A Framework for Chosen IV Statistical Analysis of Stream Ciphers

Håkan Englund[1], Thomas Johansson[1] and Meltem Sönmez Turan[2]

[1]Dept. of Electrical and Information Technology, Lund University, Sweden
[2]Institute of Applied Mathematics, METU, Turkey

Abstract. Saarinen recently proposed a chosen IV statistical attack, called the *d*-monomial test, and used it to find weaknesses in several proposed stream ciphers. In this paper we generalize this idea and propose a framework for chosen IV statistical attacks using a polynomial description. We propose a few new statistical attacks, apply them on some existing stream cipher proposals, and give some conclusions regarding the strength of their IV initialization. In particular, we experimentally detected statistical weaknesses in some state bits of Grain-128 with full IV initialization as well as in the keystream of Trivium using an initialization reduced to 736 rounds from 1152 rounds. We also propose some stronger alternative initialization schemes with respect to these statistical attacks.

1 Introduction

Synchronous stream ciphers are an important part of symmetric cryptosystems and they are suitable for applications where high speed and low delay are required. As examples, the stream cipher family A5 is used in the GSM standard and the cipher E0 is used to supply privacy in Bluetooth applications. In most applications, the transmission of ciphertext is assumed to be done over a noisy channel where the synchronization between sender and receiver can be lost and resynchronization is necessary.

Depending on the protocol, different resynchronization mechanisms can be used. In most applications the message is divided into frames and each frame is encrypted using different publicly known Initialization Vectors (IVs) and the same secret key. In such systems, the ciphers should be designed to resist attacks that use many short keystreams generated by random or chosen IVs.

In [1], an attack on nonlinear filter generators with linear resynchronization and filter function with few inputs is presented and this attack is extended to the case where the filter function is unknown in [2]. More extensions of the resynchronization attack is available in [3].

To avoid such attacks, the initialization of stream ciphers in which the internal state variables are determined using the secret key and the public IV should be designed carefully. In most ciphers, firstly the key and IV are loaded into the state variables, then a next state function is applied to the internal state iteratively for a number of times without producing any output. The number of iterations play an important role on both security and the efficiency of the cipher. It should be chosen so that each key and IV bit affect each initial state bit in a complex way. On the other hand, using a large number of iterations is inefficient and may hinder the speed for applications requiring frequent resynchronizations.

In [4], tests were introduced to evaluate the statistical properties of symmetric ciphers using the number of the monomials in the Boolean functions that simulate the action of a given cipher. In [5], Saarinen recently proposed to extend these ideas to a chosen IV statistical attack, called the d-monomial test, and used it to find weaknesses in several proposed stream ciphers.

In this paper we generalize this idea and propose a framework for chosen IV statistical attacks using a polynomial description. The basic idea is to select a subset of IV bits as variables. Assuming all other IV values as well as the key being fixed, we can write a keystream symbol as a Boolean function of the selected IV variables. By running through all possible values of these bits and creating a keystream output for each of them, we create the truth table of this Boolean function. We now hope that this Boolean function has some statistical weaknesses that can be detected. We describe the d-monomial test in this framework, and then we propose two new tests, called the monomial distribution test and the maximal degree monomial test.

We then apply them on some existing stream cipher proposals, and give some conclusions regarding the strength of their IV initialization. In particular, we experimentally detected statistical weaknesses in the keystream of Grain-128 with IV initialization reduced to 192 rounds as well as in the keystream of Trivium using an initialization reduced to 736 rounds. Furthermore, we repeat our experiments to study the statistical properties of internal state bits. Here we could detected statistical weaknesses in some state bits of Grain-128 with full IV initialization. In the context, we also propose alternative initial loadings for some of the ciphers so that the diffusion is satisfied in fewer rounds.

The paper is organized as follows. In the next section, some background information about hypothesis testing and Boolean functions are given. In Section 3, the suggested framework for chosen IV statistical attacks is presented, and in Section 4 some results are presented for reduced round initializations of the ciphers Grain [6], Trivium [7] and Decim [8]. Finally we conclude the paper in Section 5.

2 Preliminaries

2.1 Hypothesis Testing

Assume we have independently and identically distributed (i.i.d.) random variables X_i, the sum is a new random variable, denoted by Y, i.e., $Y = \sum_{i=0}^{n} X_i$. According to the central limit theorem Y is approximately normally distributed if n is large. Let y denote an observation from Y, and assume that we have r observations of random variables Y, i.e., $y_0, \ldots, y_{r-1}$, then the chi-square statistic is

$$\chi^2 = \sum_{k=0}^{r-1} \frac{\big(y_k - E(Y)\big)^2}{E(Y)} \xrightarrow{d} \chi_r^2$$

where $\xrightarrow{d}$ means convergence in distribution, and r is called the degrees of freedom (i.e., number of independent pieces of information).

Our two hypothesis are

- H_0 : $z = 0$, $y_0, \ldots, y_{r-1}$ are samples from Y,
- H_1 : $z \neq 0$, $y_0, \ldots, y_{r-1}$ are not samples from Y.

For a one-sided χ^2-Goodness of fit test, the hypothesis is rejected if the test statistics χ^2 is greater than the tabulated $\chi^2(1-\alpha;\ r)$ value, for some significance level α with r degrees of freedom.

2.2 Algebraic Normal Form of a Boolean Function

Let $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ be a mapping from n binary input bits into one output bit, then f is called a Boolean function. There are many representations of a Boolean function, but in this paper we are mainly interested in the so called Algebraic Normal Form (ANF). The *ANF* is the polynomial

$$f(x_1, x_2, \ldots, x_n) = a_0 \oplus a_1 x_1 \oplus \ldots \oplus a_n x_n \oplus a_{n+1} x_1 x_2 \oplus \ldots \oplus a_{2^n-1} x_1 x_2 \ldots x_n$$

with unique a_i's in $\mathbb{F}_2$.

2.3 Computation of Algebraic Normal Form

Assume the truth table of an n-variable Boolean function is represented in a vector v of size 2^n and the ANF of the Boolean function can be calculated with complexity $O(n2^n)$ using the algorithm presented in Figure 1, which uses two auxiliary vectors t and u, both of size 2^{n-1}.

```
Compute ANF(v)

for $i = 1, \ldots, n$
    for $j = 1, \ldots, 2^{n-1}$
        $t_j = v_{2j-1}$
        $u_j = v_{2j-1} \oplus v_{2j}$
    end for
    $v = t||u$
end for
```

Fig. 1. Algorithm to compute the ANF in vector v from the truth table in v.

2.4 Properties of a Random Boolean Function

Let $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ be a Boolean function, and let the number of monomials in the ANF of f be denoted by M. If f is randomly chosen, each monomial is included with probability one half, i.e., a Bernoulli distribution. The sum of Bernoulli distributed random variables is Binomially distributed, hence $M \in$Bin$(2^n, \frac{1}{2})$, with expected value $E(M) = 2^{n-1}$. Let's denote the number of monomials of degree k by M_k, i.e., $M = \sum_{k=0}^{n} M_k$. The distribution of M_k is Bin$(\binom{n}{k}, \frac{1}{2})$ with $E(M_k) = \frac{1}{2}\binom{n}{k}$.

Let m_k be an observation from M_k, then

$$\chi^2 = \sum_{k=0}^{n} \frac{\left(m_k - \frac{1}{2}\binom{n}{k}\right)^2}{\frac{1}{2}\binom{n}{k}} \xrightarrow{d} \chi^2_{n+1}, \text{ when } \binom{n}{k} \to \infty.$$

If $\binom{n}{k}$ is large enough, methods described in Section 2.1 can be used to perform a hypothesis test to decide if the function in question has a deviant number of monomials of degree k.

3 A Framework for Chosen IV Statistical Attacks

For an additive synchronous stream cipher, let $K = (k_0, \ldots, k_{N-1})$ denote the secret key. Furthermore, let $IV = (iv_0, \ldots, iv_{M-1})$ denote the public IV value used, and, finally, let $Z = z_0, z_1, \ldots$ denote the keystream sequence. We assume that the attacker has received a number of different keystream sequences generated using different (possibly chosen) IV values.

Different tests have been introduced to evaluate statistical properties of sequences from symmetric ciphers and hash functions. The tests are usually based on taking one long keystream sequence and then applying different statistical tests, like the NIST statistical test suit used in the AES evaluation [9].

However, recently several researchers have noted the possibility to instead generate a lot of short keystream sequences, from different chosen IV values and look at the statistical properties of, say, only the first output symbol of each keystream. One such example is the observation by Shamir and Mantin that the second byte in RC4 is strongly biased [10].

Based on work in [4], Saarinen [5] recently proposed the d-monomial IV distinguisher. The behavior of the keystream is analyzed using a function of n IV bits, i.e., $z = f(iv_0, \ldots, iv_{n-1})$. All other IV and key bits are considered to be constants. In [5] among a few other tests, Saarinen suggested the d-monomial test. For a chosen parameter d (set to be a small value), the test counts the number of monomials of weight d in the ANF of f and compares it to its expected value $\frac{1}{2}\binom{n}{d}$, using the χ^2-Goodness of Fit test with one degree of freedom.

In this paper, we will instead sum the test statistics for each d and evaluate the result using $n+1$ degrees of freedom. The algorithm for the d-Monomial test is summarized in Figure 2.

The complexity of this attack is $O(n2^n)$ operations and it needs memory $O(n2^n)$. The downside of this method is that statistical deviations for lower

```
d-Monomial Test

for $iv = 1, \ldots, 2^n - 1$
    Initialize cipher with $iv$
    $v[iv]$=first keystream bit after initialization
end for
Compute ANF of vector $v$ and store result in $v$.
for $i = 1, \ldots, 2^n - 1$
    if $v[i] = 1$
        $weight$= weight of monomial $i$
        distr[$weight$] + +
end for
for $d = 0, \ldots, n$
    $\chi^2 += \frac{(\mathrm{distr}[d] - \frac{1}{2}\binom{n}{d})^2}{\frac{1}{2}\binom{n}{d}}$.
if $\chi^2 > \chi^2(1-\alpha; n+1)$
    return cipher
else
    return random
```

Fig. 2. Summary of the d-monomial test, complexity $O(n2^n)$.

and higher degree monomials are hard to detect since their numbers are few. So even if the maximal degree monomial never occurs, the test does not detect this anomaly. In the next section we will present alternative attacks that solves this problem.

3.1 A generalized approach

We suggest to use a generalized approach. Instead of analyzing just one function in ANF form, we can study the behavior of more polynomials so that monomials that are more (or less) probable than others can be detected.

Let us select n IV values, denoted $iv_0, \ldots, iv_{n-1}$, as our *variables*. The remaining IV values as well as key bits are kept constant. Using the first output symbol, $z_0 = f_1(iv_0, \ldots, iv_{n-1})$, for each choice of $iv_0, \ldots, iv_{n-1}$, the ANF of f_1 can be constructed.

The new approach is now to do the same again, but using some other choice on IV values outside the IV variables. Running through each choice of $iv_0, \ldots, iv_{n-1}$ in this case gives us a new function f_2. Continuing in this way, we derive P different Boolean functions $f_1, f_2, \ldots, f_P$ in ANF form. In some situations, it might also be possible to obtain polynomials from different keys, where the same IVs have been used.

Having P different polynomials in our possession we can now design any test that looks promising, taken over all polynomials. The d-monomial test would appear for the special case $P = 1$, and the test being counting the number of weight d monomials. We now propose in detail two different tests.

3.2 The Monomial Distribution Test

The attack scenario is similar to the d-monomial test, but instead of counting the number of monomials of a certain degree, we generate P polynomials and calculate in how many of the polynomials each monomial is present. That is, we generate P polynomials of the form (1) and count the number of occurrences of $a_i = 1,\ 0 \leq i \leq 2^n - 1$

$$f = a_0 + a_1 x_1 + \ldots + a_{n+1} x_1 x_2 + \ldots + a_{2^n - 1} x_1 x_2 \ldots x_{n-1} x_n \tag{1}$$

Denote the number of occurrences of coefficient a_i by M_{a_i}, since each monomial should be included in a function with probability 1/2, i.e., $P(a_i = 1) = 0.5,\ 0 \leq i \leq 2^n - 1$, the number of occurrences is binomially distributed with expected value $E(M_{a_i}) = P/2$ for each monomial. We will as previously perform a χ^2-Goodness of fit test with 2^n degrees of freedom, as described by Equation (2).

$$\chi^2 = \sum_{i=0}^{2^n - 1} \frac{(M_{a_i} - \frac{P}{2})^2}{\frac{P}{2}} \tag{2}$$

If the observed amount is larger than some tabulated limit $\chi^2(1 - \alpha; 2^n)$, for some significance level α, we can distinguish the cipher from a random one. The pseudo-code of the monomial distribution test is given in Figure 3.

```
Monomial Distribution Test

for $j = 1, \ldots, P$
    for $iv = 1, \ldots, 2^n - 1$
        Initialize cipher with $iv$
        $v[iv]$=first keystream bit after initialization
    end for
    Compute ANF of vector $v$ and store result in $v$.
    for $i = 1, \ldots, 2^n - 1$
        if $v[i] = 1$
            $M_{a_i} ++$
    end for
end for
for $d = 0, \ldots, 2^n - 1$
    $\chi^2 += \frac{(M_{a_d} - \frac{P}{2})^2}{\frac{P}{2}}$.
end for
if $\chi^2 > \chi^2(1 - \alpha; 2^n)$
    return cipher
else
    return random
```

Fig. 3. Summary of Monomial distribution test, complexity $O(Pn2^n)$.

This algorithm has a higher computational complexity than the d-Monomial attack, $O(Pn2^n)$, and needs the same amount of memory, $O(n2^n)$. On the other hand, if for a cipher some certain monomials are highly non-randomly distributed, the attack may be successful with less number of IV bits, i.e., smaller n, compared to the d-monomial test. Additionally, although this attack is originally proposed for the chosen IV scenario of a fixed unknown key, it is also possible to apply the test for different key values, if the same IV bits are considered.

3.3 The Maximal Degree Monomial

A completely different and very simple test is to see if the maximal degree monomial can be produced by the keystream generator. The maximal degree monomial is the product of all IV bits and can hence only occur if all the IV bits have been properly mixed. In hardware oriented stream ciphers the IV loading is usually as simple as possible to save gates, e.g., the IV bits are loaded into different memory cells. The update function is then performed a number of steps to produce proper diffusion of the bits, intuitively it will take many clockings before all IV bits meet in the same memory cell and even more clocking before they spread to all the memory cells and are mixed nonlinearly. The aim of the Maximal Degree Monomial is to check in a simple way whether the number of initial clockings are sufficient. Since the maximal degree monomial is unlikely to exist if lower degree monomials do not exist, this is our best candidate to study. Hence, the existence of the maximal degree term in ANFs is a good indication to the satisfaction of diffusion criteria, especially completeness.

According to the Reed-Muller transform the maximal degree monomial can be calculated as the XOR of all entries in the truth table. So the test is similar to the previous tests performed by initializing the cipher with all possible combinations for n IV bits, $z^{iv_0,\ldots,iv_{n-1}} = f(iv_0,\ldots,iv_{n-1})$, all other bits are considered to be constants. The existence of the maximal degree monomial can be checked by XORing the first keystream bit from each initialization, following the notation from Section 2.2, this is equivalent to determining a_{2^n-1}.

$$a_{2^n-1} = \bigoplus_{iv_0,\ldots,iv_{n-1}} z^{iv_0,\ldots,iv_{n-1}}.$$

By for example changing some other IV bit we receive a new polynomial and perform the same procedure again, this is repeated for P polynomials, if the maximal degree polynomial never occurs in any of the polynomials or if it occurs in all of the polynomials we successfully distinguish the cipher. Hence we can, with low complexity, and more importantly, almost no memory, check whether the maximal degree monomial can exist in the output from the cipher. It is possible, with the same complexity, to consider other weak monomials, the coefficient can be calculated according to the Reed-Muller transform. The complexity of the Maximal Degree Attack is $O(P2^n)$ and it only requires $O(1)$ memory. The description of the test is given in Figure 4.

```
Maximal Degree Monomial Test

for $j = 1, \ldots, P$
    $a_{2^n-1} = 0$
    for $iv = 1, \ldots, 2^n - 1$
        Initialize cipher with $iv$
        $z$ = first keystream bit after initialization
        $a_{2^n-1} = a_{2^n-1} \oplus z$
    end for
    if $a_{2^n-1} = 1$
        ones++
end for
if  ones=0 or ones=P
    return cipher
else
    return random
```

Fig. 4. Summary of Maximal Degree Test, complexity $O(P2^n)$.

3.4 Other possible tests

We have proposed two specific tests that we will use in the sequel to analyze different stream ciphers. Our framework gives us the possibility to design many other interesting tests. As an example, a monomial distribution test restricted to only monomials with very high weight could be an interesting test. Another possibility would be to examine properties of the Walsh transform of each polynomial. These tests have not been experimentally examined in this work.

4 Experimental Results

We applied the proposed tests described above on some of the Phase III eSTREAM candidates to evaluate their efficiency of initializations. We evaluated their security margin by testing reduced round versions of the ciphers. We also presented some results on the statistical properties of the internal state variables.

The significance level of the hypothesis tests is chosen to be approximately $1 - \alpha = 1 - 2^{-10}$. The tabulated results have a success rate of at least 90%. The required number of IVs, polynomials and the amount of memory needed to attack the ciphers are given in tables. Also, the results for initial state variables are presented with the percentage of weak initial state variables.

Hardware oriented stream ciphers use simple initial key and IV loading compared to software oriented ciphers. Generally, key and IV bits affect one initial state variable. Therefore, they require a large number of clockings to satisfy the diffusion of each input bit on each state bit. We repeated some of our simulations using alternative key/IV loadings in which each IV bit is assigned to more than one internal state bit and compared the results to the original settings. In the alternative loadings the hardware complexity is slightly higher, however on the other hand the cipher has more resistance to chosen IV attacks.

4.1 Grain-128

Grain-128 [6] is a hardware oriented stream cipher using a LFSR and a NFSR together with a nonlinear filter function. In the initialization of Grain, a 128 bit key is loaded into the NFSR and a 96 bit IV is loaded into the first 96 positions of the LFSR, the rest of the LFSR is filled with ones. The cipher is then clocked 256 times and for each clock the output bit is fed back into both the LFSR and the NFSR.

In Table 1, the results obtained for reduced version of Grain are given. The highest number of rounds, we succeeded to break is 192 out of the original 256 which corresponds to the 75% of the initialization phase.

Rounds	*d*-Monomial test			Monomial distr. test			Max. degree monomial		
	P	IVs	Memory	P	IVs	Memory	P	IVs	Memory
160	1	14	2^{14}	2^6	7	2^7	2^5	11	1
192	1	25	2^{25}	2^6	22	2^{22}	2^5	22	1

Table 1. Number of IV bits needed to attack the first keystream bit of Grain-128 for different number of rounds in the initialization (out of 256 rounds).

In Table 2, the results of the experiments for initial state variables are presented. The number of weak initial state variables are three times better in the maximum degree test compared to the *d*-monomial test. The statistical deviations in state bits remain even after full initialization. These weak state bits are located in the left most positions of the feedback shift registers. To remove the statistical deviations in state variables, at least 320 initial clockings are needed. It is possible that if we use larger number of IV bits, the weaknesses in state variables may also be observed from the keystream bits.

Rounds	*d*-Monomial test				Monomial distr. test				Max. degree monomial			
	P	IVs	Memory	Fraction	P	IVs	Memory	Fraction	P	IVs	Memory	Fraction
256	1	14	2^{14}	33/256	2^6	8	2^8	20/256	2^5	14	1	108/256
256	1	16	2^{16}	40/256	2^6	10	2^{10}	35/256	2^5	16	1	120/256
256	1	20	2^{20}	56/256	2^6	15	2^{15}	44/256	2^5	20	1	138/256
288	1	20	2^{20}	0/256	2^6	20	2^{20}	0/256	2^5	20	1	73/256

Table 2. Number of IV bits needed to attack the initial state variables Grain-128 for different number of rounds in the initialization (out of 256 rounds).

Alternative Key/IV Loading for Grain-128 Here we propose an alternative Key/IV loading in which only the loading of the first 96 bits of the NFSR is different from the original. Instead of directly assigning the key, we assign the modulo 2 summation of IV and the first 96 bits of the key. The proposed loading is very similar to the original and the increase in number of gates required is approximately 10-15%. In an environment where many resynchronizations are expected, one can reduce the number of initial clockings by using some more gates in the hardware implementation. In the new loading, each IV bit affects two internal state variables. We repeated our experiments using the new loading and the results are given in Table 3 and Table 4. Using alternative loading, Grain shows more resistance to the presented attacks, but still the statistical deviations in the state bits remain after full initialization.

Rounds	*d*-Monomial test			Monomial distr. test			Max. degree monomial		
	P	IVs	Memory	P	IVs	Memory	P	IVs	Memory
160	1	19	2^{19}	2^6	20	2^{20}	2^5	21	1

Table 3. Number of IV bits needed to attack the first keystream bit of Grain-128 with alternative Key/IV loading for different number of rounds in the initialization (out of 256 rounds).

Rounds	*d*-Monomial test				Monomial distr. test				Max. degree monomial			
	P	IVs	Memory	Fraction	P	IVs	Memory	Fraction	P	IVs	Memory	Fraction
256	1	14	2^{14}	1/256	2^6	8	2^8	4/256	2^5	14	1	100/256
256	1	16	2^{16}	5/256	2^6	10	2^{10}	10/256	2^5	20	1	108/256
288	1	20	2^{20}	0/256	2^6	20	2^{20}	0/256	2^5	20	1	47/256

Table 4. Number of IV bits needed to attack the initial state variables of Grain-128 with alternative Key/IV loading for different number of rounds in the initialization (out of 256 rounds).

4.2 Trivium

Trivium [7] is another hardware oriented stream cipher based on NFSRs. The state is divided into three registers which in total stores 288 bits. During the initialization the 80-bit key is inserted into the first register while an 80-bit IV is inserted into the second register. The cipher is clocked 4 full cycles before producing any keystream, i.e., 1152 clockings.

The results for Trivium are given in Table 5 and Table 6. The attacks on 736 and more rounds, the *d*-Monomial and the Monomial distribution attacks suffer from too large memory requirements. The maximal degree monomial test can be used to attack even 736 rounds (approximately 64% of initialization) using 33 IV bits, the attack on 736 rounds has only been performed a handful of times so the success rate is still an open issue in this case. The percentage of weak initial state variables for Trivium are approximately same using d-monomial and maximal degree tests.

Rounds	d-Monomial test			Monomial distr. test			Max. degree monomial		
	P	IVs	Memory	P	IVs	Memory	P	IVs	Memory
608	1	12	2^{12}	2^5	9	2^9	2^5	9	1
640	1	15	2^{15}	2^6	13	2^{13}	2^5	13	1
672	1	20	2^{20}	2^8	18	2^{17}	2^5	18	1
704	1	27	2^{27}	2^6	23	2^{23}	2^5	24	1
736	–	–	–	–	–	–	–	33 $*$	1

Table 5. Number of IV bits needed to attack the first keystream bit of Trivium for different number of rounds in the initialization (out of 1152 rounds).

Rounds	d-Monomial test				Monomial distr. test				Max. degree monomial			
	P	IVs	Memory	Fraction	P	IVs	Memory	Fraction	P	IVs	Memory	Fraction
608	1	12	2^{12}	144/288	2^5	12	2^{12}	105/288	2^5	12	1	169/288
640	1	12	2^{12}	57/288	2^5	12	2^{12}	29/288	2^5	12	1	86/288
672	1	15	2^{15}	87/288	2^5	15	2^{15}	0/288	2^5	15	1	108/288
704	1	20	2^{20}	74/288	2^5	20	2^{20}	12/288	2^5	20	1	76/288

Table 6. Number of IV bits needed to attack the initial state variables of Trivium for different number of rounds in the initialization (out of 1152 rounds).

Alternative Key/IV Loading for Trivium In the original key/IV loading, 128 bits of the initial state are assigned to constants and the key and IV bits affect only one state bit. Here, we propose an alternative initial Key/IV loading in which the first register is filled with the modulo 2 summation of key and IV, the second register is filled with IV and the last register is filled with the complement of key plus IV. In this setting, each IV bit affects 3 internal state bits, therefore the diffusion of IV bits to the state bits is satisfied in less number of clockings. We repeated the tests using the alternative loading and obtained the results given in Table 7 and Table 8. In the alternative loading, the required number of IV bits and memory needed to attack Trivium are approximately 50 percent more compared to the original loading.

Rounds	d-Monomial test			Monomial distr. test			Max. degree monomial		
	P	IVs	Memory	P	IVs	Memory	P	IVs	Memory
608	1	18	2^{18}	2^5	22	2^{22}	2^5	17	1
640	1	23	2^{23}	–	–	–	2^5	21	1

Table 7. Number of IV bits needed to attack the first keystream bit of Trivium with alternative Key/IV loading for different number of rounds in the initialization (out of 1152 rounds).

Rounds	d-Monomial test				Monomial distr. test				Max. degree monomial			
	P	IVs	Memory	Fraction	P	IVs	Memory	Fraction	P	IVs	Memory	Fraction
608	1	12	2^{12}	4/288	2^5	12	2^{12}	2/288	2^5	12	1	21/288
640	1	18	2^{18}	17/288	2^5	18	2^{18}	19/288	2^5	18	1	24/288
672	1	20	2^{20}	0/288	2^5	20	2^{20}	0/288	2^5	20	1	0/288

Table 8. Number of IV bits needed to attack the initial state variables of Trivium with alternative Key/IV loading for different number of rounds in the initialization (out of 1152 rounds).

4.3 Decim

Decim-v2 [8] is also a hardware oriented stream cipher based on a nonlinearly filtered LFSR and the irregularly decimation mechanism, ABSG. The internal state size of Decim-v2 is 192 bit and it is loaded with 80 bit Key and 64 bit IV. The first 80 bits of the LFSR are filled with the key, the bits between 81 and 160 are filled with linear functions of key and IV and the last 32 bits are filled with a linear function of IV bits.

The results we obtained for Decim-v2 are given in Table 9 and Table 10. The security margin for Decim against chosen IV attacks is very large, the cipher can only be broken when not more than about 3% of the initialization is used. This is mainly because of the initial loading of key and IV in which each IV bits affect 3 state variables and the high number of quadratic terms in the filter function. The weakness in initial state variables can be observed for higher number of clockings. The number of weak initial state variables are approximately same for all attacks.

Rounds	d-Monomial test			Monomial distr. test			Max. degree monomial		
	P	IVs	Memory	P	IVs	Memory	P	IVs	Memory
20	1	16	2^{16}	2^5	13	2^{13}	2^5	19	1

Table 9. Number of IV bits needed to attack the first keystream bit of Decim-v2 for different number of rounds in the initialization (out of 768 rounds).

Rounds	*d*-Monomial test				Monomial distr. test				Max. degree monomial			
	P	IVs	Memory	Fraction	P	IVs	Memory	Fraction	P	IVs	Memory	Fraction
160	1	12	2^{12}	47/192	2^5	17	2^{17}	47/192	2^5	12	1	44/192
192	1	20	2^{20}	18/192	2^5	20	2^{20}	13/192	2^5	20	1	17/192

Table 10. Number of IV bits needed to attack the initial state variables of Decim-v2 for different number of rounds in the initialization (out of 768 rounds).

4.4 Lex

Lex [11] is a software oriented stream cipher that is based on the block cipher AES. It uses 128 bit keys and IVs, the IV is taken as the initial state and to initialize the cipher one full AES round, i.e., 10 round operations, is applied to the IV before producing any keystream, then one more round is performed before the first leakage of keystream bits.

The maximal degree test on Lex relates to the saturation attack on AES, this attack uses the property that the sum of one output byte for all possible inputs to one byte of the cipher is zero after three rounds of AES. According to the Reed-Muller transform, the maximal degree monomial is calculated as the sum of all entries in the truth table, this means that a degree 8 polynomial can never occur from the same byte after 3 round operations. This also means that we can easily create a chosen IV distinguisher by considering eight IV bits that go into the same byte, the maximal degree of the output polynomial for three rounds is seven. After four rounds this property in general disappears, but the attack can be extended to attack 6 rounds of AES by using guess and determine techniques on other rounds.

Rounds	*d*-Monomial test			Monomial distr. test			Max. degree monomial		
	P	IVs	Memory	P	IVs	Memory	P	IVs	Memory
2	1	8	2^8	2^6	2	2^2	2^5	2	1
3	1	18	2^{18}	2^6	8	2^8	2^5	8	1

Table 11. Number of IV bits needed to attack the first keystream of Lex for different number of rounds in the initialization (out of 11 rounds).

As we see the monomial distribution attack and the maximal degree monomial attack performs best on Lex, a slight advantage for the Maximum Degree Monomial attack because of the low memory requirement. The *d*-Monomial test fails to find the anomaly that the degree 8 monomial can not exist since there is only one monomial of this degree.

5 Conclusions

In this study, we generalize the idea of d-monomial attacks and propose a framework for chosen IV statistical analysis. The proposed framework can be used as an instrument for designing good initialization procedures. It can be used to verify the effectiveness of the initialization, but also to help designing a well-balanced initialization, e.g., prevent an unnecessary large number of initial clockings or even reduce the number of gates used in an hardware implementation by being able to use a simpler loading procedure.

Also, we propose a few new statistical attacks, apply them on some existing stream cipher proposals, and give some conclusions regarding the strength of their IV initialization. In particular, we experimentally detected statistical weaknesses in the keystream of Trivium using an initialization reduced to 736 rounds as well as in some state bits of Grain-128 with full IV initialization. It is an open question how to utilize these weaknesses of state bits to attack the cipher.

For ciphers Grain and Trivium, we also propose alternative initialization schemes with slightly higher hardware complexity. In the proposed loadings, each IV and key bit affects more than one state bit and the resistance of the ciphers to the proposed attacks increases about 50%. Decim seems to have a high security margin and it is an interesting question whether a simpler loading procedure could be used in Decim which could mean a smaller footprint in hardware, fewer intial clockings could also be used for a faster intialization procedure.

References

1. Joan Daemen, René Govaerts, and Joos Vandewalle. Resynchronization weaknesses in synchronous stream ciphers. In *EUROCRYPT*, pages 159–167, 1993.
2. Jovan Dj. Golic and Guglielmo Morgari. On the resynchronization attack. In Thomas Johansson, editor, *FSE*, volume 2887 of *Lecture Notes in Computer Science*, pages 100–110. Springer, 2003.
3. Frederik Armknecht, Joseph Lano, and Bart Preneel. Extending the resynchronization attack. In Helena Handschuh and M. Anwar Hasan, editors, *Selected Areas in Cryptography*, volume 3357 of *Lecture Notes in Computer Science*, pages 19–38. Springer, 2004.
4. E. Filiol. A new statistical testing for symmetric ciphers and hash functions. In V. Varadharajan and Y. Mu, editors, *International Conference on Information, Communications and Signal Processing*, volume 2119 of *Lecture Notes in Computer Science*, pages 21–35. Springer-Verlag, 2001.
5. M.J. O. Saarinen. Chosen-iv statistical attacks on estream stream ciphers. eSTREAM, ECRYPT Stream Cipher Project, Report 2006/013, 2006. *http://www.ecrypt.eu.org/stream.*
6. M. Hell, T. Johansson, A. Maximov, and W. Meier. A stream cipher proposal: Grain-128. ISIT, Seattle, USA, 2006. available at *http://www.ecrypt.eu.org/stream.*
7. C. De Cannière and B. Preneel. Trivium - specifications. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/030, 2005. available at *http://www.ecrypt.eu.org/stream.*

8. C. Berbain, O. Billet, A. Canteaut, N. Courtois, B. Debraize, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Pornin, and H. Sibert. Decim v2. eSTREAM, ECRYPT Stream Cipher Project, Report 2006/004, 2006. http://www.ecrypt.eu.org/stream.
9. A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications. 2001. http://www.nist.gov.
10. S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. *Lecture Notes in Computer Science*, 2259:1–24, 2001.
11. A. Biryukov. A new 128 bit key stream cipher : Lex. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/013, 2006. available at *http://www.ecrypt.eu.org/stream.*