Improved Attack on Full-round Grain-128

Ximing Fu[1], and Xiaoyun Wang[1,2,3,4★], and Jiazhe Chen[5], and Marc Stevens[6], and Xiaoyang Dong[2]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[2] Institute for Advanced Study, Tsinghua University, Beijing 100084, China
`xiaoyunwang@mail.tsinghua.edu.cn`
[3] School of Mathematics, Shandong University, Jinan 250100, China
[4] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan 250100, China,
[5] China Information Technology Security Evaluation Center, Beijing 100085, China,
[6] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

Keywords: Stream ciphers, Grain-128, Polynomial reduction, IV representation, Dynamic cube attack

Abstract. In this paper, we propose a series of techniques that can be used to determine the missing IV terms of a complex multivariable Boolean polynomial. Using these techniques, we revisit the dynamic cube attack on Grain-128. Based on choosing one more nullified state bit and one more dynamic bit, we are able to obtain the IV terms of degree 43, combined with various of reduction techniques, fast discarding monomial techniques and IV representation technique for polynomials, so that the missing IV terms can be determined. As a result, we improve the time complexity of the best previous attack on Grain-128 by a factor of 2^{16}. Moreover, our attack applies to all keys.

1 Introduction

Most cryptanalytic problems of symmetric ciphers can be reduced to the problem of solving large non-linear multivariate polynomial systems. This problem is NP-complete [?,?]. Solving the system using linearization or relinearization methods directly will result in space and time complexities that are beyond the power of current computers. As a result, algebraic attacks such as cube attack [?], cube tester [?,?] and dynamic cube attack [?] were proposed in order to reduce the complexities.

Stream cipher Grain-128 [?] is a refined version of Grain scheme ciphers, one of the finalists of eSTREAM Project. The output bit is a high degree Boolean function over initial vector (IV) bits and key bits. Since the proposal of Grain-128, a number of cryptanalytic results have been presented in the literatures. Fischer et al. applied the statistical analysis to recover the key of reduced

★ Corresponding author.

Grain-128 up to 180 out of 256 iterations with a complexity slightly better than brute force [?]. They pointed out that Grain-128 may be immune to this attack due to the very high degree of the output polynomial. Knellwolf et al. proposed the conditional differential cryptanalysis on NFSR-based stream ciphers including Grain scheme ciphers [?]. Conditional differential cryptanalysis exploited the message modification technique introduced in [?,?], which controlled the diffusion by controlling the plaintexts. It was applied to recover 3 bit's key of the Grain-128 reduced to 213 rounds with a probability up to 0.59 and distinguish Grain-128 reduced to 215 rounds from random primitives [?]. Another message controlling method is to nullify some state bits which may be more important than the others for reducing the degree or enhancing the density. After nullifying some state bits, the representation of the output bit with IV bits can be simplified; and the output bit becomes nonrandom. This technique is called dynamic cube attack which combines the conditional differential and cube tester. With dynamic cube attack, Dinur and Shamir [?] proposed two key-recovery attacks on reduced-round Grain-128 for arbitrary keys and an attack on the full Grain-128 that holds for 2^{-10} of the key space. Furthermore, Dinur et al. [?] improved the dynamic cube attack on full Grain-128, and tested the main component with a dedicated hardware. Their experimental results showed that for about 7.5% of the keys, the proposed attack beat the exhaustive search by a factor of 2^{38}.

In this paper, we further improve the attack in [?] by obtaining the missing IV terms.

Our Contributions. The contributions of this paper are five fold. Firstly, we exploit the nullification technique introduced in [?] and improve the nullification by a better choice of nullified state bits and a carefully selected nullification of IV bits. Secondly, we give several fast discarding monomial techniques for Boolean functions in order to reduce the number of terms we need to process. Thirdly, with the aforementioned techniques, we mathematically compute the IV terms of degree 43, combined with IV representation technique. The major difference between our attack and the previous dynamic cube attacks is that we focus on obtaining the IV terms mathematically instead of choosing the suitable (sub)cubes with testing technique. Finally, we present an attack with a complexity 2^{16} faster than the best result before. In this way, our method not only improves the previous attacks but also can naturally be applied to all keys. So the successful ratio of our attack is 100%.

Due to the difference, we also simplified the attack procedure of the dynamic cube attack. Briefly, our attack can be decomposed to the following phases:

1. Determine the dynamic variables, state/IV bits to be nullified, as well as the key bits to be guessed. Calculate the IV terms of degree 43 afterwards. The cube can be chosen freely from those disappeared IV terms. This is the preprocessing phase of the attack. However, most of the dedicated work in this paper contributes to this phase. In this phase, we use various techniques to obtain the exact IV terms. We exploit degree evaluation, degree reduction and low-frequency IV bits, which help discard monomials dramatically. We also use IV representation to compute the IV terms of degree 43.

2. Guess the key bits, and get the output bits with IVs chosen according to the principle in the previous phase. The summation of the output bits over the cube can be used as a distinguisher since for the correct key the summation will always be 0, while for wrong keys 0 and 1 are supposed to occur with the same probability. This is the only on-line phase of our attack. The time complexity of our attack is about 2^{74} cipher executions, which is applicable to all keys of Grain128.

The details of our attack will be demonstrated in the rest of the paper. In Section **??**, the outline of Grain-128 and basic concepts of Boolean functions will be introduced. The related techniques including cube attacks/tester, dynamic cube attack and nullification will be introduced in Section **??**. In Section **??**, we will show the outline of our attack. Then the preprocessing phase of our attack will be presented in Section **??**, followed by the online attack in Section **??**. Next, we show an example of Grain128 reduced to 191 to illustrate our attack process in Section **??**. Finally, Section **??** summarizes the paper.

2 Preliminaries

In this section, we will first briefly introduce Grain-128, followed by the basic concepts about Boolean functions.

2.1 Notations

ANF	the Algebraic Normal Form
IV bit	public variables of Grain
IV term	product of certain IV bits
state bit	internal state bit in the initialization of Grain stream cipher
state term	product of certain state bits, IV bits or key bits

2.2 Outline of Grain-128

The state of Grain-128 is represented by a 128-bit linear feedback shift register (LFSR) and a 128-bit nonlinear feedback shift register (NFSR). The feedback function of the LFSR and NFSR are defined as

$$\begin{aligned} s_{i+128} =& s_i + s_{i+7} + s_{i+38} + s_{i+70} + s_{i+81} + s_{i+96}, \\ b_{i+128} =& s_i + b_i + b_{i+26} + b_{i+56} + b_{i+91} + b_{i+96} + b_{i+3}b_{i+67} + b_{i+11}b_{i+13} + \\ & b_{i+17}b_{i+18} + b_{i+27}b_{i+59} + b_{i+40}b_{i+48} + b_{i+61}b_{i+65} + b_{i+68}b_{i+84}. \end{aligned}$$

The output function is

$$\begin{aligned} z_i =& b_{i+2} + b_{i+15} + b_{i+36} + b_{i+45} + b_{i+64} + b_{i+73} + b_{i+89} + s_{i+93} + b_{i+12}s_{i+8} + \\ & s_{i+13}s_{i+20} + b_{i+95}s_{i+42} + s_{i+60}s_{i+79} + b_{i+12}b_{i+95}s_{i+95}. \end{aligned}$$

During the initialization step, the 128-bit key is loaded into the NFSR and 96 bits of IV are loaded into the LFSR, with the other IV bits setting to 1. The state runs 256 rounds with the output feeding back, and the first output bit is z_{257}. For the detail of Grain-128, we refer to [**?**].

2.3 Boolean Polynomial

The output bit and internal state bits of symmetric cryptosystem can be described as a Boolean function of the public variables and private variables, which can be applicable to all symmetric ciphers. For stream ciphers, the public variables are often IV while the private variables are keys. Supposing that there are m IV bits, i.e., $v_0, v_1, \ldots, v_{m-1}$ and n key bits, i.e., $k_0, k_1, \ldots, k_{n-1}$, the Algebraic Normal Form (ANF) of the internal state bit s could be written as the following style:

$$s = \sum_{I,J} \prod_{i \in I} v_i \prod_{j \in J} k_j, \tag{1}$$

where the sum operation is over filed $\mathbb{F}_2$. The $\prod_{i\in I} v_i \prod_{j\in J} k_j$ is denoted as a **state term** of s and $\prod_{i\in I} v_i$ is denoted as its corresponding **IV term**. Let IV term $t_I = \prod_{i\in I} v_i$ be the multiplication of v_i whose indices are within I, the ANF of s can be rewritten as

$$s = \sum_{I} t_I g_I(k), \tag{2}$$

where $g_I(k)$ is the sum of corresponding coefficient function of terms whose corresponding IV term is t_I, i.e. $g_I(k) = \prod_{l_1 \in J_1} k_{l_1} + \prod_{l_2 \in J_2} k_{l_2} + \cdots + \prod_{l_p \in J_p} k_{l_p}$. The degree of s is $\deg(s) = \mathtt{max}_I\{\deg(t_I)\}$.

Property 1. (**Repeat Property**) In the ANF of s in Equ. (**??**), suppose two state terms are equal, i.e. $\prod_{i\in I} v_i \prod_{j\in J} k_j = \prod_{i\in I'} v_i \prod_{j\in J'} k_j$, then the ANF of state bit s is simplified by removing the two state terms.

Property 2. (**Cover Property**) In Equ. (**??**), suppose there are two state terms $T_1 = t_{I_1} g_{I_1}(k)$ and $T_2 = t_{I_2} g_{I_2}(k)$. If $I_1 \subseteq I_2$, we denote that the state term T_1 is *covered* by T_2. We delete the *covered state term* T_1 from s to get a new s^*. If $T_1 = T_2$, according to the *repeat property*, the two state terms should be removed from s. However, according to the *cover property*, T_2 is still in s^*. So $\deg(s^*) \geq \deg(s)$. This property could help us estimate $\deg(s)$ efficiently.

3 Related Works

In this section, the techniques related to our work will be introduced, including cube attack/tester, dynamic cube attack and nullification technique.

3.1 Cube Attack and Cube Tester

Cube attack [**?**] exploits the IV terms whose coefficient is linear over key bits, and then retrieves the keys once enough independent linear functions are obtained, by solving linear equations. A methodology is proposed [**?**,**?**] to test if a coefficient is linear and which key bits are involved.

Cube attack and cube tester techniques can retrieve partial information about Boolean functions probabilistically, however maybe fail for some specific structures with high degrees. For example, if the coefficient function of key bits is $k_0 + k_1k_2k_3k_4k_5k_6k_7$, cube attack may return a linear function k_0 for less than 128 tests with random keys. In addition, if the coefficient function over key bits is $k_1k_2k_3k_4k_5k_6k_7$, the result of cube tester may be always 0 for less than 128 tests with random keys, so that it is determined that the IV term t_I does not occur.

3.2 Dynamic Cube Attack

In [?], Dinur and Shamir proposed the dynamic cube attack to recover the secret key by exploiting distinguishers obtained from cube testers, with application to Grain-128.

In dynamic cube attack, some dynamic bits in the IV are determined by key bits are chosen to nullify some state bits so as to greatly simplify the output function. Then one expects to acquire certain nonrandom property which can be exploited by cube tester. The nonrandom property can be used as a distinguisher for key recovery.

As mentioned in Section 1, the attack in [?] is an improvement of that in [?]. The dynamic cube attacks introduced and exploited in [?,?] are based on the nullification technique. The major difference between the two works is the different choices of nullified bits[7]. As a consequence, we will detail the nullification technique in the next subsection.

3.3 Nullification Technique

For Grain-128, the first output bit z_{257} can be expresses by state bits as

$$\begin{aligned} z_{257} =& b_{269}b_{352}s_{352} + b_{352}s_{299} + s_{317}s_{336} + s_{270}s_{277} + b_{269}s_{265} + s_{350} + b_{346} + \\ & b_{330} + b_{321} + b_{302} + b_{293} + b_{272} + b_{259}. \end{aligned}$$

$b_{269}b_{352}s_{352}$ is the most vital term for us because it comprises more high degree IV terms than the others. Similarly, the most vital terms of the ANF of b_{269}, b_{352} and s_{352} are $b_{153}b_{236}s_{236}$, $b_{236}b_{319}s_{319}$ and $b_{236}b_{319}s_{319}$ respectively. The common factor is b_{236}, so b_{269}, b_{352}, s_{352} and z_{257} can be simplified if nullifying b_{236}. But b_{236} is too complex, i.e.,

$$\begin{aligned} b_{236} =& b_{120}b_{203}s_{203} + b_{203}s_{150} + b_{176}b_{192} + s_{168}s_{187} + b_{169}b_{173} + b_{148}b_{156} + b_{135}b_{167} + \\ & b_{125}b_{126} + b_{119}b_{121} + b_{111}b_{175} + s_{121}s_{128} + b_{120}s_{116} + b_{204} + s_{201} + b_{199} \\ & + b_{197} + b_{181} + b_{172} + b_{164} + b_{153} + b_{144} + b_{134} + b_{123} + b_{110} + b_{108} + 1. \end{aligned}$$

Nullifying b_{236} needs too many guessed key bits, so the scheme in [?] retrieved a subset of 2^{-10} of all possible keys by fixing 10 key bits to be zero.

[7] The authors also experimentally verified the main component of the attack by a dedicated hardware in [?]

Another approach is to simplify b_{236} by nullifying b_{203}, which is adopted by [?]. b_{203} is still too complex to be nullified directly, i.e.,

$$\begin{aligned}b_{203} =& b_{87}b_{170}s_{170} + b_{170}s_{117} + b_{143}b_{159} + s_{135}s_{154} + b_{136}b_{140} + b_{115}b_{123} + b_{102}b_{134} \\ &+ b_{92}b_{93} + b_{86}b_{88} + b_{78}b_{142} + s_{88}s_{95} + b_{87}s_{83} + b_{171} + s_{168} + b_{166} + b_{164} + \\ & b_{148} + b_{139} + b_{131} + b_{120} + b_{111} + b_{101} + b_{90} + b_{77} + b_{75} + s_{75}.\end{aligned}$$

In order to nullify b_{203}, one should first nullify b_{170}, b_{159}, b_{138}, s_{135}, b_{136}, b_{134}, b_{133}, and b_{131}. All of these bits but b_{170} can be nullified directly by choosing IV bits. We know that

$$\begin{aligned}b_{170} =& b_{54}b_{137}s_{137} + b_{137}s_{84} + b_{110}b_{126} + s_{102}s_{121} + b_{103}b_{107} + b_{82}b_{90} + b_{69}b_{101} + \\ & b_{59}b_{60} + b_{53}b_{55} + b_{45}b_{109} + s_{55}s_{62} + b_{54}s_{50} + b_{138} + s_{135} + b_{133} + b_{131} + b_{115} \\ &+ b_{106} + b_{98} + b_{87} + b_{78} + b_{68} + b_{57} + b_{44} + b_{42} + s_{42}.\end{aligned}$$

In order to nullify b_{170}, b_{137} can be nullified first by setting s_9 to be a dynamic value.

In fact, b_{170} can be nullified directly as well since its expression over IV and key bits can be obtained directly in a PC. However, nullification of b_{137} can not only nullify b_{170} but also simplify b_{253} which contributes a lot to the degree and IV terms. The term $b_{143}b_{159}$ can be nullified by nullifying either b_{143} or b_{159}. The authors chose to nullify b_{159} because it can help reduce b_{275} whose ANF significant term is $b_{159}b_{242}s_{242}$. b_{145}, b_{153} and b_{176} are also nullified in order to simplify s_{261}, b_{269} and b_{292}. The nullified state bits and dynamic IV bits of [?] are shown in **??**.

Table 1. Nullification Scheme in [?]

nullification	$b_{131}, b_{133}, b_{134}, s_{135}, b_{136}, b_{137}, b_{138}, b_{145}, b_{153}, b_{159}, b_{170}, b_{176}, b_{203}$
dynamic bits	$s_3, s_5, s_6, s_{77}, s_8, s_9, s_{10}, s_{17}, s_{25}, s_{31}, s_{42}, s_{83}, s_1$

4 Basic Ideas

4.1 Outline of Our Attack

In this paper, our basic idea is to find the missing IV terms in the ANF of the first output bit z_{257} of Grain128. For example, the output bit is polynomial p over six secret variables $k_1, k_2, ..., k_6$ and five public variables v_1, v_2, v_3, v_4, v_5, as shown in **??**.

$$p = k_1k_2v_1v_2v_3v_4v_5 + k_3v_2v_3v_4v_5 + k_5v_1v_3v_4v_5 + k_6v_1v_2v_4v_5 + k_2v_1v_2v_3 + ... \quad (3)$$

Obviously, the IV term $v_1v_2v_3v_4$ does not exist in p. So we can construct cube-sum distinguisher by setting $v_5 = 0$ and selecting cube $C_T = \{v_1, v_2, v_3, v_4\}$. Then the cube sum of p is always zero. So to determine the missing product of certain IV bits (missing IV term) is in fact a way to find the missing cube variables combination from a set of candidate cube variables, i.e. the five public variables. In our attack on Grain128, there are totally 96 public variables (IV bits), in order to reduce the ANFs of the internal state bits and then reduce the output polynomial, we study the nullification technique comprehensively and give a ***new nullification scheme***, which includes:

i. We use 14 dynamic variables to nullify 14 internal state bits. When comparing with Dinur *et al.*'s nullification scheme, we use one more dynamic variable s_{15} to nullify an additional internal state bit b_{143} . Thus there are 96-14=82 free public variables (IV bits) left.
ii. Then calculate the frequency of occurrence of the 82 IV bits in the internal state bits and state terms. We nullify the high frequency IV bits to reduce the internal state bits and state terms as much as possible. We choose 36 IV bits to be nullified (set to zero). Then there are 82-36=46 free IV bits left.
iii. Thus, we have to find the missing IV terms from a candidate cube set with size 46. In this paper, we are going to find the 43-degree missing IV terms in the output polynomial z_{257}.

After determining the nullification scheme, we have to reduce the output polynomial z_{257} in order to find the 43-degree missing IV terms. The procedures are as follows shown in **??**:

- **Step 1.** Initialize LFSR and NFSR with the new nullification scheme. We compute forward to express the ANF of some internal state bits over IV bits and key bits. Thus, b_i and s_i $(0 \leq i \leq 222)$ are computed and their degrees are compute directly.
- **Step 2.** During the iteratively computing the ANF representation of the output bit z in the backward direction (decryption), we introduce the ***fast discarding monomial technique*** in Section **??**, which includes the following techniques:
  - First, we propose the ***degree evaluation*** algorithm to obtain the degree bounds of internal state bits. As the monomials of z's ANF is a product of these internal state bits, the degree of a monomial is bounded by the sum of the degrees of the multiplied internal state bits, which is regarded as the degree estimation of the monomial. If the estimated degrees of monomials are lower than 43, they are discarded directly.
  - Second, we find in Grain128 encryption scheme the high degree state terms are in the form of b_is_i. We pre-compute the ***degree reduction***s of those products, which is $d_t = \deg(b_i) + \deg(s_i) - \deg(b_is_i)$. Thus, the degree of a monomial is upper bounded by difference value between the sum of the multiplied internal state bits and the corresponding degree reduction d_t. If it is smaller than 43, the monomial is discarded.

- Third, we find 7 low-frequency IV bits out of the 46 IV bits, which means that the 7 IV bits have less probability being involved in the internal state bits. ***We are going to find the 43-degree missing IV terms that contain the 7 low frequency IV bits.*** So if a monomial does not contain the 7 low-frequency IV bits simultaneously, we discard it. For detailed description, we refer to Section **??**.

– **Step 3.** For the left monomials of z's ANF, we introduce ***IV representation technique*** in Section **??** to continue to discard monomials and finally find the 43-degree missing product of certain 43 IV bits (missing IV term). In IV representation technique, the symbolic key bits in the internal state bits are removed and only IV bits are left. We combine IV representation technique with the following two techniques:
  - Combining with removing covered IV terms, we can simplify monomials without losing the degree information. It helps us to determine a more accurate upper bound degree of the monomials of z. And then if the degree of a monomial is smaller than 43, delete it.
  - Combining with repeated IV term removing algorithm, we can simplify monomials of z's ANF without losing the missing IV term information. If we find an IV term is not in the IV representation of z, we can conclude that it is also not in z.

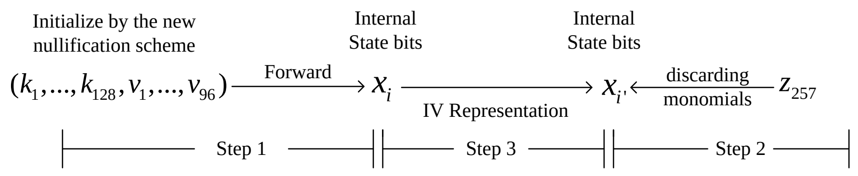

Fig. 1. Framework of the Preprocessing Phase

Thus, our attack includes two phases, which are the preprocessing phase and the online attack phase.

– In the preprocessing phase, we determine the 43-degree missing products of certain 43 IV bits (missing IV terms) in z_{257}, and select those 43-bit IV sets as cubes, which are summarised in **??**.

– In the online phase, we guess the key bits in the 14 dynamic variables, and compute the cube sums of the cubes produce in preprocessing phase:
  a. For the right key guessing, the coefficient is zero. Thus the cube sums must be zero.
  b. For the wrong key guessing, the output is random, the cube sums are not always zero.

5 Preprocessing Phase of the Attack on Full-round Grain128

5.1 New Nullification Scheme

In **Step 1** of **??**, we have to introduce a nullification scheme to initialize the state. We obtain the nullification scheme of state bits and the corresponding 14 dynamic IV bits in **??**. More details of the nullification are shown in **??**.

Table 2. Our Nullification Scheme

nullification	$b_{131}, b_{133}, b_{134}, s_{135}, b_{136}, b_{137}, b_{138}, b_{143}, b_{145}, b_{153}, b_{159}, b_{170}, b_{176}, b_{203}$
dynamic bits	$s_3, s_5, s_6, s_{77}, s_8, s_9, s_{10}, s_{15}, s_{17}, s_{25}, s_{31}, s_{42}, s_{83}, s_1$

In **??**, the first column are the state bits to be nullified and the second column are the corresponding equations. The third column are the subkey bits guessed for nullifications. For example, in order to nullify b_{131}, we just need to set s_3 to $b_{15}b_{98} + \underline{b_{98}}s_{45} + b_{71}b_{87} + s_{63}s_{82} + b_{64}b_{68} + b_{43}b_{51} + b_{30}b_{62} + b_{20}b_{21} + b_{14}b_{16} + b_6b_{70} + s_{16}s_{23} + \underline{b_{15}}s_{11} + b_{99} + b_{94} + b_{92} + b_{76} + b_{67} + b_{59} + b_{48} + b_{39} + b_{29} + b_{18} + b_5 + b_3 + 1$, where b_{98} and b_{15} underlined are the key bits guessed. Besides these two bits, one expression on key bits indicated by $*$, that is $b_{15}b_{98} + b_{71}b_{87} + b_{64}b_{68} + b_{43}b_{51} + b_{30}b_{62} + b_{20}b_{21} + b_{14}b_{16} + b_6b_{70} + b_{99} + b_{94} + b_{92} + b_{76} + b_{67} + b_{59} + b_{48} + b_{39} + b_{29} + b_{18} + b_5 + b_3$ should be guessed. Hence, three bits need to be guessed to nullify b_{131}. Totally, 40 bits should be guessed for nullifying the state bits in Table **??**.

After setting 14 dynamic IV bits, there are 96-14=82 free IV bits left. We calculate the frequency of occurrence of the 82 IV bits in the internal state bits and state terms. We nullify the high frequency IV bits to reduce the internal state bits and state terms as much as possible. We choose 36 IV bits shown in **??** to be nullified (set to zero). There are 82-36=46 free IV bits left.

After the initialization of **Step 1** in **??**. We are going to determine the 43-degree missing IV terms. It is divided into two steps as shown in **??**: **Step 2**, Fast Discarding Monomials; **Step 3**, IV Representation. The two steps are shown in **??**.

5.2 Fast Discarding Monomials

After initializing the states in **??**, we are going to iteratively compute and reduce z_{257} in **Step 2**. In each iteration, many state terms of z_{257} are produced. There are 46 free IV bits left and we will find missing IV terms of degree 43. We introduce the following three ways to discard monomials fast shown in **??**:

– Removing state terms according to degree evaluation;

Table 3. Nullification Scheme

Nullified bits	Equations for nullification	Subkey bits guessed
b_{203}	$s_1 = b_{115}b_{123} + b_{92}b_{93} + b_{86}b_{88} + s_{88}s_{95} + \underline{b_{87}}s_{83} + b_{111}b_{127} + b_{104}b_{108} + b_{83}b_{91} + b_{48} + b_{60}b_{61} + b_{54}b_{56} + b_{46}b_{110} + s_{56}s_{63} + b_{107} + b_{99} + b_{88} + b_{79} + b_{58} + b_{45} + b_{43} + s_{14}s_{21} + s_{43} + \underline{b_{52}}s_{48} + b_{113} + b_{104} + b_{85} + b_{55} + b_{42} + s_{78} + s_{47} + s_{40} + b_{106}b_{122} + b_{99}b_{103} + b_5 + b_{78}b_{86} + b_{65}b_{97} + b_{55}b_{56} + b_{49}b_{51} + b_{41}b_{105} + b_{50}s_{46} + b_{102} + b_{83} + b_{64} + b_{53} + \underline{b_{96}}s_{43} + b_{40} + b_{104}b_{120} + b_{97}b_{101} + b_{76}b_{84} + b_{63}b_{95} + b_{53}b_{54} + b_{47}b_{49} + b_{39}b_{103} + b_{120} + s_3\underline{b_{13}b_{96}} + b_{101} + b_{90} + b_{125} + b_{109} + b_{100} + b_{92} + b_{81} + b_{72} + b_{62} + b_{51} + b_{36} + b_{77} + b_{75} + s_{82} + s_{75} + b_{32}b_{115} + \underline{b_{115}}s_{62} + b_{88}b_{104} + s_{80} + b_{81}b_{85} + b_{60}b_{68} + b_{47}b_{79} + b_{37}b_{38} + s_{39} + s_8 + b_{31}b_{33} + b_{23}b_{87} + b_{111} + b_{109} + b_{93} + b_{84} + b_{76} + b_{65} + b_{56} + b_{46} + b_{70}b_{102} + b_{16} + b_3 + b_{35} + b_{22} + b_{20} + \underline{b_{100}}s_{47} + s_{18}s_{25} + b_{17}s_{13} + b_{78} + b_{50} + b_{41} + \underline{b_{13}}s_9 + s_{94} + b_{90} + b_{65} + b_{20} + b_7 + s_{86} + s_{43} + s_{12} + s_5\underline{b_{15}b_{98}} + \underline{b_{98}}s_{45} + \underline{b_{15}}s_{11} + b_{92} + b_{67} + b_{46} + b_{37} + b_{18} + b_{39} + s_{73} + s_{10}$	$b_{87}, b_{52}, b_{115}, b_{96}, b_{13}, *$
b_{176}	$s_{83} = s_{48} + b_{48} + b_{74} + b_{104} + s_{11} + b_{11} + b_{37} + b_{67} + b_{14}b_{78} + b_{22}b_{24} + b_{28}b_{29} + b_{38}b_{70} + b_{51}b_{59} + b_{72}b_{76} + b_{79}b_{95} + b_{13} + b_{26} + b_{47} + b_{56} + b_{75} + b_{100} + s_{19}\underline{b_{23}} + b_{23}b_{106} + b_{16} + b_{42} + b_{72} + b_{19}b_{83} + b_{27}b_{29} + b_{33}b_{34} + b_{43}b_{75} + b_{56}b_{64} + b_{77}b_{81} + b_{84}b_{100} + b_{18} + b_{31} + b_{52} + b_{61} + b_{80} + b_{89} + b_{105} + b_{28}b_{111} + s_{58}\underline{b_{111}} + b_{51}b_{115} + b_{59}b_{61} + b_{65}b_{66} + b_{75}b_{107} + b_{88}b_{96} + b_{109}b_{113} + s_4b_{116} + b_4b_{116} + b_{30}b_{116} + b_{60}b_{116} + b_{95}b_{116} + b_{100}b_{116} + b_7b_{71}b_{116} + b_{15}b_{17}b_{116} + b_{21}b_{22}b_{116} + b_{31}b_{63}b_{116} + b_{44}b_{52}b_{116} + b_{65}b_{69}b_{116} + b_{72}b_{88}b_{116} + b_6b_{116} + b_{19}b_{116} + b_{40}b_{116} + b_{49}b_{116} + b_{68}b_{116} + b_{77}b_{116} + b_{93} + b_{93}b_{116} + s_{12}\underline{b_{16}b_{116}} + b_{16}b_{99}b_{116} + b_{116} + s_{46}\underline{b_{99}b_{116}} + b_{50} + b_{63} + b_{121} + s_{13} + b_{15} + b_{28} + b_{49} + b_{58} + b_{77} + b_{86} + s_{21}\underline{b_{25}} + b_{25}b_{108} + s_{73}s_{92} + 1$	$b_{23}, b_{111}, b_{16}b_{116}, b_{25}, b_{99}b_{116}, *$
b_{170}	$s_{42} = b_{110}b_{126} + b_{103}b_{107} + b_{82}b_{90} + b_{69}b_{101} + b_{59}b_{60} + b_{53}b_{55} + b_{45}b_{109} + s_{55}s_{62} + b_{54}s_{50} + b_{115} + b_{106} + b_{98} + b_{87} + b_{78} + b_{68} + b_{57} + b_{44} + b_{42} + 1$	$*$
b_{159}	$s_{31} = b_{43}b_{126} + \underline{b_{126}}s_{73} + b_{99}b_{115} + s_{91} + b_{92}b_{96} + b_{71}b_{79} + b_{58}b_{90} + b_{48}b_{49} + b_{42}b_{44} + b_{34}b_{98} + s_{44}s_{51} + \underline{b_{43}}s_{39} + b_{127} + b_{122} + b_{120} + b_{104} + b_{95} + b_{87} + b_{76} + b_{67} + b_{57} + b_{46} + b_{33} + b_{31} + 1$	$b_{43}, b_{126}, *$
b_{153}	$s_{25} = b_{37}b_{120} + b_{120}s_{67} + b_{93}b_{109} + s_{85} + b_{86}b_{90} + b_{65}b_{73} + b_{52}b_{84} + b_{42}b_{43} + b_{36}b_{38} + b_{121} + b_{28}b_{92} + s_{38}s_{45} + b_{37}s_{33} + b_{116} + b_{114} + b_{98} + b_{89} + b_{81} + b_{70} + b_{61} + b_{51} + b_{40} + b_{27} + b_{25} + 1$	$*$
b_{145}	$s_{17} = b_{29}b_{112} + \underline{b_{112}}s_{59} + b_{85}b_{101} + s_{77} + b_{78}b_{82} + b_{57}b_{65} + b_{44}b_{76} + b_{34}b_{35} + b_{28}b_{30} + b_{20}b_{84} + s_{30}s_{37} + \underline{b_{29}}s_{25} + b_{113} + b_{108} + b_{106} + b_{90} + b_{81} + b_{73} + b_{62} + b_{53} + b_{43} + b_{32} + b_{19} + b_{17} + 1$	$b_{29}, b_{112}, *$
b_{143}	$s_{15} = b_{27}b_{110} + \underline{b_{110}}s_{57} + b_{83}b_{99} + s_{75}s_{94} + b_{76}b_{80} + b_{55}b_{63} + b_{42}b_{74} + b_{32}b_{33} + b_{26}b_{28} + b_{18}b_{82} + s_{28}s_{35} + \underline{b_{27}}s_{23} + b_{111} + b_{106} + b_{104} + b_{88} + b_{79} + b_{71} + b_{60} + b_{51} + b_{41} + b_{30} + b_{17} + b_{15} + 1$	$b_{27}, b_{110}, *$
b_{138}	$s_{10} = b_{22}b_{105} + \underline{b_{105}}s_{52} + b_{78}b_{94} + s_{70}s_{89} + b_{71}b_{75} + b_{50}b_{58} + b_{37}b_{69} + b_{27}b_{28} + b_{21}b_{23} + b_{13}b_{77} + s_{23}s_{30} + \underline{b_{22}}s_{18} + b_{106} + b_{101} + b_{99} + b_{83} + b_{74} + b_{66} + b_{55} + b_{46} + b_{36} + b_{25} + b_{12} + b_{10} + 1$	$b_{22}, b_{105}, *$
b_{137}	$s_9 = b_{21}b_{104} + b_{104}s_{51} + b_{77}b_{93} + s_{69}s_{88} + b_{70}b_{74} + b_{49}b_{57} + b_{36}b_{68} + b_{26}b_{27} + b_{20}b_{22} + b_{12}b_{76} + s_{22}s_{29} + \underline{b_{21}}s_{17} + b_{105} + b_{100} + b_{98} + b_{82} + b_{73} + b_{65} + b_{54} + b_{45} + b_{35} + b_{24} + b_{11} + b_9 + 1$	$b_{21}, *$
b_{136}	$s_8 = b_{20}b_{103} + b_{103}s_{50} + b_{76}b_{92} + s_{68}s_{87} + b_{69}b_{73} + b_{48}b_{56} + b_{35}b_{67} + b_{25}b_{26} + b_{19}b_{21} + b_{11}b_{75} + s_{21}s_{28} + b_{20}s_{16} + b_{104} + b_{99} + b_{97} + b_{81} + b_{72} + b_{64} + b_{53} + b_{44} + b_{34} + b_{23} + b_{10} + b_8 + 1$	$*$
s_{135}	$s_{77} = b_{19}b_{102} + \underline{b_{102}}s_{49} + s_{67}s_{86} + s_{20}s_{27} + \underline{b_{19}}s_{15} + b_{96} + s_{88} + b_{80} + s_7 + b_{71} + b_{52} + s_{45} + b_{43} + b_{22} + s_{14} + b_9$	$b_{19}, b_{102}, *$
b_{134}	$s_6 = b_{18}b_{101} + \underline{b_{101}}s_{48} + b_{74}b_{90} + s_{66}s_{85} + b_{67}b_{71} + b_{46}b_{54} + b_{33}b_{65} + b_{23}b_{24} + b_{17}b_{19} + b_9b_{73} + s_{19}s_{26} + b_{18}s_{14} + b_{102} + b_{97} + b_{95} + b_{79} + b_{70} + b_{62} + b_{51} + b_{42} + b_{32} + b_{21} + b_8 + b_6 + 1$	$b_{101}, *$
b_{133}	$s_5 = b_{17}b_{100} + \underline{b_{100}}s_{47} + b_{73}b_{89} + s_{65}s_{84} + b_{66}b_{70} + b_{45}b_{53} + b_{32}b_{64} + b_{22}b_{23} + b_{16}b_{18} + b_8b_{72} + s_{18}s_{25} + \underline{b_{17}}s_{13} + b_{101} + b_{96} + b_{94} + b_{78} + b_{69} + b_{61} + b_{50} + b_{41} + b_{31} + b_{20} + b_7 + b_5 + 1$	$b_{17}, b_{100}, *$
b_{131}	$s_3 = b_{15}b_{98} + \underline{b_{98}}s_{45} + b_{71}b_{87} + s_{63}s_{82} + b_{64}b_{68} + b_{43}b_{51} + b_{30}b_{62} + b_{20}b_{21} + b_{14}b_{16} + b_6b_{70} + s_{16}s_{23} + \underline{b_{15}}s_{11} + b_{99} + b_{94} + b_{92} + b_{76} + b_{67} + b_{59} + b_{48} + b_{39} + b_{29} + b_{18} + b_5 + b_3 + 1$	$b_{15}, b_{98}, *$

[1] Each $*$ in this table indicates a different expression of partial key bits.

Table 4. Nullified IV Bits

nullified bits	$s_{14}, s_{16}, s_{20}, s_{22}, s_{23}, s_{24}, s_{28}, s_{30}, s_{32}, s_{33}, s_{35}, s_{36}, s_{38}, s_{41}, s_{44}, s_{50}$ $s_{51}, s_{53}, s_{55}, s_{56}, s_{61}, s_{64}, s_{67}, s_{68}, s_{69}, s_{70}, s_{71}, s_{75}, s_{76}, s_{79}, s_{81}, s_{82}$ $s_{84}, s_{85}, s_{86}, s_{94}$

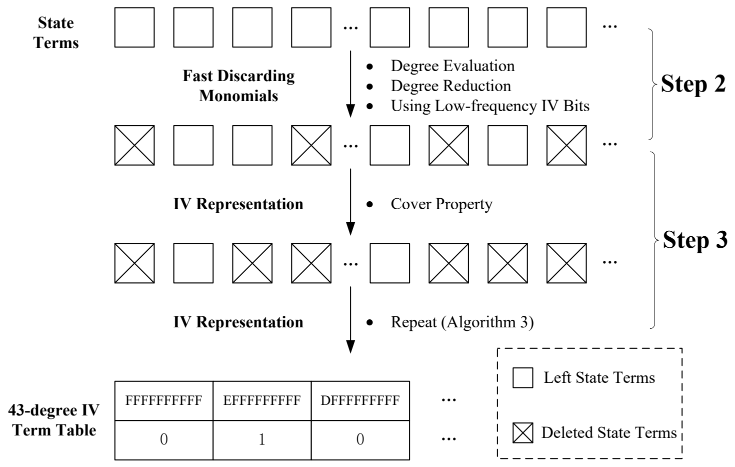

Fig. 2. Framework of Our Work

– Removing state terms according to degree reduction;
– Removing the state terms that do not contain all the 7 low-frequency IV bits.

Degree Evaluation Since we are determining the 43-degree missing IV terms, the state terms of z_{257} with degree less than 43 are removed without consideration, because they do not contain those 43-degree IV terms certainly. The exact degrees of state bits b_i and s_i for $i \in [0, 222]$ can be obtained in a PC and are shown in Table **??**. The degrees of b_i for $0 \le i < 128$ and s_j for $96 \le j < 128$ is 0 as they are constant.

In order to obtain the degree of state bit b_r or s_r for $r > 222$, we decompose the state bits until the state bits are the product of state bits b_i and s_j for $i < end$ and $j < end$. Obviously, we will get a more accurate estimation, if we choose a smaller value for *end*[8]. However, the smaller the *end* is, the more computing resource we will need. In the cryptanalysis of Grain128, we choose

[8] Suppose we are going to estimate the degree of $b_i = b_{i-3} + b_{i-1}b_{i-2}$.

$$\begin{aligned} \deg(b_i) &= \deg(b_{i-3} + b_{i-1}b_{i-2}) \\ &= \mathtt{max}\{\deg(b_{i-3}), \deg(b_{i-1}b_{i-2})\} \\ &\le \mathtt{max}\{\deg(b_{i-3}), \deg(b_{i-1}) + \deg(b_{i-2})\} \end{aligned} \tag{4}$$

Table 5. Degree of partial state bits

i	0	1	2	3	4	5	6	7	8	9	10	11
$deg(s_i)$	1	2	1	1	1	1	2	1	0	1	1	1
i	12	13	14	15	16	17	18	19	20	21	22	23
$deg(s_i)$	1	1	0	1	0	1	1	1	0	1	0	0
i	24	25	26	27	28	29	30	31	32	33	34	35
$deg(s_i)$	0	0	1	1	0	1	0	1	0	0	1	0
i	36	37	38	39	40	41	42	43	44	45	46	47
$deg(s_i)$	0	1	0	1	1	0	0	1	0	1	1	1
i	48	49	50	51	52	53	54	55	56	57	58	59
$deg(s_i)$	1	1	0	0	1	0	1	0	0	1	1	1
i	60	61	62	63	64	65	66	67	68	69	70	71
$deg(s_i)$	1	0	1	1	0	1	1	0	0	0	0	0
i	72	73	74	75	76	77	78	79	80	81	82	83
$deg(s_i)$	1	1	1	0	0	1	1	0	1	0	0	2
i	84	85	86	87	88	89	90	91	92	93	94	95
$deg(s_i)$	0	0	0	1	1	1	1	1	1	1	0	1
i	128	129	130	131	132	133	134	135	136	137	138	139
$\deg(b_i)$	1	2	1	0	1	0	0	1	0	0	0	1
$\deg(s_i)$	1	2	2	1	1	1	1	0	1	1	1	1
i	140	141	142	143	144	145	146	147	148	149	150	151
$\deg(b_i)$	2	2	2	0	1	0	1	1	1	1	0	2
$\deg(s_i)$	2	1	2	0	1	1	1	1	1	1	1	2
i	152	153	154	155	156	157	158	159	160	161	162	163
$\deg(b_i)$	1	0	2	2	1	1	1	0	2	2	3	3
$\deg(s_i)$	1	1	2	2	1	1	1	0	2	2	3	3
i	164	165	166	167	168	169	170	171	172	173	174	175
$\deg(b_i)$	2	2	2	2	2	1	0	1	2	3	3	3
$\deg(s_i)$	2	2	1	2	2	1	1	1	2	3	3	3
i	176	177	178	179	180	181	182	183	184	185	186	187
$\deg(b_i)$	0	2	2	2	2	2	2	2	3	2	2	2
$\deg(s_i)$	2	2	2	2	2	2	2	2	3	2	2	2
i	188	189	190	191	192	193	194	195	196	197	198	199
$\deg(b_i)$	3	2	2	2	2	3	3	5	4	3	3	3
$\deg(s_i)$	3	2	2	2	2	3	3	5	4	3	3	3
i	200	201	202	203	204	205	206	207	208	209	210	211
$\deg(b_i)$	3	3	3	0	3	3	5	4	4	3	4	4
$\deg(s_i)$	3	3	3	2	3	3	5	4	4	3	4	4
i	212	213	214	215	216	217	218	219	220	221	222	
$\deg(b_i)$	4	3	3	4	3	5	4	4	4	5	5	
$\deg(s_i)$	4	3	3	4	3	5	4	4	4	5	4	

end $= r - 66$ for tradeoff. We estimate the degree upper bound for b_{223} as an example, where *end* $= 223 - 66 = 157$:

- **a)** First, we express $b_{223} = b_{95} + b_{121} + b_{151} + b_{186} + b_{191} + s_{95} + b_{98}b_{162} + b_{106}b_{108} + b_{112}b_{113} + b_{122}b_{154} + b_{135}b_{143} + b_{156}b_{160} + b_{163}b_{179} + b_{97} + b_{110} + b_{131} + b_{140} + b_{159} + b_{168} + b_{184} + s_{188} + b_{107} + +b_{190}s_{137} + s_{155}s_{174} + b_{107}b_{190}s_{190}$ according to the iterative function of Grain128.
- **b)** According to Table **??** highlighted in red, initialize $d = \texttt{max}\{\deg(b_{151}), \deg(b_{186}), \deg(b_{191}), \deg(s_{95}), \deg(b_{162}), \deg(b_{154}), \deg(b_{156})+\deg(b_{160}), \deg(b_{163})+\deg(b_{179}), \deg(\deg(b_{140}), \deg(b_{168}), \deg(b_{184}), \deg(s_{188}), \deg(b_{190})+\deg(s_{137}), \deg(s_{155})+ \deg(s_{174}),$
  $\deg(b_{190})+\deg(s_{190})\} = \texttt{max}\{2, 2, 2, 1, 3, 2, 1+2, 3+2, 2, 2, 3, 3, 2+1, 2+3, 2+ 2\} = 5$. The other state terms are discarded directly because their degrees are zeros.
- **c)** Discarding the state terms of degree lower than $d = 5$, we get $b^*_{223} = b_{163}b_{179} + s_{155}s_{174}$. Iteratively compute b^*_{223} and discard state terms with degree lower than 5, we get $b^{**}_{223} = b_{47}b_{130}b_{142}s_{130}+b_{47}b_{130}b_{140}s_{130}+b_{47}b_{130}b_{146}s_{93}s_{130}+ b_{47}b_{63}b_{130}b_{146}s_{130}s_{146} + b_{141}s_{88}s_{155} + b_{58}b_{141}s_{141}s_{155}$.
- **d)** Note that there is no state bit in round that is bigger than *end*=157, the expression ends and there are still state terms that survive. Then the current degree $d = 5$ is the estimated degree of b_{223}.
- **e)** Note that, if there is no state item in b^{**}_{223} surviving, which means the degree must be less than 5. We reset $d = 4$ and continue the above steps **c)** to **d)** to get a more accurate degree bound.

We summarise the above 5 steps as Algorithm **??** and the degrees of state bit b_r or s_r for $223 \leq r \leq 320$ are shown in **??**.

Using degree evaluation to discard state terms in advance. As the monomials of z_{257}'s ANF is a product of these internal state bits, the degree of a monomial is bounded by the sum of the degrees of the multiplied internal state bits, which is regarded as the degree estimation of the monomial. If the degree estimation of the monomial is smaller than 43, we delete the monomial directly. For example, if $\deg(b_i s_i) \leq \mathcal{DEG}(b_i) + \mathcal{DEG}(s_i) < 43$, delete $b_i s_i$. If $\mathcal{DEG}(b_i) + \mathcal{DEG}(s_i) \geq 43$, continue to consider degree reduction in the following section.

Degree Reduction According to the observation of Grain128 encryption scheme, the high degree state terms are in the form of $b_i s_i$. Define the degree reduction

If we continue to decompose b_i, we find

$$\begin{aligned} b_{i-1}b_{i-2} &= (b_{i-4} + b_{i-2}b_{i-3})(b_{i-5} + b_{i-3}b_{i-4}) \\ &= b_{i-4}b_{i-5} + b_{i-3}b_{i-4} + b_{i-2}b_{i-3}b_{i-5} + b_{i-2}b_{i-3}b_{i-4}, \end{aligned} \quad (5)$$

If $\deg(b_{i-1}) = \deg(b_{i-2}b_{i-3})$ and $\deg(b_{i-2}) = \deg(b_{i-3}b_{i-4})$, then in **??**, $\deg(b_{i-1}) + \deg(b_{i-2})$ may add $\deg(b_{i-3})$ twice. So in order to obtain a more accurate degree estimation, we are willing to decompose b_i for several rounds backwards.

Algorithm 1 Degree Evaluation Algorithm ($\mathcal{DEG}$) of State Bit

```
Input: The value $r$ which indicates the state bit $b_r$.
Output: $\mathcal{DEG}(b_r)$=$d$.
1: Initialize the degree bound $d$ similar to step b) of the above example, the end point
   $end$ and the number of state terms $len \leftarrow 0$.
2: while $len = 0$ do
3:     Iteratively express $b_r$ using state bits in rounds less than $end$. During each ex-
       pression, discard the state terms of degree lower than $d$. Let $len$ be the number
       of remaining state terms.
4:     if $len = 0$ then
5:         $d \leftarrow d - 1$
6:     end if
7: end while
8: Return $d$
```

Table 6. Degree Estimation of State Bits

i	223	224	225	226	227	228	229	230	231	232	233	234
$\mathcal{DEG}(b_i)$	5	4	4	5	5	8	5	5	5	6	6	6
$\mathcal{DEG}(s_i)$	5	3	4	5	5	8	5	5	5	6	6	6
i	235	236	237	238	239	240	241	242	243	244	245	246
$\mathcal{DEG}(b_i)$	6	4	6	6	8	8	8	6	7	9	9	6
$\mathcal{DEG}(s_i)$	6	4	6	6	8	8	8	6	7	9	9	6
i	247	248	249	250	251	252	253	254	255	256	257	258
$\mathcal{DEG}(b_i)$	5	8	6	7	9	6	6	7	10	12	9	10
$\mathcal{DEG}(s_i)$	5	8	6	7	9	5	6	7	10	12	9	10
i	259	260	261	262	263	264	265	266	267	268	269	270
$\mathcal{DEG}(b_i)$	8	11	11	11	11	11	12	11	13	12	8	13
$\mathcal{DEG}(s_i)$	8	11	11	11	11	11	12	11	13	12	7	13
i	271	272	273	274	275	276	277	278	279	280	281	282
$\mathcal{DEG}(b_i)$	13	15	15	14	9	14	17	17	15	12	16	13
$\mathcal{DEG}(s_i)$	13	15	15	14	9	14	17	17	15	12	16	13
i	283	284	285	286	287	288	289	290	291	292	293	294
$\mathcal{DEG}(b_i)$	15	17	12	11	15	21	23	18	20	14	19	21
$\mathcal{DEG}(s_i)$	15	17	12	10	15	21	23	18	20	13	19	21
i	295	296	297	298	299	300	301	302	303	304	305	306
$\mathcal{DEG}(b_i)$	21	20	21	22	23	25	22	17	24	25	29	26
$\mathcal{DEG}(s_i)$	21	20	21	22	23	25	22	17	24	25	29	26
i	307	308	309	310	311	312	313	314	315	316	317	318
$\mathcal{DEG}(b_i)$	24	20	27	31	33	29	23	31	27	32	32	27
$\mathcal{DEG}(s_i)$	24	20	27	31	33	29	23	31	27	32	32	27
i	319	320										
$\mathcal{DEG}(b_i)$	22	30										
$\mathcal{DEG}(s_i)$	22	30										

d_t^r as

$$d_t^r = \deg(b_r) + \deg(s_r) - \deg(b_r s_r). \tag{6}$$

The degree reduction can be obtained using Algorithm **??**. For Algorithm **??**, we choose $end = r - 48$ considering the efficiency tradeoff of the computation.

Algorithm 2 Degree Reduction Algorithm of State Term

```
Input: The value $r$ which indicates the state term degree reduction.
Output: The degree reduction $d_t = \mathcal{DEG}(b_r) + \mathcal{DEG}(s_r) - \deg(b_r s_r)$.
1: Initialize the degree bound $d = \mathcal{DEG}(b_r) + \mathcal{DEG}(s_r)$, degree reduction $d_t = 0$, end
   point $end$ and number of survived state terms $len$.
2: while $len = 0$ do
3:     Express the state term $b_r s_r$ using state bits in rounds less than $end$, discard the
       state terms of degree lower than $d - d_t$. Let $len$ be the number of remaining state
       terms.
4:     if $len = 0$ then
5:         $d_t \leftarrow d_t + 1$
6:     end if
7: end while
8: Return $d_t$
```

The degree reduction can help discard state terms of lower degree dramatically, as it can help predict the change of degree before expression operation. We take the state term $b_{223}s_{223}$ as an example to illustrate the process to compute the degree reduction d_t. Algorithm **??** is first used to obtain the degree of state bits as shown in Table **??** and **??**.

Let end be $223 - 48 = 175$. The degree bound d is initialized as $d = \mathcal{DEG}(b_{223}) + \mathcal{DEG}(s_{223}) = 10$ shown in **??** and $d_t = 0$. Express the $b_{223}s_{223}$ discard the state terms of degree lower than $d - d_t = d$, there is 1 state term surviving, i.e., $b_{163}b_{179}s_{155}s_{174}$. Continue to compute iteratively, the remaining 5 state terms are $b_{142}b_{163}s_{155}s_{174} + b_{140}b_{163}s_{155}s_{174} + b_{146}b_{163}s_{93}s_{155}s_{174} + b_{163}s_{130}s_{155}s_{174} + b_{63}b_{146}b_{163}s_{146}s_{155}s_{174}$. There is no state bits in rounds more than 175 in all the state terms, hence the expression ends. Degree reduction $d_t = 0$ is returned.

The degree reduction of $b_i s_i$ for $i \in [0, 222]$ can be obtained directly in **??** as the degrees of $b_i s_i$, b_i and s_i can be determined exactly. Degree reductions of $b_i s_i$ for $i \in [223, 320]$ are given by **??** and shown in **??**.

Using degree reduction to discard state terms in advance. The degree of a monomial is upper bounded by difference value between the sum of the multiplied internal state bits and the corresponding degree reduction d_t. For example, reconsider monomial $b_i s_i$, if $\mathcal{DEG}(b_i) + \mathcal{DEG}(s_i) > 43$, degree evaluation introduced in **??** could not delete monomial $b_i s_i$. Thus, according to degree reduction, $\deg(b_i s_i) \leq \mathcal{DEG}(b_i) + \mathcal{DEG}(s_i) - d_t^i$, if it is smaller than 43, delete $b_i s_i$. If not, continue to use low-frequency IV bits in the following section to delete monomials.

Table 7. Degree Reduction for Partial State bits

i	129	140	142	151	154	155	160	161	162
d_t^i	1	1	1	1	1	1	1	1	1
i	163	164	165	167	168	172	173	174	175
d_t^i	2	1	1	1	1	1	1	2	2
i	180	181	182	183	184	188	193	194	195
d_t^i	1	1	1	1	1	1	2	1	2
i	196	197	198	199	206				
d_t^i	3	1	1	1	1				

Table 8. Degree Reduction of State Terms

i	223	224	225	226	227	228	229	230	231	232	233	234
d_t^i	0	0	0	0	0	2	0	0	0	1	0	0
i	235	236	237	238	239	240	241	242	243	244	245	246
d_t^i	1	0	2	0	0	1	2	1	2	3	4	0
i	247	248	249	250	251	252	253	254	255	256	257	258
d_t^i	0	2	0	0	3	0	0	0	0	4	2	3
i	259	260	261	262	263	264	265	266	267	268	269	270
d_t^i	0	6	1	0	3	3	6	1	4	4	0	5
i	271	272	273	274	275	276	277	278	279	280	281	282
d_t^i	4	3	6	8	0	4	6	7	6	5	5	3
i	283	284	285	286	287	288	289	290	291	292	293	294
d_t^i	2	5	2	0	4	6	10	7	8	0	10	8
i	295	296	297	298	299	300	301	302	303	304	305	306
d_t^i	6	9	14	10	5	6	7	1	9	5	9	7
i	307	308	309	310	311	312	313	314	315	316	317	318
d_t^i	6	2	8	11	12	11	15	13	7	8	11	9
i	319	320										
d_t^i	4	7										

Low-frequency IV Bits In our nullification scheme, 36 IV bits are nullified (set to zeros). There are remaining 46 IV bits. By testing the frequency of the occurrence of the 46 IV bits in the internal state bits of b_{128} to b_{222} and s_{128} to s_{222}, we find out 7 IV Bits, who appear in the internal state bits or terms with lower frequency than others. We list the 7 IV Bits in **??**, along with the state bits containing those 7 IV Bits.

Here, we illustrate how to exploit the 7 low-frequency bits to help discard monomials. We are finding 43-degree missing IV terms in the output polynomial z_{257}. In order to discard monomials, we add the conditions that, the 43-degree missing IV terms must contain all the 7 low-frequency bits. That means, the corresponding 43-dimension cube contains the 7 IV bits as cube variables. Thus, we do not need to consider the monomials that do not contain all the 7 low-frequency bits. Those monomials could be deleted in advance.

For example, express z_{257} using internal state bits b_i and s_i $i \in [128, 222]$. Suppose $b_{135}b_{218}s_{218}$ is a monomial of z_{257}, however s_{80} is not in b_{135}, b_{218}, or s_{218}, thus the monomial $b_{135}b_{218}s_{218}$ must not contain s_{80}. So we delete this monomial.

5.3 IV Representation

In **??**, after **Step 2**, we introduce IV representation to study each left monomials.

In the cryptanalysis of stream ciphers, the output is a Boolean function over key and IV bits. But obtaining the exact expression is hard, then the *IV representation* technique is proposed to reduce the computing complexity.

Definition 1. *(IV representation) Given a state bit* $s = \sum_{I,J} \prod_{i\in I} v_i \prod_{j\in J} k_j$, *the IV representation of* s *is* $s_{IV} = \sum_{I} \prod_{i\in I} v_i$.

For example, if a Boolean polynomial is $s = v_0k_1 + v_0k_0k_2 + v_1k_1k_2 + v_0v_1k_2$, then its IV representation is $s_{IV} = v_0 + v_0 + v_1 + v_0v_1$.

IV representation with repeated IV terms Removing Algorithm Due to the neglection of key bits, there are lots of repeated IV terms. According to Property **??**, repeated *state terms* should be removed to simplify the polynomial. However, it is different when the polynomial is in IV representation. In the above example, after removing repeated IV terms of s_{IV}, the result should be $v_0 + v_1 + v_0v_1$, instead of $v_1 + v_0v_1$ given by Property **??**. Here we give the Algorithm **??** to remove the repeated IV terms of s_{IV}. This algorithm is based on a Hash function. First, an empty hash set is initialized. For each IV term T_i, compute the hash value as $H(T_i)$ (Line 3), then determine if T_i is already in H. If not, then insert T_i into H (Lines 4-5).

Property 3. If we find an IV term is not in s_{IV}, we can conclude that it is also not in s.

Table 9. Low frequency bits

low-frequency bits	corresponding state bits
s_0	$b_{128}, s_{128}, b_{160}, s_{160}, b_{161}, s_{161}, b_{163}, s_{163}, b_{165}, s_{165}, b_{167}, s_{167}, b_{172}, s_{172}, s_{175}, b_{177}, s_{177}, b_{183}, s_{183}, s_{186}, b_{188}, s_{188}, b_{189}, s_{189}, b_{191}, s_{191}, b_{193}, s_{193}, b_{194}, s_{194}, b_{195}, s_{195}, b_{196}, s_{196}, b_{197}, s_{197}, b_{198}, s_{198}, b_{200}, s_{200}, b_{202}, s_{202}, b_{204}, s_{204}, b_{205}, s_{205}, b_{206}, s_{206}, b_{207}, s_{207}, b_{208}, s_{208}, b_{209}, s_{209}, b_{210}, s_{210}, b_{211}, s_{211}, b_{212}, s_{212}, b_{214}, s_{214}, b_{216}, s_{216}, s_{218}, b_{219}, s_{219}, b_{220}, s_{220}, b_{221}, s_{221}, b_{222}, s_{222}$
s_2	$b_{130}, s_{130}, b_{162}, s_{162}, b_{163}, s_{163}, b_{165}, s_{165}, b_{167}, s_{167}, b_{169}, s_{169}, b_{174}, s_{174}, s_{177}, b_{179}, s_{179}, b_{185}, s_{185}, s_{188}, b_{190}, s_{190}, b_{191}, s_{191}, b_{193}, s_{193}, b_{195}, s_{195}, b_{196}, s_{196}, b_{198}, s_{198}, b_{199}, s_{199}, b_{200}, s_{200}, b_{202}, s_{202}, b_{204}, s_{204}, b_{206}, s_{206}, b_{207}, s_{207}, b_{208}, s_{208}, b_{209}, s_{209}, b_{210}, s_{210}, b_{211}, s_{211}, b_{212}, s_{212}, b_{213}, s_{213}, b_{214}, s_{214}, b_{216}, s_{216}, b_{218}, s_{218}, s_{220}, b_{221}, s_{221}, b_{222}, s_{222}$
s_{37}	$b_{157}, s_{157}, s_{158}, b_{165}, s_{165}, b_{189}, s_{189}, b_{190}, s_{190}, b_{191}, s_{191}, b_{192}, s_{192}, b_{193}, s_{193}, b_{194}, s_{194}, b_{196}, s_{196}, b_{197}, s_{197}, b_{198}, s_{198}, b_{200}, s_{200}, b_{201}, s_{201}, b_{202}, s_{202}, b_{204}, s_{204}, s_{205}, s_{206}, b_{209}, s_{209}, b_{212}, s_{212}, b_{214}, s_{214}, s_{215}, s_{216}, b_{217}, s_{217}, b_{218}, s_{218}, b_{220}, s_{220}, b_{222}, s_{222}$
s_{43}	$s_{133}, b_{163}, s_{163}, s_{164}, s_{165}, b_{168}, s_{168}, b_{171}, s_{171}, s_{180}, b_{182}, s_{182}, s_{191}, b_{195}, s_{195}, b_{196}, s_{196}, b_{197}, s_{197}, b_{198}, s_{198}, b_{199}, s_{199}, b_{200}, s_{200}, b_{201}, s_{201}, b_{202}, s_{202}, b_{204}, s_{204}, b_{205}, s_{205}, b_{206}, s_{206}, b_{207}, s_{207}, b_{208}, s_{208}, b_{210}, s_{210}, b_{211}, s_{211}, b_{212}, s_{212}, b_{213}, s_{213}, b_{214}, s_{214}, b_{215}, s_{215}, b_{217}, s_{217}, b_{218}, s_{218}, b_{219}, s_{219}, b_{221}, s_{221}$
s_{60}	$b_{146}, s_{146}, s_{150}, b_{178}, s_{178}, b_{179}, s_{179}, b_{180}, s_{180}, b_{181}, s_{181}, s_{182}, b_{183}, s_{183}, b_{185}, s_{185}, b_{188}, s_{188}, b_{190}, s_{190}, s_{193}, s_{197}, b_{199}, s_{199}, b_{201}, s_{201}, s_{204}, b_{206}, s_{206}, b_{207}, s_{207}, s_{208}, b_{209}, s_{209}, b_{211}, s_{211}, b_{212}, s_{212}, b_{213}, s_{213}, b_{214}, s_{214}, b_{215}, s_{215}, b_{216}, s_{216}, b_{217}, s_{217}, b_{218}, s_{218}, b_{219}, s_{219}, b_{220}, s_{220}, b_{221}, s_{221}, b_{222}, s_{222}$
s_{80}	$b_{129}, s_{129}, s_{138}, b_{148}, s_{148}, b_{161}, s_{161}, b_{162}, s_{162}, b_{164}, s_{164}, b_{166}, s_{166}, b_{168}, s_{168}, b_{173}, s_{173}, s_{176}, b_{178}, s_{178}, b_{180}, s_{180}, b_{181}, s_{181}, b_{183}, s_{183}, b_{184}, s_{184}, b_{185}, s_{185}, s_{187}, b_{188}, s_{188}, b_{190}, s_{190}, b_{192}, s_{192}, b_{194}, s_{194}, b_{195}, s_{195}, s_{196}, b_{197}, s_{197}, b_{198}, s_{198}, b_{199}, s_{199}, b_{200}, s_{200}, b_{201}, s_{201}, s_{203}, b_{205}, s_{205}, b_{206}, s_{206}, s_{207}, b_{222}, s_{222}$
s_{90}	$s_{137}, s_{148}, b_{158}, s_{158}, s_{169}, b_{172}, s_{172}, b_{181}, s_{181}, b_{183}, s_{183}, s_{184}, b_{186}, s_{186}, b_{190}, s_{190}, b_{191}, s_{191}, b_{193}, s_{193}, b_{195}, s_{195}, b_{197}, s_{197}, b_{198}, s_{198}, s_{201}, b_{202}, s_{202}, b_{205}, s_{205}, s_{206}, b_{209}, s_{209}, b_{210}, s_{210}, b_{211}, s_{211}, b_{213}, s_{213}, b_{214}, s_{214}, b_{215}, s_{215}, b_{216}, s_{216}, b_{217}, s_{217}, b_{218}, s_{218}, b_{219}, s_{219}, b_{220}, s_{220}, b_{221}, s_{221}, b_{222}, s_{222}$

After applying Algorithm **??** to s_{IV}, it is easy to know that if an IV term exists in s, it must also exist in s_{IV}. But the opposite is not right. For example, $x_1 = v_0(k_1k_2 + k_0k_2) + v_1 + v_0v_1k_2$, $x_2 = v_2k_0k_1 + v_1v_2k_1$ and $s = x_1x_2$. We use the IV representations of x_1 and x_2 to approximate the IV representation of s. Thus, $x_{1IV} = v_0 + v_1 + v_0v_1$, $x_{2IV} = v_2 + v_1v_2$, and $s_{IV} = x_{1IV}x_{2IV} = v_0v_2 + v_1v_2 + v_0v_1v_2$. However, $s = x_1x_2 = v_1v_2(k_0k_1 + k_1)$. We use Property **??** to determine the missing IV terms in the output ANF of Grain128, i.e., we compute the IV representation s_{IV} of the output s, find the missing IV terms of s_{IV} and those IV terms are missing IV terms of s.

Algorithm 3 Repeated-IV term Removing Algorithm

```
Input: The vector $\boldsymbol{T}$ with $n$ IV terms, i.e., $T_1$, $T_2$, …, $T_n$.
Output: Updated $\boldsymbol{T}$ with $m$ IV terms, where $m \le n$.
1: Initialize an empty Hash set H.
2: for $i \leftarrow 1 : n$ do
3:     Compute the Hash value of $T_i$, i.e., $H(T_i)$.
4:     if $H.contains(T_i)$ is false then
5:         $H.insert(T_i)$.
6:     end if
7: end for
```

After using IV representation combined with Algorithm **??**, all the existent IV terms are left by ignoring their repetition. With collision-resistent hash function H, the time complexity of Algorithm **??** is $O(n)$ for processing n IV terms. It needs several minutes to apply Algorithm **??** on 1 billion IV terms on a single core.

IV representation with Covered IV Term Removing Method We also use IV representation by removing covered IV terms. As shown in Property **??**, if we are estimating the degree of a polynomial s, instead of detecting the missing IV terms, the covered IV terms have no effect. For example, $s = v_0k_1 + v_0k_0k_2 + v_1k_1k_2 + v_0v_1k_2$, then $s_{IV} = v_0 + v_0 + v_1 + v_0v_1$, after use Property **??** to remove covered IV terms from s_{IV}, we get $s^*_{IV} = v_0v_1$. It is obviously that $\deg(s) = \deg(s_{IV}) = \deg(s^*_{IV})$. This method can help discard (state, IV) terms dramatically, since we could get a more accurate degree evaluation of s than degree evaluation technique and degree reduction technique.

Determining the 43-degree Missing IV Terms As shown in **??**, after **Step 2**, the degrees of left state terms are possibly bigger or equal to 43. For the left state terms, we are going to use IV representation to determine the missing 43-degree IV terms.

– First, we use IV representation technique combined with covered IV term removing method for each left state terms. It could maintain the degree information by ignoring the low degree IV terms and get a more accurate degree. Thus, if the degree of a monomial is smaller than 43, the monomial is deleted.

- Second, for the left monomials, we use IV representation technique combined with Algorithm **??**. It could not only maintain the highest degree IV terms, but also the lower degree IV terms by ignoring the repetitions. Then we maintain a table 43-bit indexed by all possible 43-degree IV terms who contain the 7 low-frequency IV bits in **??**. As there are 7 fixed bits, the number of freedom variables is 39, so that the table size is $\binom{46-7}{43-7} = 9139$. Initial the table with 0. Then, if a 43-degree IV term exists, the entry turns to 1. At last, the indexes with entries equal to 0 are the missing 43-degree IV terms. According to Property **??**, if an IV term is not in z_{257}'s IV representation, it must be also not in z_{257}.
- Choose the missing 43-degree IV terms as cubes shown in **??**, whose cube sums must be zero.

Table 10. 43-dimension Cubes

Cube	Index of Cube Variables
1	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,88,89,90,91
2	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,88,89,90,92
3	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,88,89,90,95
4	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,88,90,91,92
5	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,88,90,91,93
6	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,88,90,91,95
7	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,88,90,92,93
8	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,88,90,92,95
9	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,88,90,93,95
10	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,89,90,91,92
11	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,89,90,91,95
12	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,89,90,92,93
13	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,89,90,92,95
14	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,89,90,93,95
15	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,90,91,92,93
16	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,90,91,92,95
17	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,90,91,93,95
18	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,90,92,93,95
19	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,88,89,90,91,92
20	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,88,89,90,91,95
21	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,88,89,90,92,93
22	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,88,89,90,92,95
23	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,88,89,90,93,95
24	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,88,90,91,92,93
25	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,88,90,91,92,95
26	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,88,90,91,93,95
27	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,88,90,92,93,95
28	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,89,90,91,92,95
29	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,89,90,92,93,95
30	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,90,91,92,93,95
31	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,80,87,88,89,90,91,92
32	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,80,87,88,89,90,91,95
33	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,80,87,88,89,90,92,93
34	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,80,87,88,89,90,92,95
35	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,80,87,88,89,90,93,95
36	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,80,87,88,90,91,92,93
37	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,80,87,88,90,91,92,95
38	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,80,87,88,90,91,93,95
39	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,80,87,88,90,92,93,95
40	0,2,4,7,11,12,13,18,19,21,26,27,29,34,37,39,40,43,45,46,47,48,49,52,54,57,58,59,60,62,63, 65,66,72,73,74,78,80,87,88,89,90,93

6 Online Attack of the Key-recovery on Full-round Grain128

In the online attack phase, we first guess the 40 key bits and choose the dynamic bits to nullify partial state bits. For the cubes in **??**:

– When the guessed 40-bit key is right, the cube sums must be zero;
– When the guessed key is wrong, that means the nullification scheme failed and output polynomial z_{257} become very complex, the cube sums are assumed to be almost 0-1 balanced[9].

Date Collection. Since in the nullification scheme, 36 IV bits are fixed as zero. We first collect plaintext-cipertext (i.e. IV bits and z_{257}) pairs by traversing the 96-36=60 bits IV bits and store the value IV and z_{257} in a table $\mathcal{T}$. Hence, the date complexity is 2^{60}.

Algorithm 4 On-line Attack

```
 1: List all possible 40-bit key in a table $\mathcal{T}_k$.
 2: for $i$th (from 1 to 40) cube $Cube_i$ in ?? do
 3:    for Each guessed $key$ in $\mathcal{T}_k$ do
 4:       Initialize cube sum of $Cube_i$ as $Sum = 0$
 5:       for For each 43-bit IV value in 43-dimension cube $Cube_i$ do
 6:          Initialize the 96-bit IV using the nullification scheme
 7:          Choose IV and the corresponding $z_{257}$ from $\mathcal{T}$
 8:          Update $Sum = Sum + z_{257}$
 9:       end for
10:       if The $Sum$ is 1 then
11:          Delete $key$ from $\mathcal{T}_k$.
12:       end if
13:    end for
14: end for
```

Online Attack. The online attack is shown in **??**. When $i = 1$, the size of $\mathcal{T}_k$ is 2^{40}. In step 10, about half of key guessing in $\mathcal{T}_k$ makes the cube sum be 1, which is because that the cube sums are assumed to be almost 0-1 balanced under wrong key guessing. After step 12, about $2^{40}/2 = 2^{39}$ candidate key $\mathcal{T}_k$ are left. Then $i = 2$, similarly, about $2^{39}/2 = 2^{38}$ candidate key are left in $\mathcal{T}_k$ after step 12. So after $i = 40$, it is expected that only 1 key is left.

Time Complexity. Sum over the first cube needs $2^{40} \cdot 2^{43} = 2^{83}$ bit operations. After the first cube, about half wrong key guesses are dropped off, that is $(2^{40}-1)/2$ wrong key guesses and 1 right key remain. So there are $(2^{40}-1)/2+1$ guesses for the second cube and the time complexity for the second sum is $((2^{40} - 1)/2 + 1) \cdot 2^{43}$. Generally, there are $(2^{40} - 1)/2^{i-1} + 1$ key guesses for

[9] We test this property in a reduced Grain128 in **??**. For 1000 wrong 40-bit key guessing, the cube sums are almost 0-1 balanced.

the $Cube_i$. So the time complexity for the i-th sum is $2^{43}((2^{40}-1)/2^{i-1}+1)$. Totally, the time complexity is

$$\sum_{i=1}^{40} 2^{43}((2^{40}-1)/2^{i-1}+1) \approx 2^{84}.$$

According to the estimation in [?], one encryption needs at least 1000 bit operations, which is equivalent to 2^{10} Grain-128 encryptions. So the attack needs about 2^{74} cipher executions.

Data Complexity. The data complexity is 2^{60}.

After recovering the 40 key bits, there are various methods to recover the remaining key bits. For example, b_{236} can be easily nullified with 23 key guesses. Nonexistent IV terms of lower dimensions can be obtained using the techniques in previous sections. For example, missing IV terms of dimension 42 can be chosen as distinguishers. Then the complexity to recover these bits is about $23 \cdot 2^{23} \cdot 2^{42} \approx 2^{72}$ bit operations. Then the other key bits can be recovered by guessing with a complexity of 2^{65}. As a result, the complexity of our attack is dominated by the recovery of the first 40 bits.

7 Example

In order to make the attack more clear, we use an example to illustrate the process of obtaining the missing IV terms, with the same nullification and IV choosing schemes in previous sections. In order to exploit the nullification and IV choosing schemes, we choose to obtain the missing IV terms of z_{191}, where the result is very easy to verify on a single core.

We express z_{191} as the following formula after nullifications,

$$z_{191} = b_{193} + b_{206} + b_{227} + b_{236} + b_{255} + b_{264} + b_{280} + s_{284} + s_{204}s_{211} + b_{286}s_{233} + s_{251}s_{270}.$$

If we consider the missing IV terms of degree more than 11, then the other state terms except $b_{264}+b_{280}+s_{284}+b_{286}s_{233}+s_{251}s_{270}$ can be discarded directly because their degrees are less than 11, under the degree estimation in Table **??**. After that, we decompose again and preserve the state terms of degree more than 11. Then we discard those state terms that can not deduce the low-frequency bits which are shown in Table **??**. The number of state terms drops dramatically. We decompose again and preserve the state terms that are of degree more than 11 and that can deduce the low-frequency bits. After all the state bits are within the range of $[b_0, b_{159}]$ and $[s_0, s_{159}]$, we use IV representation, combined with Algorithm **??**, to obtain the missing IV terms. The highest degree is 15 and there are only 70 existent IV terms of degree 15. Furthermore, there is a large number of missing IV terms of degree 11. For example, we choose the 40 missing IV terms as distinguishers in Table **??**. Each hexadecimal number in this table indicates a multiplication of 43 IV bits. Let $H = H_0H_1H_2H_3H_4H_5H_6H_7H_8H_9$,

Table 11. Nonexistent IV terms of degree 11 in z_{191}

2008100800	2000300800	2000500800	2000900800	2000110800	2000120800	2000140800	4000500800
2000180800	2000101800	2000102800	2000104800	2000108800	2000100900	2000100A00	4000900800
2000100C00	2000100810	2000100820	2000100840	2000100880	2000100801	2000100802	4000110800
2000100804	C000100800	4100100800	4200100800	4400100800	4800100800	4010100800	4000120800
4020100800	4040100800	4080100800	4001100800	4002100800	4004100800	4008100800	4000180800

Table 12. Test results for z_{191}

Cube	Number of 0s	Number of 1s
2008100800	495	505
2000300800	486	514
2000500800	493	507
2000900800	505	495
2000110800	499	501
2000120800	468	532
2000140800	543	457
4000500800	500	500
2000180800	546	454
2000101800	479	521

where H_i is a hexadecimal number with the range of $[0, 15]$. As there are 39 bits, so H_9 is within the range of $[0, 7]$. Define h_{ij} as the j-th lowest bit of H_i. Let S be the vector whose elements are 4, 7, 11, 12, 13, 18, 19, 21, 26, 27, 29, 34, 39, 40, 45, 46, 47, 48, 49, 52, 54, 57, 58, 59, 62, 63, 65, 66, 72, 73, 74, 78, 87, 88, 89, 91, 92, 93 and 95 sequently, then the cube defined by H is $v_0 v_2 v_{37} v_{43} v_{60} v_{80} v_{90} \prod_{i\in[0,9]} v_{S_{i*4+j}}^{h_{ij}}$.

We test the results above in a PC, which are shown in Table **??**. The first column are the chosen cubes. For right key guess, sum over the cubes are zeros. For each cube, we random select 1000 keys, the second column and third column are the numbers of 0s and 1s for the corresponding cubes. The source code for the test is given in *main.cpp* in the attached file.

8 Conclusion

In this paper, we improve the attack on full-round Grain-128. Our attack is based on the knowledge that a lot of IV terms will disappear, after nullifying some state bits and IV bits. In addition, we find out the low-frequency IV bits and exploit them in the high degree terms. We also propose a series of methods to discard the monomials of lower degree, and exploit the IV representation to obtain the IV terms with much lower complexity. Then the missing IV terms are used as distinguishers so that we improved the attack in [**?**] by a factor of 2^{16}. Our attack is not based on any key information, so we can attack Grain-128 with any arbitrary selected keys. Although the missing IV terms can be tested by cube tester technique on super computers, our method can also work for higher dimensions, in which case the computing complexities for cube tester are beyond our ability.

In this paper, we have various of strategies in choosing IV bits such as choosing the low-frequency IV bits, so that the IV terms of degree 43 are very sparse. We believe that attacker can enhance the density with lower degree, which is much more complex, where more nullified IV bits and low-frequency IV bits should be exploited. So finding the lowest degree for sparse IV terms is an open problem for further research.