Dynamic Cube Attack on Grain-v1

Majid Rahimi, Mostafa Barmshory

2013

Abstract

This article aims to present dynamic cube attack on Grain-v1. Dynamic cube attack finds the secret key by using distinguishers gained from structures weakness. The main idea of dynamic cube attack lies in simplifying the output function. After making it easier, dynamic cube attack will be able to exploit distinguishing attack for recovering the secret key. In this paper, we investigate Grain-v1 to which key recovery attack has never been applied because its feedback function is so sophisticated. we apply dynamic cube attack on it by utilizing both intelligent choices of Initial Value variables and appropriate simplifications. Our attack is done in feasible time complexity, and it recovers all bits of the key while the number of initialization rounds in Grain-v1 is decreased to 100. Moreover, it is the first key recovery attack on reduced version of Grain-v1. This attack is faster than exhaustive search by a factor 2^{32}.

Keywords: stream cipher, Grain-v1, dynamic cube attack, key recovery attack.

1 Introduction

Grain-v1[1] is a hardware stream cipher resisting against all previous cryptanalytic attack because not only does it have suitable tap positions and update functions, but also the number of its initialization rounds provides it with high non-linear degree. It causes this algorithm to make its way to final phase of eSTREAM project.

Distinguishing attack is one of the most effective attacks implemented on stream ciphers. It is closely associated with key recovery attack since if it finds a serious weakness, it may lead to key recovery attack. This attack is applied by different methods in which chosen IV attack is one of the most successful ones[2]. Not any cryptanalyst is now capable of directly analyzing the ANF (algebraic normal form) representation of any cryptosystem because of its high complexity, whereas chosen IV gives cryptanalysts the chance to analyze it indirectly via choosing some bits of IV as variable and considering remaining bits of IV in addition to all bits of key as constant. The structure of cube tester, cube attack and dynamic cube attack is based on chosen IV, and each one applies chosen IV with a particular purpose. This paper concentrates dynamic cube attack on Grain-v1 and elaborates on how to use it in key recovery attack.

The aim of cube tester is to distinguish an ANF of cipher from a random function where every monomial is present by 0.05 probability, i.e. a random function in terms of public and secret variables is algebraically dense. Hence, cube tester targets the number of monomials in an ANF of cryptosystem, where cube tester has just access to black box representation. For this purpose, four main steps are involved. Firstly, a subset of IV is chosen named cube variables. Secondly, all conditions of this subset are loaded as input of the black box. Thirdly, all outputs are summed. Finally, cube tester exploits the resultant sums to evaluate whether the number of monomials is either sparse or dense. If it is sparse, it will be more likely distinguishable via testing two characteristics, being balanced and including low degree monomials.

Cube attack[3] is identical to cube tester, and it is applied to any cryptographic scheme having public and secret variables, whose goal is to distinguish monomials in which every single bit of the key presents. In other words, it detects linear monomials based on the key in the ANF representation. After finding such monomials, cube attack will be capable of easily using them for key recovery attack because such monomials lead to linear equations based on the key that can be solved by Gaussian elimination method. Since cube attack was published, it has been the most successful attack on Trivium[3]. However, cube attack is just practical in a small number of cryptosystem because linear monomials exist in a high-degree ANF representation with low probability. This issue causes attacker to follow a method making cube attack more general. Much of cryptanalytic attacks exploit structure of the cipher whereas cube attack recovers the key, and cube tester distinguishes an ANF of cipher just through a large number of queries[3]. It is plainly visible that if cryptanalysts find a way to exploit structure of cipher in either cube tester or cube attack, they will be more successful. One of the best ways is simplifying the ANF representation of cipher by using the structure of cipher, and then applying either cube tester or cube attack.

While the key and IV of cryptosystem are not completely combined, existence of high-degree monomials, playing a key role in cipher resistant against cube tester, will heavily depend on few non-linear operations. In this way, cryptanalyst can simplify the ANF representation of cipher through setting to zero the state bits existing in such non-linear operation. In dynamic cube attack[4], attacker tries to both find and set to zero these state bits via utilizing the dedicated public bits named dynamic variables. The best, previous attack[5] applied to reduced variant of Grain-128[6], where a number of initial rounds are reduced from 256 to 213, can just recover 2 key bits. Whereas, dynamic cube attack is applied to two reduced variants of Grain-128[4] in which it obtains the full 128-bit key faster than exhaustive search, i.e. dynamic cube attack breaks these two variants. First and second attack are applied to 207 and 250 from 256 respectively that second attack is faster than exhaustive search by a factor of roughly 228[4]. More importantly, when 10 bits of the key are considered as constant, dynamic cube attack can be applied to the original version of Grain-128 recovering the full 118-bit key faster than exhaustive search by a factor of roughly 215[4]. In these attacks, cryptanalysts focus on nullifying a particular non-linear operation existing in output function of Grain-128 since this term is the most non-linear term which high-degree monomials in the ANF representation arises from. For nullifying this term, they set certain public variables, existing in the feedback function of NFSR, as dynamic variables. This nullification enables cryptanalysts to use cube tester in order to recover the key of Grain-128[4].

This paper presents dynamic cube attack on Grain-v1. Unlike Grain-128, high-degree monomials in the ANF representation of Grain-v1 stem from most of, instead of one, non-linear operations in update function of the NFSR. It seems that Grain-v1 will resist against dynamic cube attack because nullifying all these non-linear operations is very complex and not faster than exhaustive search. However, this article aims to write whole these non-linear operations by using the recursive description of the ciphers update function and analyze state bits that are involved in them to make the attack applicable. It leads to a significant point if only seven state bits are set to zero, high-degree monomials in the ANF representation will be omitted. Although seven state bits must be set to zero, cryptanalysts are faced with a sophisticated problem in neutralizing more than one state bit. According to dynamic cube attack, when desiring to set one state bit to zero, attackers divide the IV bits into three categories dynamic variables, cube variables and constant assigned variables[4], hence it is more likely that a special IV bits is chosen as dynamic variable in neutralizing one of the state bit and as cube variable in neutralizing other one at the same time. Such conflicts cause dynamic cube attack to fail since during the attack every IV bit can belong to one category. We address this problem via intelligent choices of the IV bits without any conflict, and then we are capable of recovering the key of Grain-v1 when its initialization rounds are decreased to 100. It should be noted that the process of choosing IV is a complex, manual one.

2 Dynamic cube attack

The idea of attack is simplifying the ANF representation of the cipher in order to intensify either the bias of cube tester or the non-random features of the ANF representation. To elaborate this idea, it is considered that cryptanalysts are able to write the ANF representation P as follows into which it is divided the three polynomials P_1, P_2 and P_3:

$$P = P_1P_2 + P_3 \tag{1}$$

P_1 is both certain and simple polynomial, P_2 is uncertain, dense polynomial and P_3 is partly simple polynomial. Being dense, P_2 is more likely behaves randomly and resists against cube testers, thus P is likewise immune from cube testers. Understandably, if cryptanalysts can set P_1 to zero, they derive $P = P_3$. P is now non-random polynomial and can be distinguished from random one by cube tester since P_3 is partly simple polynomial.

In both cube attack and cube tester, the public variables are classified into two categories. First those variables cryptanalyst sums the output of the cipher over them named cube variables, second those ones are not summed over and are considered as constant (most of the time zero) named constant assigned variables. In dynamic cube attack, the number of public variables that do not belong to cube variables is not considered as constant, named dynamic variables, since cryptanalyst sets P_1 to zero by assigning a function to each one of them. These functions are made up of a few, usually one, cube public variables and some expression of private variables. The next section clarifies how assigning such functions to dynamic variables causes P_1 to be omitted.

3 Outline of dynamic cube attack on Grain-v1

In contrast to Grain-128, the update function of the NFSR is denser which is composed of non-linear operations of wide range of degrees from two to six (instead of two and three). It leads to presence of more high-degree monomials in the ANF representation, thus cryptanalysts are forced to simplify it more in order to enable themselves to apply cube tester to Grain-v1. Most likely, simplifying entire these non-linear operations is infeasible since this process is very time consuming and has a higher complexity than exhaustive search. For getting through this trouble, we seek an optimum technique by which we are able to simplify the ANF representation with the least nullification in the update function of the NFSR. Hence, we write all non-linear operations in terms of their recursive descriptions, then we precisely investigate them to find such simplification. It can be observed that setting only seven state bits to zero will incredibly simplify the ANF representation. However, it seems that nullifying these state bits will be still very complex due to two significant issues: classification of the IV variables and high complexity of the attack.

In the case of the former, in dynamic cube attack, attacker sets a state bit to zero through assigning a special amount to much of IV variables existing in recursive description, where recursive description are equal to zero while attacker replaces these amounts in recursive description. Thus, the category of each IV is completely associated with the structure of recursive description. Regarding different kind of amounts, this process classifies the IV variables to three groups[4] dynamic variables, cube variables and constant assigned variables. Since recursive description of every state bit is independent, nullifying more than one state (seven state bits) may lead to conflict in assigning IV variables. In other words, it is more likely that a special IV bits is chosen as dynamic variable in neutralizing one of the state bit and as cube variable in neutralizing other one at the same time. Such conflicts cause attacker to be unsuccessful in setting all state bits to zero as during the attack every IV bit can be assigned to one category. This paper tackles this problem via intelligent choices of the IV bits without any conflict.

In relation to the latter, in dynamic cube attack, complexity relies on the number of guesses and cube variables[4]. The key idea behind our attack is existence of trade-off between the number of guesses and cube variables, i.e. growth in guesses triggers reduction in cube variables and vice versa. Since, escalating the state bits which must be nullified cause the number of guesses to increase , and the ANF representation to simplify more accordingly . It directly impacts on reducing cube variables[4], thus this trade-off prevents from overgrowth of complexity while we desire to set seven state bits to zero. More significantly, dynamic cube attack is capable of retrieving the key bits via those guesses which are equivalent to linear expression in terms of key[4]. Hence, when the number of guesses (or the number of state bits have to be nullified) grows, the chance of recovering more key bits will grow. This issue will increasingly enhance the complexity of attack.

Our attack is a feasible full key recovery attack on a new variant of Grain-v1 using 100 initialization rounds rather than 160, and this attack exploits output bits of 100-110. It includes two phases, preprocessing and online, that each one has two steps.

4 Description of Grain-v1

This section provides a brief explanation on Grain-v1, precise description has been cited in [1]. The cipher is made up of three main building blocks, an LFSR, an NFSR and an output function. Its state comprises a 80-bit LFSR and a 80-bit NFSR named s_i, $s_i + 1$,. . . , $s_i + 79$ and bi, $b_i + 1$,. . . , $b_i + 79$ respectively. The update functions of the LFSR and NFSR are respectively defined as follows:

$$s_{i+80} = s_i + s_{i+13} + s_{i+23} + s_{i+38} + s_{i+51} + s_{i+62}. \tag{2}$$

$$\begin{aligned} b_{i+80} =& s_i + b_i + b_{i+9} + b_{i+14} + b_{i+21} + b_{i+28} + b_{i+33} + b_{i+37} + b_{i+45} + \\ & b_{i+52} + b_{i+60} + b_{i+62} + b_{i+9}b_{i+15} + b_{i+33}b_{i+37} + b_{i+60}b_{i+63} + \\ & b_{i+21}b_{i+28}b_{i+33} + b_{i+45}b_{i+52}b_{i+60} + b_{i+15}b_{i+21}b_{i+60}b_{i+63} + \\ & b_{i+33}b_{i+37}b_{i+52}b_{i+60} + b_{i+9}b_{i+28}b_{i+45}b_{i+63} + \\ & b_{i+9}b_{i+15}b_{i+21}b_{i+28}b_{i+33} + b_{i+37}b_{i+45}b_{i+52}b_{i+60}b_{i+63} + \\ & b_{i+21}b_{i+28}b_{i+33}b_{i+37}b_{i+45}b_{i+52}. \end{aligned} \tag{3}$$

The output function is shown as below in which the variables x_0, x_1, x_2, x_3 and x_4 are equivalent to the tap positions $s_i + 3$, $s_i + 25$, $s_i + 46$, $s_i + 64$ and $b_i + 63$ respectively.

$$\begin{aligned} & z_i = \sum_{k \in A} b_{i+k} + h(s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63}) \\ & A = 1, 2, 4, 10, 31, 43, 56. \end{aligned} \tag{4}$$

Grain-v1 generates output from a 80-bit key and a 64-bit IV loaded in initialization process, where the bits of key and IV are put in NFSR and LFSR respectively, and then the value of 1 is placed in the remaining 16 bits of LFSR. It is consecutively clocked 160 times without any output in order to integrate the key and IV.

5 Description of dynamic cube attack on Grain-v1

5.1 Preprocessing phase

In preprocessing phase, we select the state bits to set them to zero, then we elaborate how to do so by assigning special amount to IV variables. Preprocessing phase performs once for every round with different secret keys. When the secret key is altered, the online phase must be iterated.

Step 1 This phase is a sophisticated, manual process because attackers must exactly analyze the ANF representation where they can not fully automate this analysis. We would desire to decompose the ANF representation of Grain-v1 into three polynomials as we mentioned above while applying dynamic cube attack to this cipher. Similar to Grain-128, the ANF representation of Grain-v1 is so complex to be decomposed in such way. Thus, this paper exploits the recursive description of the Grain-v1's update functions to find the best decomposition.

As for Grain-v1, there are more non-linear operations in the update function of NFSR (from degree two to degree six) that they have effect on high degree monomials appearance in the ANF representation in comparison to Grain-128. In this situation, simplification of the higher degree operations does not cause the ANF representation to become vulnerable to cube tester, e.g. simplifying either $b_{i+21}b_{i+28}b_{i+33}b_{i+37}b_{i+45}b_{i+52}$ or $b_{i+37}b_{i+45}b_{i+52}b_{i+60}b_{i+63}$. The solution lies in focusing on writing the recursive description for entire state bits taking part in non-linear terms of NFSR's update function. These state bits are as follows:

$\{b_{i+9}, b_{i+15}, b_{i+21}, b_{i+28}, b_{i+33}, b_{i+37}, b_{i+45}, b_{i+52}, b_{i+60}, b_{i+63}\}$

In this approach, we aim to simplify (not nullify) these state bits because setting to zero just one state bit in upper rounds is impractical. According to the structure of Grain-v1, we are able to write every state bit by applying the recursive description as bellow:

$$\begin{aligned} b_{i+j_{0\leqslant j\leqslant 79}} \quad = \quad & b_{i-80+j+21}b_{i-80+j+28}b_{i-80+j+33}b_{i-80+j+37}b_{i-80+j+45} \\ & b_{i-80+j+52} + b_{i-80+j+9}b_{i-80+j+15}b_{i-80+j+21}b_{i-80+j+28} \\ & b_{i-80+j+33} + b_{i-80+j+37}b_{i-80+j+45}b_{i-80+j+52}b_{i-80+j+60} \\ & \cdots + b_{i-80+j+63} + b_{i-80+j+21} + s_{i-80+j} \end{aligned} \tag{5}$$

We write these state bits by using equation 5, and then we inspect them in order to find the state bits by which we can increasingly simplify state bits of NFSR's update function. Those state bits which were calculated at the earlier stage of initialization step can be simplified much easier. Hence, we just specify how to simplify one of the simple ones (e.g. b_{i+21}). Regarding other state bits, the main conclusions are presented, and precise explanation for these state bits is proposed in appendix A. Regarding equation 6 and 7, making up monomials which maximally include six state bits, both b_{i+9} and b_{i+15} are simple and are not required to be simplified.

$$\begin{aligned} b_{i+9} \quad = \quad & b_{i-50}b_{i-43}b_{i-38}b_{i-34}b_{i-26}b_{i-19} + b_{i-62}b_{i-4356}b_{i-50}b_{i-43} \\ & b_{i-38} + \cdots \end{aligned} \tag{6}$$

$$\begin{aligned} b_{i+15} \quad = \quad & b_{i-44}b_{i-37}b_{i-32}b_{i-28}b_{i-20}b_{i-13} + b_{i-56}b_{i-50}b_{i-44}b_{i-37} \\ & b_{i-32} + b_{i-28}b_{i-20}b_{i-13}b_{i-5}b_{i-2} + \cdots \end{aligned} \tag{7}$$

As for equation 8, if $\mathbf{b_{i-7}}$ is nullified, the significant terms of degree five and six are nullified and b_{i+21} makes up monomials which maximally include 14 state bits rather than 15. Since the ANF of the earlier b_{i-7} is much simpler to assess and participate in other state bits (b_{i+28}, b_{i+45} and b_{i+52}) simultaneously, b_{i-7} is the best choice for nullification. This way is true for nullifying other state bits. According to appendix A, b_{i+28} is identical to b_{i+21}, i.e. if we set $\mathbf{b_{i-7}}$ to zero, the significant terms (highest non-linear degree terms) are set to zero accrodingly.

$$\begin{aligned} b_{i+21} \quad = \quad & b_{i-38}b_{i-31}b_{i-26}b_{i-22}b_{i-14}\underline{b_{i-7}} + b_{i-50}b_{i-44}b_{i-38}b_{i-31} \\ & b_{i-26} + b_{i-22}b_{i-14}\underline{b_{i-7}}(b_{i-58}b_{i-51}b_{i-46}b_{i-42}b_{i-34}b_{i-27} + \\ & b_{i-70}b_{i-64}b_{i-58}b_{i-51}b_{i-46} + b_{i-42}b_{i-34}b_{i-27}b_{i-19}b_{i-16}) \\ & (b_{i-55}b_{i-48}b_{i-43}b_{i-39}b_{i-31}b_{i-24} + b_{i-67}b_{i-61}b_{i-55}b_{i-48} \\ & b_{i-43}b_{i-39}b_{i-31}b_{i-24}b_{i-16}b_{i-13}) + \cdots \end{aligned} \tag{8}$$

In relation to other state bits, nullifying $\mathbf{b_{i-10}}$ and $\mathbf{b_{i+13}}$ cause b_{i+33} to be significantly nullified and to make up monomials which maximally consist of nine state bits instead of twenty. Nullifying $\mathbf{b_{i-10}}$ and $\mathbf{b_{i+17}}$ leads to the best simplification of b_{i+37}, causing it to make up monomials which maximally include two state bits rather than 29. Nullifying $\mathbf{b_{i-7}}$ and $\mathbf{b_{i+17}}$ leads to the significant terms of b_{i+45}, causing it to make up monomials which maximally include 31 state bits instead of 48. Similar to b_{i+45}, b_{i+52} is significantly simplified by setting $\mathbf{b_{i-7}}$ and $\mathbf{b_{i+17}}$ to zero, causing it to make up monomials which maximally include forty state bits rather than 72. Nullifying $\mathbf{b_{i-11}}$, $\mathbf{b_{i-5}}$ and $\mathbf{b_{i+17}}$ leads to the best simplification of b_{i+60}, causing it to make up monomials which maximally include fifty state bits instead of 126. Nullifying $\mathbf{b_{i-8}}$ and $\mathbf{b_{i+20}}$ causes the significant terms of b_{i+63} to be nullified, and b_{i+63} to make up monomials which maximally include 102 state bits rather than 146.

As was mentioned above, if attackers simplify the ANF representation more, the number of secret keys which are retrieve and the chance of success for recovering them will increase . Hence, we focus on the most simplification of update function.

Lower round strategy: In lower rounds, In addition to simplifying the non-linear terms, we aim to nullify those state bits taking part in update function linearly such as $\mathbf{b_{i+31}}$, $\mathbf{b_{i+43}}$ and $\mathbf{b_{i+56}}$. The results are briefly stated in table 1. As a consequence, in lower rounds, **ten state bits** are totally nullified. The ten state bits also cause some other state bits to simplify. In table 2, some of them are mentioned.

Upper round strategy: In upper rounds, nullifying ten state bits is infeasible since we are unable to classify the IV variables without conflict. It forces us to neglect some state bits nullification due to attack's success, especially ones taking part in linear term of update function . In total,

State bit	Nullification	The level of simplification
b_{i+31}	$\mathbf{b_{i-4}}$	Makes up monomials which maximally include 14 state bits instead of 20.
b_{i+43}	$\mathbf{b_{i-4}}$	Two terms of high degrees are nullified.
b_{i+56}	$\mathbf{b_{i-3}}$ and $\mathbf{b_{i+13}}$	Makes up monomials which maximally include 48 state bits instead of 103.

Table 1: The results of those state bits participating in update function linearly.

State bit	Nullification	The level of simplification
b_{i+23}	$\mathbf{b_{i-5}}$	The most non-linear terms are nullified.
b_{i+28}	$\mathbf{b_{i-7}}$	The most non-linear terms are nullified.
b_{i+36}	$\mathbf{b_{i-11}}$ and $\mathbf{b_{i-7}}$	Makes up monomials which maximally include 18 state bits instead of 29.
b_{i+39}	$\mathbf{b_{i-7}}$ and $\mathbf{b_{i+17}}$	Makes up monomials which maximally include 27 state bits instead of 38.
b_{i+40}	$\mathbf{b_{i-7}}$ and $\mathbf{b_{i+20}}$	Makes up monomials which maximally include 22 state bits instead of 37.

Table 2: Some other state bits simplified via nullifying these ten state bits.

just **seven state bits** are nullified for upper rounds. The consequences are mentioned in table 4.

After the update function is exactly analyzed, it is realized that we can nullify ten state bits up to 100 initialization rounds. Thus, we nullify ten state bits in Grain-v1 with 100 initialization rounds and seven state bits in Grain-v1 with more than 100 initialization rounds (from 100 to 110).

Step 2 This phase is also a sophisticated, manual process. Unlike Grain128, this phase is a big challenge because entire IV variables have to be classified into three categories without conflict. In this phase, we will

State bit	Nullification
b_{i+21}	$\mathbf{b_{i-7}}$
b_{i+28}	$\mathbf{b_{i-7}}$
b_{i+33}	$\mathbf{b_{i-10}}$ and $\mathbf{b_{i+13}}$
b_{i+37}	$\mathbf{b_{i-10}}$ and $\mathbf{b_{i+17}}$
b_{i+45}	$\mathbf{b_{i-7}}$ and $\mathbf{b_{i+17}}$
b_{i+52}	$\mathbf{b_{i-7}}$ and $\mathbf{b_{i+17}}$
b_{i+60}	$\mathbf{b_{i-11}}$ and $\mathbf{b_{i+17}}$
b_{i+63}	$\mathbf{b_{i-2}}$ and $\mathbf{b_{i+20}}$

Table 3: The state bits nullified in upper rounds.

nullify ten state bits for lower rounds and seven state bits for upper ones. We will be capable of doing so via intelligent choices of IV variables.

Regarding update function of Grain-v1 [1], ANF representation of any state bit includes different kinds of monomials. We divide these monomials into four classes as belows:

1. Monomials making up one IV variable, i.e. they are linear in terms of IV.
2. Monomials making up one IV and one key variable, i.e. they are linear in terms of IV and key.
3. Monomials making up key variables, they can be of any degree.
4. Those monomials which do not belong in previous classes.

This division alleviates classification of IV variable to dynamic variables, cube variables and constant assigned variables. One IV of first class is assigned to dynamic variables and others are considered as zero. Second class must be chosen as cube variables. All monomials belonging to third class are considered as a new equation called *pr*. In forth class, in which every monomial consists of one IV variable at least, we have to nullify all of them by assigning constant value to IV variables (usually zero).

Since these state bits become too sophisticated after many initialization steps, they must be written by using recursive description as well. It causes the ANF representation of every state bit to be straightforward, and to be more simplified. Hence, this method is iterated while the ANF representation becomes simple enough. Then, cryptanalysts can easily classify the IV variables by our division.

However, due to existing common IV bits in the ANF representation of different state bits, the conflict is likely to be inevitable. For tackling the problem, we use special technique for common IV bits. First, those IV bits participating in more than state bits are identified. Then, those state bits in which such IV bits can just belong to one class determine the class of them. In order to specify how to nullify more than one state bit by intelligent choice, we explain it for the earlier state bits with 100 initialization rounds in which the most simplification is done. Precise explanations about other state bits with 100 initialization rounds are stated in appendix B.

According to step 1, $\{b_{i-11}, b_{i-10}, b_{i-8}, b_{i-7}, b_{i-5}, b_{i-4}, b_{i-3}, b_{i+13}, b_{i+17}, b_{i+20}\}$ must set to zero; b_{i-11}, b_{i-10} and b_{i-8} are earlier state bits and more easier for nullification. These three state bits are equal to b_{80+9}, b_{80+10} and b_{80+12} in output bit 100 respectively. Thus, the recursive descriptions of these state bits are as follows:

$$\begin{aligned} b_{i-11} &= b_{89} = b_{80+9} \\ &= s_9 + b_{10} + b_{11} + b_{13} + b_{18} + b_{19} + b_{23} + b_{30} + b_{37} + b_{40} + b_{42} \\ &\quad + b_{46} + b_{52} + b_{54} + b_{61} + b_{65} + b_{69} + b_{71}b_{72} + b_{18}b_{24} + \cdots \\ &\quad + b_{46} \cdots b_{72} + b_{30} \cdots b_{61} + s_{34} + s_{12} + s_{55}b_{72} + s_{12}s_{34}s_{55} \\ &\quad + s_{12}s_{55} + s_{12}s_{55}b_{72} + s_{55}b_{72} \end{aligned} \tag{9}$$

$$\begin{aligned} b_{i-10} &= b_{90} = b_{80+10} \\ &= s_{10} + b_{10} + \cdots + b_{72} + b_{73} + b_{19}b_{25} + \cdots + b_{47} \cdots b_{73} \\ &\quad + b_{31}b_{62} + s_{35} + s_{13} + s_{56} + b_{73} + s_{13}s_{35}s_{56} + s_{13}s_{56} \\ &\quad + s_{13}s_{56}b_{73} + s_{56}b_{73} \end{aligned} \tag{10}$$

$$\begin{aligned} b_{i-8} &= b_{92} = b_{80+12} \\ &= s_{12} + b_{12} + \cdots + b_{74}b_{75} + b_{21}b_{27} + \cdots + b_{49} \cdots b_{75} \\ &\quad + b_{33} \cdots b_{46} + s_{37} + s_{15} + s_{85} + b_{75} + s_{15}s_{37}s_{58} + s_{15}s_{58} \\ &\quad + s_{15}s_{58}b_{75} + s_{37}s_{58}b_{75} + s_{58}b_{75} \end{aligned} \tag{11}$$

In the case of b_{89}, we assign zero to s_{12} and s_{34}, and select s_9 as dynamic variables and s_{55} as cube variable. Therefore, the equation 9 is converted to $s_9 + pr_1 + s_{55}(1 + b_{72})$ as we mentioned above. If s_9 is chosen $pr_1 + s_{55}(1 + b_{72})$, b_{89} will be set to zero. Since pr_1 and $1 + b_{72}$ consist of key bits , their amounts are unknown and have to be guessed in online phase.

In relation to b_{90}, we assign zero to s_{13} and s_{35}, and select s_{10} as dynamic variables and s_{56} as cube variable. As a consequence, the equation 10 is converted to $s_{10} + pr_2 + s_{56}(1 + b_{73})$; b_{90} will be nullified by assigning $pr_2 + s_{56}(1 + b_{73})$ to s_{10}. In online phase, pr_2 and $1 + b_{73}$ must be guessed.

As for b_{92} , we assign zero to s_{37} and s_{58}, and select s_{15} as dynamic variables. Consequently, the equation 11 is converted to $s_{15} + pr_3$; and one polynomial must be guessed. Not that we are forced to assign zero to s_{58} since it is necessary for nullifying $b_{i+13} = b_{113}$.

In $b_{i-7} = b_{93} = b_{80+13}$ (Appendix B), we assign zero to s_{59}, and choose s_{38} as dynamic and s_{51} as cube variable. Regarding s_{16} is common IV variable and have to be selected as dynamic for nullifying $b_{i-4} = b_{96}$, we assign $p\hat{r}_5 + s_{51}$ to it. Thus, b_{i-7} is converted to $s_{38} + p\hat{r}_4 + s_{51}$. By choosing $p\hat{r}_4 + s_{51}$ for s_{38}, we can set b_{93} to zero. In online phase, $p\hat{r}_4$ must be guessed.

In $b_{i-5} = b_{95} = b_{80+15}$, we assign zero to s_{61}, and select s_{18} as dynamic variable and s_{50} as cube one. Regarding s_{40} is common IV variable and have to be selected as dynamic for nullifying b_{i+20}, we assign $p\hat{r}_{20} + s_{50}(1+ b_{67})$ to it. By choosing $p\hat{r}_5 + s_{50}(1 + b_{67})$ for s_{18}, we can set b_{95} to zero. In online phase, $p\hat{r}_5$ must be guessed; $(1 + b_{67})$ will be guessed in nullifying b_{i+20}.

According to the structure of $b_{i-4} = b_{96} = b_{80+16}$,we are unable to zero it directly, we must nullify s_{80} first. If s_{80} be nullify, conflict will happen. Thus, we simplify and convert it to $\hat{Pr}_6 + Pr_{10} + s_{51}$. Then, we assign zero to s_{19} and s_{41}, and assign one to s_{62}. Finally, we select s_{16} as dynamic variables, which is equal to $\hat{Pr}_5 + s_{51}$, and s_{51} as cube variable that are caused b_{96} to be nullified.

Nullification of the remaining state bits is more complex since we are unable to nullify them directly. In each one, we have to zero a number of other state bits. The prime results are classified in table 4 and precise explanations are stated in appendix B.

State bit	Other state bits nullified	Constant assigned variables	Cube variables	Dynamic variables	The number of guess terms
$b_{i-4} = b_{96}$	s_{80}	$Zero = \{19, 41\}, one = \{62\}$	$\{51\}$	$\{16\}$	1
$b_{i-3} = b_{97}$	b_{80}	$Zero = \{25, 46\}$	$\{49\}$	$\{0, 17\}$	2
$b_{i+13} = b_{113}$	$b_{85}, b_{89}, b_{93}, b_{95}$	$Zero = \{8, 30, 58\}, one = \{36\}$	$\{51\}$	$\{5, 33\}$	3
$b_{i+17} = b_{117}$	$b_{81}, b_{82}, b_{83}, b_{99}, b_{100}, s_{83}$	$Zero = \{4, 6, 7, 22, 26, 27, 28, 29, 41, 47\}$	$\{49, 50, 51, 55\}$	$\{1, 2, 3, 44, 45, 54\}$	7
$b_{i+20} = b_{120}$	$b_{84}, b_{86}, b_{102}, b_{103}, s_{86}, s_{87}$	$Zero = \{26, 52, 53\}$	$\{50, 55\}$	$\{31, 32, 40, 48, 57\}$	5

Table 4: The main results about nullification of other state bits

5.2 Online phase

In this phase, we illustrate how the key bits are retrieved given the parameters of preprocessing phase.

Step 1 we first select a big cub and set whole subcubes summing over them. The size of subcubes is at least $d - 3$ (considering the size of big cube is d). Then, we guess entire secret expressions that exist in dynamic variables to calculate them during the cube summations. In Grain-V1 with 100 initialization rounds, we choose $d = 9$ as the size of big cube where five IV bits were determined and others must be selected from following set. Note that some selections of big cubes give the better consequences than other which depend on the structure of Grain-V1.

$$\{11, 14, 20, 21, 23, 24, 38, 39, 42, 43, 46, 60, 63\}$$

Step 2 Given that the number of secret expressions is e, the number of guesses is 2^e. For any guess, sum over the subcubes selected in previous step with dynamic variables accordingly and the constant assigned bits in preprocessing phase. Thus, a list of sums (with dimension of 2^e) are obtained from any guess. Then, the guesses score, which is the number of one in list of summation, are calculated for any guess and sorted from the lowest score to the highest score.

In dynamic cube attack [4], the guess score is measure of non-randomness in the subcube summation. In other words, those guesses having the lowest score are most likely the correct guesses for the secret expressions. Consequently, calculation of the guess having lowest score is the simple technique for finding the value of secret expression.

However, depending on the parameters of the attack and the structure of algorithm, this technique does not always lead to correct answer since there are likely to be guesses having a score is equal to the lowest score. For instance,, unlike other rounds, we are able to usually find correct guess for different random key in 100th round due to the most simplification.

Full key recovery attack: If we find the value of the linear expressions, we are capable of retrieving those key bits existing in such expression by Gaussian elimination. Thus, we focus on finding the value of linear expressions that exactly contain only a key bit, and retrieve these key bits by assigning the corresponding value from the best guess.

In output bit 100, we gain four linear expression and retrieve four key bits accordingly. We repeat this technique for other rounds, while we nullify seven state as stated in table 2. Due to classification of IV bits is easier, we can derive more linear expression by changing both dynamic and cube variable. Note that the attack fail to retrieve the correct guess with more probability in upper rounds. Altogether, we gain 51 key bits from different linear expressions existed in output bits 100-110. The remaining key bits are calculated by exhaustive search.

Complexity: The size of subcube is at least $d-3$, and the size of secret expressions is e. Thus, the complexity of summing over all its subcubes is limited to $d^2 2^{d+e}$. We could retrieve 51 key bits by dynamic cube attack and the remaining 29 key bits by exhaustive search. Consequently, the complexity is equal to $51 \times 26^2 \times 2^{33} + 2^{29} \approx 2^{48}$.

6 Conclusion and Open Issues

In this paper, we first specified how to effectively simplify the output function of Grain-v1. Then, we could nullified more than one state bits via the suitable classification of IV bits. These classifications enabled us to apply dynamic cube attack which is the first key recovery attack on reduced version of Grain-v1. Finally, we could establish a full key recovery attack with feasible time complexity.

An important future work is applying this technique to either Grain-v1 with more initialization rounds or other algorithms with the same structure. We commence to apply this attack to other variant of Grain-v1 that needs to spend more time for stating precise explanation.

7 Appendix A

$$
\begin{aligned}
b_{i+45} =& b_{i-14}\underline{b_{i-7}}b_{i-2}[b_{i+2} = b_{i-57}b_{i-50}b_{i-45}b_{i-41}b_{i-33}b_{i-26} + b_{i-69} \\
& b_{i-63}b_{i-57}b_{i-50}b_{i-45} + b_{i-41}b_{i-33}b_{i-26}b_{i-18}b_{i-15} + \cdots][b_{i+10} \\
& = b_{i-49}b_{i-42}b_{i-37}b_{i-33}b_{i-25}b_{i-18} + b_{i-61}b_{i-55}b_{i-49}b_{i-42}b_{i-37} \\
& + b_{i-33}b_{i-25}b_{i-18}b_{i-10}b_{i-7} + \cdots][b_{i+17} = b_{i-42}b_{i-35}b_{i-30}b_{i-26} \\
& b_{i-18}b_{i-11} + b_{i-54}b_{i-48}b_{i-42}b_{i-35}b_{i-30} + b_{i-26}b_{i-18}b_{i-11}b_{i-3} \\
& b_i + \cdots] + b_{i-26}b_{i-20}b_{i-14}\underline{b_{i-7}}b_{i-2} + [b_{i+2} = b_{i-57}b_{i-50}b_{i-45} \\
& b_{i-41}b_{i-33}b_{i-26} + b_{i-69}b_{i-63}b_{i-57}b_{i-50}b_{i-45} + b_{i-41}b_{i-33} \\
& b_{i-26}b_{i-18}b_{i-15}\cdots][b_{i+10} = b_{i-49}b_{i-42}b_{i-37}b_{i-33}b_{i-25}b_{i-18}+ \\
& b_{i-61}b_{i-55}b_{i-49}b_{i-42}b_{i-37} + b_{i-33}b_{i-25}b_{i-18}b_{i-10}b_{i-7} + \cdots] \\
& [\underline{b_{i+17}} = b_{i-42}b_{i-35}b_{i-30}b_{i-26}b_{i-18}b_{i-11} + b_{i-54}b_{i-48} \\
& b_{i-42}b_{i-35}b_{i-30} + b_{i-26}b_{i-18}b_{i-11}b_{i-3}b_i\cdots][b_{i+25} = b_{i-34} \\
& b_{i-27}b_{i-22}b_{i-18}b_{i-10}b_{i-3} + b_{i-46}b_{i-40}b_{i-34}b_{i-27}b_{i-22} + b_{i-18} \\
& b_{i-10}b_{i-3}([b_{i+5} = b_{i-54}b_{i-47}b_{i-42}b_{i-38}b_{i-30}b_{i-23} + b_{i-66}b_{i-60} \\
& b_{i-54}b_{i-47}b_{i-42} + b_{i-38}b_{i-30}b_{i-23}b_{i-15}b_{i-12} + \cdots])([b_{i+8} = b_{i-51} \\
& b_{i-44}b_{i-39}b_{i-35}b_{i-27}b_{i-20} + b_{i-63}b_{i-57}b_{i-51}b_{i-44}b_{i-39} + b_{i-35} \\
& b_{i-27}b_{i-20}b_{i-12}b_{i-9} + \cdots]) + \cdots][b_{i+28} = b_{i-31}b_{i-24}b_{i-19}b_{i-15} \\
& b_{i-7}b_i + b_{i-43}b_{i-37}b_{i-31}b_{i-24}b_{i-19} + b_{i-15}b_{i-7}b_i([b_{i+8} = b_{i-51} \\
& b_{i-44}b_{i-39}b_{i-35}b_{i-27}b_{i-20} + b_{i-63}b_{i-57}b_{i-51}b_{i-44}b_{i-39} + b_{i-35} \\
& b_{i-27}b_{i-20}b_{i-12}b_{i-9} + \cdots])([b_{i+12} = b_{i-48}b_{i-41}b_{i-36}b_{i-32}b_{i-24} \\
& b_{i-17} + b_{i-60}b_{i-54}b_{i-48}b_{i-41}b_{i-36} + b_{i-32}b_{i-24}b_{i-17}b_{i-9}b_{i-6}+ \\
& \cdots]) + \cdots] + \cdots
\end{aligned}
\tag{12}
$$

$$
\begin{aligned}
b_{i+52} ={}& \underline{b_{i-7}}b_i[b_{i+5} = b_{i-54}b_{i-47}b_{i-42}b_{i-38}b_{i-30}b_{i-23} + b_{i-66}b_{i-60} \\
& b_{i-54}b_{i-47}b_{i-42} + b_{i-38}b_{i-30}b_{i-23}b_{i-15}b_{i-12} + \ldots][b_{i+9} = \\
& b_{i-50}b_{i-43}b_{i-38}b_{i-34}b_{i-26}b_{i-19} + b_{i-62}b_{i-56}b_{i-50}b_{i-43} \\
& b_{i-38} + b_{i-34}b_{i-26}b_{i-19}b_{i-11}b_{i-8} + \ldots][b_{i+17} = b_{i-42}b_{i-35} \\
& b_{i-30}b_{i-26}b_{i-18}b_{i-11} + b_{i-54}b_{i-48}b_{i-42}b_{i-35}b_{i-30} + b_{i-26} \\
& b_{i-18}b_{i-11}b_{i-3}b_i + \cdots][b_{i+24} = b_{i-35}b_{i-28}b_{i-23}b_{i-19}b_{i-11} \\
& b_{i-4} + b_{i-47}b_{i-41}b_{i-35}b_{i-28}b_{i-23} + b_{i-19}b_{i-11}b_{i-4}([b_{i+4} = \\
& b_{i-55}b_{i-48}b_{i-43}b_{i-39}b_{i-31}b_{i-24} + b_{i-67}b_{i-61}b_{i-55}b_{i-48} \\
& b_{i-43} + b_{i-39}b_{i-31}b_{i-24}b_{i-16}b_{i-13} + \ldots])([b_{i+7} = b_{i-52}b_{i-45} \\
& b_{i-40}b_{i-36}b_{i-28}b_{i-21} + b_{i-64}b_{i-58}b_{i-52}b_{i-45}b_{i-40} + b_{i-36} \\
& b_{i-28}b_{i-21}b_{i-13}b_{i-10} + \ldots]) + \ldots]b_{i-19}b_{i-13}\underline{b_{i-7}}b_i \\
& [b_{i+5} = b_{i-54}b_{i-47}b_{i-42}b_{i-38}b_{i-30}b_{i-23} + b_{i-66}b_{i-60}b_{i-54}b_{i-47} \\
& b_{i-42} + b_{i-38}b_{i-30}b_{i-23}b_{i-15}b_{i-12} + \ldots] + [b_{i+9} = b_{i-50}b_{i-43} \\
& b_{i-38}b_{i-34}b_{i-26}b_{i-19} + b_{i-62}b_{i-56}b_{i-50}b_{i-43}b_{i-38} + b_{i-34}b_{i-26} \\
& b_{i-19}b_{i-11}b_{i-8} + \ldots][\underline{b_{i+17}} = b_{i-42}b_{i-35}b_{i-30}b_{i-26}b_{i-18} \\
& b_{i-11} + b_{i-54}b_{i-48}b_{i-42}b_{i-35}b_{i-30} + b_{i-26}b_{i-18}b_{i-11}b_{i-3}b_i + \cdots] \\
& [b_{i+24} = b_{i-35}b_{i-28}b_{i-23}b_{i-19}b_{i-11}b_{i-4} + b_{i-47}b_{i-41}b_{i-35}b_{i-28} \\
& b_{i-23} + b_{i-19}b_{i-11}b_{i-4}([b_{i+4} = b_{i-55}b_{i-48}b_{i-43}b_{i-39}b_{i-31}b_{i-24}+ \\
& b_{i-67}b_{i-61}b_{i-55}b_{i-48}b_{i-43} + b_{i-39}b_{i-31}b_{i-24}b_{i-16}b_{i-13} + \ldots]) \\
& ([b_{i+7} = b_{i-52}b_{i-45}b_{i-40}b_{i-36}b_{i-28}b_{i-21} + b_{i-64}b_{i-58}b_{i-52}b_{i-45} \\
& b_{i-40} + b_{i-36}b_{i-28}b_{i-21}b_{i-13}b_{i-10} + \ldots]) + \ldots]([b_{i+32} = b_{i-27}b_{i-20} \\
& b_{i-15}b_{i-11}b_{i-3}([b_{i+4} = b_{i-55}b_{i-48}b_{i-43}b_{i-39}b_{i-31}b_{i-24} + b_{i-67} \\
& b_{i-61}b_{i-55}b_{i-48}b_{i-43} + b_{i-39}b_{i-31}b_{i-24}b_{i-16}b_{i-13} + \ldots]) + b_{i-39} \\
& b_{i-33}b_{i-27}b_{i-20}b_{i-15} + b_{i-11}b_{i-3}([b_{i+4} = b_{i-55}b_{i-48}b_{i-43}b_{i-39} \\
& b_{i-31}b_{i-24} + b_{i-67}b_{i-61}b_{i-55}b_{i-48}b_{i-43} + b_{i-39}b_{i-31}b_{i-24}b_{i-16} \\
& b_{i-13} + \ldots])([b_{i+12} = b_{i-48}b_{i-41}b_{i-36}b_{i-32}b_{i-24}b_{i-17} + b_{i-60}b_{i-54} \\
& b_{i-48}b_{i-41}b_{i-36} + b_{i-32}b_{i-24}b_{i-17}b_{i-9}b_{i-6} + \ldots])([b_{i+16} = b_{i-44} \\
& b_{i-37}b_{i-32}b_{i-28}b_{i-20}b_{i-12} + b_{i-56}b_{i-50}b_{i-44}b_{i-37}b_{i-32} + b_{i-28} \\
& b_{i-20}b_{i-13}b_{i-15}b_{i-2} + \ldots]) + \ldots])([b_{i+35} = b_{i-24}b_{i-17}b_{i-12}b_{i-8}b_i \\
& ([b_{i+7} = b_{i-52}b_{i-45}b_{i-40}b_{i-36}b_{i-28}b_{i-21} + b_{i-64}b_{i-58}b_{i-52}b_{i-45} \\
& b_{i-40} + b_{i-36}b_{i-28}b_{i-21}b_{i-13}b_{i-10} + \ldots]) + b_{i-36}b_{i-30}b_{i-24}b_{i-17} \\
& b_{i-12} + b_{i-8}b_i([b_{i+7} = b_{i-52}b_{i-45}b_{i-40}b_{i-36}b_{i-28}b_{i-21} + b_{i-64}b_{i-58} \\
& b_{i-52}b_{i-45}b_{i-40} + b_{i-36}b_{i-28}b_{i-21}b_{i-13}b_{i-10} + \ldots])([b_{i+15} = \\
& b_{i-44}b_{i-37}b_{i-32}b_{i-28}b_{i-20}b_{i-13} + b_{i-56}b_{i-50}b_{i-44}b_{i-37}b_{i-32}+ \\
& b_{i-28}b_{i-20}b_{i-13}b_{i-5}b_{i-2}])([b_{i+18} = b_{i-41}b_{i-34}b_{i-29}b_{i-25}b_{i-17} \\
& b_{i-10} + b_{i-53}b_{i-47}b_{i-41}b_{i-34}b_{i-29} + b_{i-25}b_{i-17}b_{i-10}b_{i-2}([b_{i+1} = \\
& b_{i-58}b_{i-51}b_{i-46}b_{i-42}b_{i-34}b_{i-27} + b_{i-70}b_{i-64}b_{i-58}b_{i-51}b_{i-46}+ \\
& b_{i-42}b_{i-34}b_{i-27}b_{i-19}b_{i-16} + \ldots]) + \ldots]) + \ldots]) \ldots
\end{aligned}
\tag{13}
$$

8 Appendix B

The way of nullification the state bits by classification of IV bits

$$\begin{aligned}
b_{i-11} &= b_{89} = b_{80+9} \\
&= s_9 + b_{10} + b_{11} + b_13 + b_{18} + b_{19} + b_{23} + b_{30} + b_{37} + b_{40} + b_{42} + \\
&\quad b_{46} + b_{52} + b_{54} + b_{61} + b_{65} + b_{69} + b_{71}b_{72} + b_{18}b_{24} + \cdots + b_{46}\cdots b_{72} + \\
&\quad b_{30}\cdots b_{61} + s_{34} + s_{12} + s_{55}b_{72} + s_{12}s_{34}s_{55} + s_{12}s_{55} + s_{12}s_55b_{72} + \\
&\quad s_{55}b_{72}
\end{aligned} \tag{14}$$

$$b_{89} = s_9 + Pr_1 + s_{55}(1 + b_{72}) \tag{15}$$

$$s_9 = Pr_1 + s_{55}(1 + b_{72}) \tag{16}$$

cube bites: $\{55\}$
Dynamic bits: $\{9\}$
Zero Bits: $\{12, 34\}$

$$\begin{aligned}
b_{i-10} &= b_{90} = b_{80+10} \\
&= s_{10} + b_{10} + \cdots + b_{72} + b_{73} + b_{19}b_{25} + .. + b_{47}..b_{73} + \\
&\quad b_{31}b_{62} + s_{35} + s_{13} + s_{56} + b_{73} + s_{13}s_{35}s_{56} + s_{13}s_{56} + s_{13}s_{56}b_{73} + \\
&\quad s_{56}b_{73}
\end{aligned} \tag{17}$$

$$b_{90} = s_{10} + Pr_2 + s_{56}(1 + b_{73}) \tag{18}$$

$$s_{10} = Pr_2 + s_{56}(1 + b_{73}) \tag{19}$$

cube bites: $\{56\}$
Dynamic bits: $\{10\}$
Zero Bits: $\{13, 35\}$

$$\begin{aligned}
b_{i-8} &= b_{92} = b_{80+12} \\
&= s_{12} + b_{12} + .. + b_{74}b_{75} + b_{21}b_{27} + .. + b_{49}..b_{75} + \\
&\quad b_{33}..b_{46} + s_{37} + s_{15} + s_{85} + b_{75} + s_{15}s_{37}s_{58} + s_{15}s_{58} + s_{15}s_{58}b_{75} + \\
&\quad s_{37}s_{58}b_{75} + s_{58}b_{75}
\end{aligned} \tag{20}$$

$$b_{92} = s_{15} + Pr_3 + s_{58}(1 + b_{75}) \tag{21}$$

$$s_{15} = Pr_3 \tag{22}$$

cube bites: $\{\}$
Dynamic bits: $\{15\}$
Zero Bits: $\{37, 58\}$

$$\begin{aligned} b_{i-7} &= b_{93} = b_{80+13} \\ &= s_{13} + b_{13} + .. + b_{75}b_{76} + b_{22}b_{28} + .. + b_{50}..b_{76} + \\ & \quad b_{34}..b_{65} + s_{38} + s_{16} + s_{59} + b_{76} + s_{16}s_{38}s_{59} + s_{16}s_{59} + s_{16}s_{59}b_{76} + \\ & \quad s_{38}s_{59}b_{76} + s_{59}b_{76} \end{aligned} \tag{23}$$

$$b_{93} = \hat{Pr}_5 + Pr_4 + s_{51} + s_{38} = \hat{Pr}_4 + s_{51} + s_{38} \tag{24}$$

$$s_{38} = \hat{Pr}_4 + s_{51} + s_{38} \tag{25}$$

cube bites: {51}
Dynamic bits: {38}
Zero Bits: {59} Gusse bit: {1}

$$\begin{aligned} b_{i-5} &= b_{95} = b_{80+15} \\ &= s_{15} + b_{15} + .. + b_{77} + b_{24}b_{30} + .. + b_{52}..b_{78} + \\ & \quad b_{36}..b_{67} + s_{40} + s_{18} + s_{61} + s_{18}s_{40}s_{61} + s_{18}s_{61} + s_{18}s_{61}b_{78} + \\ & \quad s_{40}s_{61}b_{79} + s_{61}b_{7} \end{aligned} \tag{26}$$

$$b_{95} = Pr_{19} + Pr_5 + s_{50}(1 + b_{67}) + s_{18} = \hat{Pr}_5 + s_{50}(1 + b_{67}) + s_{18} \tag{27}$$

$$s_{18} = \hat{Pr}_5 + s_{50}(1 + b_{67}) \tag{28}$$

cube bites: {50}
Dynamic bits: {18}
Zero Bits: {61} Gusse bit: {2}

$$\begin{aligned} b_{i-4} &= b_{96} = b_{80+16} \\ &= s_{16} + b_{16} + \cdots + b_{78} + b_{25}b_{31} + \cdots + s_{41} + s_{19}s_{80} + s_{62}s_{80} \\ & \quad + s_{19}s_{41}s_{62} + s_{19}s_{62}s_{80} + s_{62}b_{79}s_{80} + s_{80}b_{79} + b_{79} \\ & \quad + s_{19}s_{62}b_{79} + s_{41}s_{62}b_{79} \end{aligned} \tag{29}$$

$$\begin{aligned} s_{80} &= s_{62} + s_{38} + s_{23} + s_{13} + s_{3} + s_{51} + s_{0} + Pr + s_{25} + b_{63} \\ & \quad + s_{3} \times 1 + s_{46}s_{64} + 1 \times b_{63} + s_{3}s_{25}s_{46} + s_{3}s_{46} \times 1 \\ & \quad + s_{3}s_{46}b_{63} + s_{25}s_{46}b_{63} + s_{64}b_{63} \times 1 \\ &= \hat{Pr}_6 + s_{49}(1 + b_{66}) + Pr_{10} + s_{49}(1 + b_{66}) + s_{51} \\ &= \hat{Pr}_6 + Pr_{10} + s_{51} \end{aligned} \tag{30}$$

$$b_{96} = Pr_5 + s_{80} + s_{16} = \hat{P}r_5 + s_{51} + s_{16} \tag{31}$$

$$s_{16} = \hat{Pr}_5 + s_{51} \tag{32}$$

Cube bites: $\{51\}$
Dynamic bits: $\{16\}$
Zero assigned bits: $\{19, 41\}$
One assigned bits: $\{62\}$
The number of guessed term: $\{1\}$

$$\begin{aligned} b_{i-3} &= b_{97} = b_{80+17} \\ &= s_{17} + b_{17} + \cdots + b_{80} + b_{54} \cdots b_{80} + b_{36} \cdots b_{69} + s_{42} \\ &\quad + s_{20}s_{81} + s_{63}s_{81} + s_{81}b_{80} + s_{20}s_{42}s_{63} + s_{20}s_{42}s_{81} \\ &\quad + s_{20}s_{42}b_{80} + s_{42}s_{63}b_{80} + s_{63}s_{81}b_{80} \end{aligned} \tag{33}$$

$$\begin{aligned} b_{80} &= s_0 + b_0 + \cdots + b_{63} + \cdots + b_{37} \cdots b_{63} + b_{21} \cdots b_{52} \\ &\quad + s_{25} + s_3 + s_{46} + b_{63} + s_3s_{25}s_{46} + s_3s_{46} + s_3s_{46}b_{63} \\ &\quad + s_{25}s_{46}b_{63} + s_{46}b_{63} \\ &= Pr_6 + s_{46}(1 + b_{63}) + s_0 + s_3 \\ &= s_0 + Pr_6 + Pr_{10} + s_{49}(1 + b_{66}) \\ &= s_0 + \hat{Pr}_6 + s_{49}(1 + b_{66}) \end{aligned} \tag{34}$$

Cube bites: $\{49$ which was used before$\}$
Dynamic bits: $\{0\}$
Zero assigned bits: $\{25, 46\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{1\}$

$$b_{97} = s_{17} + b_{17} + \cdots + b_{79} + \cdots + b_{36} \cdots b_{69} = \hat{Pr}_7 + s_{17} \tag{35}$$

$$s_{17} = \hat{Pr}_7 \tag{36}$$

Cube bites: $\{-\}$
Dynamic bits: $\{17\}$
Zero assigned bits: $\{-\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{1\}$

$$\begin{aligned} b_{i+13} &= b_{113} = b_{80+33} \\ &= s_{33} + b_{33} + \cdots + b_{85} + b_{89} + b_{93} + b_{95} + b_{96} + \cdots \\ &\quad + b_{70} \cdots b_{96} + b_{54} \cdots b_{85} + s_{58} + s_{36}s_{97} + s_{79}s_{97} + s_{36}s_{58}s_{79} \\ &\quad + s_{36}s_{58}s_{97} + s_{36}s_{58}b_{96} + s_{58}s_{79}b_{96} + s_{79}s_{97}b_{96} \end{aligned} \tag{37}$$

$\{b_{89}, b_{93}, b_{95}\}$ which were set to zero before.

$$\begin{aligned} b_{85} &= s_5 + b_5 + \cdots + b_{68} + \cdots + b_{42} \cdots b_{68} + b_{26} \cdots b_{57} \\ &\quad + s_{30} + s_8 + s_{51} + b_{68} + s_8 s_{30} s_{51} + s_8 s_{51} + s_8 s_{51} b_{68} + s_{30} s_{51} b_{68} \\ &\quad + s_{51} b_{68} \\ &= s_5 + Pr_8 + s_{51}(1 + b_{68}) \end{aligned} \tag{38}$$

$$s_5 = Pr_8 + s_{51}(1 + b_{68}) \tag{39}$$

Cube bites: $\{51\}$
Dynamic bits: $\{5\}$
Zero assigned bits: $\{8, 30\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{2\}$

$$\begin{aligned} b_{113} &= s_{33} + b_{33} + \cdots + b_{75} \cdots b_{96} + b_{54} \cdots b_{85} + s_{36} s_{97} + s_{97} \\ &= s_{33} + Pr_8 \end{aligned} \tag{40}$$

$$s_{33} = Pr_8 \tag{41}$$

Cube bites: $\{-\}$
Dynamic bits: $\{33\}$
Zero assigned bits: $\{58\}$
One assigned bits: $\{36\}$
The number of guessed term: $\{1\}$

$$\begin{aligned} b_{i+17} &= b_{117} = b_{80+37} \\ &= s_{37} + b_{37} + b_{38} + \cdots + b_{82} + b_{89} + b_{93} + b_{97} + b_{99} + \cdots \\ &\quad + b_{74} \cdots b_{100} + b_{58} \cdots b_{89} + s_{62} + s_{40} s_{101} + s_{101} b_{100} \\ &\quad + s_{83} s_{101} + s_{40} s_{62} s_{83} + s_{40} s_{62} s_{101} + s_{40} s_{62} b_{100} \\ &\quad + s_{62} s_{83} b_{100} + s_{83} s_{101} b_{100} \end{aligned} \tag{42}$$

$\{b_{89}, b_{93}, b_{97}, s_{37}\}$ which were set to zero before.

$$\begin{aligned} s_{83} &= s_3 + s_{16} + s_{36} + s_{41} + s_{54} + 1 + b_{66} + s_6 + s_{28} + s_{49} + 1 \times b_{66} \\ &\quad + s_6 s_{28} s_{49} + s_6 s_{49} + s_6 s_{49} b_{66} + s_{28} s_{49} b_{66} + s_{49} b_{66} \\ &= s_3 + s_6 + s_{16} + s_{26} + s_{28} + s_{49} + s_{54} + s_6 s_{28} s_{49} + s_6 s_{49} \\ &\quad + s_6 s_{49} b_{66} + s_{28} s_{49} b_{66} + s_{49} b_{66} \end{aligned} \tag{43}$$

$\{s_6, s_{16}, s_{28}\}$ which were set to zero before.
$\{s_3\}$ which is dynamic variable and used in nullifying b_{100}.

$$\begin{aligned} s_{83} &= Pr_{10} + s_{26} + s_{41} + s_{49} + s_{54} \\ s_{54} &= Pr_{10} + s_{49} \end{aligned} \tag{44}$$

Cube bites: $\{49\}$
Dynamic bits: $\{54\}$
Zero assigned bits: $\{26, 41\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{1\}$

$$\begin{aligned} b_{82} &= s_2 + b_2 + \cdots + b_{65} + \cdots + b_{39} \cdots b_{65} + b_{23} \cdots b_{54} + s_{27} \\ &\quad + s_5 + s_{48} + b_{65} + s_5 s_{27} s_{48} + s_5 s_{48} + s_5 s_{48} b_{65} + s_{27} s_{48} b_{65} \\ &\quad + s_{48} b_{65} \end{aligned} \tag{45}$$

$$b_{82} = s_2 + s_{48} + pr_{12} + s_{51}(1 + b_{68}) \tag{46}$$

$$s_2 = p\hat{r}_{12} + s_{51}(1 + b_{68}) \tag{47}$$

Cube bites: $\{51\}$
Dynamic bits: $\{2\}$
Zero assigned bits: $\{-\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{1\}$

$$\begin{aligned} b_{100} &= s_{20} + b_{20} + \cdots + b_{80} + b_{82} + \cdots + b_{57} \cdots b_{83} + b_{39} \cdots b_{71} + s_{45} \\ &\quad + s_{23} s_{84} + s_{66} s_{84} + s_{84} b_{83} + s_{23} s_{45} s_{84} + s_{23} s_{45} s_{66} + s_{23} s_{45} b_{83} \\ &\quad + s_{45} s_{66} b_{83} + s_{66} s_{84} b_{83} \end{aligned} \tag{48}$$

$\{b_{80}, s_{20}\}$ which were set to zero before.
$\{b_{82}, b_{83}\}$ which will be set to zero.

$$\begin{aligned} b_{83} &= s_3 + b_3 + \cdots + b_{66} + \cdots + b_{40} \cdots b_{66} + b_{24} \cdots b_{55} + s_{28} \\ &\quad + s_6 + s_{49} + b_{66} + s_6 s_{28} s_{49} + s_6 s_{49} + s_6 s_{49} b_{66} + s_{28} s_{49} b_{66} \\ &\quad + s_{49} b_{66} \end{aligned} \tag{49}$$

$$b_{83} = s_3 + pr_{10} + s_{49}(1 + b_{66}) \tag{50}$$

$$s_3 = pr_{10} + s_{49}(1 + b_{66}) \tag{51}$$

Cube bites: $\{49\}$
Dynamic bits: $\{3\}$
Zero assigned bits: $\{6, 28\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{2\}$

$$\begin{aligned} b_{100} &= s_{45} + pr_{11} + s_{84} \\ &= + b_{67} + s_{45} + pr_{11} + s_4 + s_{17} + s_{42} + s_{55} + s_{27} + 1 + b_5 + \cdots \\ &\quad + s_7 + s_{29} + s_{50} + s_{67} + s_7 s_{50} b_{67} + s_{29} s_{50} b_{67} + s_{50} b_{67} \\ &= s_{45} + p\hat{r}_{11} + s_{50}(1 + b_{67}) + s_{55} \end{aligned} \tag{52}$$

Cube bites: $\{50, 55\}$
Dynamic bits: $\{45\}$
Zero assigned bits: $\{4, 7, 27, 29\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{1\}$

$$\begin{aligned} b_{99} &= s_{19} + b_{19} + \cdots + b_{79} + b_{81} + b_{82} + b_{82} + b_{79} + \cdots \\ &\quad + b_{82} \cdots b_{34} + \cdots + b_{82} \cdots b_{56} + \cdots + s_{44} + s_{22}s_{83} \\ &\quad + s_{65}s_{83} + s_{83}b_{82} + s_{22}s_{44}s_{83} + s_{22}s_{44}s_{65} + s_{22}s_{44}b_{82} \\ &\quad + s_{44}s_{65}b_{82} + s_{65}s_{83}s_{82} \\ &= s_{44} + pr_{13} + s_{22}s_{44} + s_{81} \end{aligned} \tag{53}$$

$$b_{99} = s_{81} \tag{54}$$

Cube bites: $\{-\}$
Dynamic bits: $\{44\}$
Zero assigned bits: $\{22\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{1\}$

$$\begin{aligned} b_{81} &= s_1 + b_1 + \cdots + b_{64} + \cdots + b_{38} \cdots b_{64} + b_{22} \cdots b_{53} \\ &\quad + s_{26} + s_4 + s_{47} + b_{64} + s_4 s_{26} s_{47} + s_4 s_{47} + s_4 s_{47} b_{64} \\ &\quad + s_{26} s_{47} b_{64} + s_{47} s_{64} \\ &= s_1 + s_{26} + s_{47} + pr_{14} \end{aligned} \tag{55}$$

$$\begin{aligned} b_{117} &= s_1 + s_{47}(1 + b_{64}) + s_{26}s_{47}b_{64} + pr + pr_{14} \\ &= s_1 + p\hat{r}_{14} \end{aligned} \tag{56}$$

Cube bites: $\{-\}$
Dynamic bits: $\{1\}$
Zero assigned bits: $\{47\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{1\}$

$$\begin{aligned} b_{i+20} &= b_{120} = b_{80+40} = s_{40} + b_{40} + b_{41} + \cdots + b_{85} + b_{92} + b_{96} \\ &\quad + b_{100} + b_{102} + \cdots + b_{77} \cdots b_{103} + b_{61} \cdots b_{92} + s_{65} + s_{43}s_{104} \\ &\quad + s_{86}s_{104} + s_{43}s_{65}s_{86} + s_{43}s_{65}s_{104} + s_{43}s_{65}b_{103} \\ &\quad + s_{65}s_{86}b_{103} \end{aligned} \tag{57}$$

$\{b_{85}, b_{92}, b_{96}, b_{100}\}$ which were set to zero before.
$\{b_{102}, b_{103}\}$ which will be set to zero.

$$\begin{aligned} b_{103} &= s_{23} + b_{23} + \cdots + b_{83} + b_{85} + \cdots + b_{60} \cdots b_{86} + b_{42} \cdots b_{74} \\ &+ s_{48} + s_{26}s_{87} + s_{69}s_{87} + s_{87}b_{86} + s_{26}s_{48}s_{87} + s_{26}s_{48}s_{69} \\ &+ s_{26}s_{48}b_{86} + s_{48}s_{69}b_{86} + s_{69}s_{87}b_{86} \end{aligned} \tag{58}$$

$\{b_{83}, b_{85}, s_{23}\}$ which were set to zero before.
$\{b_{86}, s_{87}\}$ which will be set to zero.

$$\begin{aligned} b_{86} &= s_6 + b_6 + \cdots + b_{69} + \cdots + b_{43} \cdots b_{69} + b_{27} \cdots b_{57} + s_{31} + s_9 \\ &+ s_{52} + b_{69} + s_9s_{31}s_{52} + s_9s_{52} + s_9s_{52}b_{69} + s_{40}s_{52}b_{69} + s_{52}b_{69} \end{aligned} \tag{59}$$

$\{s_9\}$ which was used as dynamic variable before.

$$b_{86} = s_{31} + Pr_{18} + s_{55}(1 + b_{72}) \tag{60}$$

$$s_{31} = Pr_{18} + s_{55}(1 + b_{72}) \tag{61}$$

Cube bites: $\{55\}$
Dynamic bits: $\{31\}$
Zero assigned bits: $\{52\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{1\}$

$$\begin{aligned} s_{87} &= s_7 + s_{10} + s_{20} + s_{30} + Pr + s_{32} + s_{45} + s_{53} + s_{58} + s_{69} \\ &+ s_{10}s_{32}s_{53} + s_{10}s_{53} + s_{10}s_{53}b_{70} + s_{32}s_{53}b_{70} + s_{53}b_{70} \end{aligned} \tag{62}$$

$\{s_7, s_{20}, s_{30}, s_{58}\}$ which were set to zero before.
$\{s_{10}, s_{45}\}$ which were used as dynamic variables before.

$$\begin{aligned} s_{87} &= Pr_{18} + s_{50}(1 + b_{67}) + s_{55} + s_{32} + s_{53} + s_{10}s_{32}s_{53} \\ &+ s_{10}s_{53} + s_{10}s_{53}b_{70} + s_{32}s_{53}b_{70} + s_{53}b_{70} \end{aligned} \tag{63}$$

$$s_{87} = Pr_{18} + s_{50}(1 + b_{67}) + s_{55} + s_{32} \tag{64}$$

$$s_{32} = Pr_{18} + s_{50}(1 + b_{67}) + s_{55} \tag{65}$$

Cube bites: $\{50, 55\}$
Dynamic bits: $\{32\}$
Zero assigned bits: $\{53\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{1\}$

$$b_{103} = s_{48} + Pr_{15} \tag{66}$$

$$s_{48} = Pr_{15} \tag{67}$$

Cube bites: $\{-\}$
Dynamic bits: $\{48\}$
Zero assigned bits: $\{26\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{1\}$

$$\begin{aligned} b_{102} &= s_{22} + b_{22} + \cdots + b_{82} + b_{84} + \cdots + b_{61} \cdots b_{85} + b_{43} \cdots b_{75} \\ &\quad + s_{47} + s_{25}s_{86} + s_{68}s_{86} + s_{86}b_{85} + s_{25}s_{47}s_{86} + s_{25}s_{47}s_{68} \\ &\quad + s_{25}s_{47}b_{85} + s_{47}s_{68}b_{85} + s_{68}s_{86}b_{85} \end{aligned} \tag{68}$$

$\{s_{22}, s_{47}, b_{85}\}$ which were set to zero before.
$\{s_{86}\}$ which will be set to zero.
$\{b_{84}\}$ which will be simplified.

$$\begin{aligned} b_{84} &= s_4 + b_4 + \cdots + b_{67} + \cdots + b_{41} \cdots b_{67} + b_{25} \cdots b_{56} \\ &\quad + s_{29} + s_7 + s_{50} + b_{67} + s_7s_{29}s_{50} + s_7s_{50} + s_7s_{50}b_{67} + s_{29}s_{50}b_{67} \\ &\quad + s_{50}b_{67} \\ &= Pr_{19} + s_{50}(1 + b_{67}) \end{aligned} \tag{69}$$

$\{s_4, s_7, s_{29}\}$ which were set to zero before.

$$\begin{aligned} s_{86} &= s_6 + s_9 + s_{19} + s_{29} + Pr + s_{31} + s_{44} + s_{52} + s_{57} + s_{68} \\ &\quad + s_9s_{31}s_{52} + s_9s_{52} + s_9s_{52}b_{69} + s_{31}s_{52}b_{69} + s_{52}b_{69} \end{aligned} \tag{70}$$

$\{s_6, s_{19}, s_{29}\}$ which were set to zero before.
$\{s_9, s_{44}\}$ which were used as dynamic variables before.

$$\begin{aligned} s_{86} &= Pr_{17} + s_{55}(1 + b_{72}) + s_{31} + s_{52} + s_{57} + s_9s_{31}s_{52} \\ &\quad + s_9s_{52} + s_9s_{52}b_{69} + s_{31}s_{52}b_{69} + s_{52}b_{69} \end{aligned} \tag{71}$$

$\{s_{52}\}$ which was set to zero before.

$$s_{86} = Pr_{17} + s_{57} \tag{72}$$

$$s_{57} = Pr_{17} \tag{73}$$

Cube bites: $\{-\}$
Dynamic bits: $\{57\}$
Zero assigned bits: $\{-\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{1\}$

$$\begin{aligned} b_{102} &= b_{84} + Pr \\ &= Pr_{19} + s_{50}(1 + b_{67}) \end{aligned} \tag{74}$$

$$
\begin{aligned}
b_{120} &= s_{40} + Pr + Pr_{19} + s_{50}(1 + b_{67}) + \hat{Pr} \\
&= s_{40} + Pr_{20} + s_{50}(1 + b_{67}) \qquad (75)
\end{aligned}
$$

$$
s_{40} = Pr_{20} + s_{50}(1 + b_{67}) \qquad (76)
$$

Cube bites: $\{50\}$
Dynamic bits: $\{40\}$
Zero assigned bits: $\{-\}$
One assigned bits: $\{-\}$
The number of guessed term: $\{1\}$

References

[1] M. Hell, T. Johansson, and W. Meier, "Grain-a stream cipher for constrained environments. estream, ecrypt stream cipher project, report 2005/010, 2005."

[2] H. Englund, T. Johansson, and M. Sönmez Turan, "A framework for chosen iv statistical analysis of stream ciphers," *Progress in Cryptology–INDOCRYPT 2007*, pp. 268–281, 2007.

[3] I. Dinur and A. Shamir, "Cube attacks on tweakable black box polynomials," *Advances in Cryptology-EUROCRYPT 2009*, pp. 278–299, 2009.

[4] I. Dinur and A. Shamir, "Breaking grain-128 with dynamic cube attacks," in *Fast Software Encryption*, pp. 167–187, Springer, 2011.

[5] S. Fischer, S. Khazaei, and W. Meier, "Chosen iv statistical analysis for key recovery attacks on stream ciphers," *Progress in Cryptology–AFRICACRYPT 2008*, pp. 236–245, 2008.

[6] M. Ågren, M. Hell, T. Johansson, and W. Meier, "A new version of grain-128 with authentication," in *Symmetric Key Encryption Workshop, SKEW (February 2011)*, 2011.