Some Randomness Experiments on TRIVIUM

Subhabrata Samajder and Palash Sarkar
Applied Statistics Unit
Indian Statistical Institute
203, B.T.Road, Kolkata, India - 700108.
{subhabrata_r,palash}@isical.ac.in

Abstract

The first output bit of TRIVIUM can be considered to be a boolean function of 80 key and 80 IV variables. Choose n ($n \leq 30$) of the key variables and set the other variables to constant values. This gives an n-variable boolean function. In this work, we experimentally find examples of such boolean functions which deviate from a uniform random n-variable boolean function in a statistically significant manner. This improves upon the previously reported experimental 'non-randomness' result using the cube testing methodology by Aumasson et al in 2009 for TRIVIUM restricted to 885 rounds. In contrast, we work with full TRIVIUM and instead of using the cube methodology we directly find the algebraic normal form of the restricted version of the first output bit of TRIVIUM. We note, however, that our work does not indicate any weakness of TRIVIUM. On the other hand, the kind of experiments that we conduct for TRIVIUM can also be conducted for other ciphers.

Keywords : stream ciphers, TRIVIUM, statistical test, non-randomness.

1 Introduction

TRIVIUM is a hardware oriented synchronous stream cipher that was submitted to the Profile II (hardware) of the eSTREAM competition by Christophe De Cannière and Bart Preneel [DCP]. TRIVIUM maintains an internal state of size 288 bits. The state is subdivided into 3 shift registers of sizes 93, 84 and 111 bits each. It uses a simple quadratic state update function with 3 AND operations as the only non-linear operations per round. There are 1152 initialization rounds. During the key generation, at each step, the state is updated and a single key bit is produced. This key bit is the XOR of 6 state bits. Over the years TRIVIUM has received much attention from the research community due to its simple structure. However, there is still no known attack on full version of TRIVIUM which works better than exhaustive search.

To gain a better understanding of the full cipher, scaled-down variants, such as Bivium A and Bivium B [Rad06], have been studied. Both Bivium A and Bivium B use two shift registers as their internal state unlike TRIVIUM which uses three. The attacks on TRIVIUM can be broadly classified into two categories. The first type analyses the scaled-down variants (Bivium A and Bivium B [Rad06]) and tries to extrapolate their results to the full TRIVIUM. The second approach has been to study the reduced-round variants of the cipher, i.e., TRIVIUM with 'r' rounds of key initialization where $r \leq 1152$.

Early results on reduced-round variants of TRIVIUM can be found in [TK07] and [Vie07]. In [TK07], Turan et al used Matsui's linear cryptanalysis method to get a linear approximation with a bias 2^{-31} for 288 rounds of key initialization. Whereas in [Vie07], Vielhaber used an IV resynchronization attack with

2^6 IV's to break 576 rounds of TRIVIUM. Englund et al in [EJT07], experimentally showed statistical weakness in the keystream of TRIVIUM when reduced to 736 rounds of key initialization. In [O'N07], O'Neal claimed that TRIVIUM with 1152 rounds of key initialization may not be secure and proposed that the initialization rounds for TRIVIUM should be increased to $4 \times 1152 = 4608$ rounds. Fischer et al in [FKM08] used a framework for chosen IV statistical distinguishing analysis of stream ciphers to extract few key bits of TRIVIUM when reduced to 672 rounds of key initializations.

The cube attack was proposed in [DS09] by Dinur et al and used to recover the key after 767 initialization rounds. The attack required 2^{45} bit operations and the authors showed that this can be further reduced to 2^{36} bit operations. In [ADMS09], Aumasson et al introduced a new class of attacks called cube testers and developed distinguishers for 790 rounds of TRIVIUM with 2^{30} complexity and were able to detect non-randomness over 885 rounds in 2^{27} complexity, improving on the original 767-round cube attack.

Recently in [FV13], Fouque and Vannet increased the number of attacked initialization rounds by improving the time complexity of computing cube. They were able to find a key recovery attack requiring 2^{39} queries for 784 initialization rounds and were also able to provide another key recovery attack up to 799 rounds with a complexity of 2^{40} for queries and 2^{62} for the exhaustive search part. In their attack, they used the Moebius Transform to improve on the time taken in the pre-processing stage of cube attack.

Our Results: The motivation for our work is the discovery of non-randomness after 885 rounds of TRIVIUM reported in [ADMS09]. We briefly discuss this result. The input key and IV variables are divided into two groups called cube variables (CV) and superpoly variables (SV). Suppose $g(x_1, \ldots, x_c; y_1, \ldots, y_s)$ denotes the boolean function representing the first keystream bit of TRIVIUM. There are $c + s$ input variables, where CV $= \{x_1, \ldots, x_c\}$ and SV $= \{y_1, \ldots, y_s\}$. Then superpoly s_{CV} of g corresponding to a cube of size c is defined as

$$s_{CV}(y_1, y_2, \ldots, y_s) = \bigoplus_{(x_1, x_2, \ldots, x_c) \in \mathbb{F}_2^c} g(x_1, x_2, \ldots, x_c; y_1, y_2, \ldots, y_s),$$

which is an s-variable boolean function in the variables SV. The details about the non-randomness of 885 rounds of TRIVIUM reported in [ADMS09] is a bit sketchy. We try to provide some more details. A cube of size 27 of IV variables mentioned in [DS09] was considered. Set all other IV variables to 0. It was *experimentally* discovered that in the superpoly corresponding to this cube, the key variables 1, 4 and 5 are neutral (i.e., changing their values does not affect the outcome of the polynomial). It is mentioned that the zero key was used which would imply that all key bits other than 1, 4 and 5 were set to zero. It was argued that the discovery of such a polynomial in the structure of TRIVIUM is an evidence of non-randomness. This claim is also well accepted in the literature.

In general terms the above example can be viewed as follows. Let $g(x_1, \ldots, x_c; y_1, \ldots, y_s)$ be the first output bit of TRIVIUM (after 885 rounds). The authors discover a transformation Φ such that the key bits 1, 4 and 5 are neutral for the boolean function $\Phi(g)$. The transformation Φ consists of applying the cube, setting IV and the other key bits to 0.

The above kind of experimentally discovered 'non-randomness' after 885 rounds reported in [ADMS09] forms the motivation for our work. We ask the question as to whether it is possible to experimentally discover some kind of 'non-randomness' in full TRIVIUM. As above, if g denotes the boolean function representing the first output bit, our goal is to discover a transformatiion Ψ such that the boolean function $\Psi(g)$ shows some statistically quantifiable deviation from a uniform random function. The Ψ that we consider does not involve evaluating a cube. The function g depends on 80 key and 80 IV variables. The transformation Ψ consists of choosing n key variables and setting the other key and IV variables to

constant values. As a result $\Psi(g)$ is a boolean function on n variables. We are able to experimentally obtain examples of $\Psi(g)$ whose deviation from a uniform boolean function is statistically significant. Here n is a parameter which is at most 30.

The main computational challenge is to obtain the algebraic normal form (ANF) of $\Psi(g)$. For this we discuss two methods. The first one symbolically evolves TRIVIUM over the full 1152 rounds. This requires a fast algorithm for multiplying two boolean functions given by their ANFs and for this task we use the implementation reported in [Sam13]. The second method proceeds by first obtaining the truth table representation for $\Psi(g)$ and then using the Moebius transformation to obtain the ANF. Either of the methods yields both the ANF and also the truth table representation of $\Psi(g)$.

Suppose u^* is a uniform random boolean function of n variables. The weight of u^* is a random variable with mean 2^{n-1}. Given a probability α, there is an interval I_α such that the weight of u^* is in I_α with probability at least α. We say that $\Psi(g)$ is unbalanced at level α if its weight lies outside the internal I_α. Similar notions of algebraic unbalancedness for $\Psi(g)$ can be defined with respect to the total number of monomials in the ANF of $\Psi(g)$ and also with respect to the number of monomials of degree d in the ANF of $\Psi(g)$. We also define a notion of unbalancedness over an l-dimensional uniform random vectorial boolean function $\tilde{u}^*$. Further details of the corresponding statistical tests are provided later.

In this work, we experimentally find concrete examples of $\Psi(g)$ for $n = 10, 20$ and 30 which are unbalanced, algebraically unbalanced with respect to the total number of monomials and also with respect to monomials of certain specific degrees. We further provide give concrete examples of $\Psi(\tilde{g})$ for $n = 20, 30$ (where $\tilde{g}$ denotes an l-dimensional vectorial boolean function) which are unbalanced. These results are obtained for level α corresponding to more than 99% probability. For lower values of α, we are able to obtain examples of $\Psi(g)$ which simultaneously fail several of the statistical tests. Our experiments consist of randomly selecting the n key variables and choosing the values for the other $160 - n$ key and IV variables. This in effect randomly chooses the transformation Ψ. The reported results are obtained by randomly choosing possibilities for Ψ several thousands of times.

We make no claims that our results exhibit a weakness of TRIVIUM. There are two implications of our work. First, our results show that to experimentally discover some 'non-random' polynomial in the structure of TRIVIUM, the complicated cube analysis technique of [ADMS09] is unnecessary. Instead one can simply look at the boolean function representing the first output bit by setting $160 - n$ of the input variables to constant values. Second, our work discovers 'non-randomness' in TRIVIUM after the full 1152 rounds of initialization whereas [ADMS09] reported results only after 885 rounds.

The method described here is not particular to TRIVIUM. It can be applied to any cipher. Whether the results will be similar to that obtained for TRIVIUM is not clear and exploring this can form possible future work.

Related Work: We are not the first to consider applying statistical tests to the algebraic normal form of the output bits of a stream cipher. An early work by Filio [Fil02] and later follow-up in [Saa06] had explored this possibility. Our approach, however, differs from that of [Fil02, Saa06] in the following way. The work considered the ANFs of the first N output bits of a stream cipher and investigated the distribution of monomials of degree d in these ANFs for $d \leq 3$. The study was thus aggregated and unlike the specific 'non-randomness' example reported in [ADMS09].

2 A Brief Description of TRIVIUM

TRIVIUM maintains a 288-bit internal state "S" denoted by $S = (s_1, s_2, \ldots, s_{288})$ and uses two algorithms, namely a key initialization algorithm, which we call the key and IV setup, and a key stream generation algorithm. The state S is further divided into 3 shift registers, namely $S_1 = (s_1, s_2, \ldots, s_{93})$, $S_2 = (s_{94}, s_{95}, \ldots, s_{177})$ and $S_3 = (s_{178}, s_{179}, \ldots, s_{288})$.

2.1 Key and IV Setup

The algorithm is initialized by loading an 80-bit key into the first 80-bits of the state S, i.e., $s_1, s_2, \ldots, s_{80}$ and an 80-bit IV into the state bits $s_{94}, s_{95}, \ldots, s_{173}$ and setting all remaining bits to 0, except for s_{286}, s_{287}, and s_{288}, which are set to 1. Each round of the iterative process extracts the values of 15 specific state bits and uses them to update 3 bits of the state. This is repeated for $4 \times 288 = 1152$ times. This can be summarized by the following pseudo-code (Algorithm 1):

Algorithm 1: TRIVIUM - Key and IV Setup.

$(s_1, s_2, \ldots, s_{93}) \leftarrow (K_1, K_2, \ldots, K_{80}, 0, \ldots, 0)$
$(s_{94}, s_{95}, \ldots, s_{177}) \leftarrow (IV_1, IV_2, \ldots, IV_{80}, 0, 0, 0, 0)$
$(s_{178}, s_{179}, \ldots, s_{288}) \leftarrow (0, \ldots 0, 1, 1, 1)$
for $i = 1$ *to* $4 \cdot 288$ **do**
  $t_1 \leftarrow s_{66} \oplus s_{91} \cdot s_{92} \oplus s_{93} \oplus s_{171}$
  $t_2 \leftarrow s_{162} \oplus s_{175} \cdot s_{176} \oplus s_{177} \oplus s_{264}$
  $t_3 \leftarrow s_{243} \oplus s_{286} \cdot s_{287} \oplus s_{288} \oplus s_{69}$
  $(s_1, s_2, \ldots, s_{93}) \leftarrow (t_3, s_1, \ldots, s_{92})$
  $(s_{94}, s_{95}, \ldots, s_{177}) \leftarrow (t_1, s_{94}, \ldots, s_{176})$
  $(s_{178}, s_{179}, \ldots, s_{288}) \leftarrow (t_2, s_{178}, \ldots, s_{287})$
end

2.2 Key Stream Generation

The key stream generation algorithm is similar to that of the key initialization algorithm except that at each round, a single bit which is a linear function of six state bits, is output before the state update. This process repeats itself until the requested $N \leq 2^{64}$ bits of key stream is generated. The complete description is given by the following pseudo-code (Algorithm 2):

3 Algebraic Normal Forms of the Output Bits of TRIVIUM

Let us denote the key K by $(k_1, k_2, \ldots, k_{80})$ and the IV by $(iv_1, iv_2, \ldots, iv_{80})$. If instead of bits, we consider the key and the IV as variables then the state is initialized as follows:

$$(s_1, s_2, \ldots, s_{93}) \leftarrow (k_1, k_2, \ldots, k_{80}, 0, \ldots, 0),$$

$$(s_{94}, s_{95}, \ldots, s_{177}) \leftarrow (iv_1, iv_2, \ldots, iv_{80}, 0, 0, 0, 0),$$

$$(s_{178}, s_{179}, \ldots, s_{288}) \leftarrow (0, \ldots 0, 1, 1, 1).$$

Algorithm 2: TRIVIUM - Key Stream Generation.

for $i = 1$ *to* N **do**
$t_1 \leftarrow s_{66} \oplus s_{93}$
$t_2 \leftarrow s_{162} \oplus s_{177}$
$t_3 \leftarrow s_{243} \oplus s_{288}$
$z_i \leftarrow t_1 \oplus t_2 \oplus t_3$
$t_1 \leftarrow t_1 \oplus s_{91} \cdot s_{92} \oplus s_{171}$
$t_2 \leftarrow t_2 \oplus s_{175} \cdot s_{176} \oplus s_{264}$
$t_3 \leftarrow t_3 \oplus s_{286} \cdot s_{287} \oplus s_{69}$
$(s_1, s_2, \ldots, s_{93}) \leftarrow (t_3, s_1, \ldots, s_{92})$
$(s_{94}, s_{95}, \ldots, s_{177}) \leftarrow (t_1, s_{94}, \ldots, s_{176})$
$(s_{178}, s_{179}, \ldots, s_{288}) \leftarrow (t_2, s_{178}, \ldots, s_{287})$
end

During each state update these state bits get multiplied and added in the boolean function ring defined over the variables K and IV. Thus, considering each state bit as a boolean function in $80 + 80 = 160$ variables, one can view each state update as performing 3 multiplications (1 for each $t_i, i = 1, 2, 3.$) and 9 additions (3 for each $t_i, i = 1, 2, 3.$). Addition is just bitwise XOR, whereas multiplication is that of two boolean functions given by their ANF's.

Handling the ANF of a boolean function on 160 variables is infeasible. Hence, we reduce the number of variables in the following manner. The key and IV bit positions which are to be treated as variables are randomly selected. These selected bit positions are then renamed as variables $k_1, k_2, \ldots, k_{n_k}$ and $iv_1, iv_2, \ldots, iv_{n_{iv}}$, such that $n_k + n_{iv} = n$. We work with $n \leq 30$. The rest of the key and IV bit positions are then set randomly to either 0 or 1. The outputs bits of TRIVIUM can then be considered to be boolean functions of n variables. We provide two methods to compute the ANFs of the output bits.

3.1 Method - 1

A symbolic computation of TRIVIUM is carried out where the state bits are treated as polynomials in $k_1, k_2, \ldots, k_{n_k}$ and $iv_1, iv_2, \ldots, iv_{n_{iv}}$. As a result, the first output bit which is the bitwise XOR of six state bits, namely $s_{66}, s_{93}, s_{162}, s_{177}, s_{243}$ and s_{288} is also a polynomial in these variables.

The main time-consuming step in the above symbolic computation is that of multiplying the ANFs of two boolean functions. We used the implementation MultANF_{64} of multiplication described in [Sam13]. Using this algorithm, two 30-variable boolean functions can be multiplied in less than 2 seconds on a 3 GHz processor. Carrying out the simulation of full 1152 rounds of TRIVIUM with $n = 30$ requires 3456 multiplications and the entire computation requires about one-and-half hours.

The complexity of Method - 1 is dominated by the number of multiplications of two n-variable boolean functions. We know from [Jou09] that the complexity for multiplying two n-variable boolean functions in their ANF is of the order of $3n2^n$. TRIVIUM uses 3 multiplications at each round. Therefore the total complexity of Method - 1 when used to evaluate an r-round variant of TRIVIUM (TRIVIUM with r rounds of key initializations) is of the order of $9rn2^n$.

3.2 Method - 2

The second method first constructs the truth table of the first output bit z_1 which is a polynomial in n variables as mentioned above. A fast implementation of TRIVIUM is used to evaluate z_1 on all possible 2^n input combinations. This provides the truth table representation of z_1. This is converted into the ANF format using the Moebius transformation (see [Jou09] for a description of this algorithm).

The complexity of Method - 2 when used to evaluate r-round variant of TRIVIUM is 2^n computations of r-round TRIVIUM plus the cost required to convert the truth table of an n-variable boolean function into its corresponding ANF. And the cost for the conversion of the truth table to its ANF is $\mathcal{O}(n2^n)$.

3.3 Complexity of the Cube Tester

Let the IV variables be divided into two parts the cube variables (CV) and the superpoly variables (SV). All the key variables are also treated as SV. Consider an r-round variant of TRIVIUM, i.e., TRIVIUM with r rounds of key initialization. The cube tester [ADMS09] selects the cube variables from the 80 available IV variables. It treats the remaining key and IV variables as SV. It further divides the SV variables into two parts. Let us call the first part of variables as the function variables (FV) and retain the name SV for the second part of variables. In the cube testers for TRIVIUM mentioned in [ADMS09] the SV variables are all set to zero whereas the FV variables are set to fixed random values. Then 2^{CV} many executions of r-round TRIVIUM is used to find the value of the superpoly at (SV, FV). This is repeated for N different choices of FV. Therefore, to test one particular superpoly the attacker needs $N \times 2^{CV}$ executions of r-round TRIVIUM. In their paper [ADMS09] the authors had mainly concentrated on the cubes that were listed in the appendix of [DS09]. As mentioned in the paper the maximum size of FV and CV considered are 5 and 30 respectively. Therefore in order to compute the complete truth table of a 5-variable function the cube tester has to compute 2^{35} r-round variant of TRIVIUM.

4 Some Elementary Statistics

Let $X_1, \ldots, X_n$ be independent Bernoulli distributed random variables with $\Pr[X_i = 0] = p$. Then $X = X_1 + \cdots + X_N$ follows $\mathsf{Bin}(N, p)$ with expected value Np. Given a probability α, there is an interval I_α which is symmetric about the mean, such that $\Pr[X \in I_\alpha] \geq \alpha$. If N is large enough, then the binomial distribution is well approximated by the normal distribution and it is quite routine to use the normal approximation to obtain $\Pr[X \in I_\alpha]$. Further, given α, the interval I_α is found by converting to standard normal and then using tables for the standard normal distribution.

Denote by u^* a uniform random n-variable polynomial. For our study, we will follow the above statistical approach for u^* in the following settings.

Total number of monomials in u^*: Any particular monomial occurs in u^* with probability $1/2$ and is independent of the occurrence of any other monomial. If we denote the 2^n possible monomials of n variables as $m_0, \ldots, m_{2^n-1}$, then we have 2^n random variables $X_0, \ldots, X_{2^n-1}$ where X_i is 1 if m_i is present in u^* and 0 otherwise. The random variables $X_0, \ldots, X_{2^n-1}$ are independent Bernoulli distributed variables with $\Pr[X_i = 1] = 1/2$. The number of monomials in u^* is $X = X_0 + \cdots + X_{2^n-1}$ and follows $\mathsf{Bin}(2^n, 1/2)$.

Number of monomials of degree d in u^*: Consider the number of monomials of degree d in u^*. There are a total of $\binom{n}{d}$ such monomials. In a manner similar to the above case, the number of monomials of

degree d in u^* follows $\mathsf{Bin}\left(\binom{n}{d}, 1/2\right)$.

Weight of u^*: For any input, the output of u^* is 0 or 1 with probability 1/2 and this is independent of the output of u^* on any other input. So, as in the case of total number of monomials, the weight of u^* follows $\mathsf{Bin}(2^n, 1/2)$.

Determining whether a given polynomial is 'non-random': Given a particular n-variable boolean function f, we can compute the total number of monomials in f. If the number of monomials turns out to be 'significantly' away from 2^{n-1}, then this is usually taken as an indication of some kind of non-randomness in f. We will use the term algebraically unbalanced to express the idea that the total number of monomials in f deviates significantly from the expected number of monomials in a uniform random polynomial.

Statistical tests will be conducted as follows. For a probability α, we first compute the interval I_α such that the total number of monomials in u^* will be in I_α with probability at least α. Then the total number of monomials in the given function f is calculated. If this lies outside the interval I_α, then we say that the function f fails the algebraic balancedness test for probability α, or that f is algebraically unbalanced at level α written as AU_α.

In a similar manner we conduct tests on f for monomials of degree d and the weight of f. Given an α, the interval I_α for the weight of f will be the same as that for the total number of monomials. On the other hand, when considering monomials of degree d, the interval $I_\alpha^{(d)}$ will depend on d. This is because the number of trials in the binomial distribution corresponding to monomials of degree d is $\binom{n}{d}$. If the number of monomials of degree d in f is outside the interval $I_\alpha^{(d)}$, then we will say that f is d-algebraically unbalanced at level α, written as d-AU_α. Similarly, if the weight of f is outside the interval I_α, then we say that f is unbalanced at level α, written as U_α.

5 Unbalancedness Over First l Output Bits

This section generalizes the test for unbalancedness of a boolean function f to vectorial boolean functions of dimension l. An l-dimensional vectorial boolean function is defined as $\tilde{f} : \mathbb{F}_2^n \to \mathbb{F}_2^l$, such that

$$\tilde{f}(x_1, x_2, \ldots, x_n) = (f_1(x_1, x_2, \ldots, x_n), f_2(x_1, x_2, \ldots, x_n), \ldots, f_l(x_1, x_2, \ldots, x_n))$$

where each $f_i(x_1, x_2, \ldots, x_n)$, $i = 1, 2, \ldots, l$ are n-variable boolean functions. Denote a uniform random vectorial boolean function by $\tilde{u^*}$, where each of its coordinates u_i^*, $i = 1, 2, \ldots, l$ behaves as uniformly and independently distributed n-variable polynomials.

Statistics Involved: Let $X_{i,j} \in \{0, 1\}, i \in \{1, 2, 3, \ldots, l\}$ and $j \in \{1, 2, 3, \ldots, N\}$, denote mutually independent random variables with $X_{i,j} \sim \mathsf{Ber}(p_i)$ for all $j \in \{1, 2, \ldots, N\}$. Let $X_i = \sum_{j=1}^{N} X_{i,j}$, $i \in \{1, 2, 3, \ldots, l\}$. Then $X_1, X_2, \ldots, X_l$ are independently distributed with each $X_i \sim \mathsf{Bin}(N, p_i)$. For sufficiently large N, X_i's are well approximated by normal distribution with mean Np_i and variance $Np_i(1 - p_i)$. Let $Y_i = \frac{X_i - Np_i}{\sqrt{Np_i(1-p_i)}}$. Then the random variables $Y_1, Y_2, \ldots, Y_l$ and hence $Y_1^2, Y_2^2, \ldots, Y_l^2$ are mutually independent with each $Y_i \sim \mathcal{N}(0, 1)$ and $Y_i^2 \sim \chi^2(1)$ (Chi-squared distribution with 1 degree of freedom). Therefore $\sum_{i=1}^{l} Y_i^2 \sim \chi^2(l)$ (Chi-squared distribution with l degree of freedom). For a given α and l degrees of freedom, we therefore can get an interval $I_\alpha = \left[0, \chi^2(l)_\alpha\right]$ such that $\Pr\left[\sum_{i=1}^{l} Y_i^2 \in I_\alpha\right] = \alpha$.

Weight of $\tilde{u}^*$: For any input and any coordinate i ($i \in \{1, 2, \ldots, l\}$), the output of u_i^* is 0 or 1 with probability 1/2 and this is independent on any other input and coordinate i. For a given coordinate i we denote the 2^n ($= N$) possible outputs of the n-variable boolean function u_i^* as $X_{i,0}, X_{i,1}, \ldots, X_{i,2^n-1}$, where $X_{i,j}$ is 1 if the j^{th} output of u_i^* is 0 and 0 otherwise. Given i, the random variables $X_{i,0}, X_{i,1}, \ldots, X_{i,2^n-1}$ are independent Bernoulli distributed variables with $\Pr[X_{i,j} = 1] = 1/2$. Thus the number of zeros in the 2^n outputs of u_i^* is $X_i = \sum_{j=0}^{2^n-1} X_{i,j}$ and follows $\mathsf{Bin}(2^n, 1/2)$. For $n \geq 5$, X_i's are well approximated by $\mathcal{N}\left(2^{n-1}, 2^{n-2}\right)$. Hence, the mutually independent random variables $Y_i^2 = \left(\frac{X_i - 2^{n-1}}{\sqrt{2^{n-2}}}\right)^2 \sim \chi^2(1)$ for all $i \in \{1, 2, \ldots, l\}$. Therefore, $\sum_{i=1}^{l} Y_i^2 \sim \chi^2(l)$.

Determining a l-dimensional vectorial boolean function to be 'non-random': Given an l-dimensional vectorial boolean function $\tilde{f}$ in variables $x_1, x_2, \ldots, x_n$, we construct its l-dimensional truth table. For each of the 2^n possible values of the variables $x_1, x_2, \ldots, x_n$, we consider the corresponding values of $\tilde{f}(x_1, x_2, \ldots, x_n)$. This corresponds to the l-dimensional truth table of $\tilde{f}$. Each coordinate i of this l-dimensional truth table individually corresponds to the truth table of f_i. Let n_i denote the number of zeros in the truth table corresponding to f_i. Compute $\sum_{i=1}^{l} \left(\frac{n_i - 2^{n-1}}{\sqrt{2^{n-2}}}\right)^2$ For a given α and l, if $\sum_{i=1}^{l} \left(\frac{n_i - 2^{n-1}}{\sqrt{2^{n-2}}}\right)^2 > \chi^2(l)_\alpha$ then we say that the l-dimensional vectorial boolean function $\tilde{f}$ is unbalanced at level α, written as $\mathsf{U}_{l,\alpha}$

6 Searching for (Algebraically) Unbalanced Polynomials

As mentioned earlier, the first output bit of Trivium can be written as a boolean function of 80 key and 80 IV variables. Since, it is infeasible to handle 160-variable boolean functions, we have used the following strategy to search for unbalanced polynomials.

1. Fix n to be an integer which is at most 30.
2. Out of the 80 key variables, choose n key variables.
3. Set the remaining $80 - n$ key variables and 80 IV variables to random binary values. This defines the first output bit to be a function f of the n key variables.
4. Use either Method-1 or Method-2 to obtain both the truth table representation and the algebraic normal form of the first output bit.
5. Determine whether f is AU_α, d-AU_α, U_α or $\mathsf{U}_{l,\alpha}$. For all the test except $\mathsf{U}_{l,\alpha}$ we have used 6 values $\alpha_1, \ldots, \alpha_6$ with $\alpha_i = 1 - 1/2^{i+1}$ to conduct the tests. These values roughly correspond to 75%, 87.5%, 93.75%, 96.88%, 98.44% and 99.22% probabilities respectively. For $\mathsf{U}_{l,\alpha}$ the value of α was set at 99.5%.

Note that the above method randomly searches for a function f which fails one or more of the tests. For a fixed n, Steps 2 and 3 above perform the task of selecting an f; Step 4 performs the task of generating the ANF of f; and finally Step 5 performs the test on f. If f fails one or more of the tests, then this f is reported.

The tests for different values of α are not independent. For $i > j$, $\alpha_i > \alpha_j$ and so, $I_{\alpha_i} \supset I_{\alpha_j}$. As a consequence, if a function f is AU_{α_i} then it is also AU_{α_j}. Similar comments hold for d-AU_α and U_α.

6.1 Experimental Results

The experiments were conducted by taking values of $n = 10, 20$ and 30. Table 1 gives some polynomials for $n = 10, 20, 30$, which are U_{α_6}, i.e., these polynomials are unbalanced at level α_6. In the Table the column "Key Variables" indicate the key bit positions that were treated as variables. The columns "Key Constant" and "IV Constant" gives the values of $80 - n$ and 80 bits of the key and IV bits which were set to constant values.

Consider once more what it means for a polynomial to be unbalanced at level α_6. With probability α_6, i.e., with about 99% probability, the weight of a uniform random function will be in the interval I_{α_6}. Here we report examples of f whose weight lies outside the interval I_{α_6}. This indicates significant unbalancedness.

Table 1: Table showing list of polynomials which are U_{α_6}. The values given in the table are for $n = 10, 20, 30$ and 1152 key initialization rounds of TRIVIUM.

n	Key Variables	Key Constant	IV Constant	Number of 0's
10	1, 4, 22, 38, 42, 44, 53, 56, 61, 78	OX452D5AA716418A9CC	OXBC925DE125682B159CB4	465
		OX1476803AD7850AD36	OXA1D62667224E6CF221CF	465
		OX31D5EC5914E3D922F	OXE24571405777B5521A	555
		OX54CD8D3B53FC0A114	OXD4702BB150946D98D944	556
		OX238009F2E69728CB8	OX68131089DB607D1981F1	556
		OX53DB1C63D36BB4FD2	OXCF5050997F8601AB88EF	558
		OX42F216A6B2AFCEC17	OX30E66D573F151F784B58	560
		OX17485DC470A73061E	OXD54A1D5A59055062EFB6	571
20	15, 16, 20, 27, 31, 37, 41, 45, 58, 73	OX27F50AF693342B6F9	OX706CCD7801037A0A49	437
		OX3625E972822DB6A	OXB2D91DF4E87047E9B8C6	522657
	0, 1, 9, 10, 14, 19, 27, 29, 41, 42, 52, 55, 62, 64, 68, 69, 71, 75, 78, 79	OX80F5C4876AADE17	OXA380363693475CFCCEB	522768
		OXB7521EE35C15C4B	OX309D70CFFD406A96299A	522860
		OXBCEFBB60D3A6BAF	OXB0EC6893275307067F03	522862
		OXCD8AC4B29BEE0B1	OX1DFF5B9FFE4363C2F1A3	522902
30	1, 4, 7, 9, 10, 12, 13, 14, 15, 21, 25, 27, 30, 31, 32, 33, 34, 44, 52, 54, 55, 56, 58, 59, 62, 66, 69, 70, 74, 79	OX290C10B0294D2	OX586A33527C2928DDE2C6	536920658
	7, 15, 20, 21, 22, 26, 29, 30, 32, 33, 34, 41, 42, 49, 52, 54, 55, 56, 57, 59, 60, 63, 64, 65, 66, 72, 75, 76, 78, 79	OX1FD41217D312F	OXC8C051B0D49C69D1A7DD	536822130
		OX12C5E491E4B6F	OX99E4748853D60D6617EC	536920867

Table 2, gives some polynomials for $n = 10, 20, 30$, which are AU_{α_6}, i.e., these polynomials are algebraically unbalanced at level α_6. Further, the entries d in the column "Monomial Degrees" indicate that the corresponding function is also d-AU_{α_6}. The entry "None" in the column "Monomial Degrees" means that the corresponding function is not d-AU_{α_6} for any value of $d = 2, 3, \ldots, n-2$. Again we note that the reported functions show algebraic unbalancedness at a level corresponding to 99.22% probability which indicates a significant deviation.

Table 2: Table showing list of polynomials which are AU_{α_6}. Some of them are also d-AU_{α_6}. The values given in the table are for $n = 10, 20, 30$ and 1152 key initialization rounds of TRIVIUM.

n	Key Variables	Key Constant	IV Constant	Monomial Degrees
10	1, 4, 22, 38, 42, 44, 53, 56, 61, 78	OX37FE4B0255D1D295C	OXD70079FAE0F0308EC206	6, 8
		OX457B6B0466DE7552E	OXD167CC3093E7E699466	None
		OX0484EB9A3E80085D	OX9B10785F6BF67CA8D5CB	None
		OX243E3DFA82D00EE44	OXB4526FDF61F96D7FCAE3	None
	15, 16, 20, 27, 31, 37, 41, 45, 58, 73	OX5EE252240CE406D5	OX3F0E2249DE7C031CF797	None
20	0, 1, 9, 10, 14, 19, 27, 29, 41, 42, 52, 55, 62, 64, 68, 69, 71, 75, 78, 79	OX56B4A18579E0D3E	OXAC576EF0BDDE67E72619	None
		OXFC12A46241151AD	OX10E6744E590F46973ADD	13
		OXADC520A5DA98587	OX77EC7B17675B6489CAD8	None
		OXC6AFA4B133A47F7	OX61207A01BCC272B683F9	None
		OX43ED55256B3CFF5	OX822E158DE22B7390747F	None
		OXAA1BE875BC0B948	OXE49D3F5E9DF3726567A	10
		OX44B684623514BE0	OX9CB0767A4B911C07655B	13
		OXF9BB1A903D2B55A	OXBEFF617BF05E74ED8172	11
30	7, 15, 20, 21, 22, 26, 29, 30, 32, 33, 34, 41, 42, 49, 52, 54, 55, 56, 57, 59, 60, 63, 64, 65, 66, 72, 75, 76, 78, 79	OXE65F1294C96A	OXEB482AFBDFE04F8DAD56	14
		OX128D80C2688E3	OX3CF5643BE9AD30EAC0C8	None
		OX199D831A8D833	OX9F7651D0129823F00C61	None
		OX1DBD945A6AD33	OXDB855A93A2834AC2FE5C	15

Experiments to test $U_{l,\alpha}$ were done by taking values of $n = 20, 30$ and $l = 8$. Tests were done by setting $\alpha = 0.995$. Table 3 gives examples of polynomials which are $\mathsf{U}_{8,0.995}$ for $n = 20, 30$. For $n = 20$, we found 4 such polynomials whereas for $n = 30$ we found 6 polynomials.

Table 3: Table showing $\mathsf{U}_{8,0.995}$ polynomials for $n = 20, 30$.

n	Key Variables	Key Constant	IV Constant
20	0, 4, 10, 11, 18, 19, 20, 21, 29, 30, 32, 35, 38, 41, 42, 43, 45, 56, 66, 69	OXE83EDFD172DA59E	OXD6985433DD11269B7EEC
		OX8976F5F031C8922	OX7F8322315CFB6675E72C
		OX671FA8E37FA1559	OX52CCAD8EF5C7C69766A
		OX25FB47658CE713C	OX73D27D4741280A814760
30	0, 2, 3, 4, 7, 8, 12, 19, 23, 26, 30, 33, 35, 37, 38, 39, 42, 43, 44, 49, 54, 58, 60, 62, 63, 64, 65, 70, 72, 76	OX3646D112845B2	OXB1E95646DCFA6FF10729
		OX259294BDB83A1	OX6028CA379F720ABC080
		OX11C4515398DDF	OXCEB11DDCCDCE6CD72BC1
		OX188CF40F48433	OXE2F81539EA2F476236B3
	0, 1, 5, 6, 7, 12, 13, 14, 21, 24, 36, 37, 40, 43, 47, 52, 55, 56, 58, 61, 63, 64, 65, 68, 72, 73, 75, 77, 78, 79	OX1067524FF3553	OXD91F545A23C53ADC5796
		OX2588D2C38E8BF	OX388A1E1866F8247F8D51

We note that our experiments did not find any polynomial which simultaneously fail the tests for balancedness, total number of monomials and monomials of certain degrees at level α_6. On the other hand, as we go down from level α_6 to level α_1, the experiments find more and more examples of polynomials simultaneously failing the tests for balancedness, total number of monomials and monomials of certain degree. Some examples are noted below and the details are given in the appendix.

1. Table 5 (Appendix A) gives a few polynomials which simultaneously fails the test for balancedness and the test for the total number of monomials at level α_4. In addition, the table also gives the monomial degrees for which the test fails. We found 3 polynomials for $n = 10$ and 2 polynomials for $n = 30$ which had failed the test. However, we did not find any example for $n = 20$.

2. Tables 4, 5, 6, 7, 8 (Appendix A) give examples of polynomials which simultaneously fail the three tests at levels $\alpha_5, \alpha_4, \alpha_3, \alpha_2$ and α_1, respectively. In case of $\alpha = \alpha_5$, we can see from Table 4 that corresponding to $n = 10$ we have only two polynomials which failed the tests, whereas we could not find any such examples for $n = 20, 30$. However, when the value of α was relaxed to α_1, we found 45, 61 and 28 polynomials for $n = 10, 20$ and 30, respectively. The tables also show a steady increase in the number of monomials of a particular degree failing the test as α decreases.

7 Conclusion

In this paper, we have reported results of experiments conducted on the boolean function representing the first output bit of full TRIVIUM. These results show that by suitably restricting some of the input variables to constant values, it is possible to obtain polynomials which deviate from a uniform random polynomial in a statistically quantifiable manner. Our results may be considered as showing some kind of 'non-randomness' in full TRIVIUM. This is to be contrasted with the experimental evidence of 'non-randomness' after 885 rounds reported in [ADMS09] using the complicated machinery of cube testers. We note on the one hand, that our results do not indicate any weakness in TRIVIUM, and on the other hand, that similar tests can be carried out on other ciphers.

At this point, we do not have any theoretical explanations for the experimentally obtained observations. Looking for such observations can form the task of future research.

References

[ADMS09] Jean-Philippe Aumasson, Itai Dinur, Willi Meier, and Adi Shamir. Cube Testers and Key Recovery Attacks on Reduced-Round MD6 and Trivium. In *Fast Software Encryption*. Springer-Varlag, 2009.

[DCP] Christophe De Cannière and Bart Preneel. Trivium-specifications. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/030 (2005).

[DS09] Itai Dinur and Adi Shamir. Cube Attacks on Tweakable Black Box Polynomials. *Advances in Cryptology-EUROCRYPT 2009*, pages 278–299, 2009.

[EJT07] Håkan Englund, Thomas Johansson, and Mettem Sönmez Turan. A Framework for Chosen IV Statistical Analysis of Stream Ciphers. In *INDOCRYPT*, pages 268–281, 2007.

[Fil02] Eric Filiol. A New Statistical Testing for Symmetric Ciphers and Hash Functions. In *Information and Communications Security*, pages 342–353. Springer, 2002.

[FKM08] Simon Fischer, Shahram Khazaei, and Willi Meier. Chosen IV Statistical Analysis for Key Recovery Attacks on Stream Ciphers. In *AFRICACRYPT*, pages 236–245, 2008.

[FV13] Pierre-Alain Fouque and Thomas Vannet. Improving Key Recovery to 784 and 799 rounds of Trivium using Optimized Cube Attacks. In *Fast Software Encryption*. Springer, 2013.

[Jou09] Antoine Joux. *Algorithmic cryptanalysis*. CRC Press, 2009.

[O'N07] Sean O'Neil. Algebraic Structure Defectoscopy. In *Special ECRYPT Workshop–Tools for Cryptanalysis*, 2007.

[Rad06] Håvard Raddum. Cryptanalytic Results on Trivium. Technical Report 2006/039, eSTREAM, ECRYPT Stream Cipher Project, Report, http://www.ecrypt.eu.org/stream/papersdir/2006/039.ps, 2006.

[Saa06] Markku-Juhani O Saarinen. Chosen-IV Statistical Attacks on eSTREAM Stream Ciphers. *Proc. Stream Ciphers Revisited SASC*, 2006.

[Sam13] Samajder, Subhabrata and Sarkar, Palash. Fast Multiplication of the Algebraic Normal Forms of Two Boolean Functions. In Lilya Budaghyan, Tor Helleseth and Matthew G. Parker, editor, *WCC 2013*, pages 373–385, 2013. (pdf available at `http://www.selmer.uib.no/WCC2013/pdfs/Samajder.pdf`).

[TK07] M Sönmez Turan and Orhun Kara. Linear Approximations for 2-round TRIVIUM. In *Proc. First International Conference on Security of Information and Networks (SIN 2007)*, pages 96–105, 2007.

[Vie07] M. Vielhaber. Breaking One.Fivium By AIDA : An Algebraic IV Differential Attack. Technical Report 2007/413, Cryptology ePrint Archive, Report, http://eprint.iacr.org/2007/413, 2007.

A Tables

Table 4: Table showing list of polynomials which are both AU_{α_5} and U_{α_5}. Some of them are also d-AU_{α_5}. The values given in the table are for $n = 10, 20, 30$ and 1152 key initialization rounds of TRIVIUM.

n	Key Variables	Key Constant	IV Constant	Monomial Degrees
10	1, 4, 22, 38, 42, 44, 53, 56, 61, 78	OX37BDD3EAD0BAFABC0	OXDB565D9DB98F4E3389C5	None
		OX47E214EB5727E04C9	OXD34F684B1055DAECE93	7

Table 5: Table showing list of polynomials which are both AU_{α_4} and U_{α_4}. Some of them are also d-AU_{α_4}. The values given in the table are for $n = 10, 20, 30$ and 1152 key initialization rounds of TRIVIUM.

n	Key Variables	Key Constant	IV Constant	Monomial Degrees
10	1, 4, 22, 38, 42, 44, 53, 56, 61, 78	OX37BDD3EAD0BAFABC0	OXDB565D9DB98F4E3389C5	None
		OX47E214EB5727E04C9	OXD34F684B1055DAECE93	7
		OX4547D85442C8D68CF	OXD08829A188F6241E7C2D	5, 8
30	7, 15, 20, 21, 22, 26, 29, 30, 32, 33, 34, 41, 42, 49, 52, 54, 55, 56, 57, 59, 60, 63, 64, 65, 66, 72, 75, 76, 78, 79	OX11D3963CFE658	OXE9F618EC66862A0DEB4E	12
		OX191766116C74F	OX1E8B7D71045E0F56A4EA	4, 24

Table 6: Table showing list of polynomials which are both AU_{α_3} and U_{α_3}. Some of them are also d-AU_{α_3}. The values given in the table are for $n = 10, 20, 30$ and 1152 key initialization rounds of TRIVIUM.

n	Key Variables	Key Constant	IV Constant	Monomial Degrees
10	1, 4, 22, 38, 42, 44, 53, 56, 61, 78	OX37BDD3EAD0BAFABC0	OXDB565D9DB98F4E3389C5	None
		OX47E214EB5727E04C9	OXD34F684B1055DAECE93	2, 7
		OX4547D85442C8D68CF	OXD08829A188F6241E7C2D	5, 8
20	0, 1, 9, 10, 14, 19, 27, 29, 41, 42, 52, 55, 62, 64, 68, 69, 71, 75, 78, 79	OX6725535534737CA	OXDDAE21B901422A1643A	None
		OXDDAE21B901422A1643A	OXF51C70E932C18D17D41	None
30	7, 15, 20, 21, 22, 26, 29, 30, 32, 33, 34, 41, 42, 49, 52, 54, 55, 56, 57, 59, 60, 63, 64, 65, 66, 72, 75, 76, 78, 79	OX11D3963CFE658	OXE9F618EC66862A0DEB4E	12
		OX3C25D092EFEF9	OXEE0C5AA49BA676F04E05	14, 15
		OX613242AFA99E	OX74996C308B57426EC1FF	4, 13
		OX191766116C74F	OX1E8B7D71045E0F56A4EA	4, 24

Table 7: Table showing list of polynomials which are both AU_{α_2} and U_{α_2}. Some of them are also d-AU_{α_2}. The values given in the table are for $n = 10, 20, 30$ and 1152 key initialization rounds of TRIVIUM.

n	Key Variables	Key Constant	IV Constant	Monomial Degrees
10	1, 4, 22, 38, 42, 44, 53, 56, 61, 78	OX37BDD3EAD0BAFABC0	OXDB565D9DB98F4E3389C5	3, 4
		OX47E214EB5727E04C9	OXD34F684B1055DAECE93	2, 7
		OX4547D85442C8D68CF	OXD08829A188F6241E7C2D	4, 5, 8
		OX22C82FA5C4FF1FFA7	OX8E6C7CCC6DA42DE02582	6
		OX46F73734324A8C3CF	OXED6F602BFE6161C4B002	5, 7, 8
		OX548E1A39B23F3483B	OX2C2B1447188F2DF15053	6, 8
		OX52A20EB6B6861FD2B	OX69EF224FC6FB72AC6C37	2, 3
		OX243E3DFA82D00EE44	OXB4526FDF61F96D7FCAE3	5, 6
		OX21FEF73EB0DC5739A	OX7598278A31B96B6E06F5	3
		OX36B1281D43A9240B3	OXE58A191A1E6C333C8EFD	3, 5

Continued on next page

Table 7 – *Continued from previous page*

n	Key Variables	Key Constant	IV Constant	Monomial Degrees
10	1, 4, 22, 38, 42, 44, 53, 56, 61, 78	OX1185DD59742FE8169	OX89B62A60C21C42A0E6B2	4,
		OX37FE4B0255D1D295C	OXD70079FAE0F0308EC206	6, 7, 8
20	0, 1, 9, 10, 14, 19, 27, 29, 41, 42, 52, 55, 62, 64, 68, 69, 71, 75, 78, 79	OX6725535534737CA	OXDDAE21B901422A1643A	8, 12, 15, 17
		OXC481FD7BC1F523	OXC37057969DFB005C79DC	5, 6, 8, 9
		OXDF7305B0CFDB228	OXC9C17DF5198908297669	8
		OX3480FFC0AD084D6	OXF51C70E932C18D17D41	9, 15
		OX80A26F93FFE786E	OX49025F652E977970AAA3	3, 5, 6, 9
		OX99997304FBA97AB	OX91A0123D835369D66539	3, 10, 11
		OX1438B4C6E410610	OXEC881E225AE17BE12D06	6, 16
		OX2B8619E6B23FD69	OXF24C75F66F5957352674	None
		OX2E93E577A837AAC	OXF50C2B06C5B100F1D712	2, 14
		OXFDAFFE872B1ECA6	OX63F0791A5BD92EA49167	2, 7, 10
		OX1E15EFE0723A1A0	OXAFF45320480C32FE05AD	11, 12, 13, 14, 17
		OX94252897FEBA	OX40B53BED60BA2A4EF7BD	7
		OX5A4644E0DCF37F1	OXAFE71BE0360E0C918B9C	3, 9, 13
		OXAA07D7C6F262C91	OX4F821468B1891D2AD371	2, 3, 4, 12
		OX748AA0B4C4431F6	OX6BB3415153E252D74428	9
		OX82641E96DDFE210	OXA045545ADF754FE49440	4, 16, 17
30	7, 15, 20, 21, 22, 26, 29, 30, 32, 33, 34, 41, 42, 49, 52, 54, 55, 56, 57, 59, 60, 63, 64, 65, 66, 72, 75, 76, 78, 79	OX11D3963CFE658	OXE9F618EC66862A0DEB4E	12, 15, 18
		OX3C25D092EFEF9	OXEE0C5AA49BA676F04E05	14, 15, 23
		OXCDCA70B4903D	OX28094F93A84519B6030	2, 12, 21, 26
		OX613242AFA99E	OX74996C308B57426EC1FF	4, 13, 15, 19, 24, 26, 27
		OX191766116C74F	OX1E8B7D71045E0F56A4EA	4, 18, 24, 25

Table 8: Table showing list of polynomials which are both AU_{α_1} and U_{α_1}. Some of them are also d-AU_{α_1}. The values given in the table are for $n = 10, 20, 30$ and 1152 key initialization rounds of TRIVIUM.

n	Key Variables	Key Constant	IV Constant	Monomial Degrees
10	1, 4, 22, 38, 42, 44, 53, 56, 61, 78	OX37BDD3EAD0BAFABC0	OXDB565D9DB98F4E3389C5	2, 3, 4, 5
		OX22C82FA5C4FF1FFA7	OX8E6C7CCC6DA42DE02582	6
		OX47E214EB5727E04C9	OXD34F684B1055DAECE93	2, 4, 7
		OX4547D85442C8D68CF	OXD08829A188F6241E7C2D	4, 5, 8
		OX46F73734324A8C3CF	OXED6F602BFE6161C4B002	5, 7, 8
		OX548E1A39B23F3483B	OX2C2B1447188F2DF15053	6, 7, 8
		OX25999FA70096CE14A	OX534E7B0E0099371CEC2B	7
		OX32C3B8564711127E2	OXC3B65FA580064682A886	2, 5, 6, 7, 8
		OX36700C6F525F4A15E	OX76C0CA72E4037279E52	2, 5, 7
		OX1104F75E2114D99C6	OX99545A9EDB9B664CF25E	5, 6

Continued on next page

Table 8 – *Continued from previous page*

n	Key Variables	Key Constant	IV Constant	Monomial Degrees
10	1, 4, 22, 38, 42, 44, 53, 56, 61, 78	OX52A20EB6B6861FD2B	OX69EF224FC6FB72AC6C37	2, 3, 6
		OX243E3DFA82D00EE44	OXB4526FDF61F96D7FCAE3	5, 6, 7
		OX21FEF73EB0DC5739A	OX7598278A31B96B6E06F5	2, 3
		OX36B1281D43A9240B3	OXE58A191A1E6C333C8EFD	2, 3, 5
		OX373444186660E5FF4	OXEB195ADDBE903C2D1056	2
		OX1185DD59742FE8169	OX89B62A60C21C42A0E6B2	4
		OX13C98DD112B9E4345	OXC02E2FE44559226743EA	2, 6
		OX25964FF8044895C95	OXCFC776D1C4E100F35C85	2, 4, 5
		OX2628BA81850F8F769	OX1FCF571CE4612534B608	4, 8
		OX043DACA1A2026DBDA	OXBDC5DCD77F921AABDF6	4, 7
		OX2126745C279E5A10C	OX1248772E03E133CE0B7B	2, 7
		OX586B4E14496FAC82	OX9C2138672ABB3DDDC1F2	5
		OX368252990744C0C7	OX74DF32D819F351C27B0E	2, 3, 4
		OX532D3BE97067D5129	OX7BED67B3ABB91C35FA5D	5
		OX1382D64B413855656	OX95972A312ED14A3BF60	6
		OX52FFFC0FF6FD88EBD	OXF66572321FFA19728935	3, 7
		OX439C148480CEF8257	OX74FB1106EE59338B8704	3
		OX14B0825030BA0A96B	OX207C5A11622E7FE89689	2, 3, 7
		OX568F9EDC3FA5CFC5	OXCD8D6086AF815B848C24	3, 6, 8
		OX37A8A2D3F4AD45193	OXD49B28D3ABC66F27C37E	3, 6
		OX457B6B0466DE7552E	OXD167CC3093E7E699466	5, 6
		OX2CE451DE26F01574	OX1C1342B3181C132E7CCF	8
		OX2188425AC63CCD33F	OX90C929DD67D3678472EE	2, 4, 5, 6, 8
		OX2454FEF2819CFDFE8	OX9E71576A5F36051743D5	6, 7
		OX04D00A4F9785AC8C4	OX8DCA7A16BF435CE5940B	5
		OX37FE4B0255D1D295C	OXD70079FAE0F0308EC206	4, 6, 7, 8
		OX53CD508F74BBC7DBE	OXC37E2A7F2F8164D022BB	5, 7, 8
		OX048DAFF69510A4B0E	OX3A1A19897D5D77691BDB	5
		OX4708A09334FCAFE4F	OX4A4C60F2FECB3B1FFA4F	2, 3, 5
		OX4559C324D7A96E402	OX76093CA6A6820F188476	6
		OX11CCD131147B71B01	OX82C85493CE4525CC267A	4, 7
		OX43C5AE65F459310FF	OXE29F6C5FF9B122017D55	6
		OX074D96A9360193375	OXFA274E2AFE2E68F3ECE6	6
		OX073A6C0377AF88B83	OX6351165578DB3B77F014	3, 5
		OX11D7C2096469B8D59	OX85164E66AB350C5BCD85	2
20	0, 1, 9, 10, 14, 19, 27, 29, 41, 42, 52, 55, 62, 64, 68, 69, 71, 75, 78, 79	OXCD8AC4B29BEE0B1	OX1DFF5B9FFE4363C2F1A3	2, 5, 8, 13, 17
		OX6725535534737CA	OXDDAE21B901422A1643A	8, 12, 15, 16, 17
		OXC481FD7BC1F523	OXC37057969DFB005C79DC	5, 6, 8, 9, 14
		OXDF7305B0CFDB228	OXC9C17DF5198908297669	4, 6, 8, 17, 18
		OXE8EF47A657A1A10	OX47B07462C84600BBD4C3	8, 12, 17
		OX3480FFC0AD084D6	OXF51C70E932C18D17D41	4, 6, 9, 14, 15

Continued on next page

Table 8 – *Continued from previous page*

n	Key Variables	Key Constant	IV Constant	Monomial Degrees
20	0, 1, 9, 10, 14, 19, 27, 29, 41, 42, 52, 55, 62, 64, 68, 69, 71, 75, 78, 79	OXF79DC891384AFF0	OXDDAF2C635F1725E5722C	2, 3, 5, 11, 15, 17
		OX63CBCE13DFA0AFF	OXC314A7271C748FEB77B	3, 11, 15
		OX80A26F93FFE786E	OX49025F652E977970AAA3	3, 4, 5, 6, 9, 14
		OX99997304FBA97AB	OX91A0123D835369D66539	3, 10, 11
		OX1438B4C6E410610	OXEC881E225AE17BE12D06	2, 6, 7, 9, 16
		OX2B8619E6B23FD69	OXF24C75F66F5957352674	10, 13
		OX3907E66406C1230	OXCC4F643389D174E09308	9, 11, 12, 17
		OX8DC9AD07E1DD5B7	OX6F925A43AABE3589016F	2, 9, 18
		OX1C2904C43FF6577	OXE73A4739E4C117D70E8E	7, 11, 12, 13, 16, 17
		OX2E93E577A837AAC	OXF50C2B06C5B100F1D712	2, 3, 5, 13, 14
		OXFDAFFE872B1ECA6	OX63F0791A5BD92EA49167	2, 5, 7, 10, 15
		OX1B3537B58870F55	OXA33A411FC4173976088F	2, 3, 5, 8, 9, 10, 14, 15, 18
		OX1E15EFE0723A1A0	OXAFF45320480C32FE05AD	2, 6, 11, 12, 13, 14, 15, 17
		OX137AD462B48819D	OXECFE45F642EA682545D6	7, 10, 17
		OX94252897FEBA	OX40B53BED60BA2A4EF7BD	7, 10, 11
		OX2EEDD3E63910450	OX33953B8FBB3E5DB89240	6, 7, 17
		OX7E51F8E40591536	OXB9C6CEF795453064BE6	9, 10, 14, 15, 17
		OXD241FF6460C6208	OX5C725E6A96F9353ECCE8	2, 5, 12, 14, 18
		OX895517354CC7691	OX9099444B625C731D4A9F	4, 13
		OX18093FF2EA21B89	OX3627A654C272E098ABA	8, 15, 17
		OX69219DE1A75C6B5	OX59FB6A44546208BBA473	5, 8, 9, 12, 13, 15
		OX8CA5215750A80AE	OXBBFE302CAEC030A95C66	2, 7, 11, 18
		OX48F6A050FA29FD4	OXF13127F0DE9243B0AA33	10, 12
		OXA869C2C7AE9EEDD	OXFC4D4C6DE13E62F8530E	2, 6, 7
		OX54D805626D001BA	OXCC3F38A3216216E4DD16	10, 15
		OXFF859A14E90D2D3	OX2AC86B44BBEF20B3A8EB	3
		OX92E4EE44F4AA9B0	OXB2E35370E80D3FA438FE	5, 7, 8, 9, 10, 11, 12, 14
		OXC906517717370E8	OX407C62FEAAD57DE22435	6, 8, 9, 15
		OX81ABA2F10E414F1	OXD8DC3E5436B30D553DFF	13, 15
		OX5E2587E52504105	OX492770685C260EB94076	3, 4, 5, 7, 9, 11, 14
		OX66BE1B647D74852	OXE2B64E025081086192F3	2, 4, 5, 7, 8, 11, 13
		OXC987FF2679818EA	OXECBC1922F4E82FE0E298	3, 9, 14, 15, 16
		OX399995668AA5766	OXADDE1B907E417C75C073	9, 18
		OXA01F4042A39D2AF	OXB42E343741AE2885A4B8	8, 10, 11, 15, 16
		OXEB000173340180D	OX9D4811B44D956C1C4122	9, 10
		OX6FEDB601C5D0F7	OXACF51C7A59AC427CBA18	2, 4, 13, 14, 16, 17, 18
		OX349068A2D3BE11B	OXF51A7B67A45C5173EDB0	2, 6, 9, 10, 11, 12
		OX9904566610C1359	OX5C921B727602478B4F1F	2, 11, 13, 18
		OX631BFA24A283F98	OX8BEE5BDF986177DEFCB7	2, 4, 7, 8, 14, 15, 17
		OX41B6D6E060B45	OXF4301269A0A373516F83	4, 6, 7, 9, 15
		OX6AE3E77490E6D0B	OXED4B6FCC7E5B1FFAA681	2, 6, 12, 15, 17

Continued on next page

Table 8 – *Continued from previous page*

n	Key Variables	Key Constant	IV Constant	Monomial Degrees
20	0, 1, 9, 10, 14, 19, 27, 29, 41, 42, 52, 55, 62, 64, 68, 69, 71, 75, 78, 79	OXF9BB1A903D2B55A	OXBEFF617BF05E74ED8172	6, 9, 11, 18
		OXADB103911781696	OX78001622345E7535AF89	2, 3, 10, 12, 13, 14
		OXD4BAF7074479E09	OX3CB7239F46CD1A18C135	5, 6, 13, 16, 18
		OX31C341E7D2823C0	OX44E6375DA9C323B5EFCF	15, 16, 17
		OX5A4644E0DCF37F1	OXAFE71BE0360E0C918B9C	3, 9, 13
		OXBCF4D6769BCEFB	OX6D344C7B4AC745BC07FC	10, 11, 12, 18
		OXD989E1257E60721	OX85F9933760D63491E6D	2, 8, 13, 18
		OXFE4EC2B5DF70C87	OX7E0D7707ABF24E5811D8	4, 7, 8, 10, 11,14, 17
		OX6BB3225B97D099	OX5BD374B78F0F10E1F552	6
		OX3666DCC1529A055	OX998A7DC1F1F94C9CAFB0	3, 9
		OXAA07D7C6F262C91	OX4F821468B1891D2AD371	2, 3, 4, 5, 7, 12
		OXCDB760C050E0196	OX2FDE16E5A5517466E581	10, 16, 18
		OX748AA0B4C4431F6	OX6BB3415153E252D74428	9, 17
		OX82641E96DDFE210	OXA045545ADF754FE49440	4, 12, 13, 16, 17
30	7, 15, 20, 21, 22, 26, 29, 30, 32, 33, 34, 41, 42, 49, 52, 54, 55, 56, 57, 59, 60, 63, 64, 65, 66, 72, 75, 76, 78, 79	OX2A3B12E5B3AAA	OX9107625D556D1E48A3B5	3, 7, 13, 16, 17, 18, 20, 23, 25
		OX11D3963CFE658	OXE9F618EC66862A0DEB4E	9, 12, 13, 14, 15, 18
		OXF8534CF8A0C4	OXCEDD5CCE04F12DF6FA42	7, 9, 10, 14, 15, 18
		OXE0D05859F75D	OXDE0D17F6F4A032F4345A	2, 4, 5, 8, 13, 15, 16, 18
		OX1767659F97A78	OXDB0E189FAA7523B7F38C	3, 5, 11, 12, 13, 16, 18, 19, 24
		OX3C25D092EFEF9	OXEE0C5AA49BA676F04E05	4, 8, 10, 14, 15, 23, 24
		OX358F63BC9862E	OX2B8A12AF7C7513BFB545	4, 5, 8, 10, 14, 15, 16, 23, 25
		OXCDCA70B4903D	OX28094F93A84519B6030	2, 6, 12, 16, 21, 24, 26
		OX186E1140CAE7A	OXE5893222F3CF2AD91C84	3, 10, 17, 21, 24
		OXF5633C0E0766	OX3A2161ED1A9A6C545C99	6, 8, 14, 16, 17, 18, 23, 25
		OX3EF1C76CC3786	OX91441019D7A5F99C0E2	5, 8, 15, 16, 19, 22, 26, 27
		OX1E3305EE66BF7	OX84052206580263DB7246	3, 9, 12, 14, 23
		OX2BFEF0DB6F4F7	OX21D64C13071A1E0AA4DF	10, 14, 24, 28
		OX22BE07DCB8255	OX14B48826D4E3EE8AA4A	10, 11, 13, 14, 17, 22, 24, 25
		OX2C53E5CA904F8	OX4CF8318FB91A7BD1C2D0	4, 5, 8, 14, 17, 18, 19, 22, 24, 27
		OX93726691E2D0	OX2C4C389C765606937AF4	6, 10, 15, 18, 19, 13, 16
		OX3ED1D244BD2B1	OXBB5B758E8FB029E57666	2, 6, 7, 8, 15, 23, 25
		OX2DF6C79AE5433	OX920D16223BEE4EB5822E	3, 4, 8, 9, 12, 13, 20
		OX378C02C3FDF2B	OX77681A2286592408308D	3, 10, 11, 13, 15, 22, 23, 26, 28
		OX3B97E24D24147	OX694F280DCB2B108F1385	2, 9, 11, 12, 24, 28
		OXC7294829B50A	OXF303799BF930108F4B0F	4, 8, 11, 15, 16
		OX378763FE6C96	OX2FBC5FB87C8125734B6E	2, 3, 6, 7, 8, 12, 16, 18, 23, 25, 28
		OX3BECF5CC75818	OXC3B33304C9FD300B28F3	5, 12, 14, 17, 19, 21, 22, 23, 24
		OX3049A0E2FB512	OX50986952B94273E8F099	9, 13, 16, 24, 26
		OX313C07B28B127	OX30005763E511714B24C0	10, 11, 17, 18, 25, 26
		OX613242AFA99E	OX74996C308B57426EC1FF	4, 5, 8, 11, 12, 13, 15, 17, 19, 21, 22, 24, 25, 26, 27

Continued on next page

Table 8 – *Continued from previous page*

n	**Key Variables**	**Key Constant**	**IV Constant**	**Monomial Degrees**
30	7, 15, 20, 21, 22, 26, 29, 30, 32, 33, 34, 41, 42, 49, 52, 54, 55, 56, 57, 59, 60, 63, 64, 65, 66, 72, 75, 76, 78, 79	OX5AE80FC1F4DB	OXD5776A122F7F7B0049B1	3, 10, 15, 21, 23, 24
		OX191766116C74F	OX1E8B7D71045E0F56A4EA	4, 13, 17, 18, 24, 25