, r_t, \Omega_t)$;

0.1  `/*Any modification goes here */`;

  $t = t + 1$;

**end**

Return $[\chi_0, \chi_1, \ldots, \chi_{r-1}]$, $[\Upsilon_0, \Upsilon_1, \ldots, \Upsilon_{r-1}]$, $\Delta Z = [\Delta z_0, \Delta z_1, \ldots, \Delta z_{r-1}], [\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_{r-1}]$

Return $[\mathcal{G}_{lin,0}, \mathcal{G}_{lin,1}, \ldots, \mathcal{G}_{lin,r-1}], [\mathcal{G}_{nlin,0}, \mathcal{G}_{nlin,1}, \ldots, \mathcal{G}_{nlin,r-1}]$

**Algorithm 1**: $\Delta_\phi$-$\mathsf{Grain}_{\mathsf{KSA}}$

The attacker observes that if all the **Type 2** conditions are satisfied then

$$\mathsf{Pr}[z_{97} \oplus z_{97}^{\phi} = 0] = \frac{1}{2} + \epsilon, \ (\epsilon \approx 0.0014) \tag{1}$$

Since the Secret Key is unknown to the attacker, any random IV that he picks is unlikely to satisfy all the **Type 2** conditions. The attacker then enumerates 32 different sets of IVs $T_i$, $0 \leq i \leq 31$, such that all the five **Type 2** conditions are satisfied in exactly one of the sets $T_i$. The attacker then computes the probability $\mathsf{Pr}[z_{97} \oplus z_{97}^{\phi} = 0]$ in each of the 32 sets. The Set $T_i$ in which he obtains a bias $\epsilon$ reveals the value of the 5 expressions in the Secret Key bits.

Prior to the work in [3], there was no theoretical proof of this attack, i.e. the attack was supported by experimental evidences only. In [3], the author uses a modified form of $\Delta_\phi$-$\mathsf{Grain}_{\mathsf{KSA}}$ to model Knellwolf's system. Note that [23] imposes specific algebraic conditions to stop the propagation of the differential into the NFSR at KSA rounds $t = 12, 34, 40$. This was modeled by modifying the definition of $\Delta_\phi$-$\mathsf{Grain}_{\mathsf{KSA}}$ at line no 0.1 of Algorithm 1 by including the following code snippet.

$$\textbf{if} \ \ t \ \in \ \{12, 34, 40\} : v_{t+n} \leftarrow 0$$

Thereafter the distributions of the differences of several internal variables were computed by utilizing the outputs of $\Delta_{37}$-$\mathsf{Grain}_{\mathsf{KSA}}$ and Equation (1) was proven.

## 4 Dynamic Cube attack on Grain v1

One of the reasons that the authors of [23, 24] had given for choosing $\phi = 37$ (i.e., the IV location where the single bit difference was introduced) was that due to the positioning of the tap locations in

the design of Grain v1, a difference placed in this location would be contained in the LFSR for the longest number of KSA rounds. This is a valid approach to look at differential cryptanalysis of Grain v1, since as the differences propagate into the NFSR, they become difficult to control and predict. However, given the fact that we have a tool to track the differential trails in Grain v1, one could try to see if any differential introduced at an arbitrary IV bit location $\phi$, $(0 \leq \phi < 64)$ leads to a bias in the distribution of $\mathsf{Pr}[z_t \oplus z_t^{\phi}]$ for any $t > 104$.

Before we outline the process of searching for a suitable difference location $\phi$ let us state the following Lemma.

**Lemma 1.** *Let $\lambda_1, \lambda_2, \ldots, \lambda_{n+1}$ be independent random variables over $GF(2)$. Let $\lambda_i \sim Ber(\frac{1}{2} + \epsilon_i)$. Where each $\epsilon_i$ is some real number in $[-\frac{1}{2}, \frac{1}{2}]$ and $Ber(\cdot)$ denotes the Bernoulli distribution (we will call $|\epsilon_i|$ the bias in the variable $\lambda_i$) Let $\gamma_1 = \oplus_{i=1}^{n} \lambda_i$ and $\gamma_2 = \oplus_{i=1}^{n+1} \lambda_i$. Let $\gamma_1 \sim Ber(\frac{1}{2} + \delta_1)$ and $\gamma_2 \sim Ber(\frac{1}{2} + \delta_2)$. Then we must have $|\delta_1| \geq |\delta_2|$.*

*Proof.* By the Piling-up lemma, it is easy to see that $\mathsf{Pr}[\lambda_1 \oplus \lambda_2 \oplus \cdots \oplus \lambda_n = 0] = \frac{1}{2} + 2^{n-1} \prod_{i=1}^{n} \epsilon_i$ and so we have $\gamma_1 \sim Ber(\frac{1}{2} + 2^{n-1} \prod_{i=1}^{n} \epsilon_i)$ and similarly we have $\gamma_2 \sim Ber(\frac{1}{2} + 2^{n} \prod_{i=1}^{n+1} \epsilon_i)$. Assuming that all $\epsilon_i \neq 0$, we have

$$\frac{|\delta_1|}{|\delta_2|} = \left| \frac{2^{n-1} \prod_{i=1}^{n} \epsilon_i}{2^{n} \prod_{i=1}^{n+1} \epsilon_i} \right| = \frac{1}{2|\epsilon_{n+1}|} \geq 1$$

The above inequality follows since $2|\epsilon_{n+1}| \leq 1$. If some $\epsilon_i = 0$, for $0 \leq i \leq n$, then $\delta_1 = \delta_2 = 0$, and if $\epsilon_{n+1} = 0$ and all other $\epsilon_i$ are non zero, we will have $\delta_2 = 0$, while $|\delta_1|$ is a positive real number and so the lemma is proven. □

So if $\lambda_1, \lambda_2, \ldots, \lambda_n$ are independent variables, each approximately biased towards zero by some positive quantity $|\epsilon|$ (i.e. each $\lambda_i \sim Ber(\frac{1}{2} + \epsilon)$), a corollary of the above result is that the $GF(2)$ sum of any $n_1$ variables $\lambda_i$ is likely to be more biased than the $GF(2)$ sum of any $n_2$ variables $\lambda_i$, if $n_1 < n_2$. We will use the result of the previous lemma as a heuristic argument to arrive at a suitable difference location $\phi$. Note that the difference in the keystream bits at round $t$ in Grain v1, given by $z_t \oplus z_t^{\phi}$ is the $GF(2)$ sum of eight random variables $\gamma_1, \gamma_2, \ldots, \gamma_8$ over $GF(2)$ which are given as follows. For $i = 1, 2, \ldots, 7$, we have $\gamma_i = x_{t+l_i} \oplus x_{t+l_i}^{\phi}$, where $l_1 = 1, l_2 = 2, l_3 = 4, l_4 = 10, l_5 = 31, l_6 = 43, l_7 = 56$. The last variable $\gamma_8$ is given as

$$\gamma_8 = h(y_{t+3}, y_{t+25}, y_{t+46}, y_{t+64}, x_{t+63}) \oplus h(y_{t+3}^{\phi}, y_{t+25}^{\phi}, y_{t+46}^{\phi}, y_{t+64}^{\phi}, x_{t+63}^{\phi}).$$

Consider the case when for some value of $\phi$, $\Delta_{\phi}$-Grain outputs the following values:

$$\chi_{t_1} : [v_{t_1+1} = 0, v_{t_1+2} = 0, v_{t_1+4} = 0, v_{t_1+10} = 0, v_{t_1+31} = 1, v_{t_1+43} = 0, v_{t_1+56} = 2]$$
$$\Upsilon_{t_1} : [u_{t_1+3} = 0, u_{t_1+25} = 1, u_{t_1+46} = 2, u_{t_1+64} = 2, v_{t_1+63} = 2]$$

and

$$\chi_{t_2} : [v_{t_2+1} = 0, v_{t_2+2} = 0, v_{t_2+4} = 2, v_{t_2+10} = 2, v_{t_2+31} = 2, v_{t_2+43} = 2, v_{t_2+56} = 2]$$
$$\Upsilon_{t_2} : [u_{t_2+3} = 0, u_{t_2+25} = 1, u_{t_2+46} = 2, u_{t_2+64} = 2, v_{t_2+63} = 2]$$

Note that the values of $\chi_{t_1}$ and $\Upsilon_{t_1}$ indicate that in two instances of Grain v1 initialized by the states $S_0$ and $S_0^{\phi}$, (note that these states differ only in the $\phi^{th}$ LFSR bit, introduced by a difference in the $\phi^{th}$ IV bit), the following events occur: at KSA round $t_1$, $x_{t_1+1} = x_{t_1+1}^{\phi}$, $x_{t_1+2} = x_{t_1+2}^{\phi}$, $x_{t_1+4} = x_{t_1+4}^{\phi}$, $x_{t_1+10} = x_{t_1+10}^{\phi}$, $x_{t_1+10} = x_{t_1+10}^{\phi}$, $x_{t_1+31} = 1 \oplus x_{t_1+31}^{\phi}$, $y_{t_1+3} = y_{t_1+3}^{\phi}$, $y_{t_1+25} = 1 \oplus y_{t_1+25}^{\phi}$ holds with probability 1. The differences of the remaining variables involved in the computation of the keystream

bit will be either 0 and 1 and depending on the exact value of $S_0$ used to initialize the cipher. Similar inferences may be made at round $t_2$, after looking at the values of $\chi_{t_2}, \Upsilon_{t_2}$. Now, the difference of the keystream bit at round $t_1$ is given by

$$\begin{aligned} z_{t_1} \oplus z_{t_1}^{\phi} &= \bigoplus_{i \in A} \left( x_{t_1+i} \oplus x_{t_1+i}^{\phi} \right) \oplus \left( h(y_{t_1+3}, y_{t_1+25}, \ldots, x_{t_1+63}) \oplus h(y_{t_1+3}^{\phi}, y_{t_1+25}^{\phi}, \ldots, x_{t_1+63}^{\phi}) \right) \\ &= 1 \oplus (x_{t_1+56} \oplus x_{t_1+56}^{\phi}) \oplus \left( h(y_{t_1+3}, y_{t_1+25}, \ldots, x_{t_1+63}) \oplus h(y_{t_1+3}^{\phi}, y_{t_1+25}^{\phi}, \ldots, x_{t_1+63}^{\phi}) \right) \\ &= 1 \oplus \lambda_{11} \oplus \lambda_{12} \end{aligned}$$

Here, $\lambda_{11} = (x_{t_1+56} \oplus x_{t_1+56}^{\phi})$ and $\lambda_{12} = h(y_{t_1+3}, y_{t_1+25}, \ldots, x_{t_1+63}) \oplus h(y_{t_1+3}^{\phi}, y_{t_1+25}^{\phi}, \ldots, x_{t_1+63}^{\phi})$. Similarly at round $t_2$, we have

$$\begin{aligned} z_{t_2} \oplus z_{t_2}^{\phi} &= \bigoplus_{i \in A} \left( x_{t_2+i} \oplus x_{t_2+i}^{\phi} \right) \oplus \left( h(y_{t_2+3}, y_{t_2+25}, \ldots, x_{t_2+63}) \oplus h(y_{t_2+3}^{\phi}, y_{t_2+25}^{\phi}, \ldots, x_{t_2+63}^{\phi}) \right) \\ &= \bigoplus_{i \in \{4,10,31,43,56\}} (x_{t_2+i} \oplus x_{t_2+i}^{\phi}) \oplus \left( h(y_{t_2+3}, y_{t_2+25}, \ldots, x_{t_2+63}) \oplus h(y_{t_2+3}^{\phi}, y_{t_2+25}^{\phi}, \ldots, x_{t_2+63}^{\phi}) \right) \\ &= \lambda_{21} \oplus \lambda_{22} \oplus \lambda_{23} \oplus \lambda_{24} \oplus \lambda_{25} \oplus \lambda_{26} \end{aligned}$$

where the values of $\lambda_{2i}'$s have been assigned as at round $t_1$. Note that the randomness in $z_{t_1} \oplus z_{t_1}^{\phi}$ comes from only two random variables $\lambda_{11}$ and $\lambda_{12}$ whereas the randomness in $z_{t_2} \oplus z_{t_2}^{\phi}$ comes from 6 random variables. Now this is certainly not a conclusive proof (as we still do not know the quantity of the bias of each variable $\lambda_{1i}/\lambda_{2i}$ or if they are independent or not), but one can make a heuristic argument that if the biases in the $\lambda_{1i}'$s and $\lambda_{2i}'$s are not too different, and if one can intuitively/empirically determine that the $\lambda_{1i}'$s, $\lambda_{2i}'$s are independent, then the bias of the variable $z_{t_1} \oplus z_{t_1}^{\phi}$ is likely to be much higher than that of $z_{t_2} \oplus z_{t_2}^{\phi}$, following the arguments outlined immediately after Lemma 1.

### 4.1 Search for a suitable candidate for $\phi$

Given two distributions $Ber(\frac{1}{2} + \epsilon)$ and $Ber(\frac{1}{2})$ (i.e., the Uniform distribution over $GF(2)$) and an efficient algorithm to extract multiple samples from these distributions, it usually takes number of samples proportional to $\left(\frac{1}{\epsilon}\right)^2$ to distinguish these with a constant probability of success. Therefore our goal was to find some $\phi \in [0, 63]$ so that the bias of $z_t \oplus z_t^{\phi}$, for some $t > 104$, would be around $2^{-14}$ to $2^{-15}$, so that it would be possible to distinguish it from samples generated from a uniformly random distribution by employing at most $2^{28}$ to $2^{30}$ pairs of chosen IVs differing only at bit location $\phi$.

One way of searching for such a $\phi$, is by executing the engine $\Delta_{\phi}$-$\mathsf{Grain}_{\mathsf{KSA}}$ for all $\phi \in [0, 63]$, and examining the $\chi_t$ vector output by the engine for every $t > 104$. The vector $\chi_t$ contains 7 elements, if the number of probabilistic elements (i.e. number of 2′s) in $\chi_t$, (for some $t > 104$ and $\phi \in [0, 63]$) is less (say not more than 1 or 2), then as per the arguments outlined in the previous subsection, $\phi$ and $t$ are likely to be good candidates for performing the attack, as in such a case $z_t \oplus z_t^{\phi}$ is likely to have some non-negligible bias which can in all likelihood be detected within $2^{30}$ pairs of chosen IVs differing at bit position $\phi$. However there are several practical difficulties in running such an Algorithm.

1. For $t > 104$, the number of 2′s in $\chi_t$ is usually never less than 4, for any value of $\phi \in [0, 63]$.

2. This being the case one needs to impose **Type 1** and **Type 2** conditions on the Secret Key and IV bits to prevent the propagation of differences to the NFSR whenever possible, much like the

ones mentioned in Section 3.3. Thus, as mentioned in Section 3.3, to model this event, one needs to modify the routine $\Delta_\phi$-$\mathsf{Grain}_{\mathsf{KSA}}$ by inserting code snippets in Line 0.1 in Algorithm 1.

3. During the state updates of the LFSR, NFSR in the KSA, the keystream bit $z_t$ is summed with the feedback functions of both the LFSR and NFSR. Thus in the earlier KSA rounds, whenever $z_t \oplus z_t^\phi$ is probabilistic (this occurs when the keystream element $\pi_t$ produced by $\Delta_\phi$-$\mathsf{Grain}_{\mathsf{KSA}}$ is equal to 2), the differential is likely to propagate to the NFSR. In this case, we should try to impose the **Type 1** and **Type 2** conditions to prevent the propagation. Thus for an arbitrary $\phi$, the code snippet to be inserted in Line 0.1 in Algorithm 1 is as follows:

$$\textbf{if} \;\; \pi_t = 2: \;\; v_{t+n} \leftarrow 0$$

4. As a case in point, one can examine the attack presented in [23, 24], in which the attacker stops the propagation of the differential at 3 KSA rounds $t = 12, 34, 40$. To do so, the attacker needed to impose 27 **Type 1** conditions which set individual bits of the IV to 0/1. This reduced the effective IV space to $\{0,1\}^{64-27} = \{0,1\}^{37}$. Thus, this tells us that imposing **Type 1/2** conditions at more than 3 to 4 KSA rounds may well shrink the effective IV space to below $\{0,1\}^{30}$, and in such a case we will not have enough IVs to detect any bias in $z_t \oplus z_t^\phi$. Thus preventing the propagation of the difference at more than 4 KSA rounds is not feasible.

5. As we shall see shortly, another fundamental requirement to mount the attack, is the ability to enumerate the explicit algebraic expressions of $z_t \oplus z_t^\phi$ (in terms of the Secret Key and IV variables) at all the rounds $t$ where the propagation of a difference to the NFSR is to be prevented. Thereafter our goal would be to express certian internal variables of the cipher in the form $F_3(K) \oplus F_4(V)$, where $F_3, F_4$ are functions on only the Secret Key bits and the IV bits repectively. Using Computer Algebra Systems like `SAGE` [26], we were able do this for KSA rounds upto $t = 45$ on a system running on a 3.2 GHz processor and 8 GB of internal memory. Thus we did not attempt to prevent the propagation of the difference at any round $t > 45$.

6. Finally, we would also like to note that it is not necessary to run the engine $\Delta_\phi$-$\mathsf{Grain}_{\mathsf{KSA}}$ with all the modifications and constraints listed above, for all values of $\phi \in [0, 63]$. For example, due to the state updates in Grain v1, it is easy to see that $\Delta_{36}$-$\mathsf{Grain}_{\mathsf{KSA}}$ at KSA round $t-1$, will give the same result as $\Delta_{37}$-$\mathsf{Grain}_{\mathsf{KSA}}$ at round $t$. Thus it is sufficient to run the engines at all values of $\phi$ just before the tap locations i.e., for values equal to $2, 12, 22, 24, 37, 45, 50, 61, 63$.

By running the engine $\Delta_\phi$-$\mathsf{Grain}_{\mathsf{KSA}}$, under the aforementioned modifications and restrictions, for the values of $\phi$ listed above, we were able to determine that for $\phi = 61$ and $t = 105$, and after preventing the propagation of the differential at KSA rounds $t = 15, 36, 39, 42$, the value of $\chi_{105}$ was obtained as follows.

$$\chi_{105} = [v_{106} = 1,\ v_{107} = 0,\ v_{109} = 0,\ v_{115} = 0,\ v_{136} = 0,\ v_{148} = 2,\ v_{161} = 2]$$

Since there are only two $2'$s in this vector, this seemed to be a good choice of the attack parameters and we proceeded to mount the attack from here.

### 4.2 Algebraic Details of the attack for $\phi = 61$

Algebraically speaking, introducing a difference at the $61^{st}$ bit position of the IV is equivalent to analyzing two initializations of the Grain v1 cipher, one with the initial state equal to

$$X_0 = [k_0, k_1, \ldots, k_{79}], \quad Y_0 = [\nu_0, \nu_1, \ldots, \nu_{61}, \ldots, \nu_{63}, 1, 1, \ldots, 1],$$

and the other with the initial state equal to

$$X_0^{\phi} = [k_0, k_1, \ldots, k_{79}], \quad Y_0^{\phi} = [\nu_0, \nu_1, \ldots, 1 \oplus \nu_{61}, \ldots, \nu_{63}, 1, 1, \ldots, 1].$$

where $K = [k_0, k_1, \ldots, k_{79}]$, $V = [\nu_0, \nu_1, \ldots, \nu_{61}, \ldots, \nu_{63}]$, $V^{\phi} = [\nu_0, \nu_1, \ldots, 1 \oplus \nu_{61}, \ldots, \nu_{63}]$ are the formal notations for the Secret Key and the two IVs that differ in the $61^{st}$ bit position. The primary task of cryptanalysis is to monitor the propagation of the differentials across the internal states in the $t^{th}$ KSA round, i.e., between $S_t = X_t||Y_t$ and $S_t^{\phi} = X_t^{\phi}||Y_t^{\phi}$. During the execution of $\Delta_{61}$-$\mathsf{Grain_{KSA}}$, we had chosen to halt the propagation of the differentials at $t = 15, 36, 39, 42$. Let us now enumerate the algebraic conditions one would need to impose to halt the propagation of the differential at these rounds. We will again take help of the outputs of the modified $\Delta_{61}$-$\mathsf{Grain_{KSA}}$ to do so.

1. $\mathbf{t = 15}$: At this round $\chi_t = \mathcal{G}_{lin,t} = \mathcal{G}_{nlin,t} = \mathbf{0}, u_t = 0$ and $\Upsilon_t = [u_{t+3} = 0, u_{t+25} = 0, u_{t+46} = 1, u_{t+64} = 0, v_{t+63} = 0]$. This implies that of all state bits of $S_t, S_t^{\phi}$ at $t = 15$ involved in the computation of $x_t,\ x_t^{\phi}$, only $y_{64} = 1 \oplus y_{64}^{\phi}$ holds deterministically and all the differences of all other state bits is deterministically 0. So we have

$$\begin{aligned} x_{80+15} \oplus x_{80+15}^{\phi} &= [\ g(X_{15}) \oplus y_{15} \oplus z_{15}\ ] \oplus [\ g(X_{15}^{\phi}) \oplus y_{15}^{\phi} \oplus z_{15}^{\phi}\ ] = z_{15} \oplus z_{15}^{\phi} \\ &= h(y_{18}, y_{40}, y_{61}, y_{79}, x_{78}) \oplus h(y_{18}, y_{40}, 1 \oplus y_{61}, y_{79}, x_{78}) \\ &= \nu_{18}(1 \oplus \nu_{40} \oplus k_{78}) \oplus \nu_{40}k_{78} \oplus k_{78} \oplus 1 \end{aligned}$$

   To prevent the propagation of the differential $x_{80+15} \oplus x_{80+15}^{\phi}$ must be set to 0 by imposing **Type 1/2** conditions. Note that if we set $\nu_{18} = 1, \nu_{40} = 0$, then $x_{80+15} \oplus x_{80+15}^{\phi}$ becomes identically zero. Thus we have imposed two **Type1** conditions at this round.

2. $\mathbf{t = 36}$: At this KSA round we again have $\chi_t = \mathcal{G}_{lin,t} = \mathcal{G}_{nlin,t} = \mathbf{0}, u_t = 0$ and $\Upsilon_t = [u_{t+3} = 0, u_{t+25} = 1, u_{t+46} = 0, u_{t+64} = 0, v_{t+63} = 0]$. This implies that

$$\begin{aligned} x_{80+36} \oplus x_{80+36}^{\phi} &= [\ g(X_{36}) \oplus y_{36} \oplus z_{36}\ ] \oplus [\ g(X_{36}^{\phi}) \oplus y_{36}^{\phi} \oplus z_{36}^{\phi}\ ] = z_{36} \oplus z_{36}^{\phi} \\ &= h(y_{39}, y_{61}, y_{82}, y_{100}, x_{99}) \oplus h(y_{39}, 1 \oplus y_{61}, y_{82}, y_{100}, x_{99}) \\ &= 1 \oplus \nu_{39}y_{82} \oplus y_{82}x_{99} \end{aligned}$$

   To halt the differential we need $\nu_{39} = 0, y_{82} = 1, x_{99} = 1$. The first is a simple **Type 1** condition but the remaining conditions need further investigation. If we set $\nu_{48} = 0$, the algebraic expression for $y_{82}$ becomes:

$$y_{82} = k_3 \oplus k_4 \oplus k_6 \oplus k_{12} \oplus k_{33} \oplus k_{45} \oplus k_{58} \oplus \nu_2 \oplus \nu_5 \oplus \nu_{15} \oplus \nu_{25} \oplus \nu_{27} \oplus \nu_{53} \oplus 1.$$

   Therefore if we set the following **Type 2** condition $y_{82}$ becomes identically 1.

$$C_1 :\ \nu_{15} \oplus \nu_2 \oplus \nu_5 \oplus \oplus \nu_{25} \oplus \nu_{27} \oplus \nu_{53} \oplus K_1 = 0, \tag{2}$$

   where $K_1 = k_3 \oplus k_4 \oplus k_6 \oplus k_{12} \oplus k_{33} \oplus k_{45} \oplus k_{58}$. Now to set $x_{99} = 1$, we need to impose the following **Type 1** conditions $\nu_2 = \nu_{47} = \nu_{49} = \nu_{22} = \nu_{44} = \nu_5 = \nu_{27} = 0$, and the following **Type 2** condition.

$$C_2 :\ \nu_3 \oplus \nu_1 \oplus \nu_4 \oplus \nu_6 \oplus \nu_{16} \oplus \nu_{19} \oplus \nu_{28} \oplus \nu_{41} \oplus \nu_{54} \oplus K_2 = 0, \tag{3}$$

   here $K_2$ is an expression in the Secret Key bits of algebraic degree 10 and 177 monomials.

3. **t = 39**: At this round we have $\chi_t = \mathcal{G}_{lin,t} = \mathcal{G}_{nlin,t} = \mathbf{0}, u_t = 0$ and $\Upsilon_t = [u_{t+3} = 0, u_{t+25} = 0, u_{t+46} = 0, u_{t+64} = 1, v_{t+63} = 0]$. Therefore we have,

$$\begin{aligned} x_{80+39} \oplus x^{\phi}_{80+39} &= [\ g(X_{39}) \oplus y_{39} \oplus z_{39}\ ] \oplus [\ g(X^{\phi}_{39}) \oplus y^{\phi}_{39} \oplus z^{\phi}_{39}\ ] = z_{39} \oplus z^{\phi}_{39} \\ &= h(y_{42}, y_{64}, y_{85}, y_{103}, x_{102}) \oplus h(y_{42}, y_{64}, y_{85}, 1 \oplus y_{103}, x_{102}) \\ &= \nu_{42} \oplus y_{85} \oplus x_{102} \oplus \nu_{42} y_{85} \oplus y_{85} x_{102} \end{aligned}$$

To set this to zero we need to impose $\nu_{42} = 0, y_{85} = 0, x_{102} = 0$. Again the first is a simple **Type 1** condition. To set the other variables to zero we need to do the following: if we set $\nu_{51} = 0$, the expression for $y_{85}$ becomes

$$y_{85} = k_6 \oplus k_7 \oplus k_9 \oplus k_{15} \oplus k_{36} \oplus k_{48} \oplus k_{61} \oplus \nu_8 \oplus \nu_{28} \oplus \nu_{30} \oplus \nu_{43} \oplus \nu_{56}$$

Thus if the following **Type 2** condition is applied then $y_{85}$ becomes identically zero.

$$C_3 :\ \nu_{43} \oplus \nu_8 \oplus \nu_{28} \oplus \nu_{30} \oplus \nu_{56} \oplus K_3 = 0, \tag{4}$$

where $K_3 = k_6 \oplus k_7 \oplus k_9 \oplus k_{15} \oplus k_{36} \oplus k_{48} \oplus k_{61}$. If we now set $\nu_8 = \nu_{30} = \nu_{25} = \nu_{50} = \nu_{52} = 0$, and the following **Type 2** condition $C_4$ then $x_{102}$ also is nullified.

$$C_4 :\ \nu_{57} \oplus \nu_4 \oplus \nu_6 \oplus \nu_7 \oplus \nu_9 \oplus \nu_{19} \oplus \nu_{31} \oplus K_4 = 0, \tag{5}$$

$K_4$ is an expression in the Secret Key bits of algebraic degree 15 and 2612 monomials.

4. **t = 42**: At this KSA round we again have $\chi_t = \mathcal{G}_{lin,t} = \mathcal{G}_{nlin,t} = \mathbf{0}, u_t = 0$ and $\Upsilon_t = [u_{t+3} = 0, u_{t+25} = 0, u_{t+46} = 0, u_{t+64} = 1, v_{t+63} = 0]$. This implies that

$$\begin{aligned} x_{80+42} \oplus x^{\phi}_{80+42} &= [\ g(X_{42}) \oplus y_{42} \oplus z_{42}\ ] \oplus [\ g(X^{\phi}_{42}) \oplus y^{\phi}_{42} \oplus z^{\phi}_{42}\ ] = z_{42} \oplus z^{\phi}_{42} \\ &= h(y_{45}, y_{67}, y_{88}, y_{106}, x_{105}) \oplus h(y_{45}, y_{67}, y_{88}, 1 \oplus y_{106}, x_{105}) \\ &= \nu_{45} \oplus y_{88} \oplus x_{105} \oplus \nu_{45} y_{88} \oplus y_{88} x_{105} \end{aligned}$$

Again we need to set $\nu_{45} = 0, y_{88} = 0, x_{105} = 0$ to nullify this difference. If we set $\nu_{54} = 0$, the expression for $y_{88}$ becomes

$$y_{88} = k_9 \oplus k_{10} \oplus k_{12} \oplus k_{18} \oplus k_{39} \oplus k_{51} \oplus k_{64} \oplus \nu_{11} \oplus \nu_{21} \oplus \nu_{31} \oplus \nu_{33} \oplus \nu_{46} \oplus \nu_{59} \oplus 1$$

Therefore to nullify $y_{88}$ we apply the following **Type 2** condition.

$$C_5 :\ \nu_{46} \oplus \nu_{11} \oplus \nu_{21} \oplus \nu_{31} \oplus \nu_{33} \oplus \nu_{59} \oplus K_5 = 0, \tag{6}$$

where $K_5 = k_9 \oplus k_{10} \oplus k_{12} \oplus k_{18} \oplus k_{39} \oplus k_{51} \oplus k_{64} \oplus 1$. Now if we set $\nu_{11} = \nu_{33} = \nu_{28} = \nu_{53} = \nu_{55} = 0$ and the following **Type 2** condition $C_6$, then $x_{105}$ is also nullified.

$$C_6 :\ \nu_{60} \oplus \nu_1 \oplus \nu_4 \oplus \nu_7 \oplus \nu_9 \oplus \nu_{10} \oplus \nu_{12} \oplus \nu_{26} \oplus \nu_{34} \oplus K_6 = 0, \tag{7}$$

where $K_6$ is an expression in the Secret Key bits of algebraic degree 15 and 2620 monomials.

The six **Type 2** relations $C_1, C_2, \ldots, C_6$ obtained in Equations (2)-(7) are crucial to the Key recovery attack. We note that due to the several **Type 1** relations, a total of 25 of the IV bits are assigned either 0 or 1 and hence the effective IV space is reduced to $\{0,1\}^{39}$. In fact, many of the

IV bits that appear in the expressions for the **Type 2** conditions, are later set to zero in one of the following KSA rounds. So we rewrite the final expressions for the **Type 2** conditions.

$$
\begin{aligned}
&C_1 : \nu_{15} \oplus K_1 = 0, && C_2 : \nu_3 \oplus \nu_1 \oplus \nu_4 \oplus \nu_6 \oplus \nu_{16} \oplus \nu_{19} \oplus \nu_{41} \oplus K_2 = 0,\\
&C_3 : \nu_{43} \oplus \nu_{56} \oplus K_3 = 0, && C_4 : \nu_{57} \oplus \nu_4 \oplus \nu_6 \oplus \nu_7 \oplus \nu_9 \oplus \nu_{19} \oplus \nu_{31} \oplus K_4 = 0,\\
&C_5 : \nu_{46} \oplus \nu_{21} \oplus \nu_{31} \oplus \nu_{59} \oplus K_5 = 0, && C_6 : \nu_{60} \oplus \nu_1 \oplus \nu_4 \oplus \nu_7 \oplus \nu_9 \oplus \nu_{10} \oplus \nu_{12} \oplus \nu_{26} \oplus \nu_{34} \oplus K_6 = 0.
\end{aligned}
$$

Now since the attacker does not know the values of the six expressions $K_1, K_2, \ldots, K_6$, he will need to guess them in order to satisfy the **Type 2** conditions. Let $\{\nu_{15}, \nu_3, \nu_{43}, \nu_{57}, \nu_{46}, \nu_{60}\}$ be the set of dynamic cube variables. We will partition the IV space (which now has $2^{39}$ elements) into $2^6$ disjoint sets $T_i,\ 0 \leq i < 63$ as follows. Let $\mathbf{U} = [K_1, K_2, K_3, K_4, K_5, K_6]$. Then, for each $\mathbf{U} \in \{0,1\}^6$ the set $T_{\mathbf{U}}$ is generated as follows:

1. Define the Set

$$
\begin{aligned}
T_{\mathbf{U}} \leftarrow \{V \in \{0,1\}^{64} \mid\ & \nu_{18} = 1, \nu_{40} = 0, \nu_{39} = 0, \nu_{48} = 0,\ \nu_2 = 0, \nu_{47} = 0, \nu_{49} = 0, \nu_{22} = 0,\\
& \nu_{44} = 0,\ \nu_5 = 0, \nu_{27} = 0, \nu_{42} = 0, \nu_{51} = 0,\ \nu_8 = 0, \nu_{30} = 0, \nu_{25} = 0,\\
& \nu_{50} = 0, \nu_{52} = 0, \nu_{45} = 0, \nu_{54} = 0, \nu_{11} = 0, \nu_{33} = 0, \nu_{53} = 0, \nu_{55} = 0, \nu_{28} = 0\}
\end{aligned}
$$

2. For all $V \in T_{\mathbf{U}}$, adjust $\nu_{15}, \nu_3, \nu_{43}, \nu_{57}, \nu_{46}, \nu_{60}$ according to the guessed value of $\mathbf{U}$:

$$
\begin{aligned}
&\nu_{15} \leftarrow K_1, && \nu_3 \leftarrow \nu_1 \oplus \nu_4 \oplus \nu_6 \oplus \nu_{16} \oplus \nu_{19} \oplus \nu_{41} \oplus K_2,\\
&\nu_{43} \leftarrow \nu_{56} \oplus K_3, && \nu_{57} \leftarrow \nu_4 \oplus \nu_6 \oplus \nu_7 \oplus \nu_9 \oplus \nu_{19} \oplus \nu_{31} \oplus K_4,\\
&\nu_{46} \leftarrow \nu_{21} \oplus \nu_{31} \oplus \nu_{59} \oplus K_5, && \nu_{60} \leftarrow \nu_1 \oplus \nu_4 \oplus \nu_7 \oplus \nu_9 \oplus \nu_{10} \oplus \nu_{12} \oplus \nu_{26} \oplus \nu_{34} \oplus K_6.
\end{aligned}
$$

Note that if $V \in T_{\mathbf{U}}$ for some $V$ and $\mathbf{U}$, then $V^{\phi} \in T_{\mathbf{U}}$. As it turns out, if the conditions $C_1$ to $C_6$ are all satisfied then the distribution of $z_{105} \oplus z_{105}^{\phi}$ exhibits appreciable bias. It was experimentally observed that

$$
\mathsf{Pr}\left[z_{105} \oplus z_{105}^{\phi} = 0 \mid C_i \text{ is satisfied } \forall\ i \in [1,6]\right] \approx \frac{1}{2} + 0.0002, \tag{8}
$$

where the probability was calculated by the randomness generated over the Key and IV space. For a proof of Equation (8), please refer to Appendix A. Now to mount the attack, the attacker tries to compute the distribution of $z_{105} \oplus z_{105}^{\phi}$ in each of the 64 sets $T_{\mathbf{U}}$. Note that all the conditions $C_1, C_2, \ldots, C_6$ are satisfied in only one of these sets $T_{\mathbf{U_0}}$ where $\mathbf{U_0}$ is the correct value of $\mathbf{U}$. The attacker will therefore be able to observe the bias in the set $T_{\mathbf{U_0}}$, and by standard randomness assumptions, he should not be able to detect any bias in the other sets, thereby determining the values of the six expressions $K_1, K_2, \ldots, K_6$.

### 4.3 Further Issues

We have just stated that ideally the attacker should not be able to detect any bias in any set $T_{\mathbf{U}'}$ such that $\mathbf{U}' \neq \mathbf{U_0}$. But as it turns out, he will be able detect some bias in three sets other than $T_{\mathbf{U_0}}$. These sets are those where **a**) $C_5$ is not satisfied but $C_6$ is, **b**) $C_6$ is not satisfied but $C_5$ is and **c**) Neither $C_5$ nor $C_6$ is satisfied. is satisfied. This points to the fact that even if we had not halted the difference propagation at $t = 42$, we would have obtained some bias in the distribution of $z_{105} \oplus z_{105}^{\phi}$. In fact, if we had run

the engine $\Delta_{61}$-$\mathsf{Grain}_{\mathsf{KSA}}$ and opted to nullify the difference at rounds $t = 15, 36, 39$ only, we would still have obtained the value of $\chi_{105} = [v_{106} = 1,\ v_{107} = 0,\ v_{109} = 0,\ v_{115} = 0,\ v_{136} = 0,\ v_{148} = 2,\ v_{161} = 2]$. However, in spite of this, nullifying the difference at $t = 42$, results in much higher bias in $z_{105} \oplus z^{\phi}_{105}$. This is because the bias in the random variable $h(y_{108}, y_{130}, \ldots, x_{168}) \oplus h(y^{\phi}_{108}, y^{\phi}_{130}, \ldots, x^{\phi}_{168})$ is much higher if the difference is nullified at $t = 42$. As a result although the attacker detects bias in three sets other than $T_{\mathbf{U_0}}$, the bias in these sets is usually much lower than in $T_{\mathbf{U_0}}$. So the attacker can compute the distribution of $z_{105} \oplus z^{\phi}_{105}$ in all the 64 sets and deduce that the set in which he obtains the highest bias of $z_{105} \oplus z^{\phi}_{105}$ to be the correct value of $\mathbf{U}$.

Although each set $T_i$ contains $2^{32}$ pairs of IVs (i.e., $V$ and $V^{\phi}$), it has been observed that one need not use all the IV pairs in each $T_i$ to compute the distribution of $z_{105} \oplus z^{\phi}_{105}$. Instead one can do the following. Out of the 64 IV bits, we know that 25 have been assigned with constants via the **Type 1** conditions, 6 are dynamic variables attached to the **Type 2** conditions, and 1 (i.e. $\nu_{61}$) is variable over which we are computing the cube sum $z_{105} \oplus z^{\phi}_{105}$. This leaves us with 32 'free' IV variables. The attacker can further set some $n_1$ of these 32 free variables to constants and reconstruct all the Sets $T_i$, as per the methods outlined in the previous Subsection. This method reduces the cardinality of each set $T_i$ by a factor of $2^{n_1}$. But for small $n_1$ ($\leq 5$), the attacker should be able to detect maximum bias in the correct set $T_{\mathbf{U_0}}$.

### 4.4 Experimental Results

We experimented with the value of $n_1 = 5$, with around 1000 randomly generated Secret Keys. Thus the complexity of our algorithm was $2^{32-n_1+6+1} = 2^{39-n_1}$ evaluations of the Grain v1 KSA function reduced to 105 rounds using chosen IVs. The experiments were run on a 3.2 GHz Intel Xeon processor and took around 23 hours to complete for each Secret Key. The results are tabulated in Table 1. It can be seen that all the expressions $K_1, K_2, \ldots, K_6$ are determined correctly for around 92% of the Secret Keys. In the remaining cases there may be error in determining $K_5$ and/or $K_6$. In any case, the Algorithm always determines the value of $K_1, K_2, K_3, K_4$ correctly.

**Table 1.** Experimental results for $n_1 = 5$

| Results | Percentage |
|---|---|
| All $K_1, K_2, \ldots, K_6$ determined correctly | 92% |
| $K_1, K_2, \ldots, K_5$ determined correctly but $K_6$ determined incorrectly | 3% |
| $K_1, K_2, \ldots, K_4, K_6$ determined correctly but $K_5$ determined incorrectly | 3% |
| $K_1, K_2, \ldots, K_4$ determined correctly but $K_5, K_6$ determined incorrectly | 2% |

## 5 Conclusion

In this paper we provide a framework to attack reduced round Grain v1. Selecting cubes for single or higher dimensional differential cube attacks, is not an exact science, and generally the attacker is able to arrive at a suitable cube by aid of intuition or testing a lot of random cubes. In this work, we have made a primitive attempt to provide some structure to the search for suitable cubes as far as the stream cipher Grain v1 is concerned. Our attack retrieves 6 expressions in the Secret Key bits (of which three are linear and three of higher algebraic degree) for a version of Grain v1 in which the KSA is reduced to 105 (out of 160) initialization rounds. The attack uses $2^{39-n_1}$ evaluations (here $n_1 \approx 5$) of the 105 round KSA function of Grain v1 with chosen IVs and takes around 23 CPU hours to complete. This is an improvement of 8 KSA rounds over the previously best attack published against this cipher.

## References


1. The ECRYPT Stream Cipher Project. eSTREAM Portfolio of Stream Ciphers. Revised on September 8, 2008.
2. M. Ågren, M. Hell, T. Johansson and W. Meier. A New Version of Grain-128 with Authentication. Symmetric Key Encryption Workshop 2011, DTU, Denmark, February 2011.
3. S. Banik. Some Insights into Differential Cryptanalysis of Grain v1. To appear in ACISP 2014.
4. S. Banik, S. Maitra and S. Sarkar. A Differential Fault Attack on Grain family under reasonable assumptions. In Indocrypt 2012, LNCS, Vol. 7668, pp. 191-208, 2012.
5. S. Banik, S. Maitra and S. Sarkar. A Differential Fault Attack on the Grain Family of Stream Ciphers. In CHES 2012, LNCS, Vol. 7428, pp. 122-139, 2012.
6. S. Banik, S. Maitra, S. Sarkar and M. S. Turan. A Chosen IV Related Key Attack on Grain-128a. In ACISP 2013, LNCS, Vol. 7959, pp. 13-26, 2013.
7. C. Berbain, H. Gilbert and A. Maximov. Cryptanalysis of Grain. In FSE 2006, LNCS, Vol. 4047, pp. 15–29, 2006.
8. A. Berzati, C. Canovas, G. Castagnos, B. Debraize, L. Goubin, A. Gouget, P. Paillier, S. Salgado. Fault Analysis of Grain-128. In: IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 7–14, 2009.
9. T. E. Bjørstad. Cryptanalysis of Grain using Time/Memory/Data tradeoffs (v1.0 / 2008-02-25). Available at `http://www.ecrypt.eu.org/stream`.
10. C. De Cannière, O. Küçük and B. Preneel. Analysis of Grain's Initialization Algorithm. In AFRICACRYPT 2008, LNCS, Vol. 5023, pp. 276–289, 2008.
11. I. Dinur, T. Güneysu, C. Paar, A. Shamir, R. Zimmermann. An Experimentally Verified Attack on Full Grain-128 Using Dedicated Reconfigurable Hardware. In ASIACRYPT 2011, LNCS Vol. 7073, pp. 327–343, 2011.
12. I. Dinur, A. Shamir. Breaking Grain-128 with Dynamic Cube Attacks. In FSE 2011, LNCS, Vol. 6733, pp. 167–187, 2011.
13. I. Dinur, A. Shamir. Cube Attacks on Tweakable Black Box Polynomials. In EUROCRYPT 2009, LNCS, Vol. 5479, pp. 278-299, 2009.
14. I. Dinur, A. Shamir. Applying cube attacks to stream ciphers in realistic scenarios. In Cryptography and Communications, Vol. 4, Issue 3-4, pp 217-232, 2012.
15. 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.
16. P. Fouque and T. Vannet. Improving Key Recovery to 784 and 799 rounds of Trivium using Optimized Cube Attacks. To appear in FSE 2013.
17. 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.
18. 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`.
19. M. Hell, T. Johansson and W. Meier. A Stream Cipher Proposal: Grain-128. In IEEE International Symposium on Information Theory (ISIT 2006), 2006.
20. K. Khoo and C. Tan. New time-memory-data trade-off attack on the estream finalists and modes of operation of block ciphers. In 7th ACM Symposium on Information, Computer and Communications Security, ASIACCS, pp. 20-21, 2012.
21. M. Lehmann, W. Meier: Conditional Differential Cryptanalysis of Grain-128a. In CANS 2012, LNCS, Vol. 7712, pp. 1–11, 2012.
22. 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`
23. S. Knellwolf. Cryptanalysis of Hardware-Oriented Ciphers, The Knapsack Generator, and SHA-1. PhD Dissertation, 2012. Available at `http://e-collection.library.ethz.ch/eserv/eth:5999/eth-5999-02.pdf`
24. 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.
25. P. Stankovski. Greedy Distinguishers and Nonrandomness Detectors. In INDOCRYPT 2010, LNCS, Vol. 6498, pp. 210–226, 2010.
26. 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).


## Appendix A: Computing the distribution of $z_{105} \oplus z_{105}^{\phi}$ for $\phi = 61$ if all $C_1, C_2, \ldots, C_6$ are satisfied

We will prove the bias in the distribution of $z_{105} \oplus z_{105}^{\phi}$ along similar lines of the proof of the bias of $z_{97} \oplus z_{97}^{37}$ reported in [3]. The probability values we calculate are computed over the randomness due to the Key bits and the those IV bits not assigned by the **Type 1, 2** relations. However, these results

also hold, even if the Key is fixed, and the randomness comes only from the IV bits. First we state a straightforward lemma without proof.

**Lemma 2.** *Let $F$ be an $i$-variable Boolean function, with $wt(F) = \mathsf{w}$. If the vector $X$ is chosen uniformly from $\{0,1\}^i$ then $\mathsf{Pr}[F(X) = 0] = 1 - \frac{\mathsf{w}}{2^i}$.*

We begin by inspecting the output of $\Delta_{61}$-$\mathsf{Grain}_{\mathsf{KSA}}$ at round $t = 105$, in which the difference is nullified at rounds $t = 15, 36, 39, 42$. At $t = 105$, we have

$$\chi_{105} = [v_{106} = 1,\ v_{107} = 0,\ v_{109} = 0,\ v_{115} = 0,\ v_{136} = 0,\ v_{148} = 2,\ v_{161} = 2]$$

$$\Upsilon_{105} = [u_{108} = 1,\ u_{130} = 1,\ u_{151} = 2,\ u_{169} = 2,\ v_{168} = 2]$$

This implies that of all the bits of $S_{105}$, $S^{\phi}_{105}$ involved in the computation of $z_{105}$ and $z^{\phi}_{105}$ respectively, the relations between only i) $x_{148}, x^{\phi}_{148}$ ii) $x_{161}, x^{\phi}_{161}$ iii) $y_{151}, y^{\phi}_{151}$ iv) $y_{169}, y^{\phi}_{169}$ v) $x_{168}, x^{\phi}_{168}$ are probabilistic. Therefore we have

$$\begin{aligned} z_{105} \oplus z^{\phi}_{105} = 1 \oplus [x_{148} \oplus x^{\phi}_{148}] \oplus [x_{161} \oplus x^{\phi}_{161}] \oplus \\ [h(y_{108}, y_{130}, y_{151}, y_{169}, x_{168}) \oplus h(1 \oplus y_{108}, 1 \oplus y_{130}, y^{\phi}_{151}, y^{\phi}_{169}, x^{\phi}_{168})] \end{aligned} \tag{9}$$

We assume that the random variables $x_{148} \oplus x^{\phi}_{148}$, $x_{161} \oplus x^{\phi}_{161}$, $y_{151} \oplus y^{\phi}_{151}$, $y_{169} \oplus y^{\phi}_{169}$ and $x_{168} \oplus x^{\phi}_{168}$ are statistically mutually independent of one another. It is difficult to prove this assumption theoretically but extensive computer simulations have shown that one can make this assumption.

### Calculating $\mathsf{Pr}[x_{148} \oplus x^{\phi}_{148} = 0]$

To find this distribution we need to look at the state of our modified $\Delta_{61}$-$\mathsf{Grain}_{\mathsf{KSA}}$ at $t = 148 - 80 = 68$. At this we have $u_{68} = 0, \Upsilon_{68} = \mathbf{0}$, $\mathcal{G}_{lin,68} = [v_{68} = 0, v_{82} = 0, v_{130} = 1], \mathcal{G}_{nlin,68} = \mathbf{0}$, and

$$\chi_{68} = [v_{69} = 0, v_{70} = 0, v_{72} = 0, v_{78} = 0, v_{99} = 0, v_{111} = 0, v_{124} = 2]$$

So we have

$$\begin{aligned} x_{148} \oplus x^{\phi}_{148} &= [g(X_{68}) \oplus y_{68} \oplus z_{68}] \oplus [g(X^{\phi}_{68}) \oplus y^{\phi}_{68} \oplus z^{\phi}_{68}] \\ &= 1 \oplus (x_{124} \oplus x^{\phi}_{124}) \end{aligned}$$

So in order to compute the above probability we need to compute the distribution of $(x_{124} \oplus x^{\phi}_{124})$ first. At $t = 124 - 80 = 44$, we have $u_{44} = 0, \Upsilon_{44} = [u_{47} = 0, u_{69} = 0, u_{90} = 1, u_{108} = 1, v_{107} = 0], \chi_{44} = \mathbf{0}$, $\mathcal{G}_{lin,44} = [v_{44} = 0, v_{58} = 0, v_{106} = 1], \mathcal{G}_{nlin,68} = \mathbf{0}$. Note that $y_{47} = \nu_{47} = 0$ according to one of the **Type 1** conditions, and $y_{69} = 1$ as defined by the padding rule of Grain v1. So we have

$$\begin{aligned} x_{124} \oplus x^{\phi}_{124} &= [g(X_{44}) \oplus y_{44} \oplus z_{44}] \oplus [g(X^{\phi}_{44}) \oplus y^{\phi}_{44} \oplus z^{\phi}_{44}] \\ &= 1 \oplus h(y_{47}, y_{69}, y_{90}, y_{108}, x_{107}) \oplus h(y_{47}, y_{69}, 1 \oplus y_{90}, 1 \oplus y_{108}, x_{107}) \\ &= x_{107} \oplus y_{90} \cdot x_{107} \oplus y_{90} \oplus y_{108} \cdot x_{107} \oplus y_{108} \end{aligned}$$

Since $x_{107} \oplus y_{90} \cdot x_{107} \oplus y_{90} \oplus y_{108} \cdot x_{107} \oplus y_{108}$ is a Boolean Function of weight 6, assuming independence of the component variables we have $\mathsf{Pr}[x_{124} \oplus x^{\phi}_{124} = 0] = 1 - \frac{6}{8} = \frac{1}{4}$. This implies that $\mathsf{Pr}[x_{148} \oplus x^{\phi}_{148} = 0] = \frac{3}{4}$.

### Calculating $\mathsf{Pr}[y_{151} \oplus y^{\phi}_{151} = 0]$

To find this distribution we need to look at the state of our modified $\Delta_{61}$-$\mathsf{Grain}_{\mathsf{KSA}}$ at $t = 151-80 = 71$. At this we have $\Upsilon_{71} = [u_{74} = 0, u_{96} = 0, u_{117} = 0, u_{135} = 2, v_{134} = 2]$, $\mathcal{F}_{71} = \mathbf{0}, \chi_{71} = \mathbf{0}$. So we have

$$\begin{aligned} y_{151} \oplus y^{\phi}_{151} &= [f(Y_{71}) \oplus z_{71}] \oplus [f(Y^{\phi}_{71}) \oplus z^{\phi}_{71}] \\ &= h(y_{74}, y_{96}, y_{117}, y_{135}, x_{134}) \oplus h(y_{74}, y_{96}, y_{117}, y^{\phi}_{135}, x^{\phi}_{134}) \end{aligned}$$

Define the set of functions $h_{ij} = h(y_{74}, y_{96}, y_{117}, y_{135}, x_{134}) \oplus h(y_{74}, y_{96}, y_{117}, y_{135} \oplus i, x_{134} \oplus j)$, for $i, j \in \{0, 1\}$. Now assuming independence or the random variables involved, we can write,

$$\mathsf{Pr}[y_{151} \oplus y^{\phi}_{151} = 0] = \sum_{i,j} \mathsf{Pr}[h_{ij} = 0] \cdot \mathsf{Pr}[y_{135} \oplus y^{\phi}_{135} = i] \cdot \mathsf{Pr}[x_{134} \oplus x^{\phi}_{134} = j] \tag{10}$$

To compute this probability, we would need to calculate the individual probabilities $\mathsf{Pr}[y_{135} \oplus y^{\phi}_{135} = 0]$ and $\mathsf{Pr}[x_{134} \oplus x^{\phi}_{134} = 0]$. At $t = 135 - 80 = 55$, we have $\mathcal{F}_{55} = [u_{55} = 0, u_{68} = 0, u_{78} = 0, u_{93} = 0, u_{106} = 1, u_{117} = 0]$, $\chi_{55} = \mathbf{0}$, $\Upsilon_{55} = [u_{58} = 0, u_{80} = 0, u_{101} = 0, u_{119} = 1, v_{118} = 0]$. So we have

$$\begin{aligned} y_{135} \oplus y^{\phi}_{135} &= [f(Y_{55}) \oplus z_{55}] \oplus [f(Y^{\phi}_{55}) \oplus z^{\phi}_{55}] \\ &= 1 \oplus h(y_{58}, y_{80}, y_{101}, y_{119}, x_{118}) \oplus h(y_{58}, y_{80}, y_{101}, 1 \oplus y_{119}, x_{118}) \\ &= 1 \oplus y_{58}y_{101} \oplus y_{58} \oplus y_{101}x_{118} \oplus y_{101} \oplus x_{118} \end{aligned}$$

This represents a Boolean Function of weight 2, and hence we have $\mathsf{Pr}[y_{135} \oplus y^{\phi}_{135} = 0] = 1 - \frac{2}{8} = \frac{3}{4}$. Now at $t = 134 - 80 = 54$, we have $\chi_{54} = \mathbf{0}$, $\mathcal{G}_{lin,54} = \mathbf{0}$, $u_{54} = 0$, $\Upsilon_{54} = [u_{57} = 0, u_{79} = 0, u_{100} = 0, u_{118} = 1, v_{117} = 0]$ and all elements of $\mathcal{G}_{nlin,54}$ are zeros except $v_{106} = 1$. So we have

$$\begin{aligned} x_{134} \oplus x^{\phi}_{134} &= [g(X_{54}) \oplus y_{54} \oplus z_{54}] \oplus [g(X^{\phi}_{54}) \oplus y^{\phi}_{54} \oplus z^{\phi}_{54}] \\ &= [g(\ldots, x_{106}, \ldots) \oplus g(\ldots, 1 \oplus x_{106}, \ldots)] \oplus \\ &\quad [h(y_{57}, y_{79}, y_{100}, y_{118}, x_{117}) \oplus h(y_{57}, y_{79}, y_{100}, 1 \oplus y_{118}, x_{117})] \end{aligned}$$

We have to set $x_{99} = 1$, in the above equation since it is one of the conditions imposed at $t = 36$ to nullify a difference. Hence we have $\mathsf{Pr}[y_{135} \oplus y^{\phi}_{135} = 0] = \frac{35}{64}$. Now turning back to our original problem, we know that $\mathsf{Pr}[h_{00} = 0] = 1$, $\mathsf{Pr}[h_{01} = 0] = \frac{1}{2}$, $\mathsf{Pr}[h_{10} = 0] = \frac{1}{4}$, $\mathsf{Pr}[h_{11} = 0] = \frac{1}{2}$. Putting these values in Equation (10), we get $\mathsf{Pr}[y_{151} \oplus y^{\phi}_{151} = 0] \approx 0.6709$.

### Calculating $\mathsf{Pr}[y_{169} \oplus y^{\phi}_{169} = 0]$, $\mathsf{Pr}[x_{168} \oplus x^{\phi}_{168} = 0]$ and $\mathsf{Pr}[x_{161} \oplus x^{\phi}_{161} = 0]$

To compute these probabilities, we will need to look at outputs of $\Delta_{61}$-$\mathsf{Grain}_{\mathsf{KSA}}$ at $t = 81, 88, 89$. However at these rounds both $\chi_t$ and $\Upsilon_t$ have many elements equal to 2 and hence at this point we have to delve into several lower KSA rounds and compute the distributions of several intermediate variables and work our way up from there. Since this is slightly tedious, we omit extensive analysis of these two distributions and simply state the results.

$$\mathsf{Pr}[y_{169} \oplus y^{\phi}_{169} = 0] = 0.5015, \ \mathsf{Pr}[x_{168} \oplus x^{\phi}_{168} = 0] = 0.5, \ \mathsf{Pr}[x_{161} \oplus x^{\phi}_{161} = 0] = 0.495$$

### Calculating $\mathbf{Pr}[h(y_{108}, y_{130}, y_{151}, y_{169}, x_{168}) \oplus h(1 \oplus y_{108}, 1 \oplus y_{130}, y^{\phi}_{151}, y^{\phi}_{169}, x^{\phi}_{168}) = 0]$

For the sake of conciseness, let this expression be denoted by the symbol $H$, and define the Boolean Functions $\mathcal{H}_{ijk} = h(y_{108}, y_{130}, y_{151}, y_{169}, x_{168}) \oplus h(1 \oplus y_{108}, 1 \oplus y_{130}, i \oplus y_{151}, j \oplus y_{169}, k \oplus x_{168})$, for all $i, j, k \in \{0, 1\}$. Assuming independence, it is easy to see that

$$\mathsf{Pr}[H = 0] = \sum_{i,j,k} \mathsf{Pr}[\mathcal{H}_{ijk} = 0] \cdot \mathsf{Pr}[y_{151} \oplus y^{\phi}_{151} = i] \cdot \mathsf{Pr}[y_{169} \oplus y^{\phi}_{169} = j] \cdot \mathsf{Pr}[x_{168} \oplus x^{\phi}_{168} = k]$$

As it turns out, all the functions $\mathcal{H}_{ijk}$ are balanced except $\mathcal{H}_{011}$ and $\mathsf{Pr}[\mathcal{H}_{011} = 0] = \frac{3}{4}$. By plugging these values into the above equation we get $\mathsf{Pr}[H = 0] \approx 0.542$.

### Calculating $\mathbf{Pr}[z_{105} \oplus z^{\phi}_{105} = 0]$

From Equation (9), we can write

$$\begin{aligned}\mathsf{Pr}[z_{105} \oplus z^{\phi}_{105} = 0] &= 1 - \sum_{i \oplus j \oplus k = 0} \mathsf{Pr}[x_{148} \oplus x^{\phi}_{148} = i] \cdot \mathsf{Pr}[x_{161} \oplus x^{\phi}_{161} = j] \cdot \mathsf{Pr}[H = k] \\ &\approx 0.5002\end{aligned}$$

This concludes our proof for the distribution of $z_{105} \oplus z^{\phi}_{105} = 0$.