Improved Conditional Differential Attacks on Round-Reduced Grain v1

Jun-Zhi Li* and Jie Guan
Zhengzhou Institute of Information Science and Technology
Zhengzhou, Henan 450001 - China
[e-mail: lijunzhi1998@163.com]
*Corresponding author: Jun-Zhi Li

Received October 6, 2017; revised December 6, 2017; accepted April 2, 2018; published September 30, 2018

Abstract

Conditional differential attack against NFSR-based cryptosystems proposed by Knellwolf *et al.* in Asiacrypt 2010 has been widely used for analyzing round-reduced Grain v1. In this paper, we present improved conditional differential attacks on Grain v1 based on a factorization simplification method, which makes it possible to obtain the expressions of internal states in more rounds and analyze the expressions more precisely. Following a condition-imposing strategy that saves more IV bits, Sarkar's distinguishing attack on Grain v1 of 106 rounds is improved to a key recovery attack. Moreover, we show new distinguishing attack and key recovery attack on Grain v1 of 107 rounds with lower complexity $O(2^{34})$ and appreciable theoretical success probability 93.7%. Most importantly, our attacks can practically recover key expressions with higher success probability than theoretical results.

Keywords: Conditional differential attack, Grain v1, distinguisher, key recovery attack, stream cipher

This work is supported by the National Natural Science Foundations of China under grant NO. 61572516 and 61272488.

http://doi.org/10.3837/tiis.2018.09.023 **ISSN : 1976-7277**

1. Introduction

Grain v1 [1] is one of eSTREAM [2] hardware-oriented finalists proposed by Hell, Johansson and Meier in 2005. It is a bit-oriented cipher designed for constrained hardware environments. It uses an 80-bit secret key and a 64-bit initial value (IV) and has a compact structure. Grain v1 consists of an 80-bit non-linear feedback shift register, an 80-bit linear feedback shift register (LFSR) and a filter function that combined the two registers. Several kinds of cryptanalysis against Grain v1 have been proposed so far such as conditional differential attack (CDA) [3-8], related-key chosen IV attack [9], near collision attack [10] and differential fault attack [11, 12]. In this paper, we focus on CDA.

CDA against NFSR-based cryptosystems was introduced by Knellwolf *et al.* in Asiacrypt 2010[3]. In CDA, attackers introduce a difference on initial state and track the difference of internal states. After setting some conditions on initial states to limit the propagation of difference in internal states, there will be a detectable unbalance in the keystream. Then the cryptosystem can be distinguished from random numbers. Once the conditions involve key expressions, these conditions can be used to recover key information. This method has been successfully used to analyze Grain v1 [3], Trivium [13] and KATAN [13].

CDA has been widely used for analyzing round-reduced Grain v1. In [3], Knellwolf *et al.* gave distinguishing attack and key recovery results for 97 rounds and extended the attack to 104 rounds with time complexity 2^{35} and advantage of about 0.45. However, the results of [3] were only based on experiments. In [4], a theoretical framework was provided to prove the biases of 97 rounds. Later, Banik [8] gave CDA results on Grain v1 of 105 rounds and recovered six key expressions and nine key bits extracted from the expressions with time complexity about $2^{37.2}$. Recently, a new distinguisher of 106-round Grain v1 has been presented with time complexity 2^{30} and advantage of about 0.63 in [5]. However, this work could not recover key expressions due to the lack of free IV variables. In [6], Ma *et al.* improved CDA on Grain v1 of 107 rounds with time complexity 2^{42} and 110 rounds with time complexity 2^{47} based on new difference-searching and IV-saving condition-imposing strategies. As they highly focused on saving IV, it needs to guess more key expressions, leading to the higher complexity. In [7], CDA on Grain v1 up to 114 rounds in 2^{40} weak key subspace has been studied.

In this paper, we present improved CDA on Grain v1 based on a factorization simplification method. This method makes it possible to obtain the expressions of internal states in more rounds and analyze the expressions more precisely. Then a meticulous analysis conducted on the expressions enables us to obtain the condition with more IV bits saved. Using our proposed factorization simplification method and condition-imposing strategy, Sarkar's distinguishing attack of 106 rounds on Grain v1 is improved to a key recovery attack. Moreover, we present new distinguishing attack and key recovery attack on Grain v1 to 107 rounds with lower complexity $O(2^{34})$ and appreciable theoretical success probability 93.7%. Experiments verify that our attack can recover key expressions practically with a high success probability.

This paper is organized as follows. Section 2 presents the description of Grain v1 and some notes. Our proposed factorization simplification method and condition-imposing strategy are introduced in Section 3. Section 4 gives our attacks on Grain v1 of 106 and 107 rounds. Then Section 5 conclude our paper.

2. Description of Grain v1

Grain v1 uses an 80-bit secret key and a 64-bit initial value (IV). Grain v1 consists of an 80-bit NFSR, an 80-bit LFSR and a filter function combined the two registers. The state of NFSR is denoted by $n_0,\ldots,n_{79}$ and that of LFSR by $l_0,\ldots,l_{79}$. The registers are filled by Key and IV bits as follows.

$$n_i = k_i \text{ for } (0 \le i \le 79)$$
$$l_i = iv_i \text{ for } (0 \le i \le 63)$$
$$l_i = 1 \text{ for } (64 \le i \le 79)$$

The update function of LFSR is $l_{i+80} = l_i + l_{i+13} + l_{i+23} + l_{i+38} + l_{i+51} + l_{i+62}$ and the update function of NFSR is

$$\begin{aligned} n_{i+80} &= g(l_i, n_i, n_{i+9}, n_{i+14}, n_{i+15}, n_{i+21}, n_{i+28}, n_{i+33}, n_{i+37}, n_{i+45}, n_{i+52}, n_{i+60}, n_{i+62}, n_{i+63}) \\ &= l_i + n_i + n_{i+9} + n_{i+14} + n_{i+21} + n_{i+28} + n_{i+33} + n_{i+37} + n_{i+45} + n_{i+52} \\ &\quad + n_{i+60} + n_{i+62} + n_{i+63}n_{i+60} + n_{i+37}n_{i+33} + n_{i+15}n_{i+9} + n_{i+60}n_{i+52}n_{i+45} \\ &\quad + n_{i+33}n_{i+28}n_{i+21} + n_{i+63}n_{i+45}n_{i+28}n_{i+9} + n_{i+60}n_{i+52}n_{i+37}n_{i+33} \\ &\quad + n_{i+63}n_{i+60}n_{i+21}n_{i+15} + n_{i+63}n_{i+60}n_{i+52}n_{i+45}n_{i+37} + n_{i+33}n_{i+28}n_{i+21}n_{i+15}n_{i+9} \\ &\quad + n_{i+52}n_{i+45}n_{i+37}n_{i+33}n_{i+28}n_{i+21} \end{aligned} \tag{1}$$

The keystream generating function is

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

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

$$\begin{aligned} h(l_{i+3}, l_{i+25}, l_{i+46}, l_{i+64}, n_{i+63}) &= l_{i+25} + n_{i+63} + l_{i+3}l_{i+64} + l_{i+46}l_{i+64} + l_{i+64}n_{i+63} + l_{i+3}l_{i+25}l_{i+46} \\ &\quad + l_{i+3}l_{i+46}l_{i+64} + l_{i+3}l_{i+46}n_{i+63} + l_{i+25}l_{i+46}n_{i+63} + l_{i+46}l_{i+64}n_{i+63} \end{aligned} \tag{2}$$

The cipher outputs keystream after the initialization phase. During the initialization phase, the cipher is clocked 160 times without outputting keystream. Instead, the output z_i is xored feedback in both LFSR and NFSR.

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.

3. Improvement of CDA on Grain v1

This section presents techniques used to improve CDA on Grain v1.

3.1 Factorization simplification method

In CDA, it requires distinct expressions of internal states or keystream to impose right conditions to control the propagation of difference. Usually, the internal states expressions are iteratively computed over $GF(2)$. During this process, the algebraic complexity of the expressions increases rapidly with each increasing round. Therefore, it is hard to obtain expressions after a few clocks, not even to analyze them carefully. Herein, we propose a factorization simplification method to improve this situation partially.

As with most software, the storage space always determines whether a polynomial could be computed. At the same time, the storage complexity of computing a polynomial mainly depends on the total number of algebraic symbols used to express the algebraic expression of the polynomial over $GF(2)$. If this number could decrease, a reasonable result is that polynomials of internal states of more rounds can be obtained.

The total number of algebraic symbols used to express the algebraic expression of polynomial f over $GF(2)$ is denoted by T_f. In the following paragraphs, polynomial f is complicated means that T_f is large. The main idea of factorization simplification method is to decrease T_f. Main steps of factorization simplification method are presented as follows:

Step1: Simplify the update function.

Let update function f be a complicated polynomial for key bits and IV bits and f could be factored as

$$f = f_1 \cdot f_2 + f_3,$$

Where f_1, f_2 and f_3 are simpler than f. Then, a compact form is used to compute and store instead of a complicated one. This operation on Grain v1 has an evident effect as Example 1.

Example 1. For the update function *g* and the filter function *h* of Grain v1, it is clear that T_g =52, T_h =23.

The update function of NFSR *g* and filter function *h* are factored as follows:

$$\begin{aligned} n_{i+80} &= g'(l_i, n_i, n_{i+9}, n_{i+14}, n_{i+15}, n_{i+21}, n_{i+28}, n_{i+33}, n_{i+37}, n_{i+45}, n_{i+52}, n_{i+60}, n_{i+62}, n_{i+63}) \\ &= l_i + n_i + n_{i+14} + n_{i+21} + n_{i+28} + n_{i+33} + n_{i+37} + n_{i+52} + n_{i+60} + n_{i+62} \\ &+ n_{i+63}n_{i+60}\left(1 + n_{i+21}n_{i+15} + n_{i+52}n_{i+45}n_{i+37}\right) + n_{i+9}\left(1 + n_{i+15} + n_{i+63}n_{i+45}n_{i+28}\right) \\ &+ \left(n_{i+45} + n_{i+33}n_{i+37}\right)\left(1 + n_{i+60}n_{i+52}\right) + n_{i+33}n_{i+28}n_{i+21}\left(1 + n_{i+15}n_{i+9} + n_{i+52}n_{i+45}n_{i+37}\right) \end{aligned} \tag{3}$$

$$\begin{aligned} h'(l_{i+3}, l_{i+25}, l_{i+46}, l_{i+64}, n_{i+63}) &= l_{i+3}l_{i+64} + l_{i+25}l_{i+46} + (l_{i+3} + n_{i+63})(l_{i+25} + l_{i+64}n_{i+63}) \\ &+ (l_{i+64} + n_{i+63})(l_{i+46} + l_{i+64} + 1) \end{aligned} \tag{4}$$

After factorization simplification, $T_{g'}$ =35 and $T_{h'}$ =13.

T_g has decreased by about 33% after factorization (52 to 35) and T_h has decreased by about 50% (23 to 13).

Considering that after few clocks, hundreds of these functions are used to compute expressions, the effect will be more significant.

Step2: Keep the compactness of expressions round by round.

The general method for computing the expressions iteratively is to substitute variables with an un-simplified polynomial and expand them over $GF(2)$ iteratively. Unlike the previous method, we substitute variables with their simplified expressions for key and IV instead of a complicated polynomial. Meanwhile, the compactness of expressions is kept round by round without expanding them over $GF(2)$. On the one hand, this technique makes it easier to analyze the expressions and impose simplicity conditions. On the other hand, it really saves time and storage when computing expressions of internal states for large rounds.

Step3: Factor the polynomials further.

After the target expressions are obtained, they are further factored to the form $f = f_1 \cdot f_2 + f_3$. Then, expressions can be analyzed carefully for imposing appropriate conditions. Different attackers may obtain different factorization results, but the main purpose is making it easy to impose conditions.

When applying factorization simplification method on Grain v1, the expressions of at least 55 rounds could be obtained, which are fifteen rounds more than [5] and ten rounds more than [8].

3.2 Proposed condition-imposing strategy

If the target expression is factored to the form $f = f_1 \cdot f_2 + f_3$, the next step is to impose appropriate conditions in CDA for eliminating f. Compared to eliminating f directly, the elimination of f_1, f_2 and f_3 is always easier and takes less condition IV bits. There are two cases according to f_3.

If f_3=1, then set: f_1= f_2= 1;

Otherwise, set: f_3=0 and the simpler one in f_1 and f_2 to 0.

This process can be repeated when certain polynomials are too complicated to be eliminated.

As mentioned in [3], 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 could be controlled by setting IV bits to certain value. Type 1 conditions can be used to recover the key information. Complexity will increase when imposing this condition due to the guessing of the right values of key expressions. Whether the Type 2 condition is satisfied depends on the specific key. If Type 2 conditions are proposed, the attack succeeds under a weak key hypothesis. Fortunately, our attack on Grain v1 only contains conditions of Type 0 and Type 1.

With the number of Type 1 conditions increasing, the bias keeps unchanged while the complexity increases exponentially under the same differential conditions.

Our strategy is to **minimize the number of Type 1 conditions while keeping the whole number of conditions as small as possible**. Based on the method of Section 3.1, this strategy can be applied on Grain v1 of 106 and 107 rounds successfully. This enables us to improve the distinguishing attack of 106-round Grain v1 in [5] to a key recovery attack. We also perform distinguishing and key recovery attacks on 107-round Grain v1.

Our strategy is contrast to that of [6], which is to maximize the number of Type 1 conditions while keeping total number of conditions as small as possible. Consequently, our distinguishing attack takes lower complexity but can recover less key expressions.

4. CDAs on Round-Reduced Grain v1

This section provides our attacks on 106-round and 107-round Grain v1. A single bit difference is set on IV of position *i*, which denoted by $\Delta v_i = 1$. Suppose Δz_j is the difference of *j*-th keystream bit. In [5], Sarkar experimentally found that if Δv_{62} =1, then

$$\Pr(\Delta z_{106} = 0 \mid \Delta z_{16} = 0 \,\&\, \Delta z_{34} = 0 \,\&\, \Delta z_{37} = 0 \,\&\, \Delta z_{40} = 0) = 0.500245 .$$

Analogously, if Δv_{63} =1, then

$$\Pr(\Delta z_{107} = 0 \mid \Delta z_{17} = 0 \,\&\, \Delta z_{35} = 0 \,\&\, \Delta z_{38} = 0 \,\&\, \Delta z_{41} = 0) = 0.500246 .$$

In this paper, the differential characteristics described above are used to perform CDAs on 106-round and 107-round of Grain v1. The details of attacks are presented as follows.

4.1 CDA on 106-round Grain v1

The expressions of z_i are computed literally with a compact form using the factorization simplification method introduced in Section 3.1. If Δz_i must be set to zero, we compute the coefficient of v_{62} in z_i and impose Type 0 or Type 1 conditions to eliminate the coefficient. The attacking steps are as follows:

Step 1. For $i=16$, the coefficient of v_{62} in z_{16} is the polynomial below:

$$\begin{aligned}
&k_1+k_{10}+k_2+k_{31}+k_4+k_{43}+k_{56}+k_{79}+v_0+v_{13}+v_{23}+v_{25}+v_3\\
&+v_{38}+v_{51}+v_{46}\cdot(k_{63}+1)+v_{46}\cdot(k_{63}+v_3)\cdot(k_{63}+v_{25}+1)+1+\\
&(k_{79}+v_{19})\cdot(k_1+k_{10}+k_2+k_{31}+k_4+k_{43}+k_{56}+k_{79}+v_0+v_{13}+v_{23}+v_{25}+v_3\\
&+v_{38}+v_{51}+v_{46}\cdot(k_{63}+1)+v_{46}\cdot(k_{63}+v_3)\cdot(k_{63}+v_{25}+1)+v_{41})
\end{aligned}\tag{5}$$

It is clear that if $k_{79}+v_{19}=1$ and $v_{41}=1$, the value of formula (5) will be zero. So conditions are set as below:

Type 0 condition:

$$v_{41}=1,$$

Type 1 condition:

$$v_{19}=k_{79}+1.$$

This saves two IV bits comparing with [5] to achieve the same target, which highlights the advantage of our factorization simplification method.

Step 2. For $i=34$, the coefficient of v_{62} in z_{34} is $(f_1+1)\cdot(f_2+v_{59})+v_{59}+1$, where f_1 and f_2 are two polynomials for key and IV. If $v_{59}=1$ and $f_1=1$, Δz_{34} will be zero. So conditions are set as below:

Type 0 conditions:

$$v_{46}=v_{47}=v_{63}=v_{20}=0, v_{59}=1 \ ,$$

$$v_{25}=v_0+v_3,$$

$$v_{14}=v_1+v_{24}+v_{26}+v_{39}+v_4+v_{52}$$

Type 1 condition:

$v_{37}=v_{17}+v_{42}+F_1$, where F_1 is a polynomial of keys only.

Step 3. For $i=37$, the coefficient of v_{62} in z_{37} is $f_3\cdot f_4+1$, where f_3 and f_4 are two polynomials for key and IV. So conditions are set to impose $f_3=1$ and $f_4=1$. Then there are Type 0 conditions:

$$v_{23}=v_{48}=v_{49}=v_{50}=v_{45}=0,$$

$$v_{25}=v_0+v_3,$$

$$v_{28}=v_3+v_6,$$

$$v_{29}=v_{17}+v_{27}+v_4+v_{42}+v_{55}+v_7.$$

Type 1 conditions:

$$v_{16}=v_{26}+v_{54}+k_{13}+k_{34}+k_4+k_{46}+k_5+k_{59}+k_7+1,$$

$$v_2=v_{27}+v_{40}+v_5+F_2,$$

where F_2 is a polynomial of keys only.

Step 4. For $i=40$, the coefficient of v_{62} in z_{40} is $(f_5+1)\cdot(f_6+1)+1$, where f_5 and f_6 are two polynomials for key and IV. So conditions are set to impose $f_5=0$ and $f_6=0$. Then there are Type 0 conditions:

$$v_{26}=v_{51}=v_{52}=v_{53}=0 \ ,$$

$$v_{31}=v_6+v_9,$$

$$v_{10} = v_{30} + v_{32} + v_{58} + v_7 ,$$

$$v_{28} = v_3 + v_6 ,$$

Type 1 conditions

$$v_{57} = v_{29} + v_{44} + k_{10} + k_{16} + k_{37} + k_{49} + k_{62} + k_7 + k_{79} + k_8 ,$$

$$v_5 = v_{30} + v_{43} + v_8 + F_3 ,$$

where F_3 is a polynomial of keys only.

Thus, all the conditions are rearranged and 21 Type 0 conditions and 6 Type 1 conditions are obtained as below:

$$v_{41} = v_{59} = 1, v_j = 0, j \in \{20,23,26,45,46,47,49,48,50,51,52,53,63\},$$

$$v_{10} = v_{30} + v_{32} + v_{58} + v_7 ,$$

$$v_{25} = v_0 + v_3 ,$$

$$v_{28} = v_3 + v_6 ,$$

$$v_{31} = v_6 + v_9$$

$$v_{14} = v_1 + v_{24} + v_{39} + v_4 ,$$

$$v_{55} = v_{17} + v_{27} + v_4 + v_{42} + v_{29} + v_7$$

$$v_{37} = v_{17} + v_{42} + F_1 ,$$

$$v_2 = v_{27} + v_{40} + v_5 + F_2 ,$$

$$v_{43} = v_{30} + v_5 + v_8 + F_3 ,$$

$$v_{19} = F_4 , where\ \ F_4 = k_{79} + 1,$$

$$v_{16} = v_{26} + v_{54} + F_5 , where\ \ F_5 = k_{13} + k_{34} + k_4 + k_{46} + k_5 + k_{59} + k_7 + 1,$$

$$v_{57} = v_{29} + v_{44} + F_6 , where\ \ F_6 = k_{10} + k_{16} + k_{37} + k_{49} + k_{62} + k_7 + k_{79} + k_8 .$$

The total number of our conditions is 27, which decreased by 13 compared with the results of [5]. The 6 IV bits concerned with key expressions are called dynamic variables. Then there are 36 free IV variables and this is enough to recover the 6 key expressions considering the bias of Δz_{106} is about 0.000245.

Both theory derivation and experiments confirm that the expressions of F_1 to F_3 are right and only contain key bits. We omit the specific expressions for lack of space. This case is also applied for G_1 to G_4 in Section 4.2.

The algorithm of attack on 106-round Grain v1 is presented below:

Algorithm 1. CDA on 106-round Grain v1

```
Step 1. Set 64 counters corresponding to the 64 options of $F_1$ to $F_6$. Then, choose $2^{27}$ free IVs randomly and set the IV bits involved in Type 0 conditions and Type 1 conditions to appropriate values.
Step 2. Compute $\Delta z_{106}$ with $\Delta v_{62} = 1$. If $\Delta z_{106}$ is zero, the counter of corresponding options of $F_1$ to $F_6$ is summed by 1.
Step 3. The values of 64 counters are sorted and the max counter is denoted by $sum_{\max}$. Then
    If $sum_{\max} / 2^{27} \geq 0.500133$
        Conclude the values of $F_1$ to $F_6$ corresponding to the max counter are the right values of key expressions.
    Else
        For the given $2^{27}$ IVs, the right values of $F_1$ to $F_6$ could not be recovered and output ‘failure’.
```

The data complexity of our attack is $O(2^{27})$ and the compute complexity is $2^{27}\cdot 2^{6}\cdot 2 = O(2^{34})$. The distributions of wrong and right guesses are normal distribution $N(\frac{N}{2},\frac{N}{4})$ and normal distribution $N(N\cdot\rho_1, N\cdot\rho_1\cdot(1-\rho_1))$ respectively, where $\rho_1 = 0.500245$. The probability that none of 63 wrong guesses passes the test is $(\phi(\frac{0.500133\cdot 2^{27}-0.5\cdot 2^{27}}{\sqrt{2^{27}/4}}))^{63}\approx 0.939$. The probability that right guess passes the test is $\phi(\frac{0.500133\cdot 2^{27}-\rho_1\cdot 2^{27}}{\sqrt{2^{27}/4}})\approx 0.995$. Therefore, the success probability of Algorithm 1 is about $0.995\cdot 0.939\approx 0.934$. From experiments with 100 random keys, we successfully recover all the key expressions for 94 times and the success probability is about 94%. The experiments were run on a 3.2 GHz Intel Core i5 processor and took about 1.1 hours to complete for each attack.

4.2 CDA on 107-round Grain v1

For 107-round Grain v1, single difference is set in v_{63} and the conditions are $\Delta z_{17}=0\ \&\ \Delta z_{35}=0\ \&\ \Delta z_{38}=0\ \&\ \Delta z_{41}=0$. The same as Section 4.1, in order to eliminate Δz_i, we compute the coefficient of v_{63} in z_i and impose Type 0 or Type 1 conditions to eliminate the coefficient. The attacking steps are as follows:

Step 1. For $i=17$, the coefficient of v_{63} in z_{17} is the polynomial $(g_2+1)\cdot(g_1+v_{42})+v_{42}+1$, where g_1 and g_2 are two polynomials for key and IV. If $g_2=1$ and $v_{42}=1$, the coefficient will be zero. So conditions are set as below:

Type 0 conditions:

$$v_{42}=1, v_{46}=0\ ,$$

Type 1 condition:

$v_{20}=v_0+v_{25}+v_3+G_1$, where G_1 is a polynomial of keys only.

Step 2. For $i=35$, the coefficient of v_{63} in z_{35} is $(g_4+1)\cdot(g_3+v_{60})+v_{60}+1$, where f_3 and f_4 are two polynomials for key and IV. If $v_{60}=1$ and $g_4=1$, Δz_{34} will be zero. So conditions are set as below:

Type 0 conditions:

$$v_{21}=v_{22}=v_{43}=v_{47}=v_{48}=v_{49}=0, v_{60}=1\ ,$$
$$v_{15}=v_2+v_{25}+v_{40}+v_5+v_{27}+v_{53}\ ,$$
$$v_{26}=v_1+v_4,$$

Type 1 condition:

$$v_{38}=\ v_0\ +\ v_{25}\ +\ v_3\ +\ v_{18}\ +G_2\ ,$$

where G_2 is a polynomial of keys only.

Step 3. For $i=38$, the coefficient of v_{63} in z_{38} is $g_5\cdot g_6+1$, where g_5 and g_6 are two polynomials for key and IV. So conditions are set to impose $g_5=1$ and $g_6=1$. Then there are Type 0 conditions:

$$v_{50}=v_{51}=v_{24}=0\ \ ,$$
$$v_{29}=v_4+v_7\ ,$$
$$v_{26}=v_1+v_4,$$
$$v_{56}=v_{18}+v_{28}+v_{30}+v_5+v_8\ ,$$

and Type 1 conditions:

$$v_{17} = v_{27} + v_{55} + 1 + k_{14} + k_{35} + k_{47} + k_5 + k_6 + k_{60} + k_8\,,$$

$$v_{41} = v_{28} + v_3 + v_6 + G_3\,,$$

where G_3 is a polynomial of keys only.

Step 4. For $i = 41$, the coefficient of v_{63} in z_{41} is $g_7 \cdot g_8 + g_9$, where g_7, g_8 and g_9 are polynomials for key and IV. So conditions are set to impose $g_9 = 0$ and $g_7 = 0$. Then there are Type 0 conditions:

$$v_{27} = v_{52} = v_{53} = v_{54} = 0\ \ ,$$

$$v_{29} = v_4 + v_7\,,$$

$$v_{59} = v_{11} + v_{31} + v_{33} + v_8\,,$$

$$v_{58} = v_0 + v_{25} + v_3 +\ \ v_{30} + v_{45}\,,$$

and Type 1 conditions:

$$v_{10} = v_{20} + v_{30} + v_{32} + v_{45} + v_{58} + v_7 + k_{11} + k_{17} + k_{38} + k_{50} + k_{63} +\ \ k_8 + k_9 + 1\,,$$

$$v_9 = v_0 + v_{25} + v_3 + v_{31} + v_{44} + v_6 + G_4\,.$$

where G_4 is a polynomial of keys only.

Thus, all the conditions are rearranged and 22 Type 0 conditions and 6 Type 1 conditions are obtained as below:

$$v_{42} = v_{60} = 1, v_j = 0, j \in \{21,22,24,27,43,46,47,48,49,50,51,52,53,54\}\,,$$

$$v_{15} = v_2 + v_{25} + v_{40} + v_5\,,$$

$$v_{26} = v_1 + v_4\,,$$

$$v_{29} = v_4 + v_7\,,$$

$$v_{56} = v_{18} + v_{28} + v_{30} + v_5 + v_8\,,$$

$$v_{59} = v_{11} + v_{31} + v_{33} + v_8\,,$$

$$v_{58} = v_0 + v_{25} + v_3 + v_{30} + v_{45}$$

$$v_{20} = v_0\ +\ v_{25}\ +\ v_3\ +\ G_1\,,$$

$$v_{38} =\ v_0\ +\ v_{25}\ +\ v_3\ +\ v_{18}\ + G_2\,,$$

$$v_{41} = v_{28} + v_3 + v_6 + G_3\,,$$

$$v_9 = v_0 + v_{25} + v_3 + v_{31} + v_{44} + v_6 + G_4\,,$$

$$v_{17} = v_{27} + v_{55} + G_5, where\ G_5 = k_{14}\ + k_{35} + k_{47} + k_5 + k_6 + k_{60} + k_8 + 1,$$

$$v_{10} = v_{20} + v_{30} + v_{32} + v_{45} + v_{58} + v_7 + G_6, where\ G_6 = k_{11} + k_{17} + k_{38} + k_{50} + k_{63} + k_8 + k_9 + 1\,.$$

The total number of our conditions is 28 and there are 35 free IV variables, which is enough to recover the 6 key expressions considering the bias of Δz_{107} is about 0.000246.

The algorithm of attack on 107-round Grain v1 is presented below:

Algorithm 2. CDA on 107-round Grain v1

Step 1. Set 64 counters corresponding to the 64 options of G_1 to G_6. Then, choose 2^{27} free IVs randomly and set the Type 0 condition and Type 1 condition IV to appropriate value.

Step 2. Compute Δz_{107} with $\Delta v_{63} = 1$. If Δz_{107} is zero, the counter of corresponding options of G_1 to G_6 is summed by 1.

Step 3. The values of 64 counters are sorted and the max counter is denoted by $sum_{\max}$. Then

If $sum_{\max} / 2^{27} \geq 0.500133$

Conclude the values of G_1 to G_6 correspond to the max counter are the right values of

```
key expressions.
    Else
        For the given $2^{27}$ IVs, the right values of $G_1$ to $G_6$ could not be recovered and output
‘failure’.
```

The data complexity of our attack is $O(2^{27})$ and the compute complexity is $2^{27} \cdot 2^6 \cdot 2 = O(2^{34})$. The distributions of wrong and right guesses are normal distribution $N(\frac{N}{2}, \frac{N}{4})$ and normal distribution $N(N \cdot \rho_2, N \cdot \rho_2 \cdot (1 - \rho_2))$ respectively, where $\rho_2 = 0.500246$. The probability that none of 63 wrong guesses passes the test is $(\phi(\frac{0.500133 \cdot 2^{27} - 0.5 \cdot 2^{27}}{\sqrt{2^{27}/4}}))^{63} \approx 0.939$. The probability that right guess passes the test is $\phi(\frac{0.500133 \cdot 2^{27} - \rho_2 \cdot 2^{27}}{\sqrt{2^{27}/4}}) \approx 0.998$. Therefore, the probability of Algorithm 2 is about $0.998 \cdot 0.939 \approx 0.937$. From experiments with 100 random keys, we successfully recover all the key expressions for 95 times. The experiments were run on a 3.2 GHz Intel Core i5 processor and took about 1.1 hours to complete for each attack.

4.3 Comparison and discussion

Table 1 summarizes the results of our improved CDAs on Grain v1 compared with the previous reported references. It can be seen that we improve Sarkar’s [5] distinguishing attack on Grain v1 of 106 rounds to a key recovery attack. And our attack on 106-round Grain v1 requires fewer Type 0 conditions than [5] and [8], indicating our method saves more IV bits. In addition, we perform distinguishing and key recovery attack on 107-round Grain v1 with both fewer Type 1 conditions and lower time complexity of distinguishing than [6]. Most importantly, our attacks can practically recover key expressions with slightly higher success probability than theoretical results.

Table 1. Results of CDAs on Grain v1

Rounds	Type of attack	Number of recovered key expressions	Time complexity of distinguishing	Number of conditions (Type0,Type1)	Reference
104	Distinguishing and key recovery	15	$O(2^{40})$	(14,15)	[6]
105	Distinguishing and key recovery	6 (and 9 key bits extracted from expressions)	$O(2^{34})$	(25,6)	[8]
106	Distinguishing only	-	$O(2^{30})$	(34,6)	[5]
106	Distinguishing and key recovery	6	$O(2^{34})$	(21,6)	Sect.4.1
107	Distinguishing and key recovery	6	$O(2^{34})$	(22,6)	Sect.4.2
107	Distinguishing and key recovery	12	$O(2^{42})$	(12,12)	[6]
110	Distinguishing and key recovery	15	$O(2^{47})$	(17,15)	[6]
114	Distinguishing and key recovery at *40 bits weak key* assumption	1 key bit	$O(2^{33})$	(24,1)	[7]

5. Conclusion

A factorization simplification method is presented in this paper. This method makes it possible to obtain the expressions of states in more rounds and analyze the expressions more precisely. A meticulous analysis can be further conducted on the expressions to obtain the conditions with more IV bits saved. Using our proposed factorization simplification method and condition-imposing strategy, the distinguishing attack of 106 rounds on Grain v1 proposed by Sarkar is improved to a key recovery attack. Moreover, a new distinguishing attack and key recovery attack on Grain v1 of 107 rounds is presented with lower complexity. Experiments verify that our attack can recover key expressions practically with a high success probability. Hopefully, similar techniques may be used to attack Grain v1 for more rounds and even other NFSR based cryptosystems, which is one of our future work.

References

[1] M. Hell, T. Johansson, and W. Meier, "Grain: a stream cipher for constrained environments," *International Journal of Wireless and Mobile Computing*, vol. 2, no. 1, pp. 86-93, May, 2007. Article (CrossRef Link)

[2] M. Robshaw, "The eSTREAM Project," *New Stream Cipher Designs: The eSTREAM Finalists*, pp. 1-6, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. Article (CrossRef Link)

[3] S. Knellwolf, W. Meier, and M. Naya-Plasencia, "Conditional differential cryptanalysis of NLFSR-based cryptosystems," in *Proc. of 16th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 130-145, December 5-9, 2010. Article (CrossRef Link)

[4] S. Banik, "Some Insights into Differential Cryptanalysis of Grain v1," in *Proc. of Information Security and Privacy: 19th Australasian Conference*, pp. 34-49, July 7-9, 2014. Article (CrossRef Link)

[5] S. Sarkar, "A New Distinguisher on Grain v1 for 106 Rounds," in *Proc. of Information Systems Security: 11th International Conference*, pp. 334-344, December 16-20, 2015. Article (CrossRef Link)

[6] Z. Ma, T. Tian, and W. F. Qi, "Improved conditional differential attacks on Grain v1," *IET Information Security*, vol. 11, no. 1, pp. 46-53, January, 2017. Article (CrossRef Link)

[7] Y. Watanabe, Y. Todo, and M. Morii, "New Conditional Differential Cryptanalysis for NLFSR-based Stream Ciphers and Application to Grain v1," in *Proc. of Information Security (AsiaJCIS), 2016 11th Asia Joint Conference on*. IEEE, pp. 115-231, August 4-5, 2016. Article (CrossRef Link)

[8] S. Banik, "Conditional differential cryptanalysis of 105 round Grain v1," *Cryptography and Communications*, vol. 8, no. 1, pp. 113-137, January, 2016. Article (CrossRef Link)

[9] Y. Lee, K. Jeong, J. Sung, and S. Hong, "Related-Key Chosen IV Attacks on Grain-v1 and Grain-128," in *Proc. of Information Security and Privacy: 13th Australasian Conference*, pp. 321-335, July 7-9, 2008. Article (CrossRef Link)

[10] B. Zhang, Z. Li, D. Feng, and D. Lin, "Near Collision Attack on the Grain v1 Stream Cipher," in *Proc. of Revised Selected Papers of Fast Software Encryption: 20th International Workshop*, pp. 518-538, March 11-13, 2013. Article (CrossRef Link)

[11] S. Banik, S. Maitra, and S. Sarkar, "A Differential Fault Attack on the Grain Family of Stream Ciphers," in *Proc. of Cryptographic Hardware and Embedded Systems – CHES 2012: 14th International Workshop*, pp. 122-139, September 9-12, 2012. Article (CrossRef Link)

[12] S. Sarkar, S. Banik, and S. Maitra, "Differential Fault Attack against Grain family with very few faults and minimal assumptions," *IEEE Transactions on Computers*, vol. 64, no. 6, pp. 1647-1657, June, 2015. Article (CrossRef Link)

[13] S. Knellwolf, W. Meier, and M. Naya-Plasencia, "Conditional Differential Cryptanalysis of Trivium and KATAN," In *Revised Selected Papers of Selected Areas in Cryptography: 18th International Workshop*, pp. 200-212, August 11-12, 2011 . Article (CrossRef Link)

Jun-zhi Li received B.S. degree from Zhengzhou Information Science and Technology Institute in 2015. He is studying for M.S. degree in cryptography in the same university. His main research interests include the design and cryptanalysis of stream cipher.

Jie Guan is a professor of Zhengzhou Information Science and Technology. Her main research interests include the theory of information security and her main teaching lies in the areas of information systems and the theory of cryptography. She received Ph.D. degree in cryptography from Zhengzhou Information Science and Technology Institute in 2004.