Determining the Nonexistent Terms of Non-linear Multivariate Polynomials: How to Break Grain-128 More Efficiently

Ximing Fu[1], and Xiaoyun Wang[1,2,3,4⋆], and Jiazhe Chen[5]

[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

Abstract. In this paper, we propose a reduction technique that can be used to determine the density of IV terms of a complex multivariable boolean polynomial. Using this technique, 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 with various of complicated reduction techniques for polynomials, so that the nonexistent 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.

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

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 [6]. 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 [3], cube tester [5,1] and dynamic cube attack [4] were proposed in order to reduce the complexities.

Stream cipher Grain-128 [7] is a refined version of Grain scheme, one of the finalists of eSTREAM Project. The output bit is a high degree boolean function in 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.

⋆ Corresponding author.

applied the statistical analysis to recover the key of reduced Grain-128 up to 180 out of 256 iterations with a complexity slightly better than brute force [5]. They pointed out that Grain-128 may be immune to this attack due to the very high degree of the polynomial of the output bit. Knellwolf et al. proposed the conditional differential cryptanalysis on NLFSR-based stream ciphers including Grain scheme ciphers [8]. Conditional differential cryptanalysis exploited the message modification technique introduced in [10,9], which controlled the diffusion by controlling the plaintexts. It was applied to recover 3 bits' 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 [8]. 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 sparsity. 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 [4] 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. [2] 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 [2] by obtaining the IV terms. The time complexity of our attack is about 2^{74} cipher executions, which is applicable to all keys of Grain128.

Our Contributions. The contributions of this paper are three fold. Firstly, we exploit the nullification technique introduced in [2] and improve the nullification by a better choice of nullified state bits and dynamic IV bits. Secondly, we propose IV controlling techniques to reduce the IV terms of high degrees. Thirdly, we give several reduction techniques for boolean functions in order to reduce the number of terms we need to process. The primary techniques are those to remove the repeated terms and covered terms. Furthermore, we present the data partition techniques so that we do the removal in parallel with super computers.

With the aforementioned techniques, we mathematically compute the IV terms of degree 43. 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. In this way, our method not only improves the previous attacks but also can naturally be applied to all keys with the same complexity.

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 exploit various of techniques to obtain the exact IV terms. We exploit two algorithms to remove the repeated and covered terms, which help reduce the polynomial dramatically. We also use IV representation to calculate 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 detail of our attack will be demonstrated in the rest of the paper. In Section 2, the outline of Grain-128 and basic concepts of cube attacks, cube tester and dynamic cube attack will be introduced. In Section 3, we present a couple of methods to control the IV terms and reduction techniques in order to reduce the number of terms to be processed. The preprocessing phase is introduced in Section 3 as well. Section 4 will give the on-line phase of the attack. Finally, Section 5 summarizes the paper.

2 Preliminaries

In this section, we will first briefly introduce Grain-128. Then the techniques related to our work will be introduced, including cube attack/tester, dynamic cube attack and nullification technique.

2.1 Outline of Grain-128

The state of Grain-128 is represented by a 128-bit LFSR and a 128-bit NLFSR. The feedback function of the LFSR and NLFSR 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 NLFSR 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 [7].

The output bit z_{257} is a boolean function over IV bits and key bits. We define in this paper the degree of the polynomial in terms of the IV variables, since the key bits are constant though they are unknown in our attack.

Next, we will give some more definitions used throughout the paper. Assuming that there are n IV bits, i.e., $v_1, v_2, \ldots, v_n$ and m key bits, i.e., $k_1, k_2, \ldots, k_m$, the output bit x can be illustrated as

$$x = \sum_{I,J} t_I g_I(k), \tag{1}$$

where t_I is the multiplication of all the IV bits whose indices are in I, i.e., it can be represented by $t_I = \prod_{i \in I} v_i$, I and J are subsets of $\{1, 2, \ldots, n\}$ and $\{1, 2, \ldots, m\}$ respectively. $g_I(k)$ is defined as the coefficient function, which is a function over key bits:

$$g_I(k) = \prod_{l_1 \in J_1} k_{l_1} + \prod_{l_2 \in J_2} k_{l_2} + \cdots + \prod_{l_s \in J_p} k_{l_p}, \tag{2}$$

where $J_1, J_2, \ldots, J_p$ are subsets of $\{1, 2, \ldots, m\}$. In the rest of the paper, we will call it coefficient for short, when there is no ambiguity. Each $t_I \prod_{l_j \in J_j} k_{l_j}$ in (1) is a term of x, and t_I is called an IV term. Note that any term has a unique IV term. But an IV term may occur in a large number of terms because $g_I(k)$ is probably very complicated.

2.2 Cube Attack and Cube Tester

Cube attack [3] 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 [3,1] to test if a coefficient is linear and which key bits are involved.

Even if the coefficient is nonlinear but only involves a small number of key bits (where the degree on key bits is low), the coefficient can also be obtained by the technique called cube tester [1]. This technique is constitute of two steps: First testing which key bits are involved in the coefficient, and then determining the specific expression (if it is linear, then the first step is enough).

2.3 Dynamic Cube Attack

In [4], 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 that are determined by key bits are chosen in order to nullify some state bits that will 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 [2] is an improvement of that in [4]. The dynamic cube attacks introduced and exploited in [4,2] are based on the nullification technique. The major difference between the two work is the different choices of nullified bits[6]. As a consequence, we will detail the nullification technique in the next subsection.

2.4 Nullification Technique

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

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

The most significant term of its ANF (algebraic normal form) is $b_{269}b_{352}s_{352}$. In fact, the terms resulted from $b_{269}b_{352}s_{352}$ are much more than those from the other terms. Similarly, the most significant 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 complicated, 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 [4] retrieved a subset of 2^{-10} of all possible keys by fixing 10 key bits to be zero.

Another approach is to simplify b_{236} by nullifying b_{203}, which is adopted by [2]. b_{203} is still too complicated 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 it is not too complicated. 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

[6] The authors also experimentally verified the main component of the attack by a dedicated hardware in [2]

Table 1. Nullification Scheme in [2]

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	$s_3, s_5, s_6, s_{77}, s_8, s_9, s_{10}, s_{17}, s_{25}, s_{31}, s_{42}, s_{83}, s_1$

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 [2] are shown in Table 1.

3 Obtaining the IV terms of Boolean Polynomials

In this paper, the main purpose is to obtain all the IV terms of degree 43. In order to achieve this goal, we first propose new nullification (Section 3.1) and IV choosing (Section 3.2) techniques. Then term reduction techniques (Section 3.3) are also presented to remove the terms that will not contribute to the IV terms of degree 43 in the polynomial. With these techniques, we are able to obtain the exact IV terms of degree 43 in the output bit in Section 3.4.

3.1 Nullification of State Bits

Motivated by the nullification technique in [4,2], we choose to nullify some state bits. In order to simplify $b_{269}b_{352}s_{352}$, we also need to nullify b_{203}. In addition to the nullified state bits in Table 1, b_{143} is also nullified in our attack in order to simplify b_{259}, where

$$\begin{aligned}b_{259} =& b_{143}b_{226}s_{226} + b_{226}s_{173} + b_{199}b_{215} + s_{191}s_{210} + b_{192}b_{196} + b_{171}b_{179} + b_{158}b_{190}\\ &+ b_{148}b_{149} + b_{142}b_{144} + b_{134}b_{198} + s_{144}s_{151} + b_{143}s_{139} + b_{227} + s_{224} + b_{222}+\\ &b_{220} + b_{204} + b_{195} + b_{187} + b_{176} + b_{167} + b_{157} + b_{146} + b_{133} + b_{131} + s_{131}.\end{aligned}$$

Note that the nullification of b_{143} only leads to 1 more guessed key bit. The state bits to be nullified are shown in Table 2.

Next we will explain why we choose to nullify these state bits. First, in order to nullify b_{203}, some other bits should be nullified. We substitute the terms and preserve the terms with high degree, and then calculate the frequency of occurrence of each state bit involved. Nullifying the high frequency state bits and setting them to 0, then the corresponding terms will disappear. As a result, the nullification of these state bits decreases the degree and terms of high degree dramatically. Then we substitute again and repeat the procedure above. Finally we can determine the dynamic IV bits and the corresponding key bits that have to be guessed during the on-line attack. The dynamic IV bits are shown in Table 2 as well. The more details of the nullification are shown in Table 7 in Appendix A.

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	$s_3, s_5, s_6, s_{77}, s_8, s_9, s_{10}, s_{15}, s_{17}, s_{25}, s_{31}, s_{42}, s_{83}, s_1$

3.2 IV Choosing Techniques

After the nullifications shown in Table 2, there are 82 IV bits left, which probably leads to a very high degree of the output bit polynomial. Thus in this subsection we will show how to choose some IV bits to reduce the degree and make some IV terms disappear as well.

For Grain-128, the output is generated by iteration of IV and key bits, some IV or key bits tend to (dis)appear simultaneously in high degree terms. For example, for state bits in the first 32 initialization rounds, the only degree 2 terms are $s_{i+13}s_{i+20}$ and $s_{i+60}s_{i+79}$. Choosing to nullify s_{i+13} may result in disapperance of s_{i+20}.

Furthermore, the degrees of some state bits are decreased to 1 by nullifying some IV bits of the terms. However, some state bits may have higher degrees due to the higher-degree dynamic bits. As a result, the high frequency IV bits in the high frequency state bits and terms are chosen to be nullified, in order to make as many dynamic bits disappear as possible.

Since some IV bits will disappear in high degree terms after setting some other IV bits to zero, IV terms with these IV bits such as s_{80} will be sparse in high degree terms. We call these common IV bits that lead to sparse terms *low-frequency bits*. We find out at least 7 of them, which are listed in Table 5 along with the corresponding state bits in the first 32 rounds.

In addition, the IV nullification is a iterative process since nullifying any IV bit may result in changing the degree of many state bits.

In summary, carefully choosing the nullified IV bits will result in the disappearance of many IV terms. The fact of the disappearance of IV terms can be used to derive distinguishers.

3.3 Reduction Techniques for Polynomial Terms

We presented the methods to reduce the boolean function of the output bit in Section 3.1 and Section 3.2. However, the specific IV terms that appear are under consideration, so that distinguishers can be deduced. Thus we propose in this subsection the manners to obtain the IV terms of boolean functions of Grain-128.

Degree estimation of state bits After the nullification and IV choosing scheme introduce in Section 3.1 and Section 3.2, we first estimate the degree of some state bits. Degrees of some state bits are shown in Table 3, which can be

Table 3. Degree of partial state bits

i	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146
$\deg(b_i)$	1	2	1	0	1	0	0	1	0	0	0	1	2	2	2	0	1	0	1
$\deg(s_i)$	1	2	2	1	1	1	1	0	1	1	1	1	2	1	2	0	1	1	1
i	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165
$\deg(b_i)$	1	1	1	0	2	1	0	2	2	1	1	1	0	2	2	3	3	2	2
$\deg(s_i)$	1	1	1	1	2	1	1	2	2	1	1	1	0	2	2	3	3	2	2
i	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184
$\deg(b_i)$	2	2	2	1	0	1	2	3	3	3	0	2	2	2	2	2	2	2	3
$\deg(s_i)$	1	2	2	1	1	1	2	3	3	3	2	2	2	2	2	2	2	2	3
i	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203
$\deg(b_i)$	2	2	2	3	2	2	2	2	3	3	5	4	3	3	3	3	3	3	0
$\deg(s_i)$	2	2	2	3	2	2	2	2	3	3	5	4	3	3	3	3	3	3	2
i	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222
$\deg(b_i)$	3	3	5	4	4	3	4	4	4	3	3	4	3	5	4	4	4	5	5
$\deg(s_i)$	3	3	5	4	4	3	4	4	4	3	3	4	3	5	4	4	4	5	4

Table 4. $\deg(b_i) + \deg(s_i) - \deg(b_i s_i)$

i	129	140	142	151	154	155	160	161	162	163	164	165	167
$\deg(b_i) + \deg(s_i) - \deg(b_i s_i)$	1	1	1	1	1	1	1	1	1	2	1	1	1
i	168	172	173	174	175	180	181	182	183	184	188	193	194
$\deg(b_i) + \deg(s_i) - \deg(b_i s_i)$	1	1	1	2	2	1	1	1	1	1	1	2	1
i	195	196	197	198	199	206							
$\deg(b_i) + \deg(s_i) - \deg(b_i s_i)$	2	3	1	1	1	1							

obtained directly on PC by obtaining their boolean function in IV bits and key bits. In fact, the exact degree of state bits before round 150 can be obtained, while the others can be estimated by substitution.

In addition, we can obtain $\deg(b_i) + \deg(s_i) - \deg(b_i s_i)$, which can help reduce terms in advance and is shown in Table 4.

Remove the Repeated Terms We know that it is much easier to illustrate the output with the state bits than using the key and IV bits directly. Moreover, the state bits can be expressed by earlier ones that are simpler functions of the IV and key bits. As a consequence, we can iteratively express the output bits. During this procedure, we will try to reduce the complexity of the polynomial by removing the so called **repeated terms** (RT).

Two terms are repeated terms if they are the same by ignoring s_i ($i \in [96, 127]$) (which are all 1s). It is obvious that the boolean addition of two RT is zero.

Before introducing our RT algorithm, we will introduce a simple but useful property for Grain-128.

Table 5. Low frequency bits

low-frequency bits	corresponding state bits
s_0	$b_{128}, s_{128}, b_{160}, s_{160}$
s_2	b_{130}, s_{130}
s_{37}	$b_{157}, s_{157}, s_{158}$
s_{43}	s_{133}
s_{60}	$b_{146}, s_{146}, s_{150}$
s_{80}	$b_{129}, s_{129}, s_{138}, b_{148}, s_{148}$
s_{90}	$s_{137}, s_{148}, s_{158}$

Property 1. Let $b_{i+128} = b'_{i+128} + z_i$ and $s_{i+128} = s'_{i+128} + z_i$, then $b_{i+128}s_{i+128} = b'_{i+128}s'_{i+128} + z_i(b'_{i+128} + s'_{i+128}) + z_i$, where z_i is the feedback bit, b'_{i+128} and s'_{i+128} are the state bits before the feedback of z_i.

Property 1 holds because there is a collision, i.e., $z_i \cdot z_i = z_i$, which can result in a number of repeated terms when substituting the terms of z_i. Actually we find that the repeated term are mostly caused from collisions. Property 1 is trivial, however, it helps a lot when we remove the RT with computer program but not manually before our RT removing algorithm, which is shown in Algorithm 1. Suppose that there are n terms, the complexity of Algorithm 1 is $O\left(n^2\right)$.

Algorithm 1 Repeated-term Removing Algorithm

Input: The vector $\boldsymbol{T}$ with n terms, i.e., T_1, T_2, ..., T_n.
Output: Updated $\boldsymbol{T}$ with m terms, where $m \leq n$.

```
for i ← 1 : n − 1 do
    for j ← i + 1 : n do
        if (T_i = T_j) then
            T_i ← 0
            T_j ← 0
        end if
    end for
end for
m ← 1
for i ← 1 : n do
    if T_i ≠ 0 then
        T_m ← T_i
        m ← m + 1
    end if
end for
```

In our attack, we hope to use Algorithm 1 as much as possible. However, the number of terms can grow dramatically in the process of substitution even though we have removed some of them. Applying Algorithm 1 to one million terms needs about half a day with a single core, which is practical. However, there are billions of terms; the complexity is beyond the power of a single PC.

In this paper, we present a method to partition the terms into non-overlapping sets so as to apply Algorithm 1 in parallel. In order to guarantee that all the repeated terms can be removed, we propose to partition the terms according to the common factor. The reason is the fact that two repeated terms must have common factor(s). Then the terms can be partitioned into different sets according to the common factor, which is shown in Algorithm 2.

Algorithm 2 Term Partition Algorithm

Input: The vector $\boldsymbol{T}$ of n terms, i.e., T_1, T_2, …, T_n.
Output: m sets $\mathfrak{s}_1, \mathfrak{s}_2, \ldots, \mathfrak{s}_m$ with $n_1, n_2, \ldots, n_m$ terms.

```
for k ← 1 : m do
    Calculate the occurrence frequency of each state bit in T.
    Select a state bit x with frequency n_k that is closest to n/(m−(k−1)).
    Include the terms with common factor x in set s_k.
    n ← n − n_k
end for
```

The complexity of Algorithm 2 is

$$m \times \frac{n}{m} + (m-1) \times \frac{n}{m} + \cdots + \frac{n}{m} = \frac{m+1}{2} n.$$

If n is large, then m may be also large because the number of terms in a set cannot be too large due to the computing ability of a single core. Then we refine the partition scheme as follows. Suppose that n terms are going to be partitioned into $m = m_1 \times m_2$ sets, n terms can be partitioned into m_1 sets first, then each set can be partitioned into m_2 sets. The complexity of the first partition is $\frac{m_1+1}{2}n$ and the complexity of the second partition is $\frac{(m_2+1)n}{2m_1}$ on average in a single core.

After partitioning n terms into m sets. Algorithm 1 is applied to each set, which will run in parallel with m cores. The total computing complexity on m cores is about $\frac{1}{m}$ of the original total complexity.

Identifying the IV Terms of Degree 43 Removing the RT is not enough since the number of terms still increases dramatically, which pushes us to reduce the terms furthermore.

IV representation. Replacing the terms with their corresponding IV terms is called *IV representation*. Due to the neglect of the key information, using IV representation will result in repeated IV terms.

Repeated IV terms. When two IV terms are the same, then one of them can be removed directly, which is shown in Algorithm 3. In Line 4 of Algorithm 3, the repeated IV term is set to -1 but not 0 so that the constant information will not be lost.

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 $m \leq n$.

for $i \leftarrow 1 : n-1$ **do**
  for $j \leftarrow i+1 : n$ **do**
    if $(T_i = T_j)$ **then**
      $T_i \leftarrow -1$
    end if
  end for
end for
$m \leftarrow 1$
for $i \leftarrow 1 : n$ **do**
  if $T_i \neq -1$ **then**
    $T_m \leftarrow T_i$
    $m \leftarrow m+1$
  end if
end for

Covered IV terms. In addition to repeated IV terms, we give the definition of covered IV terms: If $I \subseteq J$, then IV term $s_1 = \prod_{i \in I} v_i$ is said to be covered by IV term $s_2 = \prod_{j \in J}$, where I and J are subsets of $\{0, 1, \ldots, 95\}$ respectively. So repeated IV terms are special cases of covered IV terms.

Covered IV terms will show up frequently using IV representation, so we develop an algorithm to remove the covered IV terms, which is shown in Algorithm 4.

Here we give an example to illustrate the use of IV representation, repeated IV term removing and covered IV term removing. Assume that $x_1 = v_0(k_1 + k_0k_2) + v_0v_1k_2$, $x_2 = v_1k_0 + v_1v_2k_1$, then the IV representations of x_1 and x_2 are $v_0 + v_0v_1$ and $v_1 + v_1v_2$ respectively. After removing the repeated IV terms of $x_1x_2 = v_0v_1 + v_0v_1v_2 + v_0v_1 + v_0v_1v_2$ using Algorithm 3, the resultant IV terms are v_0v_1 and $v_0v_1v_2$, so that the IV terms can be determined. If only the highest degree is under consideration, then Algorithm 4 can be used. After removing the covered IV terms, $\hat{x}_1 = v_0v_1$, $\hat{x}_2 = v_1v_2$. Then $\hat{x}_1\hat{x}_2 = v_0v_1v_1v_2 = v_0v_1v_2$. So the highest degree of x_1x_2 is 3, within the IV term $v_0v_1v_2$. Using the IV representation and Algorithm 4, only 1 IV multiplication is needed for this concrete example, while 4 IV multiplication and 6 key multiplication are needed in the trivial way.

Since the coefficient function on key bits is usually quite complicated, the proposed method reduces the computing complexity dramatically, which enables us to obtain the IV terms of degree 43.

The computational complexity of Algorithm 3 and Algorithm 4 is $O(n^2)$ if there are n IV terms. This is the worst-case complexity; while for Grain-128 the complexity will be much lower due to the fact that there are large number of repeated IV terms and that an higher degree term may cover a larger number of lower degree ones. Normally, processing 10 million IV terms by Algorithm 3

Algorithm 4 Covered-(IV)term Removing Algorithm

Input: The vector $\boldsymbol{T}$ of n terms, i.e., T_1, T_2, …, T_n.
Output: Updated $\boldsymbol{T}$ with m terms, where $m \leq n$.

for $i \leftarrow 1 : n-1$ **do**
    if $T_i \neq 0$ **then**
        for $j \leftarrow i+1 : n$ **do**
            if $T_j \neq 0$ **then**
                if (T_i is covered by T_j) **then**
                    $T_i \leftarrow 0$
                    $i \leftarrow i+1$
                else if (T_j is covered by T_i) **then**
                    $T_j \leftarrow 0$
                    $j \leftarrow j+1$
                end if
            end if
        end for
    end if
end for
$m \leftarrow 1$
for $i \leftarrow 1 : n$ **do**
    if $T_i \neq 0$ **then**
        $T_m \leftarrow T_i$
        $m \leftarrow m+1$
    end if
end for

and 30 million IV terms by Algorithm 4 only needs about several minutes on a single core, which is quite efficient.

It is obvious that Algorithm 3 will not lose any information of IV terms. Now we have to make sure that the above method does not lose any information about the degree, i.e., the degree of the original polynomial will be bounded by the deduced degree.

Property 2. The degree of the multiplication of two state bits is strictly bounded by the estimated one deduced by *IV representation* and Algorithm 4.

Proof. This proof is composed of two steps. First, we need to prove that IV representation of the state bits with Algorithm 4 will guarantee the keeping of the IV terms with the highest degree. Then, we need to prove that applying Algorithm 4 on the multiplication of two IV-represented state bits will guarantee the keeping of the IV terms with the highest degree. It is obvious right for the latter one, we just need to prove the former one.

Consider state terms $A = a_1 + a_2 + \cdots + a_m$ and $B = b_1 + b_2 + \cdots + b_n$, where a_i and b_j are IV terms. Assuming that a_{i_1} is covered by a_{i_2}, then $a_{i_1}B$ will be covered by $a_{i_2}B$. Hence, a_{i_1} can be removed from A, which is done by Algorithm 4. The removal of IV terms in B is similar.□

This property can be easily extended to the multiplication of multiple state bits, which holds by iteratively using Property 2.

Actually, Algorithm 4 can be used to remove the so called **covered terms** (CT), where CT is defined as follows:

Let state terms

$$t_1 = \prod_{i_1 \in I_1} b_{i_1} \prod_{i_2 \in I_2} b_{i_2} \prod_{i_3 \in I_3} s_{i_3}$$

and

$$t_2 = \prod_{j_1 \in J_1} b_{j_1} \prod_{j_2 \in J_2} b_{j_2} \prod_{j_3 \in J_3} s_{j_3},$$

where $I_1, J_1 \subseteq [0, 127]$, $I_2, J_2 \subseteq [128, 352]$, $I_3, J_3 \subseteq [0, 95] \bigcup [128, 352]$. Term t_1 is covered by t_2 if $I_2 \subseteq J_2$ and $I_3 \subseteq J_3$.

If the degree of a state term can be bounded by a bound using *IV representation*, then IV terms produced by the covered state terms will also be covered and hence will be removed using Algorithm 4. So covered state terms can be removed first. As a result, the state terms are partitioned into two sets, of which degrees of state terms in the first one are bounded by our bound while degrees of state terms in the other may be higher. Of course, most state terms will be in the first set. Algorithm 4 can be applied to the first set and most terms would be dropped off.

When there are billions of terms, removing the covered terms by executing Algorithm 4 on all terms with a single core is difficult as well, so we remove the covered terms in parallel. Similarly, here we propose a partition scheme that guarantees all covered terms can be removed. A term covers another if only the degree of the first is no less than that of the second. So the terms can be partitioned into different sets according to their degrees. First we apply Algorithm 4 to the highest degree terms, and remove the second highest degree terms that are covered by the highest ones. Then we operate Algorithm 4 on second highest degree terms, and so on, until all covered terms are removed. Algorithm 4 is much faster than Algorithm 1 although they have the same computational complexity in worst case. Normally, operating Algorithm 4 on 10 million terms needs just several minutes in a single core when analyzing Grain-128. This is because a term may cover a large number of terms. Therefore, these terms will be set to 0 after a scan of all the terms. In fact, two terms are repeated terms only if they have the same degree. So the partition scheme for covered terms can also be used to partition terms into different sets with no repeated terms. However, removing repeated terms in this way will lead to a higher complexity since it cannot partition the terms uniformly. Thus we will not use it for repeated terms removing.

For state terms in the second set, we need to substitute the terms and remove the repeated terms using Algorithm 1 until the IV terms can be obtained using *IV representation*. Combined with Algorithm 3, all IV terms of degree 43 can be obtained.

3.4 Preprocessing Phase of the Proposed Attack

Now we are ready to describe the preprocessing phase of our attack, using the techniques proposed in the previous three subsections.

1. We deduce the key bits to be guessed (the number is 40), as well as the corresponding dynamic IV bits, to nullify the state bits shown in Table 2. The nullified IV bits and low-frequency bits shown in Table 6 are also chosen.
2. Iteratively express the output bit and discard the terms whose degrees are likely to be below certain threshold. Algorithm 1 is then used to remove the repeated terms. Note that there is no information lost in Algorithm 1.
3. Use Algorithm 4 to reduce the number of state terms, and to deduce the IV terms that may exist.
4. Obtain the IV terms that may exist using IV representation, combined with Algorithm 3 and Algorithm 4.

In Step 2, we will remove as many repeated terms as possible within our computing and storage ability. After Step 2, a large number of state terms are discarded. However, most of the terms are of degrees lower than our bound, which is actually proved by using Algorithm 4.

The preprocessing process is quite complicated, so we use a computer cluster with 740 nodes (8880 cores in total) to do most of the time consuming work. A dynamic number of cores are used (which are between 600 and 4000), depending on the specific program to be paralleled. Finally, we can determine all the IV terms that may appear, which are shown in Appendix B. The IV terms

Table 6. Nullified and low-frequency IV bits.

nullified bits	$s_{14}, s_{16}, s_{20}, s_{22}, s_{23}, s_{24}, s_{28}, s_{30}, s_{32}, s_{33}, s_{35}, s_{36}, s_{38}, s_{41}, s_{44}, s_{50}, s_{51}, s_{53}$ $s_{55}, s_{56}, s_{61}, s_{64}, s_{67}, s_{68}, s_{69}, s_{70}, s_{71}, s_{75}, s_{76}, s_{79}, s_{81}, s_{82}, s_{84}, s_{85}, s_{86}, s_{94}$
low-frequency bits	$s_0, s_2, s_{37}, s_{43}, s_{60}, s_{80}, s_{90}$

in Appendix B occur with probability 1/2 while the others do not occur with probability 1 in z_{257}.

4 On-line Phase of the Attack

In this section, we will introduce the on-line phase of our attack on Grain-128 and analyze its complexity. Actually, the on-line phase of our attack is much simpler than the preprocessing phase.

Since we determine that totally 2581 IV terms of degree 43 are possible to be existent, which means there are $C_{46-7}^{43-7} - 2581 = 6558$ nonexistent IV terms. The density of IV terms is about 28%. We choose cubes from these nonexistent IV terms. It is known that summing over each of the cubes, the output will be always zero, which results in distinguishers for key recovery. Since for the correct key guess the summation is always 0, while for the wrong ones the summation could be 0 or 1 with random probability. On average, $40 \log 40 \approx 213$ cubes are needed to get the correct key.

Thus, in this phase, we first guess the 40 key bits and sum over the output bits with the first chosen cube. If the result is 1, then we conclude that the guess is wrong. About half of the keys will be discarded in this way. For the remaining keys, we repeat the procedure with the second cube. And so on. Then after all the 213 cubes are used, it is supposed that only the correct key will be kept.

The time complexity is about $2^{43}\left(2^{40} + 1 + \frac{2^{40}-1}{2} + 1 + \frac{2^{40}-1}{2^2} + \cdots 1 + \frac{2^{40}-1}{2^{213}}\right) \approx 2^{84}$ bit operations, equivalent to 2^{74} cipher executions. The data complexity is $213 \cdot 2^{14} \cdot 2^{43} \approx 2^{65}$.

After recovering the 40 key bits, there are various of methods to recover the remaining key bits. For example, b_{236} can be easily nullified with 23 key guesses. Then cubes of dimension 42 can be chosen as distinguishers. So 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.

5 Conclusion

In this paper, we improved 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 method to cancel the terms with lower degree, and exploit the IV representation to obtain the IV terms with much lower complexity. Then the nonexistent IV terms are used as distinguishers so that we improved the attack in [2] 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 nonexistent 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 sparsity with lower degree, which is much more complicated. So finding the lowest degree for sparse IV terms is an open problem for further research.

References

1. Aumasson, J., Dinur, I., Meier, W., Shamir, A.: Cube testers and key recovery attacks on reduced-round MD6 and Trivium. In: Fast Software Encryption, 16th International Workshop, FSE 2009. pp. 1–22. Springer (2009)
2. Dinur, I., Güneysu, T., Paar, C., Shamir, A., Zimmermann, R.: An experimentally verified attack on full Grain-128 using dedicated reconfigurable hardware. In: Advances in Cryptology-ASIACRYPT2011. pp. 327–343. Springer (2011)
3. Dinur, I., Shamir, A.: Cube attacks on tweakable black box polynomials. In: Joux, A. (ed.) Advances in Cryptology–EUROCRYPT2009. LNCS, vol. 5479, pp. 278–299. Springer, Heidelberg (2009)
4. Dinur, I., Shamir, A.: Breaking Grain-128 with dynamic cube attacks. In: Fast Software Encryption - 18th International Workshop, FSE 2011. pp. 167–187 (2011)
5. Fischer, S., Khazaei, S., Meier, W.: Chosen IV statistical analysis for key recovery attacks on stream ciphers. In: Progress in Cryptology - AFRICACRYPT 2008. pp. 236–245. Springer (2008)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
7. Hell, M., Johansson, T., Maximov, A., Meier, W.: A stream cipher proposal: Grain-128. In: IEEE International Symposium on Information Theory (ISIT 2006) (2006)
8. Knellwolf, S., Meier, W., Naya-Plasencia, M.: Conditional differential cryptanalysis of NLFSR-based cryptosystems. In: Advances in Cryptology-ASIACRYPT2010. pp. 130–145. Springer (2010)
9. Wang, X., Yin, Y.L., Yu, H.: Finding collisions in the full SHA-1. In: Advances in Cryptology - CRYPTO 2005. pp. 17–36. Springer (2005)
10. Wang, X., Yu, H.: How to break MD5 and other hash functions. In: Advances in Cryptology-EUROCRYPT 2005. pp. 19–35. Springer (2005)

A The details of nullifications

The nullification detail is shown in Table 7. The first column is the state bits to be nullified and the second column is the corresponding equation. The third column is 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 2.

B All IV terms of degree 43 in z_{257}

The resulted IV terms that may appear are shown in Table 8, 9, 10, 11, 12and 13. Each hexadecimal number in this table indicates a multiplication of 43 IV bits. Let $H = H_0H_1H_2H_3H_4H_5H_6H_7H_8H_9$, 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 $v_0v_2v_{37}v_{43}v_{60}v_{80}v_{90}\prod_{i\in[0,9]} v_{S_{i*4+j}}^{h_{ij}}$. We use this expression due to the convenience for programming.

Table 7. Nullification equations

Nullified bits	Equations for nullification	Subkey bits guessed
b_{203}	$s_1 = b_{115}b_{123} + b_{92}b_{93} + b_{86}b_{88} + s_{88}s_{95} + \underline{b_{87}}s_{83} + b_{111}b_{127} + b_{104}b_{108} + b_{83}b_{91} + b_{48}$ $+b_{60}b_{61} + b_{54}b_{56} + b_{46}b_{110} + s_{56}s_{63} + b_{107} + b_{99} + b_{88} + b_{79} + b_{58} + b_{45} + b_{43} + s_{14}s_{21}$ $+s_{43} + \underline{b_{52}}s_{48} + b_{113} + b_{104} + b_{85} + b_{55} + b_{42} + s_{78} + s_{47} + s_{40} + b_{106}b_{122} + b_{99}b_{103} + b_5$ $+b_{78}b_{86} + b_{65}b_{97} + b_{55}b_{56} + b_{49}b_{51} + b_{41}b_{105} + b_{50}s_{46} + b_{102} + b_{83} + b_{64} + b_{53} + \underline{b_{96}}s_{43}$ $+b_{40} + b_{104}b_{120} + b_{97}b_{101} + b_{76}b_{84} + b_{63}b_{95} + b_{53}b_{54} + b_{47}b_{49} + b_{39}b_{103} + b_{120} + s_3\underline{b_{13}b_{96}}$ $+b_{101} + b_{90} + b_{125} + b_{109} + b_{100} + b_{92} + b_{81} + b_{72} + b_{62} + b_{51} + b_{36} + b_{77} + b_{75} + s_{82}$ $+s_{75} + b_{32}b_{115} + \underline{b_{115}}s_{62} + b_{88}b_{104} + s_{80} + b_{81}b_{85} + b_{60}b_{68} + b_{47}b_{79} + b_{37}b_{38} + s_{39} + s_8$ $+b_{31}b_{33} + b_{23}b_{87} + b_{111} + b_{109} + b_{93} + b_{84} + b_{76} + b_{65} + b_{56} + b_{46} + b_{70}b_{102} + b_{16} + b_3$ $+b_{35} + b_{22} + b_{20} + \underline{b_{100}}s_{47} + s_{18}s_{25} + b_{17}s_{13} + b_{78} + b_{50} + b_{41} + \underline{b_{13}}s_9 + s_{94} + b_{90} + b_{65}$ $+b_{20} + b_7 + s_{101} + s_{86} + s_{43} + s_{12} + s_5\underline{b_{15}b_{98}} + \underline{b_{98}}s_{45} + \underline{b_{15}}s_{11} + b_{92} + b_{67} + b_{46} + b_{37}$ $+b_{18} + b_{39} + s_{73} + s_{10} + 1$	$b_{87}, b_{52}, b_{115}, b_{96},$ $b_{13}, *$
b_{176}	$s_{83} = s_{48} + b_{48} + b_{74} + b_{104} + s_{11} + b_{11} + b_{37} + b_{67} + b_{14}b_{78} + b_{22}b_{24} + b_{28}b_{29} + b_{38}b_{70}$ $+b_{51}b_{59} + b_{72}b_{76} + b_{79}b_{95} + b_{13} + b_{26} + b_{47} + b_{56} + b_{75} + b_{100} + s_{19}\underline{b_{23}} + b_{23}b_{106} + b_{16}$ $+b_{42} + b_{72} + b_{19}b_{83} + b_{27}b_{29} + b_{33}b_{34} + b_{43}b_{75} + b_{56}b_{64} + b_{77}b_{81} + b_{84}b_{100} + b_{18} + b_{31}$ $+b_{52} + b_{61} + b_{80} + b_{89} + b_{105} + b_{28}b_{111} + s_{58}\underline{b_{111}} + b_{51}b_{115} + b_{59}b_{61} + b_{65}b_{66} + b_{75}b_{107}$ $+b_{88}b_{96} + b_{109}b_{113} + s_4b_{116} + b_4b_{116} + b_{30}b_{116} + b_{60}b_{116} + b_{95}b_{116} + b_{100}b_{116} + b_7b_{71}b_{116}$ $+b_{15}b_{17}b_{116} + b_{21}b_{22}b_{116} + b_{31}b_{63}b_{116} + b_{44}b_{52}b_{116} + b_{65}b_{69}b_{116} + b_{72}b_{88}b_{116} + b_6b_{116}$ $+b_{19}b_{116} + b_{40}b_{116} + b_{49}b_{116} + b_{68}b_{116} + b_{77}b_{116} + b_{93} + b_{93}b_{116} + s_{12}\underline{b_{16}b_{116}} + b_{16}b_{99}b_{116}$ $+b_{116} + s_{46}\underline{b_{99}b_{116}} + b_{50} + b_{63} + b_{121} + s_{13} + b_{15} + b_{28} + b_{49} + b_{58} + b_{77} + b_{86} + s_{21}\underline{b_{25}}$ $+b_{25}b_{108} + s_{73}s_{92}$	$b_{23}, b_{111}, b_{16}b_{116},$ $b_{25}, b_{99}b_{116}, *$
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 8. IV terms of degree 43 in z_{257}-part 1

FFFFFFDFB6	FFBFFFFFB6	FFFFFFDBB7	FFFFFFDFA7	FFDFFFDFB7	FFBFFFFBB7
FFBFFFFFA7	FF9FFFFFB7	BFFFFFDFB7	FDFFFFDFB7	BFBFFFFFB7	FDBFFFFFB7
FFFFFFD7B7	FFBFFFF7B7	FFFFBFDFB7	FFFFFDDFB7	FFBFBFFFB7	FFBFFDFFB7
FFBFFFDFB7	FFFFFFDFB3	F7FFFFDFB7	FEFFFFDFB7	FFBFFFFFB3	F7BFFFFFB7
FEBFFFFFB7	EFFFFFDFB7	EFBFFFFFB7	FFFFFFDF97	FFFFDFDFB7	FFEFFFDFB7
FFFFFFDEB7	7FFFFFDFB7	FFBFFFFF97	FFBFDFFFB7	FFAFFFFFB7	FFBFFFFEB7
7FBFFFFFB7	BFEFFFEFF7	9FFFFFEFF7	9FEFFFFFF7	DFEFFFEFF7	BFFDFFEFF7
BFEDFFFFF7	9FFDFFFFF7	DFFDFFEFF7	DFEDFFFFF7	FFEFFFFEF6	FFEFFFFCF7
FFFFFFF6F6	FFFFBFFEF6	FFFFFDFEF6	FFFFFFFEF2	F7FFFFFEF6	FEFFFFFEF6
FFFFFFEEF6	FFFFFFFEE6	FFFDFFFEF6	FFBFFFFEF6	FFFFFFFED6	FDFFFFFEF6
FFFFFFDEF6	FFFFDFFEF6	FFFFF7FEF6	FFFFFFFCF3	FFFFFFECF7	FFFFFFFCE7
FFFDFFFCF7	F7FFFFFCF7	FEFFFFFCF7	FFFFFFF4F7	FFFFBFFCF7	FFFFDFFCF7
FFFFFDFCF7	FFBFFFFCF7	FFEDFFEFF7	FFFFFFFC77	FFEFDFFDF7	FFEFFFFDF3
7FEFFFFDF7	FFEFFFF9F7	EFEFFFFDF7	F7EFFFFDF7	FFEFFFF5F7	FFEFBFFDF7
FEEFFFFDF7	FFEDFFFDF7	FFAFFFFDF7	FFEFFFFDE7	FFCFFFFDF7	FFEFFDFDF7
7FFFFFFDF3	FFFFFFF9F3	EFFFFFFDF3	FFFFFFF5F3	FFFFBFFDF3	FFDFFFFDF3
FFFFFDFDF3	FEFFFFFDF3	FFFFFFFDE3	FFFDFFFDF3	FFBFFFFDF3	FFFFDFFDF3
7FFFFFFDE7	7FFDFFFDF7	77FFFFFDF7	7EFFFFFDF7	FFFFFFF9E7	FFFDFFF9F7
F7FFFFF9F7	FEFFFFF9F7	EFFFFFFDE7	EFFDFFFDF7	E7FFFFFDF7	EEFFFFFDF7
FFFDFFF5F7	FFFDBFFDF7	FFFDFDFDF7	7FFFFFF5F7	FFFFFFF1F7	EFFFFFF5F7
7FFFBFFDF7	FFFFBFF9F7	EFFFBFFDF7	7FFFDFFDF7	7FFFFDFDF7	7FBFFFFDF7
FFFFDFF9F7	FFFFFDF9F7	FFBFFFF9F7	EFFFDFFDF7	EFFFFDFDF7	EFBFFFFDF7
F7FFFFF5F7	FEFFFFF5F7	F7FFBFFDF7	FEFFBFFDF7	FFFDFFFDE7	FFDFFFFDE7
FFDDFFFDF7	FFBDFFFDF7	F6FFFFFDF7	F7FDFFFDF7	FEFDFFFDF7	F7BFFFFDF7
FEBFFFFDF7	F7FFFFFDE7	FEFFFFFDE7	F7DFFFFDF7	FEDFFFFDF7	F7FFFDFDF7
FEFFFDFDF7	F7FFDFFDF7	FFFFDFF5F7	FFFFFFF5E7	FFDFFFF5F7	FFBFFFF5F7
FFFF9FFDF7	FFFFBFFDE7	FFDFBFFDF7	FFBFBFFDF7	FEFFDFFDF7	FFFDDFFDF7
FFBFDFFDF7	FFFFDFFDE7	FFDFDFFDF7	FFFFDDFDF7	FFFFFDFDE7	FFDFFDFDF7
FFBFFFFDE7	FF9FFFFDF7	FFBFFDFDF7	FFEFFFEEF7	FFEDFFFEF7	FFFDFFEEF7
DFEFFFFEF7	DFFFFFEEF7	DFFDFFFEF7	FBFFDFFEF7	FBEFFFFEF7	FBFFFFEEF7
FBFDFFFEF7	FBFFFFFAF7	EBFFFFFEF7	FBFFFFF6F7	FBFFFFFEF3	FBFFFFFEE7
FBDFFFFEF7	FBFFFDFEF7	FBBFFFFEF7	F3FFFFFEF7	FAFFFFFEF7	FBFFBFFEF7
DBFFFFFEF7	F9FFFFFEF7	FBFFFFDEF7	FBFFFFFED7	FBFFF7FEF7	BFFFDFFEF7
BFEFFFFEF7	BFFFFFEEF7	BFFDFFFEF7	BFFFFFFAF7	AFFFFFFEF7	BFFFFFF6F7
BFFFFFFEF3	BFFFFFFEE7	BFDFFFFEF7	BFFFFDFEF7	BFBFFFFEF7	B7FFFFFEF7
BEFFFFFEF7	BFFFBFFEF7	9FFFFFFEF7	BDFFFFFEF7	BFFFFFDEF7	BFFFFFFED7
BFFFF7FEF7	FFEFDFFEF7	FFFFDFEEF7	FFFDDFFEF7	DFFFDFFEF7	FFFFFFFE76
FFFFF7FF76	7FEFFFFEF7	7FFFFFEEF7	7FFDFFFEF7	FFEFFFFAF7	FFFFFFEAF7
FFFDFFFAF7	7FFFFFFAF7	EFEFFFFEF7	EFFFFFEEF7	EFFDFFFEF7	6FFFFFFEF7
EFFFFFFAF7	FFEFFFF6F7	FFFFFFE6F7	FFFDFFF6F7	FFEFBFFEF7	FFFFBFEEF7
FFFDBFFEF7	FFDFFFFEF3	FFFFFDFEF3	FFEFFFFEF3	FFFFFFEEF3	FFFFFFFEE3
FFFDFFFEF3	FFEFFFFEE7	FFDFFDFEF7	FFCFFFFEF7	FFEFFDFEF7	FFFFFFEEE7
FFDFFFEEF7	FFFFFDEEF7	FFFDFFFEE7	FFDFFFFEE7	FFFFFDFEE7	FFDDFFFEF7
FFFDFDFEF7	FFFFFFF6F3	FFFFFFF6E7	FFDFFFF6F7	FFFFFDF6F7	7FFFFFF6F7
FFFFFFF2F7	EFFFFFF6F7	7FFFFFFEF3	7FFFFFFEE7	7FDFFFFEF7	7FFFFDFEF7
FFFFFFFAF3	FFFFFFFAE7	FFDFFFFAF7	FFFFFDFAF7	EFFFFFFEF3	EFFFFFFEE7
EFDFFFFEF7	EFFFFDFEF7	FFFFFFFCF6	FFAFFFFEF7	FFBFFFEEF7	FFBDFFFEF7
F6FFFFFEF7	F7EFFFFEF7	FEEFFFFEF7	F7FFFFEEF7	FEFFFFEEF7	F7FDFFFEF7
FEFDFFFEF7	F7BFFFFEF7	FEBFFFFEF7	7FBFFFFEF7	77FFFFFEF7	7EFFFFFEF7
FFBFFFFAF7	F7FFFFFAF7	FEFFFFFAF7	EFBFFFFEF7	E7FFFFFEF7	EEFFFFFEF7
FFBFFFF6F7	F7FFFFF6F7	FEFFFFF6F7	FFBFFFFEF3	FFBFFFFEE7	FF9FFFFEF7
FFBFFDFEF7	F7FFFFFEF3	FEFFFFFEF3	F7FFFFFEE7	FEFFFFFEE7	F7DFFFFEF7
FEDFFFFEF7	F7FFFDFEF7	FEFFFDFEF7	7FFFBFFEF7	FFFFBFFAF7	EFFFBFFEF7
5FFFFFFEF7	DFFFFFFAF7	CFFFFFFEF7	FFFFBFF6F7	DFFFFFF6F7	FFFFBFFEF3
FFFFBFFEE7	FFDFBFFEF7	FFFFBDFEF7	DFFFBFFEF7	DFFFFFFEF3	DFFFFFFEE7
DFDFFFFEF7	DFFFFDFEF7	FFBFBFFEF7	F7FFBFFEF7	FEFFBFFEF7	DFBFFFFEF7
D7FFFFFEF7	DEFFFFFEF7	FDEFFFFEF7	FDFFFFEEF7	FDFDFFFEF7	FFEFFFDEF7
FFFFFFCEF7	FFFDFFDEF7	FFEFFFFED7	FFFFFFEED7	FFFDFFFED7	FFEFF7FEF7
FFFFF7EEF7	FFFDF7FEF7	DDFFFFFEF7	DFFFFFDEF7	DFFFFFFED7	DFFFF7FEF7
7DFFFFFEF7	FDFFFFFAF7	EDFFFFFEF7	FDFFFFF6F7	FDFFBFFEF7	FDFFFFFEF3
FDFFFFFEE7	FDDFFFFEF7	FDFFFDFEF7	FDFFFFFCF7	FFFFFFDCF7	FFFFFFFCD7
FFFFF7FCF7	FDBFFFFEF7	F5FFFFFEF7	FCFFFFFEF7	FFEFFFEFF6	FFEDFFFFF6
FFFDFFEFF6	DFEFFFFFF6	DFFFFFEFF6	DFFDFFFFF6	7FFFDFFEF7	7FFFFFDEF7
7FFFFFFED7	FFFFDFFAF7	FFFFFFDAF7	FFFFFFFAD7	EFFFDFFEF7	EFFFFFDEF7
EFFFFFFED7	7FFFF7FEF7	FFFFF7FAF7	EFFFF7FEF7	FFFFDFF6F7	FFFFFFD6F7
FFFFFFF6D7	FFFF9FFEF7	FFFFBFDEF7	FFFFBFFED7	FFFFDFFEF3	FFFFDFFEE7
FFDFDFFEF7	FFFFDDFEF7	FFFFFFDEF3	FFFFFFFED3	FFFFFFDEE7	FFDFFFDEF7
FFFFFDDEF7	FFFFFFFEC7	FFDFFFFED7	FFFFFDFED7	FFFFF7F6F7	FFFFB7FEF7
FFFFF7FEF3	FFFFF7FEE7	FFDFF7FEF7	FFFFF5FEF7	FFBFFFDEF7	FFBFDFFEF7
FFBFFFFED7	F7FFDFFEF7	FEFFDFFEF7	F7FFFFDEF7	FEFFFFDEF7	F7FFFFFED7
FEFFFFFED7	FFBFF7FEF7	F7FFF7FEF7	FEFFF7FEF7	FDFFDFFEF7	FDFFFFDEF7
FDFFFFFED7	FFFFDFDEF7	FFFFFFDED7	FFFFDFFED7	FFFFD7FEF7	FDFFF7FEF7

Table 9. IV terms of degree 43 in z_{257}-part 2

FFFFF7DEF7	FFFFF7FED7	FBEFFFFFF6	FBFFFFEFF6	FBFDFFFFF6	DBFFFFFFF6
FBFFDFFFF6	FBFFFFFBF6	FBFFFFF7F6	FBFFBFFFF6	FBFFFFFFF2	FBFFFFFFE6
FBDFFFFFF6	FBFFFDFFF6	FBBFFFFFF6	F3FFFFFFF6	FAFFFFFFF6	7BFFFFFFF6
BBFFFFFFF6	EBFFFFFFF6	F9FFFFFFF6	FBFFFFDFF6	FBFFFFFFD6	FBFFF7FFF6
FFEFDFFFF6	FFFFDFEFF6	FFFDDFFFF6	DFFFDFFFF6	FFFFDFFBF6	FFFFDFF7F6
FFFF9FFFF6	FFFFDFFFF2	FFFFDFFFE6	FFDFDFFFF6	FFFFDDFFF6	FFBFDFFFF6
F7FFDFFFF6	FEFFDFFFF6	7FFFDFFFF6	BFFFDFFFF6	EFFFDFFFF6	FFEFFFFBF6
FFFFFFEBF6	FFFDFFFBF6	FFEFFFF7F6	FFFFFFE7F6	FFFDFFF7F6	FFEFBFFFF6
FFFFBFEFF6	FFFDBFFFF6	FFDFFFFFF2	FFFFFDFFF2	FFEFFFFFF2	FFFFFFEFF2
FFFFFFFFE2	FFFDFFFFF2	FFEFFFFFE6	FFDFFDFFF6	FFCFFFFFF6	FFEFFDFFF6
FFFFFFEFE6	FFDFFFEFF6	FFFFFDEFF6	FFFDFFFFE6	FFDFFFFFE6	FFFFFDFFE6
FFDDFFFFF6	FFFDFDFFF6	FFFFFFF7F2	FFFFFFF7E6	FFDFFFF7F6	FFFFFDF7F6
FFFFFFF3F6	FFFFFFFBF2	FFFFFFFBE6	FFDFFFFBF6	FFFFFDFBF6	FFAFFFFFF6
FFBFFFEFF6	FFBDFFFFF6	F6FFFFFFF6	F7EFFFFFF6	FEEFFFFFF6	F7FFFFEFF6
FEFFFFEFF6	F7FDFFFFF6	FEFDFFFFF6	F7BFFFFFF6	FEBFFFFFF6	7FEFFFFFF6
7FFFFFEFF6	7FFDFFFFF6	BFEFFFFFF6	BFFFFFEFF6	BFFDFFFFF6	EFEFFFFFF6
EFFFFFEFF6	EFFDFFFFF6	6FFFFFFFF6	7FBFFFFFF6	77FFFFFFF6	7EFFFFFFF6
EFBFFFFFF6	E7FFFFFFF6	EEFFFFFFF6	FFBFFFFBF6	F7FFFFFBF6	FEFFFFFBF6
BFBFFFFFF6	B7FFFFFFF6	BEFFFFFFF6	7FFFFFFBF6	3FFFFFFFF6	BFFFFFFBF6
EFFFFFFBF6	AFFFFFFFF6	FFBFFFF7F6	F7FFFFF7F6	FEFFFFF7F6	FFBFFFFFF2
FFBFFFFFE6	FF9FFFFFF6	FFBFFDFFF6	F7FFFFFFF2	FEFFFFFFF2	F7FFFFFFE6
FEFFFFFFE6	F7DFFFFFF6	FEDFFFFFF6	F7FFFDFFF6	FEFFFDFFF6	7FFFFFF7F6
BFFFFFF7F6	EFFFFFF7F6	7FFFFFFFF2	7FFFFFFFE6	7FDFFFFFF6	7FFFFDFFF6
BFFFFFFFF2	BFFFFFFFE6	BFDFFFFFF6	BFFFFDFFF6	EFFFFFFFF2	EFFFFFFFE6
EFDFFFFFF6	EFFFFDFFF6	FFFFBFFBF6	DFFFFFFBF6	FFFFBFF7F6	DFFFFFF7F6
FFFFBFFFF2	FFFFBFFFE6	FFDFBFFFF6	FFFFBDFFF6	DFFFBFFFF6	DFFFFFFFF2
DFFFFFFFE6	DFDFFFFFF6	DFFFFDFFF6	FFBFBFFFF6	F7FFBFFFF6	FEFFBFFFF6
DFBFFFFFF6	D7FFFFFFF6	DEFFFFFFF6	7FFFBFFFF6	BFFFBFFFF6	EFFFBFFFF6
5FFFFFFFF6	9FFFFFFFF6	CFFFFFFFF6	FDEFFFFFF6	FDFFFFEFF6	FDFDFFFFF6
FFEFFFDFF6	FFFFFFCFF6	FFFDFFDFF6	FFEFFFFFD6	FFFFFFEFD6	FFFDFFFFD6
FFEFF7FFF6	FFFFF7EFF6	FFFDF7FFF6	DDFFFFFFF6	DFFFFFDFF6	DFFFFFFFD6
DFFFF7FFF6	FDFFDFFFF6	FFFFDFDFF6	FFFFDFFFD6	FFFFD7FFF6	FDFFFFFBF6
FFFFFFDBF6	FFFFFFFBD6	FFFFF7FBF6	FDFFFFF7F6	FFFFFFD7F6	FFFFFFF7D6
FFFFF7F7F6	FDFFBFFFF6	FFFFBFDFF6	FFFFBFFFD6	FFFFB7FFF6	FDFFFFFFF2
FDFFFFFFE6	FDDFFFFFF6	FDFFFDFFF6	FFFFFFDFF2	FFFFFFFFD2	FFFFFFDFE6
FFDFFFDFF6	FFFFFDDFF6	FFFFFFFFC6	FFDFFFFFD6	FFFFFDFFD6	FFFFF7FFF2
FFFFF7FFE6	FFDFF7FFF6	FFFFF5FFF6	FFBFFFDFF6	FFBFFFFFD6	FDBFFFFFF6
F5FFFFFFF6	FCFFFFFFF6	F7FFFFDFF6	FEFFFFDFF6	F7FFFFFFD6	FEFFFFFFD6
FFBFF7FFF6	F7FFF7FFF6	FEFFF7FFF6	7DFFFFFFF6	7FFFFFDFF6	FDFFFFDFF6
FFFFFFDFD6	FDFFFFFFD6	7FFFFFFFD6	7FFFF7FFF6	BDFFFFFFF6	BFFFFFDFF6
BFFFFFFFD6	BFFFF7FFF6	EDFFFFFFF6	EFFFFFDFF6	EFFFFFFFD6	EFFFF7FFF6
FDFFF7FFF6	FFFFF7DFF6	FFFFF7FFD6	FBFFFFFE77	FBFFF7FF77	BFFFFFFE77
BFFFF7FF77	7FFFFFFE77	7FFFF7FF77	BBFFDFFFF7	3BFFFFFFF7	BBFFFFFBF7
ABFFFFFFF7	BBFFFFF7F7	BBFFBFFFF7	BBFFFFFFF3	BBFFFFFFE7	BBDFFFFFF7
BBFFFDFFF7	BBBFFFFFF7	B3FFFFFFF7	BAFFFFFFF7	B9FFFFFFF7	BBFFFFDFF7
BBFFFFFFD7	BBFFF7FFF7	FFEFFFFE77	FFFFFFEE77	FFFDFFFE77	FFFFFFFA77
EFFFFFFE77	FFEFF7FF77	FFFFF7EF77	FFFDF7FF77	FFFFF7FB77	EFFFF7FF77
FFFFFFF677	FFFFFFFE73	FFFFFFFE67	FFDFFFFE77	FFFFFDFE77	FFFFF7F777
FFFFF7FF73	FFFFF7FF67	FFDFF7FF77	FFFFF5FF77	FFBFFFFE77	F7FFFFFE77
FEFFFFFE77	FFBFF7FF77	F7FFF7FF77	FEFFF7FF77	FFFFBFFE77	DFFFFFFE77
FFFFB7FF77	DFFFF7FF77	FDFFFFFE77	FDFFF7FF77	FFFFDFFE77	FFFFFFDE77
FFFFFFFE57	FFFFD7FF77	FFFFF7DF77	FFFFF7FF57	FFFFF7FE77	FFEFDFFFE7
FFEFDFFFF3	FFEFDFEFF7	FFEDDFFFF7	FFCFDFFFF7	FFEFDDFFF7	FFFDDFEFF7
FFEFDFF7F7	FFEFDFFBF7	FFAFDFFFF7	F7EFDFFFF7	FEEFDFFFF7	FFEF9FFFF7
DFEFDFFFF7	DFFFDFEFF7	DFFDDFFFF7	7FEFFFEFF7	7FEDFFFFF7	7FFDFFEFF7
FFEFFFEBF7	FFEDFFFBF7	FFFDFFEBF7	EFEFFFEFF7	EFEDFFFFF7	EFFDFFEFF7
6FEFFFFFF7	6FFFFFEFF7	6FFDFFFFF7	FFEFFFE7F7	FFEDFFF7F7	FFFDFFE7F7
FFEFBFEFF7	FFEDBFFFF7	FFFDBFEFF7	FFDFFDFFF3	FFCFFFFFF3	FFDFFFEFF3
FFFDFFFFE3	FFDFFFFFE3	FFDDFFFFF3	FFEFFDFFF3	FFFFFDEFF3	FFEFFFEFF3
FFFFFDFFE3	FFEFFFFFE3	FFFDFDFFF3	FFEDFFFFF3	FFFFFFEFE3	FFFDFFEFF3
FFCFFDFFF7	FFEFFFEFE7	FFDFFDEFF7	FFCFFFEFF7	FFEFFDEFF7	FFFDFDFFE7
FFEDFFFFE7	FFDFFDFFE7	FFCFFFFFE7	FFEFFDFFE7	FFDDFDFFF7	FFCDFFFFF7
FFEDFDFFF7	FFFDFFEFE7	FFDFFFEFE7	FFFFFDEFE7	FFDDFFEFF7	FFFDFDEFF7
FFDFFFF7F3	FFFFFDF7F3	FFEFFFF7F3	FFFFFFE7F3	FFFFFFF7E3	FFFDFFF7F3
FFEFFFF7E7	FFDFFDF7F7	FFCFFFF7F7	FFEFFDF7F7	FFFFFFE7E7	FFDFFFE7F7
FFFFFDE7F7	FFFDFFF7E7	FFDFFFF7E7	FFFFFDF7E7	FFDDFFF7F7	FFFDFDF7F7
FFEFFFF3F7	FFFFFFE3F7	FFFDFFF3F7	FFDFFFFBF3	FFFFFDFBF3	FFEFFFFBF3
FFFFFFEBF3	FFFFFFFBE3	FFFDFFFBF3	FFEFFFFBE7	FFDFFDFBF7	FFCFFFFBF7
FFEFFDFBF7	FFFFFFEBE7	FFDFFFEBF7	FFFFFDEBF7	FFFDFFFBE7	FFDFFFFBE7
FFFFFDFBE7	FFDDFFFBF7	FFFDFDFBF7	FFFFFFF3F3	FFFFFFF3E7	FFDFFFF3F7
FFFFFDF3F7	FFEFFFFDF6	FFFFFFEDF6	FFFDFFFDF6	FFAFFFEFF7	FFADFFFFF7
FFBDFFEFF7	F7AFFFFFF7	FEAFFFFFF7	F6EFFFFFF7	F6FFFFEFF7	F7EFFFEFF7

Table 10. IV terms of degree 43 in z_{257}-part 3

FEEFFFEFF7	F6FDFFFFF7	F7EDFFFFF7	FEEDFFFFF7	F7FDFFEFF7	FEFDFFEFF7
F6BFFFFFF7	F7BFFFEFF7	FEBFFFEFF7	F7BDFFFFF7	FEBDFFFFF7	FFAFFFFBF7
FFBFFFEBF7	FFBDFFFBF7	F6FFFFFBF7	F7EFFFFBF7	FEEFFFFBF7	F7FFFFEBF7
FEFFFFEBF7	F7FDFFFBF7	FEFDFFFBF7	F7BFFFFBF7	FEBFFFFBF7	FFAFFFF7F7
FFBFFFE7F7	FFBDFFF7F7	F6FFFFF7F7	F7EFFFF7F7	FEEFFFF7F7	F7FFFFE7F7
FEFFFFE7F7	F7FDFFF7F7	FEFDFFF7F7	F7BFFFF7F7	FEBFFFF7F7	FF9FFFFFF3
FFBFFDFFF3	FFAFFFFFF3	FFBFFFEFF3	FFBFFFFFE3	FFBDFFFFF3	FFAFFFFFE7
FF9FFDFFF7	FF8FFFFFF7	FFAFFDFFF7	FFBFFFEFE7	FF9FFFEFF7	FFBFFDEFF7
FFBDFFFFE7	FF9FFFFFE7	FFBFFDFFE7	FF9DFFFFF7	FFBDFDFFF7	F7DFFFFFF3
FEDFFFFFF3	F6FFFFFFF3	F7FFFDFFF3	FEFFFDFFF3	F7EFFFFFF3	FEEFFFFFF3
F7FFFFEFF3	FEFFFFEFF3	F7FFFFFFE3	FEFFFFFFE3	F7FDFFFFF3	FEFDFFFFF3
F7BFFFFFF3	FEBFFFFFF3	F6FFFFFFE7	F7EFFFFFE7	FEEFFFFFE7	F6DFFFFFF7
F6FFFDFFF7	F7DFFDFFF7	FEDFFDFFF7	F7CFFFFFF7	FECFFFFFF7	F7EFFDFFF7
FEEFFDFFF7	F7FFFFEFE7	FEFFFFEFE7	F7DFFFEFF7	FEDFFFEFF7	F7FFFDEFF7
FEFFFDEFF7	F7FDFFFFE7	FEFDFFFFE7	F7DFFFFFE7	FEDFFFFFE7	F7FFFDFFE7
FEFFFDFFE7	F7DDFFFFF7	FEDDFFFFF7	F7FDFDFFF7	FEFDFDFFF7	F7BFFFFFE7
FEBFFFFFE7	F79FFFFFF7	FE9FFFFFF7	F7BFFDFFF7	FEBFFDFFF7	FFBFFFF7F3
FFBFFFF7E7	FF9FFFF7F7	FFBFFDF7F7	F7FFFFF7F3	FEFFFFF7F3	F7FFFFF7E7
FEFFFFF7E7	F7DFFFF7F7	FEDFFFF7F7	F7FFFDF7F7	FEFFFDF7F7	FFBFFFF3F7
F7FFFFF3F7	FEFFFFF3F7	FFBFFFFBF3	FFBFFFFBE7	FF9FFFFBF7	FFBFFDFBF7
F7FFFFFBF3	FEFFFFFBF3	F7FFFFFBE7	FEFFFFFBE7	F7DFFFFBF7	FEDFFFFBF7
F7FFFDFBF7	FEFFFDFBF7	FFEFBFFBF7	FFFFBFEBF7	FFFDBFFBF7	5FEFFFFFF7
5FFFFFEFF7	5FFDFFFFF7	DFEFFFFBF7	DFFFFFEBF7	DFFDFFFBF7	CFEFFFFFF7
CFFFFFEFF7	CFFDFFFFF7	4FFFFFFFF7	FFEFBFF7F7	FFFFBFE7F7	FFFDBFF7F7
DFEFFFF7F7	DFFFFFE7F7	DFFDFFF7F7	FFDFBFFFF3	FFFFBDFFF3	FFEFBFFFF3
FFFFBFEFF3	FFFFBFFFE3	FFFDBFFFF3	FFEFBFFFE7	FFDFBDFFF7	FFCFBFFFF7
FFEFBDFFF7	FFFFBFEFE7	FFDFBFEFF7	FFFFBDEFF7	FFFDBFFFE7	FFDFBFFFE7
FFFFBDFFE7	FFDDBFFFF7	FFFDBDFFF7	DFEFBFFFF7	DFFFBFEFF7	DFFDBFFFF7
DFDFFFFFF3	DFFFFDFFF3	DFEFFFFFF3	DFFFFFEFF3	DFFFFFFFE3	DFFDFFFFF3
DFEFFFFFE7	DFDFFDFFF7	DFCFFFFFF7	DFEFFDFFF7	DFFFFFEFE7	DFDFFFEFF7
DFFFFDEFF7	DFFDFFFFE7	DFDFFFFFE7	DFFFFDFFE7	DFDDFFFFF7	DFFDFDFFF7
DFFFBFFFF3	DFFFBFFFE7	DFDFBFFFF7	DFFFBDFFF7	FFFFBFF7F3	FFFFBFF7E7
FFDFBFF7F7	FFFFBDF7F7	DFFFBFF7F7	DFFFFFF7F3	DFFFFFF7E7	DFDFFFF7F7
DFFFFDF7F7	FFFFBFF3F7	DFFFFFF3F7	FFFFBFFBF3	FFFFBFFBE7	FFDFBFFBF7
FFFFBDFBF7	DFFFBFFBF7	DFFFFFFBF3	DFFFFFFBE7	DFDFFFFBF7	DFFFFDFBF7
DFFFFFFDF6	FFAFBFFFF7	FFBFBFEFF7	FFBDBFFFF7	F6FFBFFFF7	F7EFBFFFF7
FEEFBFFFF7	F7FFBFEFF7	FEFFBFEFF7	F7FDBFFFF7	FEFDBFFFF7	F7BFBFFFF7
FEBFBFFFF7	DFAFFFFFF7	DFBFFFEFF7	DFBDFFFFF7	D6FFFFFFF7	D7EFFFFFF7
DEEFFFFFF7	D7FFFFEFF7	DEFFFFEFF7	D7FDFFFFF7	DEFDFFFFF7	D7BFFFFFF7
DEBFFFFFF7	FFBFBFFBF7	F7FFBFFBF7	FEFFBFFBF7	DFBFFFFBF7	D7FFFFFBF7
DEFFFFFBF7	FFBFBFF7F7	F7FFBFF7F7	FEFFBFF7F7	DFBFFFF7F7	D7FFFFF7F7
DEFFFFF7F7	FFBFBFFFF3	FFBFBFFFE7	FF9FBFFFF7	FFBFBDFFF7	F7FFBFFFF3
FEFFBFFFF3	F7FFBFFFE7	FEFFBFFFE7	F7DFBFFFF7	FEDFBFFFF7	F7FFBDFFF7
FEFFBDFFF7	DFBFBFFFF7	D7FFBFFFF7	DEFFBFFFF7	DFBFFFFFF3	DFBFFFFFE7
DF9FFFFFF7	DFBFFDFFF7	D7FFFFFFF3	DEFFFFFFF3	D7FFFFFFE7	DEFFFFFFE7
D7DFFFFFF7	DEDFFFFFF7	D7FFFDFFF7	DEFFFDFFF7	FDEFFFEFF7	FDEDFFFFF7
FDFDFFEFF7	FFEFFFCFF7	FFEDFFDFF7	FFFDFFCFF7	FFEFFFEFD7	FFEDFFFFD7
FFFDFFEFD7	FFEFF7EFF7	FFEDF7FFF7	FFFDF7EFF7	DDEFFFFFF7	DDFFFFEFF7
DDFDFFFFF7	DFEFFFDFF7	DFFFFFCFF7	DFFDFFDFF7	DFEFFFFFD7	DFFFFFEFD7
DFFDFFFFD7	DFEFF7FFF7	DFFFF7EFF7	DFFDF7FFF7	FDEFFFFDF7	FDFFFFEDF7
FDFDFFFDF7	FFEFFFDDF7	FFFFFFCDF7	FFFDFFDDF7	FFEFFFFDD7	FFFFFFEDD7
FFFDFFFDD7	FFEFF7FDF7	FFFFF7EDF7	FFFDF7FDF7	DDFFFFFDF7	DFFFFFDDF7
DFFFFFFDD7	DFFFF7FDF7	FBFFDFFFE7	FBFFDFFFF3	FBEFDFFFF7	FBFFDFEFF7
FBFDDFFFF7	FBDFDFFFF7	FBFFDDFFF7	FBFFDFF7F7	7BFFDFFFF7	FBFFDFFBF7
EBFFDFFFF7	FBBFDFFFF7	F3FFDFFFF7	FAFFDFFFF7	FBFF9FFFF7	DBFFDFFFF7
FFFFDFEBF7	FFFDDFFBF7	FFFFDFE7F7	FFFDDFF7F7	FFFF9FEFF7	FFFD9FFFF7
FFDFDFFFF3	FFFFDDFFF3	FFFFDFEFF3	FFFFDFFFE3	FFFDDFFFF3	FFDFDDFFF7
FFFFDFEFE7	FFDFDFEFF7	FFFFDDEFF7	FFFDDFFFE7	FFDFDFFFE7	FFFFDDFFE7
FFDDDFFFF7	FFFDDDFFF7	FFFFDFF7F3	FFFFDFF7E7	FFDFDFF7F7	FFFFDDF7F7
FFFFDFF3F7	FFFFDFFBF3	FFFFDFFBE7	FFDFDFFBF7	FFFFDDFBF7	FFBFDFEFF7
FFBDDFFFF7	F6FFDFFFF7	F7FFDFEFF7	FEFFDFEFF7	F7FDDFFFF7	FEFDDFFFF7
F7BFDFFFF7	FEBFDFFFF7	FFBFDFFBF7	F7FFDFFBF7	FEFFDFFBF7	FFBFDFF7F7
F7FFDFF7F7	FEFFDFF7F7	FFBFDFFFF3	FFBFDFFFE7	FF9FDFFFF7	FFBFDDFFF7
F7FFDFFFF3	FEFFDFFFF3	F7FFDFFFE7	FEFFDFFFE7	F7DFDFFFF7	FEDFDFFFF7
F7FFDDFFF7	FEFFDDFFF7	FFFF9FFBF7	DFFFDFFBF7	FFFF9FF7F7	DFFFDFF7F7
FFFF9FFFF3	FFFF9FFFE7	FFDF9FFFF7	FFFF9DFFF7	DFFF9FFFF7	DFFFDFFFF3
DFFFDFFFE7	DFDFDFFFF7	DFFFDDFFF7	FFBF9FFFF7	F7FF9FFFF7	FEFF9FFFF7
DFBFDFFFF7	D7FFDFFFF7	DEFFDFFFF7	7BEFFFFFF7	7BFFFFEFF7	7BFDFFFFF7
FBEFFFFBF7	FBFFFFEBF7	FBFDFFFBF7	7BFFFFFBF7	EBEFFFFFF7	EBFFFFEFF7
EBFDFFFFF7	6BFFFFFFF7	EBFFFFFBF7	FBEFFFF7F7	FBFFFFE7F7	FBFDFFF7F7
FBEFBFFFF7	FBFFBFEFF7	FBFDBFFFF7	FBDFFFFFF3	FBFFFDFFF3	FBEFFFFFF3
FBFFFFEFF3	FBFFFFFFE3	FBFDFFFFF3	FBEFFFFFE7	FBDFFDFFF7	FBCFFFFFF7

Table 11. IV terms of degree 43 in z_{257}-part 4

FBEFFDFFF7	FBFFFFEFE7	FBDFFFEFF7	FBFFFDEFF7	FBFDFFFFE7	FBDFFFFFE7
FBFFFDFFE7	FBDDFFFFF7	FBFDFDFFF7	FBFFFFF7F3	FBFFFFF7E7	FBDFFFF7F7
FBFFFDF7F7	7BFFFFF7F7	FBFFFFF3F7	EBFFFFF7F7	7BFFFFFFF3	7BFFFFFFE7
7BDFFFFFF7	7BFFFDFFF7	FBFFFFFBF3	FBFFFFFBE7	FBDFFFFBF7	FBFFFDFBF7
EBFFFFFFF3	EBFFFFFFE7	EBDFFFFFF7	EBFFFDFFF7	FBFFFFFDF6	FBAFFFFFF7
FBBFFFEFF7	FBBDFFFFF7	F2FFFFFFF7	F3EFFFFFF7	FAEFFFFFF7	F3FFFFEFF7
FAFFFFEFF7	F3FDFFFFF7	FAFDFFFFF7	F3BFFFFFF7	FABFFFFFF7	7BBFFFFFF7
73FFFFFFF7	7AFFFFFFF7	FBBFFFFBF7	F3FFFFFBF7	FAFFFFFBF7	EBBFFFFFF7
E3FFFFFFF7	EAFFFFFFF7	FBBFFFF7F7	F3FFFFF7F7	FAFFFFF7F7	FBBFFFFFF3
FBBFFFFFE7	FB9FFFFFF7	FBBFFDFFF7	F3FFFFFFF3	FAFFFFFFF3	F3FFFFFFE7
FAFFFFFFE7	F3DFFFFFF7	FADFFFFFF7	F3FFFDFFF7	FAFFFDFFF7	7BFFBFFFF7
FBFFBFFBF7	EBFFBFFFF7	5BFFFFFFF7	DBFFFFFBF7	CBFFFFFFF7	FBFFBFF7F7
DBFFFFF7F7	FBFFBFFFF3	FBFFBFFFE7	FBDFBFFFF7	FBFFBDFFF7	DBFFBFFFF7
DBFFFFFFF3	DBFFFFFFE7	DBDFFFFFF7	DBFFFDFFF7	FBBFBFFFF7	F3FFBFFFF7
FAFFBFFFF7	DBBFFFFFF7	D3FFFFFFF7	DAFFFFFFF7	F9EFFFFFF7	F9FFFFEFF7
F9FDFFFFF7	FBEFFFDFF7	FBFFFFCFF7	FBFDFFDFF7	FBEFFFFFD7	FBFFFFEFD7
FBFDFFFFD7	FBEFF7FFF7	FBFFF7EFF7	FBFDF7FFF7	D9FFFFFFF7	DBFFFFDFF7
DBFFFFFFD7	DBFFF7FFF7	F9FFDFFFF7	FBFFDFDFF7	FBFFDFFFD7	FBFFD7FFF7
79FFFFFFF7	7BFFFFDFF7	7BFFFFFFD7	7BFFF7FFF7	F9FFFFFBF7	FBFFFFDBF7
FBFFFFFBD7	FBFFF7FBF7	E9FFFFFFF7	EBFFFFDFF7	EBFFFFFFD7	EBFFF7FFF7
F9FFFFF7F7	FBFFFFD7F7	FBFFFFF7D7	FBFFF7F7F7	F9FFBFFFF7	FBFFBFDFF7
FBFFBFFFD7	FBFFB7FFF7	F9FFFFFFF3	F9FFFFFFE7	F9DFFFFFF7	F9FFFDFFF7
FBFFFFDFF3	FBFFFFFFD3	FBFFFFDFE7	FBDFFFDFF7	FBFFFDDFF7	FBFFFFFFC7
FBDFFFFFD7	FBFFFDFFD7	FBFFF7FFF3	FBFFF7FFE7	FBDFF7FFF7	FBFFF5FFF7
F9FFFFFDF7	FBFFFFDDF7	FBFFFFFDD7	FBFFF7FDF7	FBBFFFDFF7	FBBFFFFFD7
F9BFFFFFF7	F1FFFFFFF7	F8FFFFFFF7	F3FFFFDFF7	FAFFFFDFF7	F9FFFFDFF7
FBFFFFDFD7	F9FFFFFFD7	F3FFFFFFD7	FAFFFFFFD7	FBBFF7FFF7	F3FFF7FFF7
FAFFF7FFF7	F9FFF7FFF7	FBFFF7DFF7	FBFFF7FFD7	FDEFDFFFF7	FDFFDFEFF7
FDFDDFFFF7	DDFFDFFFF7	FDFFDFFBF7	FDFFDFF7F7	FDFF9FFFF7	FDFFDFFFF3
FDFFDFFFE7	FDDFDFFFF7	FDFFDDFFF7	FDBFDFFFF7	F5FFDFFFF7	FCFFDFFFF7
FDEFFFFBF7	FDFFFFEBF7	FDFDFFFBF7	FDEFFFF7F7	FDFFFFE7F7	FDFDFFF7F7
FDEFBFFFF7	FDFFBFEFF7	FDFDBFFFF7	FDDFFFFFF3	FDFFFDFFF3	FDEFFFFFF3
FDFFFFEFF3	FDFFFFFFE3	FDFDFFFFF3	FDEFFFFFE7	FDDFFDFFF7	FDCFFFFFF7
FDEFFDFFF7	FDFFFFEFE7	FDDFFFEFF7	FDFFFDEFF7	FDFDFFFFE7	FDDFFFFFE7
FDFFFDFFE7	FDDDFFFFF7	FDFDFDFFF7	FDFFFFF7F3	FDFFFFF7E7	FDDFFFF7F7
FDFFFDF7F7	FDFFFFF3F7	FDFFFFFBF3	FDFFFFFBE7	FDDFFFFBF7	FDFFFDFBF7
FDAFFFFFF7	FDBFFFEFF7	FDBDFFFFF7	F4FFFFFFF7	F5EFFFFFF7	FCEFFFFFF7
F5FFFFEFF7	FCFFFFEFF7	F5FDFFFFF7	FCFDFFFFF7	F5BFFFFFF7	FCBFFFFFF7
FDBFFFFBF7	F5FFFFFBF7	FCFFFFFBF7	FDBFFFF7F7	F5FFFFF7F7	FCFFFFF7F7
FDBFFFFFF3	FDBFFFFFE7	FD9FFFFFF7	FDBFFDFFF7	F5FFFFFFF3	FCFFFFFFF3
F5FFFFFFE7	FCFFFFFFE7	F5DFFFFFF7	FCDFFFFFF7	F5FFFDFFF7	FCFFFDFFF7
FDFFBFFBF7	DDFFFFFBF7	FDFFBFF7F7	DDFFFFF7F7	FDFFBFFFF3	FDFFBFFFE7
FDDFBFFFF7	FDFFBDFFF7	DDFFBFFFF7	DDFFFFFFF3	DDFFFFFFE7	DDDFFFFFF7
DDFFFDFFF7	FDBFBFFFF7	F5FFBFFFF7	FCFFBFFFF7	DDBFFFFFF7	D5FFFFFFF7
DCFFFFFFF7	FFFFDFFDF6	7FEFDFFFF7	7FFFDFEFF7	7FFDDFFFF7	FFEFDFDFF7
FFFFDFCFF7	FFFDDFDFF7	FFEFDFFFD7	FFFFDFEFD7	FFFDDFFFD7	BFEFDFFFF7
BFFFDFEFF7	BFFDDFFFF7	EFEFDFFFF7	EFFFDFEFF7	EFFDDFFFF7	6FFFDFFFF7
FFEFD7FFF7	FFFFD7EFF7	FFFDD7FFF7	5FFFDFFFF7	DFFFDFDFF7	DFFFDFFFD7
9FFFDFFFF7	CFFFDFFFF7	DFFFD7FFF7	7FFFDFFBF7	FFFFDFDBF7	FFFFDFFBD7
3FFFDFFFF7	BFFFDFFBF7	EFFFDFFBF7	AFFFDFFFF7	FFFFD7FBF7	7FFFDFF7F7
FFFFDFD7F7	FFFFDFF7D7	BFFFDFF7F7	EFFFDFF7F7	7FFF9FFFF7	FFFF9FDFF7
FFFF9FFFD7	BFFF9FFFF7	EFFF9FFFF7	7FFFDFFFF3	7FFFDFFFE7	7FDFDFFFF7
7FFFDDFFF7	FFFFDFDFF3	FFFFDFFFD3	FFFFDFDFE7	FFDFDFDFF7	FFFFDDDFF7
FFFFDFFFC7	FFDFDFFFD7	FFFFDDFFD7	BFFFDFFFF3	BFFFDFFFE7	BFDFDFFFF7
BFFFDDFFF7	EFFFDFFFF3	EFFFDFFFE7	EFDFDFFFF7	EFFFDDFFF7	FFFFD7F7F7
FFFF97FFF7	FFFFD7FFF3	FFFFD7FFE7	FFDFD7FFF7	FFFFD5FFF7	FFBFDFDFF7
FFBFDFFFD7	7FBFDFFFF7	77FFDFFFF7	7EFFDFFFF7	F7FFDFDFF7	FEFFDFDFF7
F7FFDFFFD7	FEFFDFFFD7	BFBFDFFFF7	B7FFDFFFF7	BEFFDFFFF7	EFBFDFFFF7
E7FFDFFFF7	EEFFDFFFF7	FFBFD7FFF7	F7FFD7FFF7	FEFFD7FFF7	FFFFFFF9F6
7FEFFFFBF7	7FFFFFEBF7	7FFDFFFBF7	FFEFFFDBF7	FFFFFFCBF7	FFFDFFDBF7
FFEFFFFBD7	FFFFFFEBD7	FFFDFFFBD7	3FEFFFFFF7	3FFFFFEFF7	3FFDFFFFF7
BFEFFFFBF7	BFFFFFEBF7	BFFDFFFBF7	3FFFFFFBF7	EFEFFFFBF7	EFFFFFEBF7
EFFDFFFBF7	6FFFFFFBF7	AFEFFFFFF7	AFFFFFEFF7	AFFDFFFFF7	2FFFFFFFF7
AFFFFFFBF7	FFEFF7FBF7	FFFFF7EBF7	FFFDF7FBF7	FFFFFFF5F6	FFFFBFFDF6
FFFFFFFDF2	FFFFFFFDE6	FFDFFFFDF6	FFFFFDFDF6	7FEFFFF7F7	7FFFFFE7F7
7FFDFFF7F7	FFEFFFD7F7	FFFFFFC7F7	FFFDFFD7F7	FFEFFFF7D7	FFFFFFE7D7
FFFDFFF7D7	BFEFFFF7F7	BFFFFFE7F7	BFFDFFF7F7	EFEFFFF7F7	EFFFFFE7F7
EFFDFFF7F7	6FFFFFF7F7	7FEFBFFFF7	7FFFBFEFF7	7FFDBFFFF7	FFEFBFDFF7
FFFFBFCFF7	FFFDBFDFF7	FFEFBFFFD7	FFFFBFEFD7	FFFDBFFFD7	BFEFBFFFF7
BFFFBFEFF7	BFFDBFFFF7	EFEFBFFFF7	EFFFBFEFF7	EFFDBFFFF7	6FFFBFFFF7
7FDFFFFFF3	7FFFFDFFF3	7FEFFFFFF3	7FFFFFEFF3	7FFFFFFFE3	7FFDFFFFF3
7FEFFFFFE7	7FDFFDFFF7	7FCFFFFFF7	7FEFFDFFF7	7FFFFFEFE7	7FDFFFEFF7

Table 12. IV terms of degree 43 in z_{257}-part 5

7FFFFDEFF7	7FFDFFFFE7	7FDFFFFFE7	7FFFFDFFE7	7FDDFFFFF7	7FFDFDFFF7
FFDFFFDFF3	FFDFFFFFD3	FFFFFDDFF3	FFEFFFDFF3	FFFFFFCFF3	FFFFFFDFE3
FFFDFFDFF3	FFFFFDFFD3	FFEFFFFFD3	FFFFFFEFD3	FFFFFFFFC3	FFFDFFFFD3
FFEFFFDFE7	FFDFFFDFE7	FFFFFDDFE7	FFDFFDDFF7	FFCFFFDFF7	FFEFFDDFF7
FFFFFFCFE7	FFDFFFCFF7	FFFFFDCFF7	FFFDFFDFE7	FFDDFFDFF7	FFFDFDDFF7
FFEFFFFFC7	FFDFFFFFC7	FFFFFDFFC7	FFDFFDFFD7	FFCFFFFFD7	FFEFFDFFD7
FFFFFFEFC7	FFDFFFEFD7	FFFFFDEFD7	FFFDFFFFC7	FFDDFFFFD7	FFFDFDFFD7
BFDFFFFFF3	BFFFFDFFF3	BFEFFFFFF3	BFFFFFEFF3	BFFFFFFFE3	BFFDFFFFF3
BFEFFFFFE7	BFDFFDFFF7	BFCFFFFFF7	BFEFFDFFF7	BFFFFFEFE7	BFDFFFEFF7
BFFFFDEFF7	BFFDFFFFE7	BFDFFFFFE7	BFFFFDFFE7	BFDDFFFFF7	BFFDFDFFF7
EFDFFFFFF3	EFFFFDFFF3	EFEFFFFFF3	EFFFFFEFF3	EFFFFFFFE3	EFFDFFFFF3
EFEFFFFFE7	EFDFFDFFF7	EFCFFFFFF7	EFEFFDFFF7	EFFFFFEFE7	EFDFFFEFF7
EFFFFDEFF7	EFFDFFFFE7	EFDFFFFFE7	EFFFFDFFE7	EFDDFFFFF7	EFFDFDFFF7
6FFFFFFFF3	6FFFFFFFE7	6FDFFFFFF7	6FFFFDFFF7	7FFFFFF7F3	7FFFFFF7E7
7FDFFFF7F7	7FFFFDF7F7	FFFFFFD7F3	FFFFFFF7D3	FFFFFFD7E7	FFDFFFD7F7
FFFFFDD7F7	FFFFFFF7C7	FFDFFFF7D7	FFFFFDF7D7	BFFFFFF7F3	BFFFFFF7E7
BFDFFFF7F7	BFFFFDF7F7	EFFFFFF7F3	EFFFFFF7E7	EFDFFFF7F7	EFFFFDF7F7
FFEFF7F7F7	FFFFF7E7F7	FFFDF7F7F7	FFEFB7FFF7	FFFFB7EFF7	FFFDB7FFF7
FFDFF7FFF3	FFFFF5FFF3	FFEFF7FFF3	FFFFF7EFF3	FFFFF7FFE3	FFFDF7FFF3
FFEFF7FFE7	FFDFF5FFF7	FFCFF7FFF7	FFEFF5FFF7	FFFFF7EFE7	FFDFF7EFF7
FFFFF5EFF7	FFFDF7FFE7	FFDFF7FFE7	FFFFF5FFE7	FFDDF7FFF7	FFFDF5FFF7
FFFFF7F7F3	FFFFF7F7E7	FFDFF7F7F7	FFFFF5F7F7	7FFFFFF3F7	FFFFFFD3F7
FFFFFFF3D7	3FFFFFF7F7	BFFFFFF3F7	EFFFFFF3F7	AFFFFFF7F7	7FFFFFFBF3
7FFFFFFBE7	7FDFFFFBF7	7FFFFDFBF7	FFFFFFDBF3	FFFFFFFBD3	FFFFFFDBE7
FFDFFFDBF7	FFFFFDDBF7	FFFFFFFBC7	FFDFFFFBD7	FFFFFDFBD7	3FFFFFFFF3
3FFFFFFFE7	3FDFFFFFF7	3FFFFDFFF7	BFFFFFFBF3	BFFFFFFBE7	BFDFFFFBF7
BFFFFDFBF7	EFFFFFFBF3	EFFFFFFBE7	EFDFFFFBF7	EFFFFDFBF7	AFFFFFFFF3
AFFFFFFFE7	AFDFFFFFF7	AFFFFDFFF7	FFFFF7F3F7	FFFFF7FBF3	FFFFF7FBE7
FFDFF7FBF7	FFFFF5FBF7	FFBFFFFDF6	F7FFFFFDF6	FEFFFFFDF6	7FFFFFFDF6
BFFFFFFDF6	EFFFFFFDF6	FFAFFFDFF7	FFBFFFCFF7	FFBDFFDFF7	FFAFFFFFD7
FFBFFFEFD7	7FAFFFFFF7	7FBFFFEFF7	FFBDFFFFD7	7FBDFFFFF7	76FFFFFFF7
77EFFFFFF7	7EEFFFFFF7	77FFFFEFF7	7EFFFFEFF7	77FDFFFFF7	7EFDFFFFF7
77BFFFFFF7	7EBFFFFFF7	F7BFFFDFF7	FEBFFFDFF7	F7BFFFFFD7	FEBFFFFFD7
F6FFFFDFF7	F7EFFFDFF7	FEEFFFDFF7	F7FFFFCFF7	FEFFFFCFF7	F7FDFFDFF7
FEFDFFDFF7	F6FFFFFFD7	F7EFFFFFD7	FEEFFFFFD7	F7FFFFEFD7	FEFFFFEFD7
F7FDFFFFD7	FEFDFFFFD7	BFAFFFFFF7	BFBFFFEFF7	BFBDFFFFF7	B6FFFFFFF7
B7EFFFFFF7	BEEFFFFFF7	B7FFFFEFF7	BEFFFFEFF7	B7FDFFFFF7	BEFDFFFFF7
B7BFFFFFF7	BEBFFFFFF7	EFAFFFFFF7	EFBFFFEFF7	EFBDFFFFF7	E6FFFFFFF7
E7EFFFFFF7	EEEFFFFFF7	E7FFFFEFF7	EEFFFFEFF7	E7FDFFFFF7	EEFDFFFFF7
E7BFFFFFF7	EEBFFFFFF7	6FBFFFFFF7	67FFFFFFF7	6EFFFFFFF7	FFAFF7FFF7
FFBFF7EFF7	FFBDF7FFF7	F6FFF7FFF7	F7EFF7FFF7	FEEFF7FFF7	F7FFF7EFF7
FEFFF7EFF7	F7FDF7FFF7	FEFDF7FFF7	F7BFF7FFF7	FEBFF7FFF7	FFBFFFDBF7
FFBFFFFBD7	7FBFFFFBF7	77FFFFFBF7	7EFFFFFBF7	F7FFFFDBF7	FEFFFFDBF7
F7FFFFFBD7	FEFFFFFBD7	3FBFFFFFF7	37FFFFFFF7	3EFFFFFFF7	BFBFFFFBF7
B7FFFFFBF7	BEFFFFFBF7	EFBFFFFBF7	E7FFFFFBF7	EEFFFFFBF7	AFBFFFFFF7
A7FFFFFFF7	AEFFFFFFF7	FFBFF7FBF7	F7FFF7FBF7	FEFFF7FBF7	FFBFFFD7F7
FFBFFFF7D7	7FBFFFF7F7	77FFFFF7F7	7EFFFFF7F7	F7FFFFD7F7	FEFFFFD7F7
F7FFFFF7D7	FEFFFFF7D7	BFBFFFF7F7	B7FFFFF7F7	BEFFFFF7F7	EFBFFFF7F7
E7FFFFF7F7	EEFFFFF7F7	FFBFFFDFF3	FFBFFFFFD3	7FBFFFFFF3	FFBFFFDFE7
FF9FFFDFF7	FFBFFDDFF7	FFBFFFFFC7	FF9FFFFFD7	FFBFFDFFD7	7FBFFFFFE7
7F9FFFFFF7	7FBFFDFFF7	77FFFFFFF3	7EFFFFFFF3	77FFFFFFE7	7EFFFFFFE7
77DFFFFFF7	7EDFFFFFF7	77FFFDFFF7	7EFFFDFFF7	F7FFFFDFF3	FEFFFFDFF3
F7FFFFFFD3	FEFFFFFFD3	F7FFFFDFE7	FEFFFFDFE7	F7DFFFDFF7	FEDFFFDFF7
F7FFFDDFF7	FEFFFDDFF7	F7FFFFFFC7	FEFFFFFFC7	F7DFFFFFD7	FEDFFFFFD7
F7FFFDFFD7	FEFFFDFFD7	BFBFFFFFF3	BFBFFFFFE7	BF9FFFFFF7	BFBFFDFFF7
B7FFFFFFF3	BEFFFFFFF3	B7FFFFFFE7	BEFFFFFFE7	B7DFFFFFF7	BEDFFFFFF7
B7FFFDFFF7	BEFFFDFFF7	EFBFFFFFF3	EFBFFFFFE7	EF9FFFFFF7	EFBFFDFFF7
E7FFFFFFF3	EEFFFFFFF3	E7FFFFFFE7	EEFFFFFFE7	E7DFFFFFF7	EEDFFFFFF7
E7FFFDFFF7	EEFFFDFFF7	FFBFF7F7F7	F7FFF7F7F7	FEFFF7F7F7	FFBFF7FFF3
FFBFF7FFE7	FF9FF7FFF7	FFBFF5FFF7	F7FFF7FFF3	FEFFF7FFF3	F7FFF7FFE7
FEFFF7FFE7	F7DFF7FFF7	FEDFF7FFF7	F7FFF5FFF7	FEFFF5FFF7	7FFFBFFBF7
FFFFBFDBF7	FFFFBFFBD7	3FFFBFFFF7	BFFFBFFBF7	EFFFBFFBF7	AFFFBFFFF7
5FFFFFFBF7	DFFFFFDBF7	DFFFFFFBD7	1FFFFFFFF7	9FFFFFFBF7	CFFFFFFBF7
8FFFFFFFF7	FFFFB7FBF7	DFFFF7FBF7	7FFFBFF7F7	FFFFBFD7F7	FFFFBFF7D7
BFFFBFF7F7	EFFFBFF7F7	5FFFFFF7F7	DFFFFFD7F7	DFFFFFF7D7	9FFFFFF7F7
CFFFFFF7F7	7FFFBFFFF3	7FFFBFFFE7	7FDFBFFFF7	7FFFBDFFF7	FFFFBFDFF3
FFFFBFFFD3	FFFFBFDFE7	FFDFBFDFF7	FFFFBDDFF7	FFFFBFFFC7	FFDFBFFFD7
FFFFBDFFD7	BFFFBFFFF3	BFFFBFFFE7	BFDFBFFFF7	BFFFBDFFF7	EFFFBFFFF3
EFFFBFFFE7	EFDFBFFFF7	EFFFBDFFF7	5FFFBFFFF7	DFFFBFDFF7	DFFFBFFFD7
9FFFBFFFF7	CFFFBFFFF7	5FFFFFFFF3	5FFFFFFFE7	5FDFFFFFF7	5FFFFDFFF7
DFFFFFDFF3	DFFFFFFFD3	DFFFFFDFE7	DFDFFFDFF7	DFFFFDDFF7	DFFFFFFFC7
DFDFFFFFD7	DFFFFDFFD7	9FFFFFFFF3	9FFFFFFFE7	9FDFFFFFF7	9FFFFDFFF7

Table 13. IV terms of degree 43 in z_{257}-part 6

CFFFFFFFF3	CFFFFFFFE7	CFDFFFFFF7	CFFFFDFFF7	FFFFB7F7F7	DFFFF7F7F7
FFFFB7FFF3	FFFFB7FFE7	FFDFB7FFF7	FFFFB5FFF7	DFFFB7FFF7	DFFFF7FFF3
DFFFF7FFE7	DFDFF7FFF7	DFFFF5FFF7	FFBFBFDFF7	FFBFBFFFD7	7FBFBFFFF7
77FFBFFFF7	7EFFBFFFF7	F7FFBFDFF7	FEFFBFDFF7	F7FFBFFFD7	FEFFBFFFD7
BFBFBFFFF7	B7FFBFFFF7	BEFFBFFFF7	EFBFBFFFF7	E7FFBFFFF7	EEFFBFFFF7
DFBFFFDFF7	DFBFFFFFD7	5FBFFFFFF7	57FFFFFFF7	5EFFFFFFF7	D7FFFFDFF7
DEFFFFDFF7	D7FFFFFFD7	DEFFFFFFD7	9FBFFFFFF7	97FFFFFFF7	9EFFFFFFF7
CFBFFFFFF7	C7FFFFFFF7	CEFFFFFFF7	FFBFB7FFF7	F7FFB7FFF7	FEFFB7FFF7
DFBFF7FFF7	D7FFF7FFF7	DEFFF7FFF7	FDFFFFFDF6	FFFFFFDDF6	FFFFFFFDD6
FFFFF7FDF6	7DEFFFFFF7	7DFFFFEFF7	7DFDFFFFF7	FDEFFFDFF7	FFEFFFDFD7
7FEFFFDFF7	FDFFFFCFF7	FFFFFFCFD7	7FFFFFCFF7	FDFDFFDFF7	FFFDFFDFD7
7FFDFFDFF7	FDEFFFFFD7	7FEFFFFFD7	FDFFFFEFD7	7FFFFFEFD7	FDFDFFFFD7
7FFDFFFFD7	7FEFF7FFF7	7FFFF7EFF7	7FFDF7FFF7	BDEFFFFFF7	BDFFFFEFF7
BDFDFFFFF7	BFEFFFDFF7	BFFFFFCFF7	BFFDFFDFF7	BFEFFFFFD7	BFFFFFEFD7
BFFDFFFFD7	BFEFF7FFF7	BFFFF7EFF7	BFFDF7FFF7	EDEFFFFFF7	EDFFFFEFF7
EDFDFFFFF7	EFEFFFDFF7	EFFFFFCFF7	EFFDFFDFF7	EFEFFFFFD7	EFFFFFEFD7
EFFDFFFFD7	EFEFF7FFF7	EFFFF7EFF7	EFFDF7FFF7	6DFFFFFFF7	6FFFFFDFF7
6FFFFFFFD7	6FFFF7FFF7	FDEFF7FFF7	FDFFF7EFF7	FDFDF7FFF7	FFEFF7DFF7
FFFFF7CFF7	FFFDF7DFF7	FFEFF7FFD7	FFFFF7EFD7	FFFDF7FFD7	5DFFFFFFF7
DDFFFFDFF7	DFFFFFDFD7	5FFFFFDFF7	DDFFFFFFD7	5FFFFFFFD7	5FFFF7FFF7
9DFFFFFFF7	9FFFFFDFF7	9FFFFFFFD7	9FFFF7FFF7	CDFFFFFFF7	CFFFFFDFF7
CFFFFFFFD7	CFFFF7FFF7	DDFFF7FFF7	DFFFF7DFF7	DFFFF7FFD7	FDFFDFFDF7
FFFFDFDDF7	FFFFDFFDD7	FFFFD7FDF7	7DFFDFFFF7	FDFFDFDFF7	FFFFDFDFD7
7FFFDFDFF7	FDFFDFFFD7	7FFFDFFFD7	7FFFD7FFF7	BDFFDFFFF7	BFFFDFDFF7
BFFFDFFFD7	BFFFD7FFF7	EDFFDFFFF7	EFFFDFDFF7	EFFFDFFFD7	EFFFD7FFF7
FDFFD7FFF7	FFFFD7DFF7	FFFFD7FFD7	FDFFFFF9F7	FFFFFFD9F7	FFFFFFF9D7
FFFFF7F9F7	7DFFFFFBF7	FDFFFFDBF7	FFFFFFDBD7	7FFFFFDBF7	FDFFFFFBD7
7FFFFFFBD7	7FFFF7FBF7	3DFFFFFFF7	3FFFFFDFF7	3FFFFFFFD7	3FFFF7FFF7
BDFFFFFBF7	BFFFFFDBF7	BFFFFFFBD7	BFFFF7FBF7	EDFFFFFBF7	EFFFFFDBF7
EFFFFFFBD7	EFFFF7FBF7	ADFFFFFFF7	AFFFFFDFF7	AFFFFFFFD7	AFFFF7FFF7
FDFFF7FBF7	FFFFF7DBF7	FFFFF7FBD7	FDFFFFF5F7	FFFFFFD5F7	FFFFFFF5D7
FFFFF7F5F7	FDFFBFFDF7	FFFFBFDDF7	FFFFBFFDD7	FFFFB7FDF7	FDFFFFFDF3
FDFFFFFDE7	FDDFFFFDF7	FDFFFDFDF7	FFFFFFDDF3	FFFFFFFDD3	FFFFFFDDE7
FFDFFFDDF7	FFFFFDDDF7	FFFFFFFDC7	FFDFFFFDD7	FFFFFDFDD7	FFFFF7FDF3
FFFFF7FDE7	FFDFF7FDF7	FFFFF5FDF7	7DFFFFF7F7	FDFFFFD7F7	FFFFFFD7D7
7FFFFFD7F7	FDFFFFF7D7	7FFFFFF7D7	7FFFF7F7F7	BDFFFFF7F7	BFFFFFD7F7
BFFFFFF7D7	BFFFF7F7F7	EDFFFFF7F7	EFFFFFD7F7	EFFFFFF7D7	EFFFF7F7F7
7DFFBFFFF7	FDFFBFDFF7	FFFFBFDFD7	7FFFBFDFF7	FDFFBFFFD7	7FFFBFFFD7
7FFFB7FFF7	BDFFBFFFF7	BFFFBFDFF7	BFFFBFFFD7	BFFFB7FFF7	EDFFBFFFF7
EFFFBFDFF7	EFFFBFFFD7	EFFFB7FFF7	7DFFFFFFF3	7DFFFFFFE7	7DDFFFFFF7
7DFFFDFFF7	7FFFFFDFF3	FDFFFFDFF3	FFFFFFDFD3	FDFFFFFFD3	7FFFFFFFD3
FDFFFFDFE7	FFFFFFDFC7	7FFFFFDFE7	FDDFFFDFF7	FFDFFFDFD7	FDFFFDDFF7
FFFFFDDFD7	7FDFFFDFF7	7FFFFDDFF7	FDFFFFFFC7	7FFFFFFFC7	FDDFFFFFD7
FDFFFDFFD7	7FDFFFFFD7	7FFFFDFFD7	7FFFF7FFF3	7FFFF7FFE7	7FDFF7FFF7
7FFFF5FFF7	BDFFFFFFF3	BDFFFFFFE7	BDDFFFFFF7	BDFFFDFFF7	BFFFFFDFF3
BFFFFFFFD3	BFFFFFDFE7	BFDFFFDFF7	BFFFFDDFF7	BFFFFFFFC7	BFDFFFFFD7
BFFFFDFFD7	BFFFF7FFF3	BFFFF7FFE7	BFDFF7FFF7	BFFFF5FFF7	EDFFFFFFF3
EDFFFFFFE7	EDDFFFFFF7	EDFFFDFFF7	EFFFFFDFF3	EFFFFFFFD3	EFFFFFDFE7
EFDFFFDFF7	EFFFFDDFF7	EFFFFFFFC7	EFDFFFFFD7	EFFFFDFFD7	EFFFF7FFF3
EFFFF7FFE7	EFDFF7FFF7	EFFFF5FFF7	FDFFF7F7F7	FFFFF7D7F7	FFFFF7F7D7
FDFFB7FFF7	FFFFB7DFF7	FFFFB7FFD7	FDFFF7FFF3	FDFFF7FFE7	FDDFF7FFF7
FDFFF5FFF7	FFFFF7DFF3	FFFFF7FFD3	FFFFF7DFE7	FFDFF7DFF7	FFFFF5DFF7
FFFFF7FFC7	FFDFF7FFD7	FFFFF5FFD7	FFBFFFDDF7	FFBFFFFDD7	FDBFFFFDF7
F5FFFFFDF7	FCFFFFFDF7	F7FFFFDDF7	FEFFFFDDF7	F7FFFFFDD7	FEFFFFFDD7
FFBFF7FDF7	F7FFF7FDF7	FEFFF7FDF7	7DFFFFFDF7	7FFFFFDDF7	FDFFFFDDF7
FFFFFFDDD7	FDFFFFFDD7	7FFFFFFDD7	7FFFF7FDF7	BDFFFFFDF7	BFFFFFDDF7
BFFFFFFDD7	BFFFF7FDF7	EDFFFFFDF7	EFFFFFDDF7	EFFFFFFDD7	EFFFF7FDF7
FDFFF7FDF7	FFFFF7DDF7	FFFFF7FDD7	FDBFFFDFF7	FFBFFFDFD7	7FBFFFDFF7
FDBFFFFFD7	7FBFFFFFD7	7DBFFFFFF7	75FFFFFFF7	7CFFFFFFF7	77FFFFDFF7
7EFFFFDFF7	F5FFFFDFF7	FCFFFFDFF7	F7FFFFDFD7	FEFFFFDFD7	7DFFFFDFF7
FDFFFFDFD7	7FFFFFDFD7	F5FFFFFFD7	FCFFFFFFD7	7DFFFFFFD7	77FFFFFFD7
7EFFFFFFD7	7FBFF7FFF7	77FFF7FFF7	7EFFF7FFF7	BFBFFFDFF7	BFBFFFFFD7
BDBFFFFFF7	B5FFFFFFF7	BCFFFFFFF7	B7FFFFDFF7	BEFFFFDFF7	BDFFFFDFF7
BFFFFFDFD7	BDFFFFFFD7	B7FFFFFFD7	BEFFFFFFD7	BFBFF7FFF7	B7FFF7FFF7
BEFFF7FFF7	EFBFFFDFF7	EFBFFFFFD7	EDBFFFFFF7	E5FFFFFFF7	ECFFFFFFF7
E7FFFFDFF7	EEFFFFDFF7	EDFFFFDFF7	EFFFFFDFD7	EDFFFFFFD7	E7FFFFFFD7
EEFFFFFFD7	EFBFF7FFF7	E7FFF7FFF7	EEFFF7FFF7	7DFFF7FFF7	7FFFF7DFF7
7FFFF7FFD7	BDFFF7FFF7	BFFFF7DFF7	BFFFF7FFD7	FFBFF7DFF7	FFBFF7FFD7
FDBFF7FFF7	F5FFF7FFF7	FCFFF7FFF7	F7FFF7DFF7	FEFFF7DFF7	FDFFF7DFF7
FFFFF7DFD7	FDFFF7FFD7	F7FFF7FFD7	FEFFF7FFD7	EDFFF7FFF7	EFFFF7DFF7
EFFFF7FFD7					