Chosen IV Statistical Analysis for Key Recovery Attacks on Stream Ciphers

Simon Fischer[1], Shahram Khazaei[2], and Willi Meier[1]

[1] FHNW, Windisch, Switzerland
[2] EPFL, Lausanne, Switzerland

Abstract. A recent framework for chosen IV statistical distinguishing analysis of stream ciphers is exploited and formalized to provide new methods for key recovery attacks. As an application, a key recovery attack on simplified versions of two eSTREAM Phase 3 candidates is given: For Grain-128 with IV initialization reduced to up to 180 of its 256 iterations, and for Trivium with IV initialization reduced to up to 672 of its 1152 iterations, it is experimentally demonstrated how to deduce a few key bits. Evidence is given that the present analysis is not applicable on Grain-128 or Trivium with full IV initialization.

Key words: Stream ciphers, Chosen IV analysis, eSTREAM, Grain, Trivium

1 Introduction

Synchronous stream ciphers are symmetric cryptosystems which are suitable in software applications with high throughput requirements, or in hardware applications with restricted resources (such as limited storage, gate count, or power consumption). For synchronization purposes, in many protocols the message is divided into short frames where each frame is encrypted using a different publicly known initialization vector (IV) and the same secret key. Stream ciphers should be designed to resist attacks that exploit many known keystreams generated by the same key but different chosen IVs. In general, the key and the IV is mapped to the initial state of the stream cipher by an initialization function (and the automaton produces then the keystream bits, using an output and update function). The security of the initialization function relies on its mixing (or *diffusion*) properties: each key and IV bit should affect each initial state bit in a complex way. This can be achieved with a round-based approach, where each round consists of some nonlinear operations. On the other hand, using a large number of rounds or highly involved operations is inefficient for applications with frequent resynchronizations. Limited resources of hardware oriented stream ciphers may even preclude the latter, and good mixing should be achieved with simple Boolean functions and a well-chosen number of rounds. In [4, 8, 9, 6], a framework for chosen IV statistical analysis of stream ciphers is suggested to investigate the structure of the initialization function. If mixing is not perfect, then the initialization function has an algebraic normal form (ANF) which can be distinguished from a uniformly random Boolean function. Particularly the coefficients of high degree monomials in the IV (*i.e.* the product of many IV bits) are suspect to some biased distribution: it will take many operations before all these IV bits meet in the same memory cell. In [4], this question was raised: "*It is an open question how to utilize these weaknesses of state bits to attack the cipher.*". The aim of this paper is to contribute to this problem and present a framework to mount key recovery attacks. As in [4, 8] one selects a subset of IV bits as variables. Assuming all other IV values as well as the key fixed, one can write a keystream symbol as a Boolean function. By running through all possible values of these bits and generating a keystream output each time, one can compute the truth table of this Boolean function. Each coefficient in the algebraic normal form of this Boolean function is parametrized by the bits of the secret key. Based on the idea of *probabilistic neural bits* from [1], we now examine if every key bit in the parametrized expression of a coefficient does occur, or more generally, how much influence each key bit does have on the value of the coefficient. If a coefficient depends on less than all key bits, this fact can be exploited to filter those keys which do not satisfy the imposed value for the coefficient. It is shown in [10] that for eSTREAM Phase 3 candidate Trivium with IV initialization reduced to 576 iterations, linear relations on the key bits can be derived for well chosen sets of variable IV bits. Our framework is more general, as it works with the concept of (probabilistic) neutral key bits, *i.e.* key bits which have no influence on the value of a coefficient with some (high) probability. This way, we can get information on the key for many more iterations in the IV initialization of Trivium, and similarly for the eSTREAM Phase 3 candidate Grain-128. On the other hand, extensive experimental evidence indicates clear limits to our approach: With our methods, it is unlikely to get information on the key faster than exhaustive key search for Trivium or Grain-128 with full IV initialization.

2 Problem Formalization

Suppose that we are given a fixed Boolean function $F(K, V) : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}$. An oracle chooses a random and unknown $K = (k_0, \ldots, k_{n-1})$ and returns us the value of $z = F(K, V)$ for every query $V = (v_0, \ldots, v_{m-1})$ of our choice (and fixed K). The function F could stand *e.g.* for the Boolean function which maps the key K and IV V of a stream cipher to the (let say) first output bit. Our goal as an adversary is to determine the unknown key K (or to distinguish F from a random function) in the chosen IV attack model only by dealing with the function F. If F mixes its inputs in a proper way, then one needs to try all 2^n possible keys in the worst case by sending $\mathcal{O}(n)$ queries to the oracle in order to find the correct key (since each query gives one bit information about the key for a balanced F). Here, we are going to investigate methods which can potentially lead to faster reconstruction of the key in the case where the function F does not properly mix its inputs. This could occur for example when the initialization phase of a stream cipher is performed through an iterated procedure for which the number of iterations has not been suitably chosen. On the other hand these methods may help to give the designers more insight to choose the required number of iterations. The existence of faster methods for finding the unknown key K highly depends on the structure of F. It may be even impossible to uniquely determine the key K. Let $F(K, V) = \sum_\kappa C_\kappa(V) K^\kappa$ where $K^\kappa = k_0^{\kappa_0} \cdots k_{n-1}^{\kappa_{n-1}}$ for the multi-index $\kappa = (\kappa_0, \ldots, \kappa_{n-1})$ (which can also be identified by its integer representation). Then the following lemma makes this statement more clear.

Lemma 1. *No adversary can distinguish between the two keys K_1 and K_2 for which $K_1^\kappa = K_2^\kappa$ for all $\kappa \in \{0, 1\}^n$ such that $C_\kappa(V) \neq 0$.*

Indeed, it is only possible to determine the values of $\{K^\kappa | \forall\kappa, C_\kappa(V) \neq 0\}$ which is not necessarily equivalent to determination of K. As a consequence of Lemma 1, the function F divides $\{0,1\}^n$ into *equivalence classes*: $\mathcal{K}_1,\ \mathcal{K}_2, \ldots, \mathcal{K}_J$ (with $J \leq 2^n$). See Ex. 3 as an application on a reduced version of Trivium.

3 Scenarios of Attacks

The algebraic description of the function $F(K,V)$ is too complex in general to be amenable to direct analysis. Therefore, from the function $F(K,V)$ and with the partition $V = (U,W)$ we derive simpler Boolean functions $C(K,W)$ with the help of the oracle. In our main example, $C(K,W)$ is a coefficient of the algebraic normal form of the function deduced from F by varying over the bits in U only, see Sect. 4 for more details. If this function $C(K,W)$ does not have a well-distributed algebraic structure, it can be exploited in cryptanalytic attacks. Let us investigate different scenarios:

1. If $C(K,W)$ is imbalanced for (not necessarily uniformly) random W and many fixed K, then the function F (or equivalently the underlying stream cipher) with unknown K can be distinguished from a random one, see [4,8,9,6].
2. If $C(K,W)$ is evaluated for some fixed W, then $C(K,W)$ is an expression in the key bits only. In [10], it was shown that in Trivium case for reduced iterations, linear relations on the key bits can be derived for a well chosen IV part.
3. If $C(K,W)$ has many key bits, which have (almost) no influence on the values of $C(K,W)$, a suitable approximation may be identified and exploited for key recovery attacks, see [1]. This is the target scenario of this paper and will be discussed in detail.

Scenario 1 has already been discussed in the introduction. In scenario 2, the underlying idea is to find a relation $C(K,W)$, evaluated for some fixed W, which depends only on a subset of t $(< n)$ key bits. The functional form of this relation can be determined with 2^t evaluations of $C(K,W)$. By trying all 2^t possibilities for the involved t key bits, one can filter those keys which do not satisfy the imposed relation. The complexity of this precomputation is 2^t times needed to compute $C(K,W)$, see Sect. 4. More precisely, if $p = \Pr\{C(K,W) = 0\}$ for the fixed W, the key space is filtered by a factor of $H(p) = p^2 + (1-p)^2$. For example, in the case of a linear function it is $p = H(p) = 1/2$. In addition, if several imposed relations on the key bits are available, it is easier to combine them to filter wrong keys if they have a simple structure, see *e.g.* [10]. In scenario 3, the main idea is to find a function $A(L,W)$ which depends on a key part L of t bits, and which is correlated to $C(K,W)$ with correlation coefficient ε, that is $\Pr\{C(K,W) = A(L,W)\} = 1/2(1+\varepsilon)$. Then, by asking the oracle N queries we get some information (depending on the new equivalence classes produced by A) about t bits of the secret K in time $N2^t$ by carefully analyzing the underlying hypothesis testing problem. We will proceed by explaining how to derive such functions C from the coefficients of the ANF of F in Sect. 4, and how to find such functions A using the concept of probabilistic neutral bits in Sect. 5.

4 Derived Functions from Polynomial Description

The function F can be written in the form $F(K,V) = \sum_{\nu,\kappa} C_{\nu,\kappa} V^{\nu} K^{\kappa}$ with binary coefficients $C_{\nu,\kappa}$. We can make a partition of the IV according to $V = (U,W)$ and $\nu = (\alpha,\beta)$ with l-bit segments U and α, and $(m-l)$-bit segments W and β . This gives the expression $F(K,V) = \sum_{\alpha,\beta,\kappa} C_{(\alpha,\beta),\kappa} U^{\alpha} W^{\beta} K^{\kappa} = \sum_{\alpha} C_{\alpha}(K,W) U^{\alpha}$ where $C_{\alpha}(K,W) = \sum_{\beta,\kappa} C_{(\alpha,\beta),\kappa} W^{\beta} K^{\kappa}$. For every $\alpha \in \{0,1\}^l$, the function $C_{\alpha}(K,W)$ can serve as a function C derived from F. Here is a toy example to illustrate the notation:

Example 1. Let $n = m = 3$ and $F(K,V) = k_1 v_1 \oplus k_2 v_0 v_2 \oplus v_2$. Let $U := (v_0, v_2)$ of $l = 2$ bits and $W := (v_1)$ of $m - l = 1$ bit. Then $C_0(K,W) = k_1 v_1$, $C_1(K,W) = 0$, $C_2(K,W) = 1$, $C_3(K,W) = k_2$. □

Note that an adversary with the help of the oracle can evaluate $C_{\alpha}(K,W)$ for the unknown key K at any input $W \in \{0,1\}^{m-l}$ for every $\alpha \in \{0,1\}^l$ by sending at most 2^l queries to the oracle. In other words, the partitioning of V has helped us to define a computable function $C_{\alpha}(K,W)$ for small values of l, even though the explicit form of $C_{\alpha}(K,W)$ remains unknown. To obtain the values $C_{\alpha}(K,W)$ for *all* $\alpha \in \{0,1\}^l$, an adversary asks for the output values of all 2^l inputs $V = (U,W)$ with the fixed part W. This gives the truth table of a Boolean function in l variables for which the coefficients of its ANF (*i.e.* the values of $C_{\alpha}(K,W)$) can be found in time $l2^l$ and memory 2^l using the Walsh-Hadamard transform. Alternatively, a *single* coefficient $C_{\alpha}(K,W)$ for a specific $\alpha \in \{0,1\}^l$ can be computed by XORing the output of F for all $2^{|\alpha|}$ inputs $V = (U,W)$ for which each bit of U is at most as large as the corresponding bit of α. This bypasses the need of 2^l memory.

One can expect that a subset of IV bits receives less mixing during the initialization process than other bits. These IV bits are called *weak*, and they would be an appropriate choice of U in order to amplify the non-randomness of C. However, it is an open question how to identify weak IV bits by systematic methods.

5 Functions Approximation

We are interested in the approximations of a given function $C(K,W) : \{0,1\}^n \times \{0,1\}^{m-l} \to \{0,1\}$ which depend only on a subset of key bits. To this end we make an appropriate partition of the key K according to $K = (L,M)$ with L containing t *significant* key bits and M containing the remaining $(n-t)$ *non-significant* key bits, and construct the function $A(L,W)$. We also use the term *subkey* to refer to the set of significant key bits. Such a partitioning can be identified by systematic methods, using the concept of probabilistic neutral bits from [1]:

Definition 1. *The neutrality measure of the key bit k_i with respect to the function $C(K,W)$ is defined as γ_i, where* $\Pr = \frac{1}{2}(1+\gamma_i)$ *is the probability (over all K and W) that complementing the key bit k_i does not change the output of $C(K,W)$.*

In practice, we will set a threshold γ, such that all key bits with $|\gamma_i| < \gamma$ are included in the subkey L (*i.e.* the probabilistic neutral key bits are chosen according to the individual values of their neutrality measure). The approximation $A(L,W)$ could be defined by $C(K,W)$ with non-significant key bits M fixed to zero. Here is another toy example to illustrate the method:

Example 2. Let $n = m = 3$, $l = 2$ and $C(K,W) = k_0k_1k_2v_0v_1 \oplus k_0v_1 \oplus k_1v_0$. For uniformly random K and W, we find $\gamma_0 = 1/8$, $\gamma_1 = 1/8$, $\gamma_2 = 7/8$. Consequently, it is reasonable to use $L := (k_0, k_1)$ as the subkey. With fixed $k_2 = 0$, we obtain the approximation $A(L,W) = k_0v_1 \oplus k_1v_0$ which depends on $t = 2$ key bits only. □

Note that, if M consists only of neutral key bits (with $\gamma_i = 1$), then the approximation A is exact, because $C(K,W)$ does not depend on these key bits. In [1] the notion of probabilistic neutral bits was used to derive an approximation function A in the case of $W = V$ and $C = F$ which lead to the first break of Salsa20/8.

6 Description of the Attack

In the precomputation phase of the attack, we need a suitable partitioning of the IV and the key (*i.e.* a function C and an approximation A). The weak IV bits are often found by a random search, while the weak key bits can be easily found with the neutrality measure for some threshold γ. Given C and A, we can find a small subset of candidates for the subkey L with a probabilistic guess-and-determine attack. In order to filter the set of all 2^t possible subkeys into a smaller set, we need to distinguish a correct guess of the subkey $\hat{L}$ from an incorrect one. Our ability in distinguishing subkeys is related to the correlation coefficient between $A(\hat{L},W)$ and $C(K,W)$ with $K = (L,M)$ under the following two hypotheses. H_0 : the guessed part $\hat{L}$ is correct, and H_1 : the guessed part $\hat{L}$ is incorrect. More precisely, the values of ε_0 and ε_1 defined in the following play a crucial role:

$$\Pr_{W}\{A(\hat{L},W) = C(K,W) | K = (\hat{L},M)\} = \frac{1}{2}(1+\varepsilon_0) \tag{1}$$

$$\Pr_{\hat{L},W}\{A(\hat{L},W) = C(K,W) | K = (L,M)\} = \frac{1}{2}(1+\varepsilon_1)\ . \tag{2}$$

In general, both ε_0 and ε_1 are random variables, depending on the key. In the case that the distributions of ε_0 and ε_1 are separated, we can achieve a small non-detection probability p_{mis} and false alarm probability p_{fa} by using enough samples. In the special case where ε_0 and ε_1 are constants with $\varepsilon_0 > \varepsilon_1$, the optimum distinguisher is Neyman-Pearson [2]. Then, N values of $C(K,W)$ for different W (assuming that the samples $C(K,W)$ are independent) are sufficient to obtain $p_{\text{fa}} = 2^{-c}$ and $p_{\text{mis}} = 1.3 \times 10^{-3}$, where

$$N \approx \left(\frac{\sqrt{2c(1-\varepsilon_0^2)\ln 2} + 3\sqrt{1-\varepsilon_1^2}}{\varepsilon_1 - \varepsilon_0}\right)^2\ . \tag{3}$$

The attack will be successful with probability $1 - p_{\text{mis}}$ and the complexity is as follows: For each guess $\hat{L}$ of the subkey, the correlation ε of $A(\hat{L},W) \oplus C(K,W)$ must be computed, which requires computation of the coefficients $A(\hat{L},W)$ by the adversary, and computation of the coefficient $C(K, W)$ through the oracle, for the same N values of W, having a cost of $N2^l$ at most. This must be repeated for all 2^t possible guesses $\hat{L}$. The set of candidates for the subkey L has a size of about $p_{\text{fa}}2^t = 2^{t-c}$. The whole key can then be verified by an exhaustive search over the key part M with a cost of $2^{t-c}2^{n-t}$ evaluations of F. The total complexity becomes $N2^l2^t + 2^{t-c}2^{n-t} = N2^{l+t} + 2^{n-c}$. Using more than one function C or considering several chosen IV bits U may be useful to reduce complexity; however, we do not deal with this case here.

Remark 1. In practice, the values of ε_0 and ε_1 are key dependent. If the key is considered as a random variable, then ε_0 and ε_1 are also random variables. However, their distribution may not be fully separated, and hence a very small p_{mis} and p_{fa} may not be possible to achieve. We propose the following non-optimal distinguisher: first, we choose a threshold $\varepsilon_0^\star$ such that $p_\epsilon = \Pr\{\varepsilon_0 > \varepsilon_0^\star\}$ has a significant value, *e.g.* 1/2. We also identify a threshold $\varepsilon_1^\star$, if possible, such that $\Pr\{\varepsilon_1 < \varepsilon_1^\star\} = 1$. Then, we estimate the sample size using Eq. 3 by replacing ε_0 and ε_1 by $\varepsilon_0^\star$ and $\varepsilon_1^\star$, respectively, to obtain $p_{\text{fa}} \leq 2^{-c}$ and effective non-detection probability $p_{\text{mis}} \cdot p_\epsilon \approx 1/2$. If $\varepsilon_0^\star$ and $\varepsilon_1^\star$ are close, then the estimated number of samples becomes very large. In this case, it is better to choose the number of samples intuitively, and then estimate the related p_{fa}.

Remark 2. It is reasonable to assume that a false subkey $\hat{L}$, which is close to the correct subkey, may lead to a larger value of ε. Here, the measure for being "close" could be the neutrality measure γ_i and the Hamming weight: if only a few key bits on positions with large γ_i are false, one would expect that ε is large. However, we only observed an irregular (*i.e.* not continuous) deviation for very close subkeys. The effect on p_{fa} is negligible because subkeys with difference of low weight are rare.

7 Application to Trivium

The stream cipher Trivium [3] is one of the eSTREAM candidates with a 288-bit internal state consisting of three shift registers of different lengths. At each round, a bit is shifted into each of the three shift registers using a non-linear combination of taps from that and one other register; and then one bit of output is produced. To initialize the cipher, the $n = 80$ key bits and $m = 80$ IV bits are written into two of the shift registers, with the remaining bits being set to a fixed pattern. The cipher state is then updated $R = 18 \times 64 = 1152$ times without producing output in order to provide a good mixture of the key and IV bits in the initial state. We consider the Boolean function $F(K, V)$ which computes the first keystream bit after r rounds of initialization. In [4], Trivium was analyzed with chosen IV statistical tests and non-randomness was detected for $r = 10 \times 64, 10.5 \times 64, 11 \times 64, 11.5 \times 64$ rounds with $l = 13, 18, 24, 33$ IV bits, respectively. In [10], the key recovery attack on Trivium was investigated with respect to scenario 2 (see Sect. 3) for $r = 9 \times 64$. Here we provide more examples for key recovery attack with respect to scenario 3 for $r = 10 \times 64$ and $r = 10.5 \times 64$. In the following two examples, weak IV bits have been found by a random search. We first concentrate on equivalence classes of the key:

Example 3. For $r = 10 \times 64$ rounds, a variable IV part U with the $l = 10$ bit positions $\{34, 36, 39, 45, 63, 65, 69, 73, 76, 78\}$, and the coefficient with index $\alpha = 1023$, we could experimentally verify that the derived function $C_\alpha(K, W)$ only depends on $t = 10$ key bits L with bit positions $\{15, 16, 17, 18, 19, 22, 35, 64, 65, 66\}$. By assigning all 2^{10} different possible values to these 10 key bits and putting those L's which gives the same function $C_\alpha(K, W)$ (by trying enough samples of W), we could determine the equivalence classes for L with respect to C_α. Our experiment shows the existence of 65 equivalence classes: one with 512 members for which $k_{15}k_{16} + k_{17} + k_{19} = 0$ and 64 other classes with 8 members for which $k_{15}k_{16} + k_{17} + k_{19} = 1$ and the vector $(k_{18}, k_{22}, k_{35}, k_{64}, k_{65}, k_{66})$ has a fixed value. This shows that C_α provides $\frac{1}{2} \times 1 + \frac{1}{2} \times 7 = 4$ bits of information about the key in average. □

Example 4. For $r = 10 \times 64$ rounds, a variable IV part U with the $l = 11$ bit positions $\{1, 5, 7, 9, 12, 14, 16, 22, 24, 27, 29\}$, and the coefficient with index $\alpha = 2047$, the derived function $C_\alpha(K, W)$ depends on all 80 key bits. A more careful look at the neutrality measure of the key bits reveals that $\max(\gamma_i) \approx 0.35$ and only 7 key bits have a neutrality measure larger than $\gamma = 0.18$, which is not enough to get a useful approximation $A(L, W)$ for an attack. However, we observed that $C_\alpha(K, W)$ is independent of the key for $W = 0$, and more generally the number of significant bits depends on $|W|$. □

It is difficult to find a good choice of variable IV's for larger values of r, using a random search. The next example shows how we can go a bit further with some insight.

Example 5. Now we consider $r = 10.5 \times 64 = 10 \times 64 + 32 = 672$ rounds. The construction of the initialization function of Trivium suggests that shifting the bit positions of U in Ex. 4 may be a good choice. Hence we choose U with the $l = 11$ bit positions $\{33, 37, 39, 41, 44, 46, 48, 54, 56, 59, 61\}$, and $\alpha = 2047$. In this case, $C_\alpha(K, W)$ for $W = 0$ is independent of 32 key bits, and $p = \Pr\{C_\alpha(K, 0) = 1\} \approx 0.42$. This is already a reduced attack which is $1/H(p) \approx 1.95$ times faster than exhaustive search. □

The following example shows how we can connect a bridge between scenarios 2 and 3 and come up with an improved attack.

Example 6. Consider the same setup as in Ex. 5. If we restrict ourself to W with $|W| = 5$ and compute the value of γ_i conditioned over these W, then $\max_i(\gamma_i) \approx 0.68$. Assigning all key bits with $|\gamma_i| < \gamma = 0.25$ as significant, we obtain a key part L with the $t = 29$ bit positions $\{1, 3, 10, 14, 20, 22, 23, 24, 25, 26, 27, 28, 31, 32, 34, 37, 39, 41, 46, 49, 50, 51, 52, 57, 59, 61, 63, 68, 74\}$. Our analysis of the function $A(L, W)$ shows that for about 44% of the keys we have $\varepsilon_0 > \varepsilon_0^\star = 0.2$ when the subkey is correctly guessed. If the subkey is not correctly guessed, we observe $\varepsilon_1 < \varepsilon_1^\star = 0.15$. Then, according to Eq. 3 the correct subkey of 29 bits can be detected using at most $N \approx 2^{15}$ samples, with time complexity $N2^{l+t} \approx 2^{55}$. Note that the condition $N < \binom{69}{5}$ is satisfied here. □

8 Application to Grain

The stream cipher Grain-128 [7] consists of an LFSR, an NFSR and an output function $h(x)$. It has $n = 128$ key bits, $m = 96$ IV bits and the full initialization function has $R = 256$ rounds. We again consider the Boolean function $F(K, V)$ which computes the first keystream bit of Grain-128 after r rounds of initialization. In [4], Grain-128 was analyzed with chosen IV statistical tests. With $N = 2^5$ samples and $l = 22$ variable IV bits, they observed a non-randomness of the first keystream bit after $r = 192$ rounds. They also observed a non-randomness in the initial state bits after the full number of rounds. In [8], a non-randomness up to 313 rounds was reported (without justification). In this section we provide key recovery attack for up to $r = 180$ rounds with slightly reduced complexity compared with exhaustive search. In the following example, weak IV bits for scenario 2 have been found again by a random search.

Example 7. Consider $l = 7$ variable IV bits U with bit positions $\{2, 6, 8, 55, 58, 78, 90\}$. For the coefficient with index $\alpha = 127$ (corresponding to the monomial of maximum degree), a significant imbalance for up to $r = 180$ rounds can be detected: the monomial of degree 7 appears only with a probability of $p < 0.2$ for 80% of the keys. Note that in [4], the attack with $l = 7$ could only be applied to $r = 160$ rounds, while our improvement comes from the inclusion of weak IV bits. □

In the following examples, our goal is to show that there exists some reduced key recovery attack for up to $r = 180$ rounds on Grain-128.

Example 8. Consider again the $l = 7$ IV bits U with bit positions $\{2, 6, 8, 55, 58, 78, 90\}$. For $r = 150$ rounds we choose the coefficient with index $\alpha = 117$ and include key bits with neutrality measure less than $\gamma = 0.98$ in list of the significant key bits. This gives a subkey L of $t = 99$ bits. Our simulations show that $\varepsilon_0 > \varepsilon_0^\star = 0.95$ for about 95% of the keys, hence $p_{\text{mis}} = 0.05$. On the other hand, for 128 wrong guesses of the subkey with $N = 200$ samples, we never observed that $\varepsilon_1 > 0.95$, hence $p_{\text{fa}} < 2^{-7}$. This gives an attack with time complexity $N2^{t+l} + 2^n p_{\text{fa}} \approx 2^{121}$ which is an improvement of a factor of (at least) $1/p_{\text{fa}} = 2^7$ compared to exhaustive search. □

Example 9. With the same choice for U as in Ex. 7 and 8, we take $\alpha = 127$ for $r = 180$ rounds. We identified $t = 110$ significant key bits for L. Our simulations show that $\varepsilon_0 > \varepsilon_0^\star = 0.8$ in about 30% of the runs when the subkey is correctly guessed. For 128 wrong guesses of the subkey with $N = 128$ samples, we never observed that $\varepsilon_1 > 0.8$. Here we have an attack with time complexity $N2^{t+l} + 2^n p_{\text{fa}} \approx 2^{124}$, *i.e.* an improvement of a factor of 2^4. □

9 Conclusion

A recent framework for chosen IV statistical distinguishers for stream ciphers has been exploited to provide new methods for key recovery attacks. This is based on a polynomial description of output bits as a function of the key and the IV. A deviation of the algebraic normal form (ANF) from random indicates that not every bit of the key or the IV has full influence on the value of certain coefficients in the ANF. It has been demonstrated how this can be exploited to derive information on the key faster than exhaustive key search through approximation of the polynomial description and using the concept of probabilistic neutral key bits. Two applications of our methods through extensive experiments have been given: A reduced complexity key recovery for Trivium with IV initialization reduced to 672 of its 1152 iterations, and a reduced complexity key recovery for Grain-128 with IV initialization reduced to 180 of its 256 iterations. This answers positively the question whether statistical distinguishers based on polynomial descriptions of the IV initialization of a stream cipher can be successfully exploited for key recovery. On the other hand, our methods are not capable to provide reduced complexity key recovery of the eSTREAM Phase 3 candidates Trivium and Grain-128 with full initialization.

Acknowledgments

The first author is supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center of the Swiss National Science Foundation under grant number 5005-67322. The third author is supported by Hasler Foundation www.haslerfoundation.ch under project number 2005. We would like to thank the anonymous reviewers for their comments.

References

1. J.-Ph. Aumasson, S. Fischer, S. Khazaei, W. Meier, and C. Rechberger. New Features of Latin Dances: Analysis of Salsa, ChaCha, and Rumba. In *FSE 2008*.
2. T. Cover and J.A. Thomas. Elements of Information Theory. Wiley series in Telecommunication. Wiley, 1991.
3. C. de Cannière and B. Preneel. TRIVIUM: A Stream Cipher Construction Inspired by Block Cipher Design Principles. In *ISC 2006*. See also [5].
4. H. Englund, T. Johansson, and M. S. Turan. A Framework for Chosen IV Statistical Analysis of Stream Ciphers. In *INDOCRYPT 2007*. See also *Tools for Cryptoanalysis 2007*.
5. eSTREAM - The ECRYPT Stream Cipher Project - Phase 3. See www.ecrypt.eu.org/stream.
6. E. Filiol. A New Statistical Testing for Symmetric Ciphers and Hash Functions. In *ICICS 2002*.
7. M. Hell, T. Johansson, A. Maximov, and W. Meier. A Stream Cipher Proposal: Grain-128. In *ISIT 2006*.
8. S. O'Neil. Algebraic Structure Defectoscopy. In *Cryptology ePrint Archive, Report 2007/378*. See also http://www.defectoscopy.com.
9. M.-J. O. Saarinen. Chosen-IV Statistical Attacks Against eSTREAM Ciphers. In *SECRYPT 2006*.
10. M. Vielhaber. Breaking ONE.FIVIUM by AIDA an Algebraic IV Differential Attack. In *Cryptology ePrint Archive, Report 2007/413*.