IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

Advanced conditional differential attack on Grain-like stream cipher and application on Grain v1

ISSN 1751-8709
Received on 30th October 2017
Revised 2nd May 2018
Accepted on 2nd November 2018
E-First on 21st January 2019
doi: 10.1049/iet-ifs.2018.5180
www.ietdl.org

Jun-Zhi Li[1], *Jie Guan*[1] ✉

[1]*Zhengzhou Institute of Information Science and Technology, 62 Kexue Road, Zhengzhou 450001, People's Republic of China*
✉ *E-mail: guanjie007@163.com*

Abstract: Conditional differential attacks against non-linear feedback shift register based cryptosystems were proposed by Knellwolf *et al.* at Asiacrypt 2010. In this study, the authors propose an advanced conditional differential attack on Grain-like stream cipher. They trace propagations of a single bit difference of internal states both inversely and forward. Methods of both searching for the longest inverse difference characteristic with probability one and deriving initial value (IV) conditions with the max inverse round are introduced. When tracing forward, conditions are imposed to limit the propagation of difference to obtain a high bias. Conditions of the proposed method are only imposed on IV bits and the proposed attack works in the single-key setting. Moreover, a method of recovering key expressions as well as bias-complexity-success probability target is presented in this study. Using the proposed method, the authors conduct a key recovery attack on 114-round Grain v1, recovering 6 key expressions with the time complexity of 2^{32}, which is also verified by experiments. With more conditions imposed, this attack can be improved to Grain v1 of 120 rounds, recovering 12 key expressions with the time complexity of $2^{42.75}$ and theoretical success probability of about 93%, which is ten rounds longer than the longest previous result of Grain v1 in the single-key setting.

1 Introduction

Conditional differential attacks (CDAs) against non-linear feedback shift register (NFSR) based cryptosystems were first proposed by Knellwolf *et al.* at Asiacrypt 2010 [1]. This method has been successfully used to analyse Grain v1 [1], Trivium [2] and KATAN [2]. CDA can be classified into two categories according to the means of deducing the probability. One kind of CDA only focuses on forward propagation of difference, such as [3–7]. Another kind of CDA concentrates on both inverse and forward propagation of difference, such as [8–10]. As differences quickly spread into keys, this method always works in the related-key setting or weak-key setting. In [8], Zhang *et al.* proposed an improved CDA of this type, without concerning the attack setting. In [9], Knellwolf conducted this method in the related-key setting. Recently, Watanabe *et al.* [10] improved this technique to work in the weak-key setting.

Grain-like stream cipher is a family of stream ciphers with the similar structure as Grain v1 [11]. Grain-like stream cipher consists of a linear feedback shift register (LFSR), a NFSR and a filter function. Grain v1, Grain 128 [12] and Grain 128a [13] are typical Grain-like stream ciphers, among which Grain v1 is the main research subject of this paper. Grain v1 is one of eSTREAM [14] hardware-oriented finalists proposed by Hell, Johansson and Meier in 2005. So far, several cryptanalytic results against Grain v1 have been proposed such as CDA [3–5], dynamic cube attack [6, 15], related-key chosen initial value (IV) attack [16], near collision attack [17], differential fault attack [18, 19] and internal state recovery attack [20, 21]. We give some CDA results on Grain v1 in this paper.

CDA has been widely used for reduced round Grain v1. In [1], Knellwolf *et al.* gave distinguishing and key recovery attacks for 97 rounds and extended the attack to 104 rounds with complexity of 2^{35} and advantage of about 0.45. However, results of [1] are only based on experiments. Then Banik provided a theoretical framework to prove the biases of 97 rounds in [3]. Describing CDA as dynamic cube attack, Banik further attacked Grain v1 to 105 rounds and recovered six key expressions with complexity of $O(2^{34})$ and successful probability of about 92% in [6]. Later, Sarkar presented a new distinguisher of 106-round Grain v1 using the similar approach with complexity of $O(2^{30})$ and advantage of about 0.63 in [4]. However, this work could not recover key expressions due to the lack of free IV variables. In [5], Ma *et al.* improved CDA on Grain v1 to 110 rounds with complexity of $O(2^{47})$. As they focused on highly saving IV, it required to guess more key expressions, leading to larger complexity. In [10], Watanabe *et al.* showed conditional differential distinguisher on Grain v1 up to 114 rounds in $O(2^{40})$ weak key subspace. In [9], Knellwolf presented a related-key CDA on 133-round Grain v1 with complexity of $O(2^{35})$.

In this paper, we propose an advanced CDA on Grain-like stream cipher in the following aspects. Starting from a single bit difference of the internal states, propagations of difference are traced both inversely and forward. When tracing inversely, conditions of IVs are imposed to obtain the longest difference trail with initial difference only involving IV bits. When tracing forward, conditions are imposed to limit the propagation of difference to obtain a high bias of the keystream output. Conditions of our method are only imposed on IV bits. So our attack works in the single-key setting, which is more feasible than previous similar attacks. Moreover, a method for recovering key expressions as well as bias-complexity-success probability target is presented in this paper. By applying our method, a distinguishing and key recovery attack is conducted on 114-round Grain v1, recovering 6 key expressions with time complexity of 2^{32}, data complexity of 2^{26} and theoretical success probability of about 96%. Experiments verify our attack that can recover key expressions in 17.6 min on personal PC with success rate of about 97%. With more conditions imposed, this attack can be improved to Grain v1 of 120 rounds, recovering 12 key expressions with time complexity of $2^{42.75}$, data complexity of $2^{29.75}$ and theoretical success probability of about 93%. The memory complexity of the two attacks is negligible. Our result, which is ten rounds longer than the previous results, is the longest rounds attack on reduced Grain v1 in the single-key setting. Table 1 summarises previous results of CDAs on Grain v1.

This paper is organised as follows. The description of Grain-like stream cipher and differential engine of Grain-like stream cipher are presented in Section 2. Section 3 provides our advanced CDA on Grain-like stream cipher. Section 4 presents our attacks on

Table 1 Results of CDA on Grain v1

Rounds	Attack setting	Type of attack	Time complexity	Data complexity	Reference
104	single-key	distinguishing and recovery (15 key expressions)	$O(2^{40})$	$O(2^{25})$	[5]
105	single-key	distinguishing and recovery (6 key expressions)	$O(2^{35})$	$O(2^{28})$	[6]
106	single-key	distinguishing only	$O(2^{30})$	$O(2^{24})$	[4]
107	single-key	distinguishing and recovery (12 key expressions)	$O(2^{42})$	$O(2^{30})$	[5]
110	single-key	distinguishing and recovery (15 key expressions)	$O(2^{47})$	$O(2^{31})$	[5]
114	*40 bits weak-key*	distinguishing and recovery (1 key bit)	$O(2^{33})$	$O(2^{32})$	[10]
114	single-key	distinguishing and recovery (6 key expressions)	$O(2^{32})$	$O(2^{26})$	**Section 4.1**
120	single-key	distinguishing and recovery (12 key expressions)	$O(2^{42.75})$	$O(2^{29.75})$	**Section 4.2**
133	*2 related-keys*	distinguishing and recovery (8 key expressions)	$O(2^{35})$	$O(2^{27})$	[9]

The italic words are used to emphasise that the attacks settings of these two attacks are different with other attacks in this table. The bold words are used to emphasise that these two results are first reported in this paper.

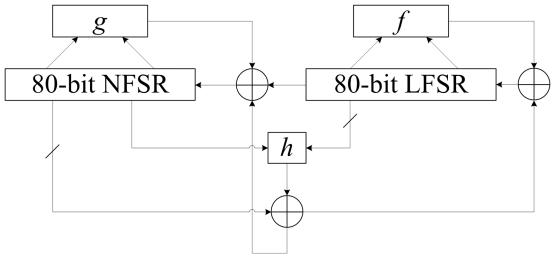

Fig. 1 *Overview of Grain v1's initialisation algorithm*

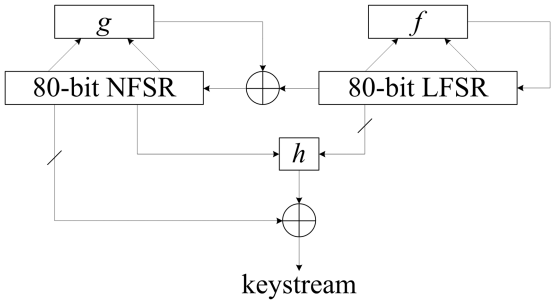

Fig. 2 *Overview of Grain v1's keystream generation algorithm*

Grain v1 of 114 and 120 rounds. Then we conclude our paper in Section 5.

2 Preliminaries

2.1 Notations

$\oplus$: addition modulo 2;
$x|_t$: tth bit of x;
$HW(\boldsymbol{X})$: *Hamming weight* of vector $\boldsymbol{X}$.

Differential forms of x and corresponding notations:

0: x is constant 0;
1: x is constant 1;
$\bar{0}$: $\Delta x = 0$ with probability 1;
$\bar{1}$: $\Delta x = 1$ with probability 1;
$\bar{2}$: $\Delta x = 1$ with probability p, $(0 < p < 1)$.
$S_i = [s_i, s_{i+1}, \ldots, s_{i+n-1}]$: LFSR internal states of ith clock;
$B_i = [b_i, b_{i+1}, \ldots, b_{i+m-1}]$: NFSR internal states of ith clock.
$\Delta_\phi - Grain$: differential engine of Grain-like stream cipher.

Notations of parameters of Algorithm 1:

r_e: inverse round;
d_e: characteristic corresponding to r_e and $d_e = \{d_1, d_2, \ldots, d_{r_e}\}$, where $d_i = \Delta z_{-i}, (1 \le i \le r_e)$;
ΔS_{-r_e}: difference in LFSR of r_e inverse round;
$r_{\max}$: max inverse round;
$d_{\max}$: characteristic corresponding to $r_{\max}$ and $d_{\max} = \{d_1, d_2, \ldots, d_{r_{\max}}\}$, where $d_i = \Delta z_{-i}, (1 \le i \le r_{\max})$;
$\Delta S_{-r_{\max}}$: difference in LFSR of the max inverse round.

2.2 Description of Grain v1

Grain v1 uses an 80-bit secret key and a 64-bit IV and has a compact structure. It consists of an 80-bit NFSR, an 80-bit LFSR and a filter function combined the two registers. The registers are filled by key and IV bits as follows:

$$\begin{aligned} b_i &= k_i \text{ for } (0 \le i \le 79) \\ s_i &= iv_i \text{ for } (0 \le i \le 63) \\ s_i &= 1 \text{ for } (64 \le i \le 79) \end{aligned}$$

The update function of LFSR is $s_{i+80} = s_i \oplus s_{i+13} \oplus s_{i+23} \oplus s_{i+38} \oplus s_{i+51} \oplus s_{i+62}$ and the update function of NFSR is

$$\begin{aligned} b_{i+80} = {} & s_i \oplus b_i \oplus b_{i+9} \oplus b_{i+14} \oplus b_{i+21} \oplus b_{i+28} \oplus b_{i+33} \\ & \oplus b_{i+37} \oplus b_{i+45} \oplus b_{i+52} \oplus b_{i+60} \oplus b_{i+62} \oplus b_{i+63}b_{i+60} \\ & \oplus b_{i+37}b_{i+33} \oplus b_{i+15}b_{i+9} \oplus b_{i+60}b_{i+52}b_{i+45} \\ & \oplus b_{i+33}b_{i+28}b_{i+21} \oplus b_{i+63}b_{i+45}b_{i+28}b_{i+9} \\ & \oplus b_{i+60}b_{i+52}b_{i+37}b_{i+33} \oplus b_{i+63}b_{i+60}b_{i+21}b_{i+15} \\ & \oplus b_{i+63}b_{i+60}b_{i+52}b_{i+45}b_{i+37} \oplus b_{i+33}b_{i+28}b_{i+21}b_{i+15}b_{i+9} \\ & \oplus b_{i+52}b_{i+45}b_{i+37}b_{i+33}b_{i+28}b_{i+21} \end{aligned} \quad (1)$$

The keystream generating function is

$$z_i = \sum_{k \in A} b_{i+k} \oplus h(s_{i+3}, s_{i+25}, s_{i+46}s_{i+64}, b_{i+63})$$

where $A = \{1, 2, 4, 10, 31, 43, 56\}$ and h is a function of degree 3 defined as

$$\begin{aligned} h(s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63}) = {} & s_{i+25} \oplus b_{i+63} \oplus s_{i+3}s_{i+64} \\ & \oplus s_{i+46}s_{i+64} \oplus s_{i+64}b_{i+63} \oplus s_{i+3}s_{i+25}s_{i+46} \oplus s_{i+3}s_{i+46}s_{i+64} \\ & \oplus s_{i+3}s_{i+46}b_{i+63} \oplus s_{i+25}s_{i+46}b_{i+63} \oplus s_{i+46}s_{i+64}b_{i+63} \end{aligned} \quad (2)$$

The cipher outputs keystream after the initialisation phase. During the initialisation phase, the cipher is clocked 160 times without outputting keystream. Instead, the output z_i is xored feedback in both the LFSR and NFSR. Figs. 1 and 2 show the structure of Grain v1.

In this paper, Grain v1 reduced to r rounds means the cipher outputs keystream after r clocks feedback and the first keystream bit is z_r.

Table 2 Xor operation between differential forms

$\oplus$	0	1	$\bar{0}$	$\bar{1}$	$\bar{2}$
0	0	1	$\bar{0}$	$\bar{1}$	$\bar{2}$
1	1	0	$\bar{0}$	$\bar{1}$	$\bar{2}$
$\bar{0}$	$\bar{0}$	$\bar{0}$	$\bar{0}$	$\bar{1}$	$\bar{2}$
$\bar{1}$	$\bar{1}$	$\bar{1}$	$\bar{1}$	$\bar{0}$	$\bar{2}$
$\bar{2}$	$\bar{2}$	$\bar{2}$	$\bar{2}$	$\bar{2}$	$\bar{2}$

2.3 Description of Grain-like stream cipher

Grain-like stream cipher consists of two FSRs, a LFSR of n bits and a NFSR of m bits. The LFSR is used to control the NFSR. The update function of LFSR is defined as

$$s_{i+n} = F(S_i) = s_i \oplus s_{i+j_{11}} \oplus s_{i+j_{12}} \oplus \cdots \oplus s_{i+j_{1a}}.$$

The update function of NFSR is defined as

$$\begin{aligned} b_{i+m} &= s_i \oplus G(B_i) = s_i \oplus G(b_i, b_{i+j_{21}}, b_{i+j_{22}}, \ldots, b_{i+j_{2c}}) \\ &= s_i \oplus b_i \oplus b_{i+j_{21}} \oplus b_{i+j_{22}} \oplus \cdots \oplus b_{i+j_{2c_0}} \\ &\quad \oplus g(b_{i+j_{2c_0+1}}, b_{i+j_{2c_0+2}}, \ldots, b_{i+j_{2c}}) \end{aligned}$$

where $b_i, b_{i+j_{21}}, b_{i+j_{22}}, \ldots, b_{i+j_{2c_0}}$ only contribute linearly.

The keystream generating function is defined as (see equation below). At the beginning of the initialisation stage, the LFSR is filled by IV bits and the NFSR is filled by key bits. In the initialisation stage, the cipher does not output keystream instead of xoring it to LFSR and NFSR. So the update function of LFSR in the initialisation stage is

$$F(S_i) \oplus H(S_i, B_i)$$

The update function of NFSR in the initialisation stage is

$$s_i \oplus G(B_i) \oplus H(S_i, B_i)$$

2.4 Differential engine of Grain-like stream cipher

In CDA, the first information that cryptanalyst should obtain is how the differences propagate in the internal states. In order to eliminate Δz_i, the exact differences of the internal states used to compute z_i must be obtained. So far, several methods are proposed to solve this problem. In [3], Banik presented a differential engine to track the differential trails of Grain family. In [5], Ma *et al.* improved the differential engine by considering the constant situation that missed by Banik. In this section, we introduce the differential engine of Grain-like stream cipher considering constant.

There are several differential forms for the difference of state bits. Differential forms of x and the corresponding notes are given in Section 2.1 [22].

Then, the xor operation between differential forms is defined in Table 2.

Without loss of generality, suppose $f(x_1, x_2, \ldots, x_p) = x_1 \oplus x_2 \oplus \cdots \oplus x_{p_1} \oplus \varphi(x_{p_1+1}, x_{p_1+2}, \ldots, x_p)$, where $x_1, x_2, \ldots, x_{p_1}$ only contribute linearly. The differential form of a Boolean function f is Df that is determined as follows.

If there are constants in $\{x_{p_1+1}, x_{p_1+2}, \ldots, x_p\}$, we substitute the constant variables with their value and rearrange function f to the form

$f'(x'_1, x'_2, \ldots, x'_{p'}) = x'_1 \oplus x'_2 \oplus \cdots \oplus x'_{p'_1} \oplus \varphi'(x'_{p'_1+1}, x'_{p'_1+2}, \ldots, x'_{p'})$,

where $x'_1, x'_2, \ldots, x'_{p'_1}$ only contribute linearly and no constant is in $\{x'_{p'_1+1}, x'_{p'_1+2}, \ldots, x'_{p'}\}$. Then, compute Df' and $Df = Df'$.

Now, suppose there is no constant in $\{x_{p_1+1}, x_{p_1+2}, \ldots, x_p\}$, according to Table 2, Df could be computed as follows:

(i) If $\exists i \in \{1, 2, \ldots, p\}$, leading to $\Delta x_j = \bar{2}$ or $\exists i \in \{p_1+1, p_1+2, \ldots, p\}$ leading to $\Delta x_i = \bar{1}$, then $Df = \bar{2}$.

(ii) If situation (i) does not happen, then $Df = \bar{0} \oplus \oplus_{j \in \{1,2,\ldots,p_1\}} Dx_j$, where Dx_j means the differential form of x_j.

3 Advanced CDA on Grain-like stream cipher

The main idea of advanced CDA on Grain-like stream cipher is as follows: starting from a single bit difference of the internal states, propagations of difference are traced both inversely and forward. When tracing inversely, conditions of IVs are imposed to obtain a difference characteristic as long as possible with the initial difference only involving IV bits. In this step, conditions are set to keep the difference from affecting NFSR. So the initial difference does not involve key bits. When tracing forward, conditions are imposed to limit the propagation of difference to obtain a high bias of the keystream. Then, a method for recovering key expressions as well as bias-complexity-success probability target is presented in this section.

3.1 Inverse technique

The main steps of the inverse technique are given below:

Step 1: Compute the inverse function.

The inverse update function of Grain-like stream cipher is given below.

The inverse keystream generating function is

$$\begin{aligned} z_{-i} = {} & s_{l_1-i} \oplus s_{l_2-i} \oplus \cdots \oplus s_{l_d-i} \oplus b_{t_1-i} \oplus b_{t_2-i} \oplus \cdots \oplus b_{t_e-i} \oplus \\ & h(s_{h_1-i}, s_{h_2-i}, \ldots, s_{h_u-i}, b_{j_1-i}, b_{j_2-i}, \ldots, b_{j_w-i}). \end{aligned}$$

The inverse update function of LFSR is

$$s_{-i} = s_{f_1-i} \oplus s_{f_2-i} \oplus \cdots \oplus s_{f_a-i} \oplus s_{n-i} \oplus z_{-i}.$$

The inverse update function of NFSR is

$$\begin{aligned} b_{-i} = {} & z_{-i} \oplus s_{-i} \oplus b_{g_1-i} \oplus b_{g_2-i} \oplus \cdots \oplus b_{g_{c0}-i} \\ & \oplus b_{m-i} \oplus g(b_{g_{c0}+1-i}, b_{g_{c0}+2-i}, \ldots, b_{g_c-i}). \end{aligned}$$

Step 2: Search for the longest inverse difference characteristic which starts from a single bit difference and matches the search principle.

Starting from a single bit difference in LFSR, $\Delta_\Phi - Grain$ is used inversely to track the inverse difference propagation. Then, we search for the characteristic that keeps no difference in NFSR and can be derived as many rounds as possible with the probability as high as possible. Step 2 only focus on the longest characteristics and the corresponding IV conditions are handled in Step 3. This

$$\begin{aligned} z_i = H(S_i, B_i) = {} & s_{i+j_{31}} \oplus s_{i+j_{32}} \oplus \cdots \oplus s_{i+j_{3d}} \oplus b_{i+j_{41}} \oplus b_{i+j_{42}} \\ & \oplus \cdots \oplus b_{i+j_{4e}} \oplus h(s_{i+j_{51}}, s_{i+j_{52}}, \ldots, s_{i+j_{5u}}, b_{i+j_{61}}, b_{i+j_{62}}, \ldots, b_{i+j_{6w}}). \end{aligned}$$

input : $(r_e, d_e, \Delta S_{-r_e}, r_{\max}, d_{\max}, \Delta S_{-r_{\max}})$
output: $(r'_{\max}, d'_{\max}, \Delta S_{-r'_{\max}})$
Set $t \leftarrow 0, r_{\max 1} \leftarrow 0, r_{\max 2} \leftarrow 0, r_{\max 3} \leftarrow 0$;
while $\Delta z_{-r_e-t} \neq \bar{2}$ *and* $\Delta z_{-r_e-t} \oplus \Delta s_{-r_e-t} \neq \bar{1}$ **do**
  $t \leftarrow t + 1$;
  Compute Δz_{-r_e-t} and ΔS_{-r_e-t} using $\Delta_\Phi - Grain$.
end
$r_{\max 1} \leftarrow r_e + t - 1$;
if $\Delta z_{-r_e-t} = \bar{2}$ **then**
  $\Delta z_{-r_e-t} \leftarrow \bar{0}$;
  if $\Delta z_{-r_e-t} \oplus \Delta s_{-r_e-t} = \bar{0}$ **then**
    Compute $(r_{\max 2}, d_{r_{\max 2}}, \Delta S_{-r_{\max 2}})$ with Algorithm 1;
  end
  $\Delta z_{-r_e-t} \leftarrow \bar{1}$;
  if $\Delta z_{-r_e-t} \oplus \Delta s_{-r_e-t} = \bar{0}$ **then**
    Compute $(r_{\max 3}, d_{r_{\max 3}}, \Delta S_{-r_{\max 3}})$ with Algorithm 1;
  end
end
$r'_{\max} \leftarrow \max\{r_{\max}, r_{\max 1}, r_{\max 2}, r_{\max 3}\}$;
Return$(r'_{\max}, d'_{\max}, \Delta S_{-r'_{\max}})$.

Fig. 3 *Algorithm 1: Search for the longest inverse difference characteristic of given input with probability one*

input : $(r_{\max}, d_{\max}, \Delta S_{-r_{\max}})$
output: Ideal of conditions
Set $J \leftarrow \emptyset$ and the initial difference if LFSR is $\Delta S_{-r_{\max}}$;
for $i \leftarrow 0$ **to** $r_{\max} - 1$ **do**
  Compute the set of conditions J_i to impose $\Delta z_i = d_{r_{\max}-i}$.
  for $C_j \in J_i$ **do**
    if $C_j = 1 \bmod \langle J \rangle$ **then**
      Return impossible characteristic.
    end
  end
  $J \leftarrow J \cup J_i$;
end

Fig. 4 *Algorithm 2: Deriving IV conditions for the given initial difference and characteristic*

paper only considers the inverse difference characteristic with probability one.

The algorithm of searching for the longest inverse difference characteristic of given input with probability one is presented in Algorithm 1 with the notations given in Section 2.1.

With the input (r_e, d_e, ΔS_{-r_e}, $r_{\max}$, $d_{\max}$, $\Delta S_{-r_{\max}}$), Algorithm 1 recursively computes the max inverse round, the corresponding characteristic and the initial difference in LFSR. The algorithm tries every possible value of uncertain Δz_{-i} to find the difference characteristic that can be derived as many rounds as possible.

Then, for all single bit differences in LFSR Δs_j $(0 \le j \le n-1)$, Algorithm 1 is run with input $(0, (\varnothing), \Delta S_0 = \Delta s_j, 0, (\varnothing), \Delta S_0)$ to find the longest difference characteristic starting from a single bit difference in LFSR. Denote the difference starting from $\Delta s_{\max}$ and inverse $r_{\max}$ round. Then, the characteristic is $d_{\max}$ and the initial difference in LFSR is $\Delta S_{-r_{\max}}$ (Fig. 3).

Step 3: Derive IV conditions for the characteristic with the max inverse round.

Suppose the initial difference in LFSR $\Delta S_{-r_{\max}}$ and the characteristic $d_{\max}$ have been obtained. We need to derive the IV conditions to satisfy the characteristic. The method of computing conditions to set uncertain Δz_{-i} to certain difference $\bar{0}$ or $\bar{1}$ is presented below.

Without loss of generality, suppose there are u_1 IV bits in $\Delta S_{-r_{\max}}$ with difference $\bar{1}$, denoted as $v_{l_1}, v_{l_2}, \ldots, v_{l_{u_1}}$. Regarding $\Delta S_{-r_{\max}}$ as the difference of round 0, the expression of $z_{-i} = z_{r_{\max}-i}$ for key and IV bits is computed and transformed to the form:

$$z_{r_{\max}-i} = \sum_{\boldsymbol{t} \in \{0,1\}^{u_1}} (v_{l_1})^{t_1} \cdot (v_{l_2})^{t_2} \cdot \cdots \cdot (v_{l_{u_1}})^{t_{u_1}} \cdot \eta_t$$

where $\boldsymbol{t} = [t_1, t_2, \ldots, t_{u_1}]$ is a vector of dimension u_1 and $t_i (1 \le i \le u_1)$ is the ith bit of $\boldsymbol{t}$. $(v)^{t_i}$ is defined as $(v)^0 = 1$ and $(v)^1 = v$. Every $\eta_t (\boldsymbol{t} \in \{0,1\}^{u_1})$ is a function independent with $v_{l_1}, v_{l_2}, \ldots, v_{l_{u_1}}$.

Denote $\boldsymbol{e}_i (1 \le i \le u_1)$ as unit vector which the ith bit is 1 and other bits are zero. So $\Delta z_{r_{\max}-i}$ can be transformed to

$$\Delta z_{r_{\max}-i} = \sum_{i=1}^{u_1} \eta_{\boldsymbol{e}_i} \oplus \sum_{\boldsymbol{t} \in \{0,1\}^{u_1} \text{and} \boldsymbol{HW}(\boldsymbol{t}) > 1} [(v_{l_1} \oplus 1)^{t_1} \cdot (v_{l_2} \oplus 1)^{t_2} \cdot \cdots \cdot (v_{l_{u_1}} \oplus 1)^{t_{u_1}} \oplus (v_{l_1})^{t_1} \cdot (v_{l_2})^{t_2} \cdot \cdots \cdot (v_{l_{u_1}})^{t_{u_1}}] \cdot \eta_t$$

If $\Delta z_{r_{\max}-i}$ should be set to $\bar{0}$, IV conditions are set to make sure that $\sum_{i=1}^{u_1} \eta_{\boldsymbol{e}_i} = 0$ and $\eta_t = 0$, where $\boldsymbol{t} \in \{0,1\}^{u_1}$ and $\boldsymbol{HW}(\boldsymbol{t}) > 1$.

If $\Delta z_{r_{\max}-i}$ should be set to $\bar{1}$, IV conditions are set to make sure that $\sum_{i=1}^{u_1} \eta_{\boldsymbol{e}_i} = 1$ and $\eta_t = 0$, where $\boldsymbol{t} \in \{0,1\}^{u_1}$ and $\boldsymbol{HW}(\boldsymbol{t}) > 1$.

As mentioned in [1], the conditions imposed in CDA can be classified into three types:

Type 0 conditions only involve bits of IV;
Type 1 conditions involve bits of IV and bits of key;
Type 2 conditions only involve bits of key.

Type 0 conditions can be easily controlled by setting IV bits to certain value. Type 1 conditions can be used to recover part of the key bits. The complexity will increase when imposing conditions of this type because right values of key expressions have to be guessed. Whether the Type 2 condition is satisfied depends on the specific key. If Type 2 conditions are proposed, the attack succeeds in a weak-key setting. In this paper, conditions of IV bits, which are all Type 0 and Type 1 conditions, are derived from given inverse characteristic. So the attack setting of the advanced CDA is the single key.

Then, Algorithm 2 derives IV conditions for the input initial difference and inverse characteristic (Fig. 4).

With the input ($r_{\max}$, $d_{\max}$, $\Delta S_{-r_{\max}}$), the conditions of the characteristic with the max inverse round can be computed by Algorithm 2. If the algorithm failed, attackers should repeat Step 2 and Step 3 to search for the conditions of the characteristic for the max inverse round with no contradictions.

After three steps above, the longest difference characteristic and corresponding conditions of only IV bits are obtained. Indeed, when these conditions are satisfied, the longest difference trail is valid with probability 1.

3.2 Forward technique

Due to difference characteristics obtained by inverse technique hold with probability 1, the forward technique is important to gain a longer difference characteristic with a larger bias of keystream. However, the previous results that contain inverse process on Grain v1 did not pay enough attention on forward technique. For example, in forward process, results of [10] did not set conditions to limit the propagation of difference and results of [9] controlled only one difference of keystream. On the contrary, we are able to control two and three differences of keystream for 114-round and 120-round Grain v1, respectively. As a consequence, advanced CDA can attack Grain v1 to more rounds with larger biases.

In this section, the method of setting conditions is familiar with that of Section 3.1 which the inverse update functions replaced by update functions, while the principle of setting conditions is different. Here, conditions are set to obtain good difference characteristics. The 'good difference characteristics' is defined by the principle that the difference characteristic makes no difference in the internal states as long as possible. This principle has a good performance when it is applied to Grain v1.

In this paper, statistic method is used to estimate the bias of difference of keystream. With 2^{32} keys and IVs randomly chosen, bias of difference of keystream is computed statistically with conditions set to appropriate values. Considering the accuracy of

input : 2^D pairs free IV
output: the right values of key expressions
Set $Counter_j = 0 (0 \le j \le 2^Y - 1)$;
for $i \leftarrow 1$ **to** 2^D **do**
  for $j \leftarrow 0$ **to** $2^Y - 1$ **do**
    for $t \leftarrow 0$ **to** $Y - 1$ **do**
      $f_t(K) \leftarrow j|_t$
    end
    Compute Δz_R with the initial difference ΔS_I under appropriate conditions.
    if $\varepsilon_0 > 0$ *and* $\Delta z_R = 0$ **then**
      $Counter_j \leftarrow Counter_j + 1$;
    else if $\varepsilon_0 < 0$ *and* $\Delta z_R = 1$ **then**
      $Counter_j \leftarrow Counter_j + 1$;
    end
  end
end
$sum_{\max} \leftarrow \max\{Counter_j | 0 \le j \le 2^Y - 1\}$;
if $sum_{\max}/2^D - 0.5 \ge \varepsilon_1$, *where* $0 < \varepsilon_1 < |\varepsilon_0|$ **then**
  Output key expressions corresponding to the sum_{max}.
else
  Judge this is random numbers.
end

Fig. 5 *Algorithm 3: CDA on R-round Grain-like cipher*

Table 3 Some specific parameters of Algorithm 3

ε_0	P_s	ε_1	Y	D
0.000143	93%	0.000100	12	28.75
0.000240	96%	0.000151	6	27
0.000478	96%	0.000301	6	25

experiences, only the keystream bits which absolute value of the biases is larger than 0.0001 are recorded and used to distinguish and recover key expressions.

3.3 Method for recovering key expressions and bias-complexity-success probability target

Suppose the initial difference, imposed conditions and the bias of difference of keystream have already been obtained. Then, the problem is how to distinguish and recover key expressions and how to determine the complexity and success probability. A method is introduced in this section to solve this problem.

The main idea of this method is that randomly choosing enough number of IVs, the statistical bias of keystream for right guess of expressions will be obviously larger than that for wrong guess. So the right guess can be distinguished from wrong guesses. The idea is natural while the key is the relation of bias, complexity and success probability.

Suppose the bias of keystream of R-round Grain-like cipher is ε_0 and the key expressions needed to be guessed are $f_t(K)(0 \le t \le 2^Y - 1)$. The initial difference is denoted by ΔS_I.

A frame work of CDA is presented in Algorithm 3 and parameters of Algorithm 3 are analysed in detail below the algorithm (Fig. 5).

Suppose when the guesses of key expressions are wrong, the difference of R-round keystream obey random distribution. Then, Theorem 1 describes the success probability of Algorithm 3.

Theorem 1: The success probability of Algorithm 3 is denoted by P_s. Then, the inequality below is true

$$P_s \ge [\Phi(\varepsilon_1 \cdot 2^{(D/2)+1})]^{2^Y - 1} \cdot \Phi((\left|\varepsilon_0\right| - \varepsilon_1) \cdot 2^{(D/2)+1}),$$

where $\Phi(x)$ denotes the standard normal distribution

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \mathrm{e}^{-(1/2)u^2} \, \mathrm{d}u \, .$$

Proof: : We only consider the situation $\varepsilon_0 > 0$ and the situation is the same for $\varepsilon_0 < 0$.

According to realistic assumptions and Algorithm 3, the probability of zero difference of keystream for wrong and right guess is $P_w = \frac{1}{2}$ and $P_r = \frac{1}{2} + \varepsilon_0$, respectively. Suppose, in Algorithm 3, the values of counters of the wrong guess and the right guess are C_w and C_r, respectively. As 2^D is large, C_w and C_r approximately obey normal distribution $N(2^D P_w, 2^D P_w(1 - P_w))$ and $N(2^D P_r, 2^D P_r(1 - P_r))$, respectively.

Hence, the probability of one counter of wrong guess that satisfies $C_w/2^D - 0.5 < \varepsilon_1$ is

$$\begin{aligned} P_1 &= \Pr\,(C_w/2^D - 0.5 < \varepsilon_1) = \Pr\left(\frac{C_w - 0.5 \cdot 2^D}{2^D} < \varepsilon_1\right) \\ &= \Pr\left(\frac{C_w - P_w \cdot 2^D}{\sqrt{2^D P_w(1 - P_w)}} < \varepsilon_1 \cdot 2^{(D/2)+1}\right) = \Phi(\varepsilon_1 \cdot 2^{(D/2)+1}) \end{aligned}$$

Supposing C_w of different guesses are independent, the probability of every counter of wrong guess satisfies $C_w/2^D - 0.5 < \varepsilon_1$ is $P_1' = P_1^{2^Y - 1}$.

Then, the probability of the counter of right guess satisfies $C_r/2^D - 0.5 \ge \varepsilon_1$ is

$$\begin{aligned} P_2 &= \Pr\,(C_r/2^D - 0.5 \ge \varepsilon_1) = \Pr\left(\frac{C_r - (0.5 + \varepsilon_0) \cdot 2^D}{2^D} \ge (\varepsilon_1 - \varepsilon_0)\right) \\ &= \Pr\left(\frac{C_w - P_r \cdot 2^D}{\sqrt{2^D \frac{1}{2}(1 - \frac{1}{2})}} \ge (\varepsilon_1 - \varepsilon_0) \cdot 2^{(D/2)+1}\right) \\ &\simeq \Pr\left(\frac{C_w - P_r \cdot 2^D}{\sqrt{2^D P_r \cdot (1 - P_r \cdot)}} \ge (\varepsilon_1 - \varepsilon_0) \cdot 2^{(D/2)+1}\right) \\ &= \Phi((\varepsilon_0 - \varepsilon_1) \cdot 2^{(D/2)+1}) \end{aligned}$$

Hence, the probability of no counter of wrong guess satisfies $C_w/2^D - 0.5 \ge \varepsilon_1$ and at the same time the counter of right guess satisfies $C_r/2^D - 0.5 \ge \varepsilon_1$ is

$$P_0 = P_1' \cdot P_2 = [\Phi(\varepsilon_1 \cdot 2^{(D/2)+1})]^{2^Y - 1} \cdot \Phi((\left|\varepsilon_0\right| - \varepsilon_1) \cdot 2^{(D/2)+1}) \, .$$

When the above situation appears, Algorithm 3 will be success.

Considering that there may exist the situation when some C_w satisfies $C_w/2^D - 0.5 \ge \varepsilon_1$ but every $C_w < C_r$, this probability is not contained by P_0 while Algorithm 3 will be success. So the success probability of Algorithm 3 is no less than $[\Phi(\varepsilon_1 \cdot 2^{(D/2)+1})]^{2^Y - 1} \cdot \Phi((\left|\varepsilon_0\right| - \varepsilon_1) \cdot 2^{(D/2)+1})$. □

In this paper, apposite ε_1 and Y are chosen to obtain a large P_s. Some specific parameters of Algorithm 3 are presented in Table 3.

4 Applications on reduced Grain v1

In this section, advanced CDA is applied to reduced Grain v1. We are able to attack 114-round and 120-round Grain v1. Conditions are imposed in inverse process and forward process. Both of the attacks share the same initial difference and difference characteristic of inverse process. The detail of inverse deduction process is shown below.

The inverse keystream generating function is

$$\begin{aligned} z_{-i} = {} & b_{1-i} \oplus b_{2-i} \oplus b_{4-i} \oplus b_{10-i} \oplus b_{31-i} \oplus b_{43-i} \oplus b_{56-i} \oplus \\ & h(s_{3-i}, s_{25-i}, s_{46-i}, s_{64-i}, b_{63-i}) \end{aligned}$$

where

$$h(s_{3-i}, s_{25-i}, s_{46-i}, s_{64-i}, b_{63-i}) = s_{25-i} \oplus b_{63-i} \oplus s_{3-i}s_{64-i} \oplus s_{46-i}s_{64-i} \oplus s_{64-i}b_{63-i} \oplus s_{3-i}s_{25-i}s_{46-i} \oplus s_{3-i}s_{46-i}s_{64-i} \oplus s_{3-i}s_{46-i}b_{63-i} \oplus s_{25-i}s_{46-i}b_{63-i} \oplus s_{46-i}s_{64-i}b_{63-i}$$

The inverse update function of LFSR is

$$s_{-i} = s_{13-i} \oplus s_{23-i} \oplus s_{38-i} \oplus s_{51-i} \oplus s_{80-i} \oplus z_{-i}.$$

The inverse update function of NFSR is

$$b_{-i} = z_{-i} \oplus s_{-i} \oplus b_{14-i} \oplus b_{62-i} \oplus b_{80-i} \oplus g(b_{9-i}, b_{15-i}, b_{21-i}, b_{28-i}, b_{33-i}, b_{37-i}, b_{45-i}, b_{52-i}b_{60-i}, b_{63-i})$$

where

$$g = b_{9-i} \oplus b_{28-i} \oplus b_{33-i} \oplus b_{37-i} \oplus b_{45-i} \oplus b_{52-i} \oplus b_{60-i} \oplus b_{63-i}b_{60-i} \oplus b_{37-i}b_{33-i} \oplus b_{15-i}b_{9-i} \oplus b_{60-i}b_{52-i}b_{45-i} \oplus b_{33-i}b_{28-i}b_{21-i} \oplus b_{63-i}b_{45-i}b_{28-i}b_{9-i} \oplus b_{60-i}b_{52-i}b_{37-i}b_{33-i} \oplus b_{63-i}b_{60-i}b_{21-i}b_{15-i} \oplus b_{63-i}b_{60-i}b_{52-i}b_{45-i}b_{37-i} \oplus b_{33-i}b_{28-i}b_{21-i}b_{15-i}b_{9-i} \oplus b_{52-i}b_{45-i}b_{37-i}b_{33-i}b_{28-i}b_{21-i}$$

Then, we use $\Delta_\phi - Grain$ to trace the inverse difference propagation. Every single bit difference in LFSR is examined to find the difference position for the longest inverse round with no difference in NFSR. Finally, the difference Δs_{23} is proved to lead to the longest inverse round, which is $r_{\max} = 24$. The detail is as follows.

According to Algorithm 1, the rounds that $\Delta z_{-i} = \bar{2}$ are $i = -2$, -5 and -23 and we set $\Delta z_{-2} = \bar{1}, \Delta z_{-5} = \bar{0}, \Delta z_{-23} = \bar{0}$. The initial difference in LFSR is $\Delta s_{22} = \Delta s_{47} = \bar{1}$.

Then, run Algorithm 2 with the parameters above.

When $i = 1$,

$$\Delta z_1 = \Delta z_{-23} = k_{64} \oplus (k_{64} \oplus v_4) \cdot (k_{64} \oplus v_{26} \oplus 1) \oplus 1 = 0. \quad (3)$$

So the conditions are $J_1 = \{v_4 = 1, v_{26} = 0\}$.

When $i = 19$

$$\Delta z_{19} = \Delta z_{-5} = v_2 \oplus v_{27} \oplus v_{44} \oplus v_5 \oplus f_1(K) \oplus v_{48} \cdot [k_{65} \oplus 1 \oplus (k_{65} \oplus v_5) \cdot (k_{65} \oplus v_{27} \oplus 1)] = 0$$

where $f_1(K)$ is a function for only key bits.

Hence, the conditions are

$$J_{19} = \{v_{48} = 0, v_2 \oplus v_{27} \oplus v_{44} \oplus v_5 \oplus f_1(K) = 0\}. \quad (4)$$

When $i = 22$

$$\Delta z_{22} = \Delta z_{-2} = v_{25} \oplus v_{30} \oplus v_5 \oplus v_8 \oplus f_2(K) \oplus v_{51} \cdot [k_{68} \oplus 1 \oplus (k_{68} \oplus v_8) \cdot (k_{68} \oplus v_{30} \oplus 1)] = 1, \quad (5)$$

where $f_2(K)$ is a function for only key bits.

Hence, the conditions are

$$J_{22} = \{v_{51} = 0, v_{25} \oplus v_{30} \oplus v_5 \oplus v_8 \oplus f_2(K) \oplus 1 = 0\}. \quad (6)$$

Above all, the conditions leading to the longest differential characteristic are

$$J = \{v_4 = 1, v_{26} = 0, v_{48} = 0, v_{51} = 0, v_2 \oplus v_{27} \oplus v_{44} \oplus v_5 \oplus f_1(K) = 0, v_{25} \oplus v_{30} \oplus v_5 \oplus v_8 \oplus f_2(K) \oplus 1 = 0\},$$

which are four Type 0 conditions and two Type 1 conditions.

The propagations of differences are shown in Table 4.

Then, using forward technique from 24th round on, several Δz_i are set to zero and the keystream bits are inspected to search for an obvious bias for as long rounds as possible. With 2^{32} keys and IVs randomly chosen, experiments show the results below:

$$\Pr(\Delta z_{114} = 0 | J \text{ is satisfied} \& \Delta z_{40} = 0 \& \Delta z_{44} = 0) = 0.500478$$

$$\Pr(\Delta z_{120} = 0 | J \text{ is satisfied} \& \Delta z_{40} = 0 \& \Delta z_{44} = 0 \& \Delta z_{50} = 0) = 0.500143$$

The two probabilities are the bases of the CDA on Grain v1 of 114 rounds and 120 rounds. The details of the two attacks are presented below.

Table 4 Propagations of differences under conditions

Round(i)	Conditions	Differences with restriction	Differences without restriction
0	∅	$\Delta s_{22}, \Delta s_{47}$	$\Delta s_{22}, \Delta s_{47}$
1	J_1	$\Delta s_{21}, \Delta s_{46}$	$\Delta s_{21}, \Delta s_{46}$
2	∅	$\Delta s_{20}, \Delta s_{45}$	$\Delta b_{79}, \Delta s_{20}, \Delta s_{45}, \Delta s_{79}$
3–18	∅	$\Delta s_{22-i}, \Delta s_{47-i}$	[a]
19	J_{19}	$\Delta s_3, \Delta s_{28}$	$\Delta s_3, \Delta s_{28}$
20	∅	$\Delta s_2, \Delta s_{27}$	$\Delta b_{79}, \Delta s_2, \Delta s_{27}, \Delta s_{79}$
21	∅	$\Delta s_1, \Delta s_{26}$	[b]
22	J_{22}	$\Delta s_0, \Delta s_{25}$	$\Delta s_0, \Delta s_{25}$
23	∅	Δs_{24}	$\Delta b_{79}, \Delta s_{24}, \Delta s_{79}$
24	∅	Δs_{23}	$\Delta b_{78}, \Delta s_{23}, \Delta s_{78}$

[a] We suppose J_1 is satisfied and just show the conditional effect of J_{19}.

[b] We suppose J_1, J_{19} are satisfied and just show the conditional effect of J_{22}.

4.1 Advanced CDA on 114-round Grain v1

In this section, conditions are set to impose $\Delta z_{40} = 0$ and $\Delta z_{44} = 0$.

For $i = 40$: The expression of Δz_{40} is $F_1 \oplus F_2 \oplus F_1 \cdot F_2$, where F_1 and F_2 are functions of key and IV bits. In order to eliminate Δz_{40}, conditions are set to impose $F_1 = 0$ and $F_2 = 0$. So the conditions are

$$J_{40} = \{v_{49} = 0, v_{52} = 0, v_{53} = 0, v_3 \oplus v_6 \oplus v_{28} = 0, v_6 \oplus v_9 \oplus v_{31} = 0, v_{19} \oplus v_{29} \oplus v_{44} \oplus k_{10} \oplus k_{16} \oplus k_{37} \oplus k_{49} \oplus k_{62} \oplus k_7 \oplus k_8 \oplus 1 = 0 v_{20} \oplus v_{10} \oplus v_{30} \oplus v_{32} \oplus v_{45} \oplus v_7 = 0, v_{43} \oplus v_{23} \oplus v_{25} \oplus f_3(K) = 0\}$$

where $f_3(K)$ is a function of only key bits.

For $i = 44$: The expression of Δz_{44} is $F_3 \oplus F_4 \cdot F_5$, where F_3, F_4 and F_5 are functions of key and IV bits. In order to eliminate Δz_{44}, conditions are set to impose $F_3 = 0$ and $F_4 = 0$. So the conditions are

$$J_{44} = \{v_{31} = 0, v_{50} = 0, v_{56} = 0, v_{57} = 0, v_{58} = 0, v_{11} \oplus v_{14} \oplus v_{36} = 0, v_{63} \oplus v_{12} \oplus v_{15} \oplus v_{25} \oplus v_{35} \oplus v_{37} = 0, v_{41} \oplus v_{28} \oplus v_{29} \oplus v_7 \oplus f_4(K) = 0, v_{33} \oplus v_{10} \oplus v_{13} \oplus v_{23} \oplus v_{35} \oplus v_{61} \oplus k_{11} \oplus k_{12} \oplus k_{14} \oplus k_{20} \oplus k_{41} \oplus k_{53} \oplus k_{66} \oplus 1 = 0\}$$

where $f_4(K)$ is a function of only key bits.

Rearrange J, J_{40} and J_{44}. All conditions are shown as follows:

$$J^1 = \{v_4 = 1, v_9 = v_6, v_{11} = v_{14} \oplus v_{36}, v_{26} = 0, v_{20} = v_{10} \oplus v_{30} \oplus v_{32} \oplus v_{45} \oplus v_7, v_{28} = v_3 + v_6, v_{31} = 0, v_{48} = 0, v_{49} = 0, v_{50} = 0, v_{52} = 0, v_{51} = 0, v_{53} = 0, v_{56} = 0, v_{57} = 0, v_{58} = 0, v_{63} = v_{12} \oplus v_{15} \oplus v_{25} \oplus v_{35} \oplus v_{37}, v_2 = v_{44} \oplus v_{27} \oplus v_5 \oplus f_1(K), v_8 = v_{30} \oplus v_5 \oplus v_{25} \oplus f_2(K), v_{43} = v_{23} \oplus v_{25} \oplus f_3(K), v_{41} = v_{28} \oplus v_{29} \oplus v_7 \oplus f_4(K), v_{19} = v_{29} \oplus v_{44} \oplus f_5(K), v_{33} = v_{10} \oplus v_{13} \oplus v_{23} \oplus v_{35} \oplus v_{61} \oplus f_6(K)\} \quad (7)$$

where

$$f_5(K) = k_{10} \oplus k_{16} \oplus k_{37} \oplus k_{49} \oplus k_{62} \oplus k_7 \oplus k_8 \oplus 1, \quad (8)$$

$$f_6(K) = k_{11} \oplus k_{12} \oplus k_{14} \oplus k_{20} \oplus k_{41} \oplus k_{53} \oplus k_{66} \oplus 1 \quad (9)$$

and other f_j are non-linear polynomials of key bits.

There are 17 Type 0 conditions and 6 Type 1 conditions. The number of free IV bits is 39. Then, with the parameters $R = 114$, $Y = 6$, $D = 25$, $\varepsilon_0 = 0.000478$, $\varepsilon_1 = 0.000301$ of Algorithm 3, the CDA on 114-rounds Grain v1 could recover six key expressions with success probability of about 96%. The time complexity and data complexity of this attack is $2^{25} \cdot 2^6 \cdot 2 = O(2^{32})$ and $O(2^{26})$, respectively. The memory complexity of this attack is negligible.

For 100 random keys, experiments successfully recover all the key expressions for 97 times, which is in good agreement with Theorem 1. The experiments are running in the platform of C++ on Computer of Intel Corei5-3470 CPU @ 3.20 GHz with Windows7 OS and it takes about 17.6 min to complete for each attack.

4.2 Advanced CDA on 120-round Grain v1

With J_{40} and J_{44} satisfied, other conditions are set to impose $\Delta z_{50} = 0$.

For $i = 50$: The expression of Δz_{50} is $F_6 \oplus F_7 \oplus F_6 \cdot F_7$, where F_6 and F_7 are functions of key and IV bits. In order to eliminate Δz_{50}, conditions are set to impose $F_6 = 0$ and $F_7 = 0$. So the conditions are

$$\begin{aligned} J_{50} = \{ & v_{19} = 0, v_{25} = 0, v_{46} = 0, v_{41} = 0, v_{59} = 0, v_{63} = 0, \\ & v_{55} \cdot (v_{12} \oplus v_{34} \oplus 1) = 0, v_{61} \cdot (v_{18} \oplus v_{40} \oplus 1) = 0, \\ & v_{52} \cdot (v_9 \oplus 1) = 0, v_0 \oplus v_{13} \oplus v_{23} \oplus v_3 \oplus v_{38} \oplus v_{62} \oplus k_1 \\ & \oplus k_{10} \oplus k_2 \oplus k_{31} \oplus k_4 \oplus k_{43} \oplus k_{56} \oplus k_{79} = 0, v_{16} \oplus v_{29} \\ & \oplus v_{39} \oplus v_{54} \oplus k_{17} \oplus k_{18} \oplus k_{20} \oplus k_{26} \oplus k_{47} \oplus k_{59} \oplus k_{72} = 0 \quad (10) \\ & v_{16} \oplus f_8(K) = 0, v_1 \oplus v_{14} \oplus v_{24} \oplus v_{39} \oplus v_{52} \oplus f_9(K) = 0, \\ & v_0 \oplus v_{12} \oplus v_{15} \oplus v_{17} \oplus v_{18} \oplus v_{20} \oplus v_{23} \oplus v_3 \oplus v_{30} \oplus v_{33} \\ & \oplus v_{34} \oplus v_{42} \oplus v_{55} \oplus v_{62} \oplus v_9 \oplus v_{55} \cdot (v_{12} \cdot v_{34} \oplus v_{12} \oplus 1) \oplus \\ & v_{61} \cdot (v_{18} \cdot v_{40} \oplus v_{18} \oplus 1) \oplus f_{10}(k) \} \end{aligned}$$

Rearrange J, J_{40}, J_{44} and J_{50} and modify some conditions to obtain more free IVs. All conditions are shown in the following equation:

$$\begin{aligned} J^2 = \{ & v_4 = 1, v_9 = v_6, v_{19} = 0, v_{25} = 0, v_{26} = 0, v_{28} = v_3 + v_6, v_{31} = 0, \\ & v_{41} = 0, v_{46} = 0, v_{48} = 0, v_{49} = 0, v_{50} = 0, v_{51} = 0, v_{53} = 0, \\ & v_{56} = 0, v_{57} = 0, v_{58} = 0, v_{59} = 0, v_{63} = 0, v_{45} = v_{10} \oplus v_{30} \\ & \oplus v_{32} \oplus v_{20} \oplus v_7, v_{52} \cdot (v_9 \oplus 1) = 0, v_{55} \cdot (v_{12} \oplus v_{34} \oplus 1) = 0, \\ & v_{61} \cdot (v_{18} \oplus v_{40} \oplus 1) = 0, v_2 = v_{44} \oplus v_{27} \oplus v_5 \oplus g_1(K), v_8 = v_{30} \\ & \oplus v_5 \oplus g_2(K), v_{44} = v_{29} \oplus g_3(K), v_{43} = v_{23} \oplus g_4(K), \\ & v_{33} = v_{10} \oplus v_{13} \oplus v_{23} \oplus v_{35} \oplus v_{61} \oplus g_5(K), v_{11} = v_{12} \oplus v_{14} \oplus v_{15} \\ & \oplus v_{35} \oplus v_{36} \oplus v_{37} \oplus g_6(K) = 0, v_7 = v_{28} \oplus v_{29} \oplus g_7(K), \\ & v_{13} = v_0 \oplus v_{23} \oplus v_3 \oplus v_{38} \oplus v_{62} \oplus g_8(K), v_{29} = v_{16} \oplus v_{39} \\ & \oplus v_{54} \oplus g_9(K), v_{16} = g_{10}(K), v_1 = v_{14} \oplus v_{24} \oplus v_{39} \oplus v_{52} \oplus g_{11}(K), \\ & v_{17} = v_0 \oplus v_{12} \oplus v_{15} \oplus v_{18} \oplus v_{20} \oplus v_{23} \oplus v_3 \oplus v_{30} \oplus v_{33} \\ & \oplus v_{34} \oplus v_{42} \oplus v_{55} \oplus v_{62} \oplus v_9 \oplus v_{55} \cdot (v_{12} \cdot v_{34} \oplus v_{12} \oplus 1) \\ & \oplus v_{61} \cdot (v_{18} \cdot v_{40} \oplus v_{18} \oplus 1) \oplus g_{12}(k) \} \end{aligned} \quad (11)$$

where

$$\begin{aligned} g_3(K) &= k_{10} \oplus k_{16} \oplus k_{37} \oplus k_{49} \oplus k_{62} \oplus k_7 \oplus k_8 \oplus 1, \\ g_5(K) &= k_{11} \oplus k_{12} \oplus k_{14} \oplus k_{20} \oplus k_{41} \oplus k_{53} \oplus k_{66} \oplus 1, \\ g_8(K) &= k_1 \oplus k_{10} \oplus k_2 \oplus k_{31} \oplus k_4 \oplus k_{43} \oplus k_{56} \oplus k_{79} = 0, \quad (12) \\ g_9(K) &= k_{17} \oplus k_{18} \oplus k_{20} \oplus k_{26} \oplus k_{47} \oplus k_{59} \oplus k_{72}, \\ g_{11}(K) &= k_{11} \oplus k_2 \oplus k_3 \oplus k_{32} \oplus k_{44} \oplus k_5 \oplus k_{57} \end{aligned}$$

and other g_j are non-linear polynomials of key bits.

There are 23 Type 0 conditions and 12 Type 1 conditions. There are 3 special Type 0 conditions which tenable probability is 0.75, so the number of free IV bits is 28.75. Then, we substitute the parameters $R = 120$, $Y = 12$, $D = 28.75$, $\varepsilon_0 = 0.000143$, $\varepsilon_1 = 0.0001$ into Algorithm 3. The advanced CDA on 120-rounds Grain v1 could recover 12 key expressions with theoretical success probability of about 93%. The time complexity and data complexity of this attack is $2^{28.75} \cdot 2^{12} \cdot 2 = O(2^{42.75})$ and $O(2^{29.75})$, respectively. The memory complexity of this attack is negligible.

The experimental complexity of full attack is beyond our capability. However, we can partly verify our attack through experiments. For 100 random keys and $2^{28.75}$ pairs of chosen IVs, with the Type 2 conditions set to right values, experiments show that about 97% keys pass the test. Considering when the Type 2 conditions set to wrong values, the difference of keystream is random. Hence, the probability that all of the wrong guesses cannot pass the test is about 95.9%. The success probability of full attack is about 93%.

5 Conclusion

In this paper, we propose an advanced CDA on Grain-like stream cipher. Propagations of a single bit difference of internal states are traced both inversely and forward. Methods of searching for the longest inverse difference characteristic with probability one and deriving IV conditions for the characteristic with the max inverse round are introduced. When tracing forward, conditions are imposed to limit the propagation of difference to obtain a high bias of keystream. Unlike previous attacks of weak-key or the related-key setting, our attack works in the single-key setting. Moreover, a method for recovering key expressions as well as bias-complexity-success probability target is presented in this paper.

Applying our method, a distinguishing and key recovery attack is conducted on 114-round Grain v1 with a high success probability and verified by experiments. With more conditions imposed, this attack can be improved to Grain v1 of 120 rounds. Based on the advanced CDAs in Sections 4.1 and 4.2, the recoveries of the full keys on Grain v1 with 114 and 120 initialisation rounds are faster than exhaustive searches by a factor of 2^6 and 2^{12}. Due to the unobvious bias of keystream and the lack of free IV bits, we think the full Grain v1 could resist advanced CDA. Hopefully, advanced CDA could be combined with other techniques and used to attack Grain v1 for more rounds and even other NFSR based cryptosystems, which is one of our future work.

6 Acknowledgments

This work is supported by the National Natural Science Foundations of China under grant nos. 61572516, 61602514 and 61272488.

7 References

[1] Knellwolf, S., Meier, W., Naya-Plasencia, M.: 'Conditional differential cryptanalysis of NLFSR-based cryptosystems'. Advances in Cryptology – ASIACRYPT 2010, Singapore, 5–9 December 2010, pp. 130–145

[2] Knellwolf, S., Meier, W., Naya-Plasencia, M.: 'Conditional differential cryptanalysis of Trivium and KATAN'. Int. Workshop on Selected Areas in Cryptography (SAC 2011), Toronto, ON, Canada, 11–12 August 2011, pp. 200–212

[3] Banik, S.: 'Some insights into differential cryptanalysis of Grain v1'. Australasian Conf. on Information Security and Privacy (ACISP 2014), Wollongong, NSW, Australia, 7–9 July 2014, pp. 34–49

[4] Sarkar, S.: 'A new distinguisher on Grain v1 for 106 rounds'. Int. Conf. on Information Systems Security (ICISS 2015), Kolkata, India, 16–20 December 2015, pp. 334–344

[5] Ma, Z., Tian, T., Qi, W.F.: 'Improved conditional differential attacks on Grain v1', *IET Inf. Sec.*, 2017, **11**, (1), pp. 46–53
[6] Banik, S.: 'Dynamic cube attack on 105 round Grain v1', *Appl. Stat.*, 2014, **34**, (2), pp. 49–50
[7] Ma, Z., Tian, T., Qi, W. F.: 'Conditional differential attacks on Grain-128a stream cipher', *IET Inf. Sec.*, 2017, **11**, (3), pp. 139–145
[8] Zhang, K., Guan, J., Fei, X.: 'Improved conditional differential cryptanalysis', *Secur. Commun. Netw.*, 2015, **8**, (9), pp. 1801–1811
[9] Knellwolf, S.: 'Cryptanalysis of hardware-oriented ciphers the Knapsack generator, and SHA-1'. PhD dissertation, ETH Zurich, 2012
[10] 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
[11] Hell, M., Johansson, T., Meier, W.: 'Grain: a stream cipher for constrained environments', *Int. J. Wireless Mob. Comput.*, 2007, **2**, (1), pp. 86–93
[12] Hell, M., Johansson, T., Maximov, A.*, et al.*: 'A stream cipher proposal: Grain-128'. 2006 IEEE Int. Symp. on Information Theory, Seattle, WA, USA, 2006, pp. 1614–1618
[13] Gren, M., Hell, M., Johansson, T.*, et al.*: 'Grain-128a: a new version of Grain-128 with optional authentication', *Int. J. Wirel. Mob. Comput.*, 2011, **5**, (1), pp. 48–59
[14] Robshaw, M.: 'The eSTREAM project', in Robshaw, M., Billet, O. (Eds.): '*New stream cipher designs: the eSTREAM finalists*' (Springer Berlin Heidelberg, Berlin, Heidelberg, 2008), pp. 1–6
[15] Rahimi, M., Barmshory, M., Mansouri, M.H.*, et al.*: 'Dynamic cube attack on Grain-v1', *IET Inf. Sec.*, 2013, **10**, (4), pp. 165–172
[16] 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 (ACISP 2008), Wollongong, Australia, 7–9 July 2008, pp. 321–335
[17] Zhang, B., Li, Z., Feng, D.*, et al.*: 'Near collision attack on the Grain v1 stream cipher'. Int. Workshop on Fast Software Encryption (FSE 2013), Singapore, 11–13 March 2013, pp. 518–538
[18] Banik, S., Maitra, S., Sarkar, S.: 'A differential fault attack on the grain family of stream ciphers'. CHES 2012: 14th Int. Workshop, Leuven, Belgium, 9–12 September 2012, pp. 122–139
[19] Sarkar, S., Banik, S., Maitra, S.: 'Differential fault attack against grain family with very few faults and minimal assumptions', *IEEE Trans. Comput.*, 2015, **64**, (6), pp. 1647–1657
[20] Mihaljević, M.J., Gangopadhyay, S., Paul, G.*, et al.*: 'Internal state recovery of Grain-v1 employing normality order of the filter function', *IET Inf. Sec.*, 2012, **6**, (2), pp. 55–64
[21] Zhen, M., Tian, T., Wenfeng, Q.: 'Internal state recovery of Grain v1 employing guess-and-determine attack', *IET Inf. Sec.*, 2017, **11**, (6), pp. 363–368
[22] Zhang, K.: 'Research on the security evaluation against mixed operation based cipher model'. PhD dissertation, Information Engineering University, 2016