11-1-2019

Distinguisher and non-randomness of Grainv1 for 112, 114 and 116 initialisation rounds with multiple-bit difference in IVs

Deepak Kumar Dalai
National Institute of Science Education and Research

Subhamoy Maitra
Indian Statistical Institute, Kolkata

Santu Pal
National Institute of Science Education and Research

Dibyendu Roy
National Institute of Science Education and Research

IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

Distinguisher and non-randomness of Grain-v1 for 112, 114 and 116 initialisation rounds with multiple-bit difference in IVs

ISSN 1751-8709
Received on 4th June 2018
Revised 13th June 2019
Accepted on 24th July 2019
E-First on 10th September 2019
doi: 10.1049/iet-ifs.2018.5276
www.ietdl.org

Deepak Kumar Dalai[1] ✉, *Subhamoy Maitra*[2], *Santu Pal*[1], *Dibyendu Roy*[1]

[1]*School of Mathematical Sciences, National Institute of Science Education and Research, HBNI, Bhubaneswar, Odisha 752050, India*
[2]*Applied Statistics Unit, Indian Statistical Institute, Kolkata 700108, India*
✉ *E-mail: deepak@niser.ac.in*

Abstract: In this study, the authors construct two different distinguishers on Grain-v1 with 112 and 114 initialisation rounds. Their first distinguisher can distinguish Grain-v1 with 112 initialisation rounds from a uniform random source for 99% of the randomly chosen keys from full key space. The second one can distinguish Grain-v1 from a random source for 73% of the randomly chosen keys for one-fourth of the total key space (2^{78} keys out of 2^{80} keys). Our results improve upon the earlier distinguishers. The technique used for the distinguishers is conditional differential cryptanalysis. The existing works in this direction considered only one bit difference in the initialisation vector. However, for the first time, they could handle complicated conditions for the 2-bit difference to obtain better cryptanalytic results. Extending their technique by allowing the 1-bit difference in the pair of keys (i.e. related keys) and the 4-bit difference in IVs, they could observe the non-randomness till 116 initialisation rounds with a success in 62% cases.

1 Introduction

Stream ciphers play an important role in symmetric key cryptography as it is used for their speed, key size, and simplicity in hardware circuitry. In 2008, Grain-v1 [1] was placed in the list of seven stream ciphers for the final candidates by eSTREAM [2]. This is one of the three ciphers selected in hardware portfolio. Hence, Grain-v1 [1] has received a lot of attention among the cryptanalysts. This cipher is a bit-oriented non-linear feedback shift register (NFSR)-based stream cipher, which uses an 80-bit NFSR, an 80-bit linear feedback shift register (LFSR) and a non-linear filter function of five variables. Grain-v1 consists of two phases, the first one is the key scheduling phase (i.e. the key-IV initialisation phase) and the second one is the keystream generation phase (i.e. the pseudorandom bit generation phase). An 80-bit secret key (K) and a 64-bit initialisation vector (IV) are used to initialise the state of the cipher in the key scheduling phase. During this phase, the process runs for 160 rounds to update the state to a random looking (or pseudorandom) state. In the pseudorandom bit generation phase, the cipher generates keystream bit as output in each step and subsequently, the state gets updated.

Several works have been published to demonstrate weaknesses in Grain-v1 with a reduced number of initialisation rounds. Since the conditional differential attack [3] suits well for NFSR-based stream ciphers, the distinguishing attack using conditional differential cryptanalysis has received a lot of attention. In this technique, the analysts propose an algorithm to distinguish the first keystream bit of the stream cipher from a random bit by imposing some conditions on the input bits in the difference function of the output function (see Section 2). All the existing conditional difference cryptanalysis on Grain-v1 are based on the difference vector of weight one which is also known as dynamic cube attack of dimension one.

- The first result in terms of distinguisher was obtained by Aumasson *et al.* [4] in 2009. They observed bias in the first keystream bit of Grain-v1 with 81 initialisation rounds.
- In 2010, Knellwolf *et al.* [3] could distinguish the first keystream bits of Grain-v1 with 97 and 104 initialisation rounds.
- In 2014, Banik [5] provided a formal way for finding a distinguisher of Grain-v1 till 105 rounds.
- Recently, in 2016, Sarkar [6] designed a distinguisher for Grain-v1 with 106 rounds, which can distinguish Grain-v1 from a random source with the success rate ∼63%.
- In 2016, Ma *et al.* [7] improved the distinguisher till 107 initialisation rounds with significant probability and till 110 initialisation rounds where the probability figure was not properly provided.
- Again in 2016, Watanabe *et al.* [8] presented a non-randomness on Grain-v1 with 114 initialisation rounds on a small subset of the key space, i.e. in a weak key setting. This can distinguish Grain-v1 till 114 initialisation rounds from a random source if the secret key belongs to a particular set of 2^{40} keys. This attack does not hold much significance as exhaustive search is possible in a set of 2^{40} keys. This is a very minor fraction $\left(2^{40}/2^{80}\right) = 2^{-40}$ of the key space.
- In 2018, Ma *et al.* [9] proposed a distinguisher for Grain-v1 with 111 initialisation rounds. Their distinguisher can distinguish Grain-v1 with 111 initialisation rounds from a random source with a success rate of ∼83%.

In the literature, there are some theoretical cryptanalytic results on full (or reduced) initialisation round of Grain-v1, here we mention few of them. In 2008, Cannière *et al.* [10] analysed grain initialisation algorithm by its sliding property. This attack is completely based on the related key setup. In the same paper, they proposed two instances of differential attacks on Grain-v1. The first instance is under related key setup on Grain-v1 with full initialisation round with 2^{71} weak keys, 2^{57} weak IVs and 2^{55} chosen IV pairs. The second instance is on Grain-v1 with 112 initialisation rounds with 2^{63} weak IVs and 2^{72} chosen IV pairs. Although the size of the complete IV space is 2^{64}. In 2017, Mihaljević *et al.* [11] presented a conditional time-memory-data trade-off attack to recover the state bits of Grain-v1 with full initialisation round. The complexity of the online phase of the attack is quite lesser than the complexity of the exhaustive key search, but the complexity of the preprocessing phase is very high ($\geq 2^{84}$), which needs to be performed before the online phase of the attack. In Eurocrypt 2018, Zhang *et al.* [12] proposed a near collision attack on Grain-v1 with full initialisation round. The authors claimed that their attack has time complexity $2^{75.7}$ and data

Input : $K = (k_0, k_1, \cdots, k_{79})$,
$IV = (iv_0, iv_1, \cdots, iv_{63})$.
Output: State $S = (n_0, \cdots, n_{79}, l_0, \cdots, l_{79})$ of Grain-v1 after key scheduling process.

Assign $n_i = k_i$ for $i = 1, \cdots, 79$; $l_i = iv_i$ for $i = 0, \cdots, 63$; $l_i = 1$ for $i = 64, \cdots, 79$;
for *r rounds* **do**
  Compute $z = \sum_{k \in \mathcal{A}} n_k + h(l_3, l_{25}, l_{46}, l_{64}, n_{63})$, for $\mathcal{A} = \{1, 2, 4, 10, 31, 43, 56\}$;
  $t_1 = z + l_{62} + l_{51} + l_{38} + l_{23} + l_{13} + l_0$;
  $t_2 = z + n_{80}$ where n_{80} is computed as Equation (2) putting $t = 0$;
  $n_i = n_{i+1}$ and $l_i = l_{i+1}$ for $i = 0, 1, \cdots, 78$;
  $l_{79} = t_1$ and $n_{79} = t_2$;
end
return $S = (n_0, n_1, \cdots, n_{79}, l_0, l_1, \cdots, l_{79})$;

Fig. 1 *Algorithm 1: KSA of Grain-v1*

complexity 2^{19}. Later in Crypto 2018, Todo *et al.* [13] proposed a fast correlation attack on Grain-v1 with full initialisation round. Their proposed attack has time complexity $2^{76.7}$ and data complexity $2^{75.1}$. The complexities of these two attacks are significantly lesser than the complexity of the exhaustive key search. Although the complexity of the near collision attack of [12] is debatable, as in [13] the authors reported that the actual time complexity of near collision attack of [12] will be $2^{86.1}$, which is even larger than the time complexity of the exhaustive key search. All these attacks have large time, data complexities, hence these attacks cannot be used to attack Grain-v1 in practical time.

Several other cryptanalytic results on Grain-v1 and other variants of Grain family are available in the literature [14–20]. More results based on related key setup are available in [21, 22]. The related keys of these works differ at multiple positions.

Our contributions: In [3, 5–7, 9], the authors considered a single bit difference in IV. This is because handling the situations for a higher number of bit differences assumed to be quite complicated in terms of handling several conditions. For the first time, we present certain distinguishers for the higher rounds of Grain-v1 with two bits difference in the IV.

- The first distinguisher can distinguish Grain-v1 with 112 initialisation rounds from a random source almost certainly (with success rate ∼99%).
- The second distinguisher can distinguish Grain-v1 with 114 initialisation rounds from a random source with success rate ∼73% for 2^{78} weak keys (i.e. one-fourth of the key space).
- Furthermore, this distinguisher has been extended to 116 initialisation rounds with 1 bit difference in the key and 4 bit difference in the IV. Here we obtain a successful result in 62% cases for 2^{75} related keys.

The second and third distinguishers are designed by extending the idea of the first technique going in the backward direction from the initial state. As a result, the last two distinguishers fall in weak and related key setup, respectively.

The organisation of the paper: The paper is organised as follows. The design specification of Grain-v1 and notations are presented in Sections 1.1 and 1.2. The broad framework of the conditional differential cryptanalysis on NFSR-based stream ciphers is presented in Section 2. Necessary statistical studies to relate a random Boolean function (coming out of complex evolution of a finite-state machine) and the normal distribution are provided in Section 2.1. To compare our work with the recent attacks, we briefly discuss existing results in Section 2.2. The distinguisher for 112 initialisation rounds of Grain-v1 is described in Section 3. In Section 4, the second distinguisher for 114 initialisation rounds is provided. The non-randomness result until 116 initialisation rounds is presented in Section 4.1. Finally, we conclude this paper with future scopes in Section 5.

1.1 Design specification of Grain-v1

Grain-v1 [1] is based on an 80-bit NFSR, an 80-bit LFSR, and a non-linear filter function on five variables. The state bits of the NFSR are denoted by n_i and the state bits of the LFSR are denoted by l_i, where $0 \le i \le 79$. In each round, the state bits of NFSR and LFSR are shifted by one position towards left. The feedback bit of LFSR and NFSR are computed using the following feedback functions:

$$l_{t+80} = l_{t+62} + l_{t+51} + l_{t+38} + l_{t+23} + l_{t+13} + l_t, \text{ for } t \ge 0. \tag{1}$$

$$\begin{aligned} n_{t+80} = {} & l_t + n_{t+62} + n_{t+60} + n_{t+52} + n_{t+45} + n_{t+37} \\ & + n_{t+33} + n_{t+28} + n_{t+21} + n_{t+14} + n_{t+9} + n_t \\ & + n_{t+63}n_{t+60} + n_{t+37}n_{t+33} + n_{t+15}n_{t+9} \\ & + n_{t+60}n_{t+52}n_{t+45} + n_{t+33}n_{t+28}n_{t+21} \\ & + n_{t+63}n_{t+45}n_{t+28}n_{t+9} \\ & + n_{t+60}n_{t+52}n_{t+37}n_{t+33} \\ & + n_{t+63}n_{t+60}n_{t+21}n_{t+15} \\ & + n_{t+63}n_{t+60}n_{t+52}n_{t+45}n_{t+37} \\ & + n_{t+33}n_{t+28}n_{t+21}n_{t+15}n_{t+9} \\ & + n_{t+52}n_{t+45}n_{t+37}n_{t+33}n_{t+28}n_{t+21}, \text{ for } t \ge 0. \end{aligned} \tag{2}$$

The algebraic normal form of the non-linear filter generator (a Boolean function) h is

$$\begin{aligned} h(x_0, x_1, x_2, x_3, x_4) = {} & x_1 + x_4 + x_0x_3 + x_2x_3 + x_3x_4 \\ & + x_0x_1x_2 + x_0x_2x_3 + x_0x_2x_4 + x_1x_2x_4 + x_2x_3x_4 . \end{aligned} \tag{3}$$

The variables x_0, x_1, x_2, x_3, and x_4 correspond to the state bits l_{t+3}, l_{t+25}, l_{t+46}, l_{t+64}, and n_{t+63}, respectively, at the tth clock. In each round, the cipher computes one keystream bit z_t using some state bits from the NFSR and output of the non-linear filter function. The algebraic expression of the keystream bit at the tth round is

$$z_t = \sum_{k \in A} n_{t+k} + h(l_{t+3}, l_{t+25}, l_{t+46}, l_{t+64}, n_{t+63}), \text{ for } t \ge 0, \tag{4}$$

where $A = \{1, 2, 4, 10, 31, 43, 56\}$.

Grain-v1 passes through two steps as key scheduling phase and pseudorandom bit generation phase. The algorithms for these phases are known as the key scheduling algorithm (KSA) and pseudorandom (bit) generation algorithm (PRGA), respectively. The KSA initialises the cipher by using one secret key (K) of 80 bits and one IV of 64 bits. The secret key bits and IV bits are denoted by k_i, $0 \le i \le 79$ and iv_i, $0 \le i \le 63$, respectively. The cipher loads the 80-bit secret key into the NFSR and 64-bit IV into the LFSR as below. One may note that the rest of the 16 bits are padded as all one pattern

$$n_i = k_i, 0 \le i \le 79$$

$$l_i = iv_i, 0 \le i \le 63$$

$$l_i = 1, 64 \le i \le 79$$

Then the cipher runs the KSA for 160 rounds, without generating any keystream bit as output bit. Instead, these keystream bits are added with the feedback bit of the NFSR and LFSR. The KSA algorithm of Grain-v1 for r rounds is described in Algorithm 1 (Fig. 1). The value of r for the full round of Grain-v1 is 160.

After the completion of the key scheduling phase, the cipher starts the pseudorandom bit generation phase, where the cipher produces keystream bits as output. These keystream bits are used for encryption/decryption of plaintext/ciphertext. The graphical view of these two phases is provided in Fig. 2.

1.2 Notations

In this subsection, we present a few notations related to Grain-v1, which will be used later.

- $\mathbb{F}_2^m$: The m-dimensional vector space on binary field $\mathbb{F}_2$.
- $+$: The addition operation in $\mathbb{F}_2$ or in usual number system used as per the requirement.
- K, IV: The secret key and public IV, respectively.
- $S_t, t \geq 0$: The state of the cipher at the tth round; S_0 being the initial state of the cipher.
- m, n, l: The length of the state, secret key, and IV respectively, where $m > n + l$.
- z: The output of the first bit of Grain-v1 after (reduced round) KSA.
- Δz: Difference (XOR) in the pair of first output bits after a certain number of KSA rounds for two different instantiations.
- z_t: The keystream bit after t KSA rounds as defined in (4).
- Δz_t: Difference (XOR) of the pair of keystream bits from two different instantiations after t KSA rounds.

2 Conditional differential attack on NFSR-based stream cipher

The technique of the conditional differential cryptanalysis works by placing certain conditions on the IV as well as the key (K) bits after introducing a differential in the IV. In distinguishing attack, a distinguisher is designed to distinguish the first keystream bit of the cipher from a uniform random source. For Grain family, distinguishing attack by using conditional differential attack was initiated by Knellwolf *et al.* [3].

The general framework of NFSR-based stream ciphers contains a state of m bits such that $m > n + l$, where n, l are the lengths of the key and IV, respectively. Let us denote the initial state as $S_0 = (s_0, s_1, \ldots, s_{m-1}) \in \mathbb{F}_2^m$. In every round, the state $S_i = (s_i, s_{i+1}, \ldots, s_{i+m-1}) \in \mathbb{F}_2^m, i \geq 0$, is updated by a non-linear feedback shift register following a recursive formula $S_{i+1} = (s_{i+1}, s_{i+2}, \ldots, s_{i+m-1}, s_{i+m})$, where $s_{i+m} = g(S_i)$ a non-linear function on the state bits. After performing the non-linear evaluation for a certain number of rounds, the cipher generates its first keystream bit z. Therefore, the first output bit z of the cipher can be represented as the output of a keyed Boolean function $f: \mathbb{F}_2^n \times \mathbb{F}_2^l \mapsto \mathbb{F}_2$, where the first n bits correspond to the secret key K and the following l bits relate to the IV, i.e. $z = f(K, \mathrm{IV})$.

For a fixed secret key K, we define the Boolean function $f_K: \mathbb{F}_2^l \mapsto \mathbb{F}_2$ as $f_K(x) = f(K, x)$. Furthermore, for a difference vector $\boldsymbol{a} \in \mathbb{F}_2^l$ on the public parameter IV, we define the difference function $\Delta_a f_K(x) = f_K(x) + f_K(x + a)$. If one runs two instances of the cipher with the same key K and the IVs with difference $\boldsymbol{a} \in \mathbb{F}_2^l$, then the non-linear differences get added in the feedback bits in every round, i.e. in every round, the difference starts affecting the state bits non-linearly by adding the non-linear differences in the feedback bits. It might be possible for the attacker to control the spread of differences by putting some conditions on the state bits involved in the non-linear function in a particular round. Then going back recursively, the conditions can be represented on the initial state bits. As per the involvement of the type of initial state bit, the conditions are classified as follows.

- **`Type I`**: Conditions involving only the bits of IV.
- **`Type II`**: Conditions involving the bits of both K and IV (but may be exploited without any information on the key bits).
- **`Type III`**: Conditions involving only the bits of K.

The **`Type I`** conditions put a restriction on the choice of the IVs, which can easily be achieved by the attacker. On the other hand, the attacker cannot do anything in the case of **`Type III`** conditions as the involved bits remain secret for the attacker. However, fixing certain secret key bits, the cryptanalyst can point out a subset of weak keys for which the attack can be implemented. In the case of **`Type II`** conditions, since some secret key bits are expressed as some bits of the IV bits, these conditions might be exploited without the knowledge of the secret key bits and consequently, that may help to expose some secret bits.

If the function f is truly random, then the Boolean function $\Delta_a f_K$ should behave like a random function in every sub-domain (i.e. a subset) of the domain $\mathbb{F}_2^l$. To control the spreading of the difference by imposing the conditions on IV and K, the domain of the IV and secret key K is shrunk. Therefore, the spreading of difference is controlled in this sub-domain and some bias is expected in the output of the difference function $\Delta_a f_K$ in this sub-domain. Since the statistical test needs to be performed to find the bias in $\Delta_a f_K$, the number of imposed conditions needs to be optimised such that the number of sample inputs should support the theoretical bound for the statistical test. Therefore, the cryptanalyst attempts to optimise the following parameters while presenting an improvement on such attack:

i. Maximisation of the number of initialisation rounds.
ii. Maximisation of the success rate.
iii. Maximisation of the space of IVs (i.e. minimisation of **`Type I`** and **`Type II`** conditions).
iv. Maximisation of the effective key space (i.e. minimisation of the **`Type III`** conditions).
v. Minimisation of the number of queries to the oracle (stream cipher) (i.e. minimisation of the **`Type II`** conditions); for α, many IV-bits related to the **`Type II`** conditions, we need to query with 2^α many IVs, and one may like to minimise this.

2.1 Randomness test of the difference function $\Delta_a f_K$

In this section, we use a statistical method of testing the randomness of the difference function $\Delta_a f_K$. This concept was also followed in [3, 6]. If the Boolean function $\Delta_a f_K: \mathbb{F}_2^l \mapsto \mathbb{F}_2$ is a random Boolean function then the output of $\Delta_a f_K$ is randomly

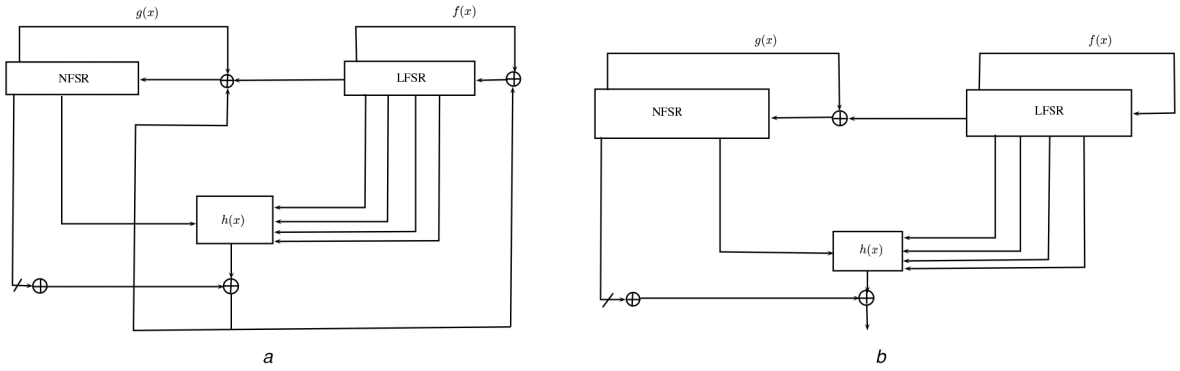

Fig. 2 *Design specification of Grain-v1*
(a) KSA of Grain-v1, ***(b)*** PRGA of Grain-v1

Input : Round r, Difference vector e_a, Type I (t_1 many), Type II (t_2 many) conditions.
Output: Grain-v1 or Random source.

An adversary $\mathcal{A}$ is given access to an oracle which outputs the keystream bit (z_r) of Grain-v1 or a random source;
for *each possibilities of free IV bits* **do**
  Consider all possible 2^{t_2} many IVs corresponding to each 0/1 values for IV bits, which are involved in Type II conditions;
  For each of the above IVs consider a bucket $\mathcal{B}_i$, $i = 0, \ldots, 2^{t_2} - 1$;
  Construct IV and $\widetilde{IV} = IV + e_a$, satisfy Type I conditions;
  Oracle outputs z_r and $\widetilde{z}_r$ for each 2^{t_2} many IV and $\widetilde{IV}$ respectively;
  Put $\Delta z_r = z_r + \widetilde{z}_r$ into their respective buckets $\mathcal{B}_i$.;
end
$\mathcal{A}$ computes the probability $p_i = Pr[\Delta z_r = 1]$, for each bucket $\mathcal{B}_i$;
$\mathcal{A}$ computes $\mathcal{W} = \sum_{0 \le i \le 2^{t_2}-1, p_i > \frac{1}{2}} \left(p_i - \frac{1}{2}\right)$;
if $\mathcal{W} > 2^{t_2} \cdot \mathcal{E}$ **then**
  return *Grain-v1*;
end
else
  return *Random source* ;
end

Fig. 3 *Algorithm 2: distinguisher for the first keystream bit of Grain-v1 with r KSA rounds*

distributed in every non-empty subset S of $\mathbb{F}_2^l$. Hence, it follows from the central limit theorem that for sufficiently large input $x_i \in S = \{x_1, x_2, \ldots, x_N\}$, the probability density function of the random variable

$$X = \sum_{x_i \in S} \Delta_a f_K(x_i),$$

approximately follows the normal distribution $\mathcal{N}(\mu, \sigma)$, i.e.

$$\phi(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2},$$

where μ and σ are the mean and standard deviation of the distribution of X. Let $b \in \mathbb{F}_2$ be a fixed value. Then, for a random Boolean function $\Delta_a f_K$ with given $\Pr[\Delta_a f_K(x) = b] \ge \frac{1}{2}$, the expectation of $(\Pr[\Delta_a f_K(x) = b] - \frac{1}{2})$ is

$$\frac{1}{\sqrt{2\pi}\sigma} \int_{\mu}^{N} e^{-(x-\mu)^2/2\sigma^2} \left(\frac{x}{N} - \frac{1}{2}\right) dx.$$

In our case, the mean $\mu = (N/2)$ and standard deviation $\sigma = (\sqrt{N}/2)$. Hence, solving the integration, we have the required expectation

$$\mathscr{E} = \frac{1}{\sqrt{8\pi N}}. \tag{5}$$

If this experiment is done for m number of times, then the sum

$$M = \sum_{\Pr[\Delta_a f_K(x) = b] \ge \frac{1}{2}} \left(\Pr[\Delta_a f_K(x) = b] - \frac{1}{2}\right)$$

is expected to be around $m\mathscr{E}$. For a specific stream cipher, if the value of the above sum (M) is greater than $m\mathscr{E}$ for a large percentage (i.e. significantly more than 50%) of keys K or lesser than $m\mathscr{E}$ for a large percentage (i.e. significantly more than 50%) of keys K then it allows to distinguish the stream cipher from a uniform random source. In case of our first distinguisher on 112 rounds, the sum is greater than $m\mathscr{E}$ for 99% of random keys (see towards the end of Section 3). For the other two distinguishers on 114 and 116 rounds, the sum is lesser than $m\mathscr{E}$ for 73% and 62% of random keys, respectively (see towards the end of Section 4 and Section4.1, respectively). We performed a similar experiment for a random source for a large number of samples. In our experiment, we considered higher rounds of Grain-v1 to verify our distinguishers. It is found that the sum M is greater (or lesser) than the expected value $m\mathscr{E}$ happens for close to 50% of samples for higher rounds (see towards the end of Section 3, Section 4 and Section 4.1). Therefore, from this experimental fact, we can distinguish Grain-v1 with the above-mentioned reduced rounds from a random source. Now we present the distinguishing technique on Grain-v1.

In the case of Grain-v1, the length of the key is $n = 80$ and the length of IV is $l = 64$. Consider there are t_1, t_2, and t_3 with many **`Type I`**, **`Type II`**, and **`Type III`** conditions, respectively. As per the involvement of state bits in **`Type I`** and **`Type III`** conditions, t_1 and t_3 with many bits in the IV and the key K need to be fixed, respectively. In the case of the **`Type II`** conditions, t_2 with many IV bits are dependent on some key bits and IV bits. Each assignment of t_2 with many IV bits provides a group where we need to check the bias. Let denote the assignments for t_2 with many IV bits as $\mathscr{A}_i, 0 \le i \le 2^{t_2} - 1$ with some order. One of the 2^{t_2} many assignments for t_2 with many IV bits must satisfy the **`Type II`** conditions. As the key bits are secret, the correct assignment is not known to the observer.

For each assignment $\mathscr{A}_i$, Grain-v1 has the domain of IV of size $2^{64-t_1-t_2}$. For all the assignments corresponding to **`Type II`** conditions, it is expected that the keystream bit z will be produced with some bias, i.e. for certain $b \in \mathbb{F}_2$, we expect a deviation on the $\left(\Pr[\Delta z = b] - \frac{1}{2}\right)$. Hence, for a random key and $b \in \mathbb{F}_2$, we calculate the probability $p_i = \Pr[\Delta z = b]$ for each assignment $\mathscr{A}_i, 0 \le i \le 2^{t_2} - 1$. As each $p_i \left(0 \le i \le 2^{t_2} - 1\right)$ might be very close to 0.5, we calculate $\mathscr{W} = \sum_{0 \le i \le 2^{t_2}-1, p_i \ge \frac{1}{2}} (p_i - 1/2)$.

If it is found that $\mathscr{W}$ is either greater than $2^{t_2}\mathscr{E}$ for a large percentage (i.e. significantly more than 50%) out of the randomly chosen keys or lesser than $2^{t_2}\mathscr{E}$ for a large percentage (i.e. significantly more than 50%) out of the randomly chosen keys then we claim that the cipher output is not pseudorandom. In that event, it is possible to distinguish Grain-v1 from a random source. This technique has earlier been exploited in [6]. We too follow this method to design the distinguishers on Grain-v1 in Sections 3 and 4. Algorithm 2 describes the distinguisher for Grain-v1 with r number of KSA rounds. The success rate of the distinguisher (described in Algorithm 2) will vary with the input of the algorithm (Fig. 3).

2.2 Existing conditional differential attacks on Grain-v1

In this subsection, we briefly present the existing works on the conditional differential attack on Grain-v1. All these works are based on a single bit difference (i.e. $wt(a) = 1$) on the IV which is also known as one-dimensional cube attack. In the case of Grain-v1, the key length $n = 80$ and the IV length $l = 64$. Let us denote $\boldsymbol{e}_i \in \mathbb{F}_2^{64}, 0 \le i \le 63$ is the unit binary vector where the ith position from the left in $\boldsymbol{e}_i$ is 1 and other positions are 0.

In 2010, Knellwolf *et al.* [3] proposed the conditional differential attack with one bit difference in IV of Grain-v1 with 97 initialisation rounds. The difference vector $\boldsymbol{a}$ was chosen as $\boldsymbol{e}_{37}$, i.e. they selected two IVs as $\mathrm{IV} = (iv_0, \ldots, iv_{37}, \ldots, iv_{63})$ and $\widetilde{\mathrm{IV}} = (iv_0, \ldots, 1 + iv_{37}, \ldots, iv_{63})$ for the difference. Let z_i and $\widetilde{z}_i$ be the ith keystream bits with IVs and $\widetilde{\mathrm{IV}}$, respectively, and $\Delta z_i = z_i + \widetilde{z}_i$ be the difference between them. To control the initial spread of the differences in the state, they imposed certain conditions on the bits of IV to make $\Delta z_{12} = 0$, $\Delta z_{34} = 0$, $\Delta z_{40} = 0$, and $\Delta z_{46} = 0$. With such

conditions, non-randomness has been observed at the 97th round. In the same paper, they extended the result to 104 initialisation rounds with single bit difference in IV.

In 2014, Banik [5] chose the difference vector $\boldsymbol{a} = \boldsymbol{e}_{61}$ and improved the result till 105 initialisation rounds. The author imposed some conditions on the bits of IV to make $\Delta z_{15} = \Delta z_{36} = \Delta z_{39} = \Delta z_{42} = 0$. Having the IV's with the imposed conditions, non-randomness in the first keystream bit of Grain-v1 with 105 initialisation rounds could be observed.

In 2015, Sarkar [6] improved the number of rounds to 106 by taking the difference vector $\boldsymbol{a} = \boldsymbol{e}_{62}$. This could be achieved by finding the conditions on the IV bits by making $\Delta z_{16} = \Delta z_{34} = \Delta z_{37} = \Delta z_{40} = 0$. The distinguisher could distinguish Grain-v1 with 106 initialisation rounds from a random source with a success rate of ∼63%.

In 2016, Ma *et al.* [7] proposed a conditional differential attack on Grain-v1 with 107 initialisation rounds. For 107 rounds, they chose three different difference vectors $\boldsymbol{e}_{34}$, $\boldsymbol{e}_{60}$, and $\boldsymbol{e}_{63}$. For the difference vector, $\boldsymbol{a} = \boldsymbol{e}_{63}$, they imposed conditions on the IV bits to make $\Delta z_{17} = \Delta z_{35} = \Delta z_{38} = \Delta z_{41} = \Delta z_{46} = 0$. With these conditions, they observed bias at the first keystream bit of Grain-v1 with 107 initialisation rounds. Similarly, for $\boldsymbol{e}_{34}$, $\boldsymbol{e}_{60}$ they imposed several conditions and observed the presence of bias in the first keystream bit of Grain-v1 with 107 initialisation rounds. In the same paper, they extended the conditional differential attack on Grain-v1 with 110 initialisation rounds by choosing the difference vector $\boldsymbol{a} = \boldsymbol{e}_{37}$. They imposed conditions on the IV bits to make $\Delta z_{12} = \Delta z_{34} = \Delta z_{40} = \Delta z_{46} = \Delta z_{48} = 0$. With these conditions, the authors have observed bias in the first keystream bit of Grain-v1 with 110 initialisation rounds. As these are experimental biases, the exact number of samples needs to be described. While this is clear in the case for 107 rounds [7, Table 4], the number of the secret key used in the case of 110 rounds [7, Table 5] is not provided.

In the same year, Watanabe *et al.* [8] proposed a conditional differential attack on Grain-v1 with 114 initialisation rounds. In this work, the authors imposed some conditions on IV bits as well as on secret key bits. Since conditions are applied on 40 secret key bits, the attack is restricted to a subset of key space of size 2^{40} whereas the size of key space is 2^{80}. If the unknown secret key K is from the set of $(2^{80} - 2^{40})$ many keys then their adversary will not be able to distinguish Grain-v1 from a random source. Furthermore, the domain of weak key space (i.e. of size 2^{40}) is immediately prone to exhaustive key search attack and thus this result does not look significant.

In 2018, Ma *et al.* [9] used the difference vector $\boldsymbol{e}_{37}$ to design a distinguisher on Grain-v1 with 111 initialisation rounds. The success rate of their distinguisher is ∼83%. They have imposed some conditions on the state bits to prevent the first five difference propagations. From these conditions on state bits, they obtained a total of 14 **`Type II`** and 15 **`Type II`** conditions. These **`Type I`** and **`Type II`** conditions provide a bias in the first keystream bit after 111 initialisation rounds.

Our distinguisher is practical and is experimentally verified. The comparison between the existing practical attacks and our present work is presented in Tables 1 and 2.

However, one may immediately note that all the works [3, 5–9] in this direction are based on the difference vector of weight 1, i.e. one bit difference in 1 of the improvement distinguishing success chance, the dimension of IV space, key space and query space. The dimension of IV space is equal to (64 – the number of **`Type I`** and **`Type II`** conditions) and the dimension of key space is equal to (80 – the number of **`Type III`** conditions). Furthermore, the dimension of the query space is proportional to (the number of **`Type II`** conditions + 1) as in each case we need to run the cipher with the same key and two different IVs. Some theoretical cryptanalytic results (see towards the end of Section 1) are available on Grain-v1 with full initialisation round which is not practical at the current time.

3 Distinguisher on Grain-v1 with 112 KSA round

The non-randomness in the first keystream bit of Grain-v1 with 97, 104, 105, 106, 110, and 111 initialisation rounds have been

Table 1 Comparison table (in the single key model)

Reference	R	#Key	#**`Type I`**, **`II`**, **`III`** conditions	#Queries	$\lvert\mathscr{K}\rvert$	Success rate
Knellwolf *et al.* [3]	97	1024	33, 5, 0	2^{31}	2^{80}	83%
	104	1024	25, 5, 0	2^{39}	2^{80}	58%
Banik [5]	105	1000	25, 6, 0	2^{39-n_1}[a]	2^{80}	92%
Sarkar [6]	106	1000	34, 6, 0	2^{30}	2^{80}	63%
Ma *et al.* [7]	104	1024	14, 15, 0	2^{40}	2^{80}	97%
	107	64	12, 12, 0	2^{42}[b]	2^{80}	99%
	110	NA	17, 15, 0	2^{47}	2^{80}	NA
Ma *et al.* [9]	111	64	14, 15, 0	2^{35}	2^{80}	83%
our work	112	2048	29, 7, 0	2^{35}	2^{80}	99%

R: number of KSA rounds.

#Key: number of random keys used in the experiment. The higher number of keys confirms the success probability better.

#Queries: number of queries used for each random key.

$\lvert\mathscr{K}\rvert$: size of the key space where distinguisher gets success.

[a] $n_1 \leq 5$ is the extra *Type I* conditions for faster implementation.

[b] Only for the difference $\boldsymbol{e}_{63}$.

Table 2 Comparison table (in the weak related key model)

Reference	R	#Key	#**`Type I`**, **`II`**, **`III`** conditions	#Queries	$\lvert\mathscr{K}\rvert$	Key model	Success rate
Watanabe *et al.* [8]	114	128	23, 1, 39	2^{32}	2^{40}	weak	NA
our work	114	2048	30, 7, 2	2^{34}	2^{78}	weak	73%
	116	4096	36, 7, 5	2^{28}	2^{75}	related[a]	62%

R: number of KSA rounds.

#Key: number of random keys used in the experiment. The higher number of keys confirms the success probability better.

#Queries: number of queries used for each random key.

$\lvert\mathscr{K}\rvert$: size of the key space where distinguisher gets success.

[a] Two related keys differ only in one place. Existing works are based on more number of key bit differences under different attack models.

observed in [3, 5–7, 9]. Furthermore, the same is done for 114 initialisation rounds in the weak key set up of key space size 2^{40} in [8]. These works exploited the transmission of bias from a single bit difference in the IV. At the current situation, improving result using a single bit difference seems exhaustive. Hence, working on single bit difference in similar direction seems difficult for higher rounds as it needs a powerful computer to study the equations generated at higher round.

Further working on multiple difference vectors, in general, it spreads more differences into the state as there are multiple numbers of non-zero positions in multiple bit difference vectors. Henceforth, it creates more situations when $\Delta z_t \neq 0$ and generates complicated equations at lower rounds. As a result, it becomes more difficult to analyse for a higher round. In contrast, there could be some difference vectors of multiple weight where the difference generated due to the different non-zero positions cancel each other to result $\Delta z_t = 0$. Hence, if such a difference vector is chosen then the attacker can make $\Delta z_t = 0$ for higher rounds and go for attacking for higher rounds.

In our work, we have improved the number of initialisation rounds by imposing a two bit difference in the IV, i.e. by using a difference vector of weight two. Let us denote vector $\boldsymbol{e}_{i,j} \in \mathbb{F}_2^{64}, 0 \leq i < j \leq 63$ such that $\boldsymbol{e}_{i,j} = \boldsymbol{e}_i + \boldsymbol{e}_j$. For our work, we choose the difference vector $\boldsymbol{e}_{20,45}$ of weight two from $(64/2) = 2016$ possibilities. The reason for choosing the specific difference vector $\boldsymbol{e}_{20,45}$ is described in Remark 1.

Remark 1: For each of 2016 possible difference vectors $\boldsymbol{e}_{i,j}$ of weight two, we experimentally checked the probability of the difference $\Delta z_t = z_t + \tilde{z}_t$ at the output of every round $t, 0 \leq t \leq 41$, for a large number of random key, IV pairs. Hence for each $\boldsymbol{e}_{i,j}$, there are disjoint partitions S_1, S_2, and S_3 of the set $\{0, 1, \ldots, 41\}$ such that $z_t = 0$ for $t \in S_1$, $z_t = 1$ for $t \in S_2$ and z_t is a non-constant function of K, IV for $t \in S_3$. For a chosen $\boldsymbol{e}_{i,j}$, we take action for these three different situations as the following.

- When $t \in S_1$: since $z_t = 0$, there is no addition of difference in the state from z_t. We need not do anything in this situation.
- When $t \in S_2$: since $z_t = 1$ (constant function), it is not possible to put any condition on state bits to stop the propagation of difference into the state. Hence, we prefer to choose such $\boldsymbol{e}_{i,j}$ where $|S_2| = 0$.
- When $t \in S_3$: since z_t is a non-constant function of IV and K, it is possible to put conditions on the bits of IV and/or K such that $z_t = 0$. As per the involvement of bits of IV and K, the conditions are classified as **`Type I`**, **`Type II`**, and **`Type III`**. Hence we need to choose $\boldsymbol{e}_{i,j}$ such that the size of set S_3 is minimised, which possibly give a minimised set of conditions.

From the initial rounds (i.e. $0 \leq t \leq 41$), we need to find the set $S_2 \cup S_3$ containing a few numbers of elements. For further refining, our aim is to choose $\boldsymbol{e}_{i,j}$ such that $|S_2| = 0$ and S_3 is a minimised set. We experimentally checked for each possible $\boldsymbol{e}_{i,j}$ for a large set of random K, IV pairs. The experimental result shows that the vectors $\boldsymbol{e}_{20,45}, \boldsymbol{e}_{23,61}, \boldsymbol{e}_{38,62}$ are having minimised set $S_2 \cup S_3$. Furthermore, our distinguisher gives the best success rate for the difference vector $\boldsymbol{e}_{20,45}$. Hence, we choose $\boldsymbol{e}_{20,45}$ as the difference vector for the distinguisher.

For the difference vector $\boldsymbol{e}_{20,45}$, the difference probabilities $\Pr[\Delta z_t \neq 0]$ are non-zero for $t = 17$, 20, 36, 37, and 38. Therefore, our aim is to find a set of conditions on IV and key bits such that the restriction $\Delta z_{17} = \Delta z_{20} = \Delta z_{36} = \Delta z_{37} = \Delta z_{38} = 0$ is satisfied. The reason for choosing the restrictions on Δz_{17}, Δz_{20}, Δz_{36}, Δz_{37}, and Δz_{38} is as follows.

Let two instances of the cipher be initialised with IV and $\widetilde{\text{IV}} = \text{IV} + \boldsymbol{e}_{20,45}$. The states at the tth round are S_t and $\tilde{S}_t$, respectively. Denote $\Delta S_t = S_t + \tilde{S}_t, t \geq 0$. The states S_0 and $\tilde{S}_0$ at the zeroth round differ exactly at two places with probability 1. As the number of rounds increases in the KSA, the number of difference positions increases with a complicated probability distribution. Our goal is to minimise the differences for maximum possible initialisation rounds by imposing specific conditions on the bit values in IV.

In the KSA, the keystream bit z_t involves the feedback bits from both the LFSR and the NFSR. The main reason for the transmission of difference into the state bits is the injection of the difference in z_t via these feedback bits. We denote the tth keystream bits of the cipher with initial state S_0 and $\tilde{S}_0$ by z_t and $\tilde{z}_t$, respectively. Since there are differences in the two bits of the initial states, the keystream bits (z_t and $\tilde{z}_t$) start differing after a certain number of rounds t. The difference of the keystream bits $\Delta z_t = z_t + \tilde{z}_t$ is a function of the bits of key K and IV. The algebraic expression of the function becomes more complicated as the number of rounds increases. We use SAGE [23] to compute the algebraic expressions of the function Δz_t for $0 \leq t \leq 41$. The conditions on IV bits are generated by imposing the condition $\Delta z_t = 0$ as follows:

`[C0.]` Case ($0 \leq t \leq 41$ and $t \neq 17, 20, 36, 37, 38$): It is observed that $\Delta z_t = 0$ for $0 \leq t \leq 16, 18 \leq t \leq 19, 21 \leq t \leq 35, 39 \leq t \leq 41$. Hence, we have nothing to impose for these rounds.

`[C1.]` Case ($t = 17$): In the 17th round, $\Delta z_{17} = P_1(K, \text{IV})$, where $P_1(K, \text{IV})$ is a polynomial involving the bits of K and IV. The algebraic normal from P_1 is provided in [24]. For a fixed key K, we need to find the set of IVs such that $P_1(K, \text{IV}) = 0$. Since finding this set is quite difficult, we choose a subset of IVs by imposing some conditions on the IV bits such that $P_1(K, \text{IV}) = \Delta z_{17} = 0$. We follow the method explained in Section 3.1 to make $\Delta z_{17} = 0$. We set $iv_{47} = iv_{63} = 0$ and $iv_1 = iv_4 + iv_{14} + iv_{24} + iv_{26} + iv_{39}$. With these conditions, the equation becomes $\Delta z_{17} = iv_{52} + F_1(K)$, where F_1 is a function involving only the secret key bits. Further, fixing $iv_{52} = F_1(K)$, we get $\Delta z_{17} = 0$. Therefore, having three **`Type I`** conditions $iv_{47} = 0$; $iv_{63} = 0$; $iv_1 + iv_4 + iv_{14} + iv_{24} + iv_{26} + iv_{39} = 0$ and one **`Type II`** condition $iv_{52} = F_1(K)$, we have a smaller set of IVs where $\Delta z_{17} = 0$.

`[C2.]` Case ($t = 20$): At this round, $\Delta z_{20} = P_2(K, \text{IV})$, where P_2 is a polynomial involving the bits of K and IV. The algebraic normal form of P_2 is provided in [24]. Similar to [C1], we set some conditions on the IV bits, so that $\Delta z_{20} = 0$. Setting $iv_{49} = 0$ and $iv_3 = iv_6 + iv_{23}$, we have the equation $\Delta z_{20} = iv_{28} + F_2(K)$, where F_2 is a function involving only the secret key bits. Furthermore, imposing an extra condition $iv_{28} = F_2(K)$, we have $\Delta z_{20} = 0$. Therefore, we set two **`Type I`** conditions $iv_{49} = 0$; $iv_3 + iv_6 + iv_{23} = 0$ and one **`Type II`** condition $iv_{28} = F_2(K)$.

`[C3.]` Case ($t = 36$): In this case, we have $\Delta z_{36} = P_3(K, \text{IV})$, where P_3 is a polynomial involving the bits of K and IV. As the algebraic expression of P_3 is very large, the algebraic normal form of P_3 is placed at [24]. The same technique (as in Section 3.1) has been followed to make $\Delta z_{36} = P_3 = 0$.

To set $\Delta z_{36} = 0$, we fix the conditions $iv_5 = iv_{14} = iv_{48} = 0; iv_2 = iv_{22} = iv_{44} = 1; iv_{15} = iv_{25}, iv_{40} = iv_{53}, iv_{16}$. $= iv_{19} + iv_{23} + iv_{24} + iv_{26} + iv_{41}$

After setting these conditions, we have $\Delta z_{36} = iv_{54} + iv_{27}iv_{54} + iv_{54}F_3(K) + iv_{27}f_1(K) + f_2(K)$ $= iv_{54}(1 + iv_{27} + F_3(K)) + iv_{27}f_1(K) + f_2(K)$.

Here, F_3, f_1 and f_2 are functions on key bits. Furthermore, setting $iv_{27} = F_3(K)$ and $iv_{54} = F_3(K)f_1(K) + f_2(K) = F_4(K)$, we get $\Delta z_{36} = 0$. Therefore, setting nine **`Type I`** conditions $iv_5 = iv_{14} = iv_{48} = iv_{15} + iv_{25} = iv_{40} + iv_{53} = 0$; $iv_{16} + iv_{19} + iv_{23} + iv_{24} + iv_{26} + iv_{41} = 0$; $iv_2 = iv_{22} = iv_{44} = 1$; and two **`Type II`** conditions $iv_{27} = F_3(K)$; $iv_{54} = F_4(K)$, we get $\Delta z_{36} = 0$.

`[C4.]` Case ($t = 37$): In the 37th round, we have $\Delta z_{37} = P_4(K, \text{IV})$, where P_4 is a polynomial involving the bits of K and IV. The algebraic normal form of P_4 is available in [24].

To make $\Delta z_{37} = 0$, we impose the conditions $iv_{24} = iv_{46} = iv_{50} = iv_{51} = iv_{62} = 0; iv_{19} = 1; iv_{34} = iv_{43} + iv_{53} + iv_{56}$; $iv_7 = iv_4 + iv_8 + iv_{18} + iv_{21} + iv_{29} + iv_{30} + iv_{59}$. After fixing these conditions, we have $\Delta z_{37} = iv_{53} + F_5(K)$. Now considering

$iv_{53} = F_5(K)$, we have $\Delta z_{37} = 0$. Therefore, at the 37th round, we set the following eight **Type I** and one **Type II** conditions.

$$\text{TypeI}: iv_{24} = iv_{46} = iv_{50} = iv_{51} = iv_{62} = 0;$$

$$iv_{19} = 1; iv_{34} + iv_{43} + iv_{53} + iv_{56} = 0;$$

$$iv_4 + iv_7 + iv_8 + iv_{18} + iv_{21} + iv_{29} + iv_{30} + iv_{59} = 0$$

$$\text{TypeII}: iv_{53} = F_5(K)\,.$$

[C5.] Case ($t = 38$): In this round, we have $\Delta z_{38} = P_5(K, \text{IV})$, where P_5 is a polynomial involving the bits of K and IV. The algebraic normal form of P_5 is available in [24].

To have $\Delta z_{38} = 0$, we impose $iv_{17} = iv_{39} = iv_{42} = iv_{55} = 0$, $iv_8 = iv_{18}, iv_{21} = iv_{30}, iv_{56} = iv_{41} + iv_{43}$. So, we have equation $\Delta z_{38} = iv_{59} + iv_{23}iv_{59} + iv_{59}F_6(K) + iv_{23}f_3(K) + f_4(K) = iv_{59}(1 + iv_{23}.$ $+ F_6(K)) + iv_{23}f_3(K) + f_4(K)$

Furthermore, imposing conditions $iv_{23} = F_6(K)$ and $iv_{59} = F_6(K)f_3(K) + f_4(K) = F_7(K)$, we have $\Delta z_{38} = 0$.

Finally, for the 38th round, we set the following seven **Type I** conditions and two **Type II** conditions to have $\Delta z_{38} = 0$

$$\text{TypeI}: iv_{17} = iv_{39} = iv_{42} = iv_{55} = iv_8 + iv_{18} =$$

$$iv_{21} + iv_{30} = iv_{41} + iv_{43} + iv_{56} = 0$$

$$\text{TypeII}: iv_{23} = F_6(K); iv_{59} = F_7(K)\,.$$

Therefore, for a fixed key K, setting the conditions proposed in **C1**, **C2**, **C3**, **C4**, and **C5** on the bits of IV, we will have $\Delta z_t = 0$, for $0 \le t \le 41$. We summarise the difference propagation and required **Type I** and **Type II** conditions in Table 3. It can be observed that unlike the results in [3, 5–7], in our case there is no t for which $\Delta z_t = 1$, where $0 \le t \le 41$. This provides an advantage to obtain an improved distinguisher by choosing the difference vector $\boldsymbol{e}_{20,45}$.

For the 42nd round, the algebraic expression of Δz_{42} is very large and complicated on the bits of K and IV. However, the Items [**C1**, **C2**, **C3**, **C4**, **C5**] contain 29 **Type I** conditions and 7 **Type II** conditions, which are listed below. Hence, with these conditions, we have $\Delta z_t = 0$ for $0 \le t \le 41$.

$$\left\{\begin{array}{c}
\texttt{TypeI:} \\
iv_5 = iv_{14} = iv_{17} = iv_{24} = iv_{39} = iv_{42} = iv_{46} = iv_{47} \\
= iv_{48} = iv_{49} = iv_{50} = iv_{51} = iv_{55} = iv_{62} = iv_{63} = 0; \\
iv_2 = iv_{19} = iv_{22} = iv_{44} = 1; iv_8 = iv_{18}; iv_{15} = iv_{25}; \\
iv_{21} = iv_{30}; iv_{40} = iv_{53}; iv_3 = iv_6 + iv_{23}; \\
iv_{41} = iv_{43} + iv_{56}; iv_{34} = iv_{43} + iv_{53} + iv_{56}; \\
iv_1 = iv_4 + iv_{14} + iv_{24} + iv_{26} + iv_{39}; \\
iv_{16} = iv_{19} + iv_{23} + iv_{24} + iv_{26} + iv_{41}; \\
iv_4 = iv_7 + iv_8 + iv_{18} + iv_{21} + iv_{29} + iv_{30} + iv_{59}; \\
\texttt{TypeII:} \\
iv_{52} = F_1(K); iv_{28} = F_2(K); iv_{27} = F_3(K); \\
iv_{54} = F_4(K); iv_{53} = F_5(K); iv_{23} = F_6(K); \\
iv_{59} = F_7(K)\,.
\end{array}\right\}$$

This set of conditions can further be simplified as

$$\left\{\begin{array}{c}
\texttt{TypeI:} \\
iv_5 = iv_{14} = iv_{17} = iv_{24} = iv_{39} = iv_{42} = iv_{46} \\
= iv_{47} = iv_{48} = iv_{49} = iv_{50} = iv_{51} = iv_{55} \\
= iv_{62} = iv_{63} = 0; \\
iv_2 = iv_{19} = iv_{22} = iv_{44} = 1; \\
iv_1 = iv_4 + iv_{26}; iv_3 = iv_6 + iv_{23}; \\
iv_4 = iv_7 + iv_{29} + iv_{59}; iv_8 = iv_{18}; \\
iv_{15} = iv_{25}; iv_{16} = 1 + iv_{23} + iv_{26} + iv_{41}; \\
iv_{21} = iv_{30}; iv_{34} = iv_{41} + iv_{53}; \\
iv_{40} = iv_{53}; iv_{41} = iv_{43} + iv_{56}; \\
\texttt{TypeII:} \\
iv_{52} = F_1(K); iv_{28} = F_2(K); iv_{27} = F_3(K); \\
iv_{54} = F_4(K); iv_{53} = F_5(K); iv_{23} = F_6(K); \\
iv_{59} = F_7(K)\,.
\end{array}\right\} \tag{6}$$

The **Type II** conditions are imposed on seven known IV bits iv_{23}, iv_{27}, iv_{28}, iv_{52}, iv_{53}, iv_{54}, and iv_{59} and a set of unknown key bits. For an unknown fixed key K and a chosen IV, if values of $F_i(K)$, $i = 1, 2, \ldots, 7$ match with the values of the above-mentioned IV bits, respectively, then all **Type II** conditions are satisfied and hence, $\Delta z_t = 0$; $0 \le t \le 41$. Consider an unknown random key K

Table 3 Differential status of Grain-v1 from round 0 to round 41

Round (i)	Δz_i	**Type-I** conditions	**Type-II** conditions	Round (i)	Δz_i	**Type-I** conditions	**Type-II** conditions
0–16	0	no conditions	no conditions	36	$P_3(K, \text{IV})$	$iv_5 = iv_{14} = iv_{48}$ $= iv_{15} + iv_{25}$ $= iv_{40} + iv_{53} = 0;$ $iv_{16} + iv_{19} + iv_{23} +$ $iv_{24} + iv_{26} + iv_{41} = 0;$ $iv_2 = iv_{22} = iv_{44} = 1;$	$iv_{27} = F_3(K);$ $iv_{54} = F_4(K);$
17	$P_1(K, \text{IV})$	$iv_{47} = iv_{63} = 0;$ $iv_1 + iv_4 + iv_{14} +$ $iv_{24} + iv_{26} + iv_{39} = 0$	$iv_{52} = F_1(K)$	37	$P_4(K, \text{IV})$	$iv_{24} = iv_{46} = iv_{50}$ $= iv_{51} = iv_{62} = 0;$ $iv_{19} = 1; iv_{34} + iv_{43}$ $+iv_{53} + iv_{56} = 0;$ $iv_4 + iv_7 + iv_8 + iv_{18}$ $+iv_{21} + iv_{29} + iv_{30}$ $+iv_{59} = 0$	$iv_{53} = F_5(K)$
18 and 19	0	no conditions	no conditions	38	$P_5(K, \text{IV})$	$iv_{17} = iv_{39} = iv_{42}$ $= iv_{55} = iv_8 + iv_{18}$ $= iv_{21} + iv_{30} = iv_{41}$ $+iv_{43} + iv_{56} = 0$	$iv_{23} = F_6(K);$ $iv_{59} = F_7(K)$
20	$P_2(K, \text{IV})$	$iv_{49} = 0;$	$iv_{28} = F_2(K)$	39–41	0	no conditions	no conditions
21–35	0	no conditions	no conditions				

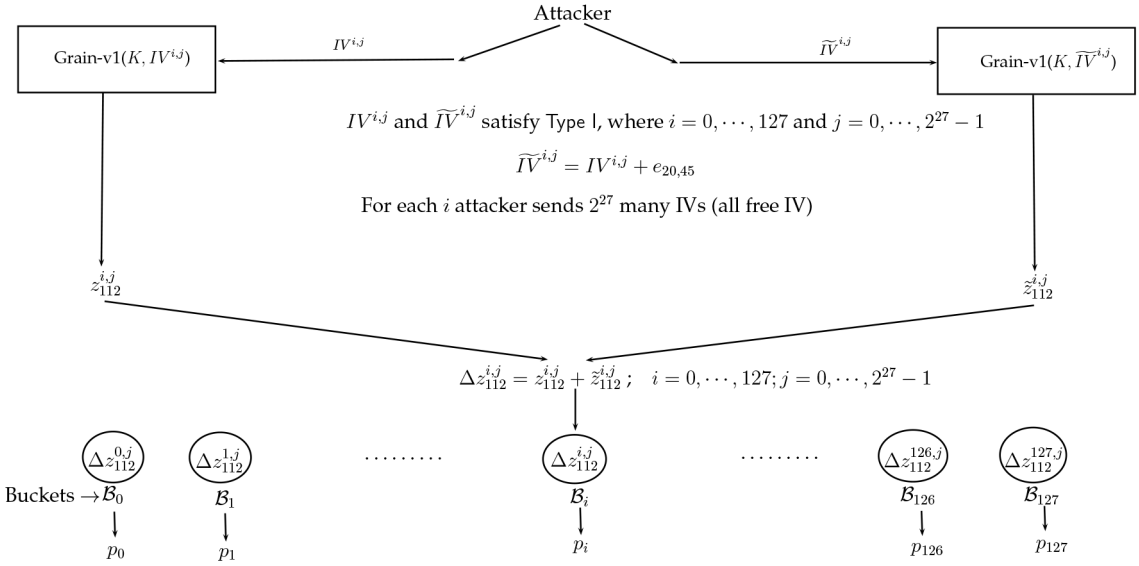

Fig. 4 *Distinguisher for Grain-v1 with 112 KSA rounds*

and IV satisfying all **`Type I`** conditions. Then, there is one possibility out of 2^7 possible assignments of seven bits iv_{23}, iv_{27}, iv_{28}, iv_{52}, iv_{53}, iv_{54}, and iv_{59} that satisfies the **`Type II`** conditions. Since the secret key K is unknown, we have to try for all 2^7 possible assignments of state bits iv_{23}, iv_{27}, iv_{28}, iv_{52}, iv_{53}, iv_{54}, and iv_{59} and there is a case for which $\Delta z_t = 0, 0 \le t \le 41$, i.e. there must be a case for which $\Pr[\Delta z_t = 0] = 1$ and for other 127 cases $\Pr[\Delta z_t = 0] \le 1$ for $0 \le t \le 41$. Note that there might be many assignments from these non-satisfying 127 assignments where $\Pr[\Delta z_t = 0] = 1$ and $0 \le t \le 41 - l$ for some small integers l.

For the high values of t, the probability $\Pr[\Delta z_t = 0]$ is expected to be (1/2). However, the existence of a case where $\Pr[\Delta z_t = 0] = 1$ for $0 \le t \le 41$ (or $0 \le t \le 41 - l$ for some integers l) motivates us to search for non-randomness at higher rounds and to construct a distinguisher. For a random key K, the probability $p_i = \Pr[\Delta z_r = 1]$ is calculated for each assignment A_i, $0 \le i \le 127$ of the IV bits iv_{23}, iv_{27}, iv_{28}, iv_{52}, iv_{53}, iv_{54}, and iv_{59} for a higher value of r. We have observed small non-randomness in some assignments for $r = 112$. As each assignment corresponds to a very small bias, we use the fact to present a distinguisher as discussed in Section 2.1.

Consider $\mathcal{W} = \sum_{0 \le i \le 127,\ p_i > \frac{1}{2}} (p_i - 1/2)$. If Grain-v1 with 112 initialisation rounds generates pseudorandom bits then the value of $\mathcal{W}$ would be expected as $\mathcal{W}_\phi = 2^7 \mathscr{E}$, where $\mathscr{E} = (1/\sqrt{8\pi N})$ and N is the size of the sample space. The number of IV bits fixed by the **`Type I`** and **`Type II`** conditions are $29 + 7 = 36$. As we take a pair $\left(\mathrm{IV}, \widetilde{\mathrm{IV}}\right)$ to create the differences, there are $64 - 36 - 1 = 27$ free IV bits that can be used to generate 2^{27} samples. Putting the sample size $N = 2^{27}$ in (5), we have $\mathcal{W}_\phi \simeq 0.0022$.

We performed experiments on 2048 random keys and it is observed that for ∼99% of keys $\mathcal{W} > 0.0022$. This took around 5 days in a machine having 120 processors of 2.8 GHz clock in a multi-user environment. With this experiment, we can distinguish Grain-v1 with 112 initialisation rounds from a random source with the success rate of ∼99%. For cross-checking, we too perform the same experiment for the round 113 and achieved the success rate of 45% which is close to 50%. To claim the success rate of 45%, we need to run the same experiment for a large number of random keys. This is not possible at this point with the present computational power we are having.

Hence, the proposed distinguisher, designed by selecting the IV differential set of dimension 2, can distinguish the stream cipher Grain-v1 with 112 initialisation rounds from a uniform random source with a significantly high success rate (∼99%).

The proposed distinguisher for Grain-v1 with 112 KSA rounds is presented as follows. The pictorial view of our distinguisher is provided in Fig. 4.

• *A distinguisher for the first keystream bit of Grain-v1 with 112 KSA round*:

i. A random key K of 80 bit is generated.
ii. An adversary $\mathscr{A}$ is given oracle access to the pseudorandom bit generator, which generates keystream bit by using K.
iii. $\mathscr{A}$ selects 64 bits IV, which satisfies **`Type I`** and **`Type II`** conditions in (6).
iv. $\mathscr{A}$ constructs another IV, $\widetilde{\mathrm{IV}} = \mathrm{IV} + \boldsymbol{e}_{20,45}$.
v. $\mathscr{A}$ considers all possible 0/1 values to seven IV bits, which satisfy **`Type II`** conditions.
vi. For every 0/1 possible values of seven IV bits (involved in **`Type II`**), $\mathscr{A}$ queries 2^{27} IV and $\widetilde{\mathrm{IV}}$ to the oracle.
vii For $2^7 \cdot 2^{27} = 2^{34}$ IV bits, the oracle returns z and $\tilde{z}$ corresponding to IV and $\widetilde{\mathrm{IV}}$.
vii For every 0/1 possible value of those seven IV bits (involved in
i. **`Type II`**) $\mathscr{A}$ segregates the keystream bits into 2^7 buckets ($\mathscr{B}_i, i = 0, \ldots, 127$). Here, each bucket contains 2^{27} z and $\tilde{z}$.
ix. $\mathscr{A}$ computes the probability $p_i = \Pr[z \ne \tilde{z}]$, for each bucket $\mathscr{B}_i$, $i = 0, \ldots, 127$.
x. $\mathscr{A}$ computes $\mathscr{W} = \sum_{0 \le i \le 127,\ p_i > \frac{1}{2}} \left(p_i - \frac{1}{2}\right)$.
xi. If $\mathscr{W} > 0.0022$, then $\mathscr{A}$ claims that the oracle is Grain-v1 with 112 KSA rounds to generate the keystream bits. Otherwise $\mathscr{A}$ claims that the oracle is generating the random bits.

It has been experimentally observed that the success rate of the adversary $\mathscr{A}$ is ∼99%.

3.1 Function reduction method

It can be observed from [24] that the algebraic normal form of functions $P_i(K, \mathrm{IV})$, $1 \le i \le 5$ is quite complicated. In Section 3, we have imposed some **`Type I`** and **`Type II`** conditions to get

$P_i(K, \mathrm{IV}) = 0$ for $1 \le i \le 5$. These **Type I** and **Type II** conditions are obtained by carefully analysing the functions. We follow the following steps to get these **Type I** and **Type II** conditions.

i. Firstly, we save the complete algebraic expression of the function into a file.
ii. We assign 0 or 1 values to some IV bits to simplify the function.
iii. Then we replace some IV bits in terms of the linear combination of some other IV bits to get a more simplified form of the function.
iv. Finally, some IV bits are substituted in terms of the secret key bits to get $P_i(K, \mathrm{IV}) = 0$.

Most of these things are done manually. Let us discuss the scenario for $P_2(K, \mathrm{IV})$, as the other functions can be tackled in the same technique. From the algebraic normal form of the function $P_2(K, \mathrm{IV})$ (can be found in [24]), one can observe that the bit iv_{49} is involved in many monomials in the algebraic normal form. Hence, the algebraic normal form is made simpler by substituting $iv_{49} = 0$. Furthermore, substituting $iv_3 + iv_6 + iv_{23} = 0$, the algebraic normal form of the function $P_2(K, \mathrm{IV})$ becomes as simple as $iv_{28} + F_2(K)$. Finally, the **Type II** condition $iv_{28} = F_2(K)$ helps us to achieve $P_2(K, \mathrm{IV}) = 0$. We have followed a similar method for the other complicated functions and naturally for the functions P_3, P_4, P_5 it took quite a bit of effort. Software to handle these issues may provide even better results that may be explored in the future.

4 Distinguisher on Grain-v1 with 114 KSA round

In this section, we design a distinguisher on Grain-v1 with 114 initialisation rounds. The idea is to increase the number of rounds followed in two steps.

- In the first step, we put conditions on $iv_{63} = iv_{62} = \cdots = iv_{63-j} = 1$ for some $j \ge 0$ and generate conditions as discussed in Section 3 to obtain a distinguisher at the rth round.
- Since we are able to run the inverse of KSA for $j + 1$ rounds as $iv_{63} = \cdots = iv_{63-j} = 1$, with some more conditions on key bits, i.e. **Type III** conditions, we can design a distinguisher for $(r + j + 1)$ KSA rounds with some more conditions on key space.

For our work, we first put three **Type I** conditions $iv_{63} = iv_{62} = iv_{61} = 1$ on last three IV bits. Then following a similar technique as in Section 3, we choose the same difference vector $\boldsymbol{a} = \boldsymbol{e}_{20,45}$ and generate the conditions as follows. We have followed the same technique, as in Section 3.1) to construct the following **Type I** and **Type II** conditions.

[C0.] Case ($0 \le t \le 41$ and $t \ne 17, 20, 36, 37, 38$): It is observed that $\Delta z_t = 0$ for $0 \le t \le 16, 18 \le t \le 19, 21 \le t \le 35, 39 \le t \le 41$. We need not require any additional condition for these rounds.

[C1.] Case ($t = 17$): In this round, $\Delta z_{17} = Q_1(K, \mathrm{IV})$, where Q_1 is a polynomial involving the bits of K and IV. From now on we will use the term Q_i in general for this. Imposing the conditions on the IV bits as

$$\mathtt{TypeI}: iv_{46} = iv_0 + iv_3 + iv_{25} = 0;$$

$$\mathtt{TypeII}: iv_{42} = G_1(K),$$

we have $\Delta z_{17} = Q_1(K, \mathrm{IV}) = 0$.

[C2.] Case ($t = 20$): In this round, $\Delta z_{20} = Q_2(K, \mathrm{IV})$. Similarly, imposing the following conditions on IV bits

$$\mathtt{TypeI}: iv_{49} = iv_3 + iv_6 + iv_{23} = 0;$$

$$\mathtt{TypeII}: iv_{28} = G_2(K),$$

we get $\Delta z_{20} = Q_2(K, \mathrm{IV}) = 0$.

[C3.] Case ($t = 36$): Here, $\Delta z_{36} = Q_3(K, \mathrm{IV})$. Here, we set the following conditions on IV bits to make $\Delta z_{36} = Q_3(K, \mathrm{IV}) = 0$.

$$\mathtt{TypeI}: iv_5 = iv_{48} = iv_{47} = 0, iv_2 = 1,$$

$$iv_{25} = iv_{15}, iv_{40} = iv_{53}, iv_{22} = iv_{44}, iv_4 = iv_{26},$$

$$iv_1 = iv_{16} + iv_{19} + iv_{23} + iv_{26} + iv_{39} + iv_{41};$$

$$\mathtt{TypeII}: iv_{27} = G_3(K), iv_{54} = G_4(K).$$

[C4.] Case ($t = 37$): Here, $\Delta z_{37} = Q_4(K, \mathrm{IV})$. We set the following conditions on IV bits to make $\Delta z_{37} = Q_4(K, \mathrm{IV}) = 0$.

$$\mathtt{TypeI}: iv_{24} = iv_{50} = iv_{51} = 0,$$

$$iv_{16} = iv_{23} + iv_{26} + iv_{41},$$

$$iv_7 = iv_8 + iv_{16} + iv_{18} + iv_{21} + iv_{23} + iv_{29} + iv_{30} + iv_{34}$$

$$+ iv_{41} + iv_{43} + iv_{44} + iv_{53} + iv_{56} + iv_{59};$$

$$\mathtt{TypeII}: iv_{53} = G_5(K).$$

[C5.] Case ($t = 38$): In this round, $\Delta z_{38} = Q_5(K, \mathrm{IV})$. We set the following conditions to make $\Delta z_{38} = Q_5(K, \mathrm{IV}) = 0$

$$\mathtt{TypeI}: iv_{34} = 0,$$

$$iv_8 = iv_{17} + iv_{18} + iv_{21} + iv_{30} + iv_{34} + iv_{43} + iv_{44} + iv_{55},$$

$$iv_{17} = iv_{55}, iv_{19} = iv_{39}, iv_{23} = iv_{41} + iv_{44};$$

$$\mathtt{TypeII}: iv_{56} = G_6(K), iv_{59} = G_7(k).$$

Hence for a fixed key K, if the IV bits satisfy the conditions [**C1**, **C2**, **C3**, **C4**, **C5**] and the initial conditions $iv_{63} = iv_{62} = iv_{61} = 1$ then $\Delta z_t = 0$ for $0 \le t \le 41$. Complete set of **Type I** and **Type II** conditions on IV bits is given in (7)

$$\left\{\begin{array}{c} \mathtt{TypeI}: \\ iv_1 = iv_5 = iv_{24} = iv_{34} = iv_{46} = iv_{47} = iv_{48} \\ = iv_{49} = iv_{50} = iv_{51} = 0; \\ iv_2 = iv_{61} = iv_{62} = iv_{63} = 1; \\ iv_0 = iv_3 + iv_{25}; iv_3 = iv_6 + iv_{23}; iv_4 = iv_{26}; \\ iv_7 = iv_{26} + iv_{29} + iv_{53} + iv_{56} + iv_{59}; \\ iv_8 = iv_{18} + iv_{21} + iv_{30} + iv_{43} + iv_{44}; \\ iv_{15} = iv_{25}; iv_{16} = iv_{26} + iv_{44}; \\ iv_{17} = iv_{55}; iv_{19} = iv_{39}; iv_{22} = iv_{44}; \\ iv_{23} = iv_{41} + iv_{44}; iv_{40} = iv_{53}; \\ \mathtt{TypeII}: \\ iv_{42} = G_1(K); iv_{28} = G_2(K); iv_{27} = G_3(K); \\ iv_{54} = G_4(K); iv_{53} = G_5(K); iv_{56} = G_6(K); \\ iv_{59} = G_7(k). \end{array}\right\} \quad (7)$$

Since the last 3 bits of the IV are 1 (i.e. $iv_{63} = iv_{62} = iv_{61} = 1$), there is a possibility to go for $t, (t \le 3)$ inverse KSA rounds to have another valid initial state keeping the last 16 bits of the initial state S_0 (i.e. the padding bits) as 1. In this manner, one can increase the round number to $(112 + t)$ rounds with few more conditions on IV and K bits. We discuss the possibility of such improvement of round numbers for $t = 1, 2, 3$ as follows.

For the first step of inversion in KSA (i.e. $t = 1$), we need to make $z + l_0 = 1$ where z is the output bit and l_0 is the feedback bit of the LFSR for inverse KSA. For this, we need to set the following conditions on key and IV bits to have $z = 1$, $l_0 = 0$

$$k_0 = k_1 + k_3 + k_9 + k_{30} + k_{42} + k_{55}, \tag{8}$$

$$iv_{12} = iv_{37} + iv_{44} + 1. \tag{9}$$

Following the same process for the second inverse round of the KSA, we set the conditions as

$$iv_{44} = 0; iv_{11} = iv_{21} + iv_{36} + iv_{41} + iv_{60}; iv_{41} = 0. \tag{10}$$

$$\begin{aligned} k_{79} = {} & k_1 + k_2 + k_3 + k_9 + k_{13} + k_{20} + k_{27} + k_{29} + k_{30} \\ & + k_{32} + k_{36} + k_{41} + k_{42} + k_{44} + k_{51} + k_{54} + k_{55} \\ & + k_{59} + k_{61} + k_8 k_{14} + k_{32} k_{36} + k_{59} k_{62} \\ & + k_{20} k_{27} k_{32} + k_{44} k_{51} k_{59} + k_8 k_{27} k_{44} k_{62} \\ & + k_{14} k_{20} k_{59} k_{62} + k_{32} k_{36} k_{51} k_{59} \\ & + k_8 k_{14} k_{20} k_{27} k_{32} + k_{36} k_{44} k_{51} k_{59} k_{62} \\ & + k_{20} k_{27} k_{32} k_{36} k_{44} k_{51}. \end{aligned} \tag{11}$$

Now if we run one more inverse KSA round then a difference vector for the secret key K is formed. Since the difference is not allowed for the secret key K, we cannot proceed for the third KSA inverse (i.e. $t = 3$). This scenario has been discussed in Section 4.1. The inclusion of two **Type III** conditions ((8) and (11)) reduces the key space by a dimension of two (i.e. one-fourth of the original key space). Including these four **Type I** and two **Type III** conditions with the constraints in (7) on the key and IV bits and further imposing two inverse KSA rounds on the state, we have the initial state S_0 for the Grain-v1. Then starting from the initial state S_0, we have a non-randomness in the first keystream bit of Grain-v1 with 114 initialisation rounds.

In this case, the total number of free IV variables is 26 and **Type II** conditions are 7. Hence, the value of $\mathscr{W}_\phi = 2^7 \times (1/\sqrt{8\pi N}) \simeq 0.00312$, where $N = 2^{26}$ is the size of the sample space (see Section 2.1 for the calculation of $\mathscr{W}_\phi$).

Furthermore, to show a non-randomness in the 114th round of KSA, as in Section 3, we compute the sum $\mathscr{W} = \sum_{0 \le i \le 127, p_i > 0.5} (p_i - 0.5)$, where $p_i = \Pr[\Delta z_{114} = 1]$ corresponds with the ith assignment $\mathscr{A}_i$ $(0 \le i \le 127)$. Each assignment $\mathscr{A}_i$ is an assignment of binary value to the seven IV bits involved in the **Type II** conditions.

From the experiment with 2048 random keys, it is observed that for ∼73% cases $\mathscr{W} < \mathscr{W}_\phi = 0.00312$. Therefore, like 112 KSA rounds, we can design a distinguisher for the first keystream bit of the Grain-v1 with 114 KSA rounds with a success rate of ∼73% in the weak key setup as the key relation provided in **Type III** conditions. For further cross-checking, we performed the same experiment for 115 rounds and note that the experimental success rate of distinguishing became ∼56% which is closer to 50%, i.e. for 2^{78} keys, the first keystream bit of Grain-v1 with 114 initialisation rounds can be distinguished from a random bit with a success rate of ∼73%.

The success rate for the case of Grain-v1 with 115 initialisation rounds is ∼56%. To claim this small success rate to use for designing a distinguisher, we need to run the same experiment for a large number of random keys, which is quiet impossible for us with our present computational power.

4.1 Non-randomness of Grain-v1 with 116 KSA round with one bit difference in keys

This section extends the distinguisher from 114 initialisation rounds to 116 rounds. We first introduce one extra **Type I** condition $iv_{60} = 1$, then start the inverse KSA of Grain-v1 with the same setup presented in Section 4. As we have already performed two inverse KSA rounds, the IV difference bits moved to the 22nd and 47th positions of the LFSR. After one more round of inverse KSA, these difference bits will move to the 23rd and 48th positions of the LFSR. During the inverse KSA, the bit at the 23rd position of LFSR is involved in the computation of the linear feedback bit of the LFSR. Hence, it will flip the feedback bit of the LFSR in this inverse KSA round. Furthermore, this feedback bit of LFSR is involved linearly in the feedback bit computation of NFSR. So it will also flip the feedback bit of the NFSR. After this inverse round, the state bits $\{l_0, l_{23}, l_{48}\}$ and $\{n_0\}$ of the present state of the cipher are flipped. As we have set $iv_{61} = 1$ (in Section 4), the last 16 bits of the LFSR remain valid (i.e. all 1). For this one round of inverse KSA, we set some conditions on IV bits and secret key bits to make $z + l_0 = 1$ with $z = 1$ and $l_0 = 0$ (as in Section 4). The following conditions are introduced here:

$$iv_{43} = 0; iv_6 = iv_{15} + 1; iv_{10} = iv_{20} + iv_{35} + iv_{59}; \tag{12}$$

$$\begin{aligned} k_{78} = {} & k_2 + k_3 + k_8 + k_9 + k_{12} + k_{19} + k_{26} + k_{28} + k_{29} \\ & + k_{30} + k_{31} + k_{35} + k_{40} + k_{41} + k_{42} + k_{43} + k_{50} \\ & + k_{53} + k_{54} + k_{55} + k_{58} + k_{60} + k_7 k_{13} + k_{31} k_{35} \\ & + k_{58} k_{61} + k_{19} k_{26} k_{31} + k_{43} k_{50} k_{58} + k_7 k_{26} k_{43} k_{61} \\ & + k_{13} k_{19} k_{58} k_{61} + k_{31} k_{35} k_{50} k_{58} + k_7 k_{13} k_{19} k_{26} k_{31} \\ & + k_{35} k_{43} k_{50} k_{58} k_{61} + k_{19} k_{26} k_{31} k_{35} k_{43} k_{50}. \end{aligned} \tag{13}$$

Furthermore, we run the inverse KSA for one more round. It can be observed that iv_{60} was free for the distinguisher on the 114th round (presented in Section 4), but here we have set $iv_{60} = 1$. For the second inverse KSA round the flipped bit at NFSR will move to $\{n_1\}$ of the present state of NFSR. As the NFSR bit $\{n_1\}$ is involved in the computation of keystream bit, the keystream bit z will be flipped (i.e. $\Delta z = 1$) in this round. As a result, the linear feedback bit of the LFSR of this inverse KSA round will also be flipped. Owing to the linear involvement of both the keystream bit and the linear feedback bit in the computation of non-linear feedback bit of the NFSR, the non-linear feedback bit remains unaffected in this inverse KSA round. After this inverse KSA round, state bits $\{l_0, l_1, l_{24}, l_{49}\}$ of the current state of LFSR and state bit $\{n_1\}$ of current state of NFSR are flipped. The last 16 bits of LFSR remain valid (i.e. all 1) as we have set $iv_{60} = 1$. In this inverse KSA round, we also set following conditions on IV bits and secret key bits to make $z + l_0 = 1$, where $z = 1$ and $l_0 = 0$ (as in Section 4)

$$iv_{21} = iv_{42} + 1; iv_9 = iv_{39} + iv_{58}; \tag{14}$$

$$k_{59} = 0; \tag{15}$$

$$\begin{aligned} k_{77} = {} & k_1 + k_2 + k_7 + k_8 + k_{11} + k_{18} + k_{25} + k_{27} + k_{28} \\ & + k_{29} + k_{30} + k_{34} + k_{39} + k_{40} + k_{41} + k_{42} + k_{49} \\ & + k_{52} + k_{53} + k_{54} + k_{57} + k_6 k_{12} + k_{30} k_{34} \\ & + k_{57} k_{60} + k_{18} k_{25} k_{30} + k_{42} k_{49} k_{57} + k_6 k_{25} k_{42} k_{60} \\ & + k_{12} k_{18} k_{57} k_{60} + k_{30} k_{34} k_{49} k_{57} + k_6 k_{12} k_{18} k_{25} k_{30} \\ & + k_{34} k_{42} k_{49} k_{57} k_{60} + k_{18} k_{25} k_{30} k_{34} k_{42} k_{49}. \end{aligned} \tag{16}$$

Now to go back further, we need to set $iv_{59} = 1$, which is not possible as this bit is involved in **Type II** conditions (see (7)). Here we have allowed 1 bit difference in the state of the NFSR, i.e. we have allowed 1 bit difference in the secret key bits for 116 rounds. For two extra inverse KSA rounds, three extra **Type III** and six extra **Type I** conditions (including $iv_{60} = 1$) are introduced. Here, we consider two states, which differ only at $\{n_1, l_0, l_1, l_{24}, l_{49}\}$ as the initial state of two ciphers. After that, we perform 116 KSA rounds on both the ciphers. Under **Type I**, **Type II**, and **Type III** conditions, we have observed the following non-randomness in Δz_{116} after 116 initialisation rounds.

It can be noticed that the sample size is reduced to 2^{20}. With this, we calculate $\mathscr{W}_\phi$, which is ∼0.02493. Now we compute $\mathscr{W} = \sum_{0 \le i \le 127, p_i > 0.5} (p_i - 0.5)$, where $p_i = \Pr[\Delta z_{116} = 1]$ corresponds with the ith assignment $\mathscr{A}_i (0 \le i \le 127)$.

For each key, we compute $\mathscr{W}$ and compare with $\mathscr{W}_\phi (\simeq 0.02493)$. We perform this experiment for 4096 random keys and it has been observed that for 62% cases $\mathscr{W} < \mathscr{W}_\phi (\simeq 0.02493)$. Hence, Grain-v1 with 116 KSA rounds can

be distinguished in a weak key setup with the 1 bit difference in the secret key and 4 bits difference in the IV. Further to cross check our distinguisher, we perform the same experiment for 117 rounds but the success rate is 52% (which is very close to 50%). To claim this success chance ($\simeq 52\%$) we need to repeat this experiment for a large number of random keys, which is an impossible task to verify with our present computation power.

Existing works under related key setup require more key bit differences than us, but the rounds are higher than 116. Our main contribution is distinguisher, and we show how the technique could be evolved when very few key bits differ.

5 Conclusion

In this study, we have introduced distinguishers for Grain-v1 with 112 and 114 initialisation rounds. The first one can distinguish Grain-v1 with 112 initialisation rounds from a random source with a 99% success rate. The second one can distinguish Grain-v1 with 114 initialisation rounds from a random source with a 73% success rate in a weak key setup for one-fourth of all the keys. Here, for the first time, we have used the difference vector of weight 2 to improve the number of rounds. The analysis in certain cases is indeed complicated and required manual intervention rather than writing computer programmes. Finally, the distinguisher for 114 rounds could be extended to 116 rounds with 1 bit and 4 bit differences in key and IV, respectively. The success rate of this distinguisher is 62%. We are presently working on similar techniques for Grain-128a. Furthermore, conditional differential attack on stream ciphers (designed like Grain) with more IV bit differences can possibly be thought of as future work. Automated handling of such scenarios will be of considerable interest.

6 Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable suggestions and comments, which considerably improved the quality of the paper.

7 References

[1] Hell, M., Johansson, T, Meier, W.: 'Grain: a stream cipher for constrained environments', *Int. J. Wirel. Mob. Comput.*, 2007, **2**, (1), pp. 86–93

[2] eSTREAM: Stream cipher project for ECrypt. Available at http://www.ecrypt.eu.org/stream/, 2005

[3] Knellwolf, S., Meier, W., Naya-Plasencia, M.: 'Conditional differential cryptanalysis of NLFSR-based cryptosystems'. Advances in Cryptology-ASIACRYPT 2010, Toronto, Canada, 2010, pp. 130–145

[4] Aumasson, J.-P., Dinur, I., Henzen, L., *et al.*: 'Efficient FPGA implementations of high-dimensional cube testers on the stream cipher grain-128'. Special-purpose Hardware for Attacking Cryptographic, SHARCS'09, Systems, Lausanne, Switzerland, 2009, p. 147

[5] Banik, S.: 'Conditional differential cryptanalysis of 105 round grain v1', *Cryptogr. Commun.*, 2016, **8**, pp. 113–137

[6] Sarkar, S.: 'A new distinguisher on Grain v1 for 106 rounds'. Int. Conf. on Information Systems Security, Kolkata, India, 2015, pp. 334–344

[7] Ma, Z., Tian, T., Qi, W.-F.: 'Improved conditional differential attacks on Grain v1', *IET Inf. Sec.*, 2017, **11**, (1), pp. 46–53

[8] Watanabe, Y., Todo, Y., Morii, M.: 'New conditional differential cryptanalysis for NLFSR-based stream ciphers and application to Grain v1'. 2016 11th Asia Joint Conf. on Information Security (AsiaJCIS), Fukuoka, Japan, 2016, pp. 115–123

[9] Ma, Z., Tian, T., Qi, W.-F.: 'A new distinguishing attack on Grain-v1 with 111 initialization rounds', *J. Syst. Sci. Complex.*, 2018, **32**, pp. 1–14

[10] Cannière, C.D., Küçük, Ö., Preneel, B.: 'Analysis of grain's initialization algorithm'. AFRICACRYPT, Casablanca, Morocco, 2008, pp. 276–289

[11] Mihaljević, M.J., Sinha, N., Gangopadhyay, S., *et al.*: 'An improved cryptanalysis of lightweight stream cipher Grain-v1'. Cryptacus: Workshop and MC meeting, Nijmegen, Netherlands, 16–18 November 2017

[12] Zhang, B., Xu, C., Meier, W.: 'Fast near collision attack on the grain v1 stream cipher'. EUROCRYPT, Tel Aviv, Israel, 2018, pp. 771–802

[13] Todo, Y., Isobe, T., Meier, W., *et al.*: 'Fast correlation attack revisited – cryptanalysis on full Grain-128a, Grain-128, and Grain-v1'. Advances in Cryptology - CRYPTO 2018, Santa Barbara, CA, USA, 2018, pp. 129–159

[14] Banik, S.: 'Some insights into differential cryptanalysis of Grain v1'. Australasian Conf. on Information Security and Privacy, Wollongong, Australia, 2014, pp. 34–49

[15] Banik, S., Maitra, S., Sarkar, S.: 'A differential fault attack on the grain family of stream ciphers'. Cryptographic Hardware and Embedded Systems, Leuven, Belgium, 2012, pp. 122–139

[16] Banik, S., Maitra, S., Sarkar, S.: 'A differential fault attack on the grain family under reasonable assumptions'. Indocrypt, Kolkata, India, 2012, pp. 191–208

[17] Ding, L., Jin, C., Guan, J., *et al.*: 'New state recovery attacks on the Grain v1 stream cipher', *China Commun.*, 2016, **13**, (11), pp. 180–188

[18] Dinur, I., Shamir, A.: 'Breaking Grain-128 with dynamic cube attacks'. Fast Software Encryption, Lyngby, Denmark, 2011, pp. 167–187

[19] Fischer, S., Khazaei, S., Meier, W.: 'Chosen IV statistical analysis for key recovery attacks on stream ciphers'. AFRICACRYPT, Casablanca, Morocco, 2008, pp. 236–245

[20] Ma, Z., Tian, T., Qi, W.-F.: 'Conditional differential attacks on Grain-128a stream cipher', *IET Inf. Sec.*, 2017, **11**, (3), pp. 139–145

[21] Knellwolf, S.: 'Cryptanalysis of hardware-oriented ciphers the Knapsack generator, and SHA-1', PhD dissertation, Zurich, 2012

[22] Lee, Y., Jeong, K., Sung, J., *et al.*: 'Related-key chosen IV attacks on Grain-v1 and Grain-128'. Australasian Conf. on Information Security and Privacy 2008, Wollongong, Australia, 2008, pp. 321–335

[23] SAGE: Open source mathematical software, http://www.sagemath.org/

[24] Supporting Materials: Available at https://drive.google.com/drive/folders/1FukMCaCeCRVidgQVMLpf_L5yFoVC60-9?usp=sharing