# Conditional Differential Cryptanalysis of NLFSR-based Cryptosystems

Simon Knellwolf*, Willi Meier, and María Naya-Plasencia**

FHNW, Switzerland

**Abstract.** Non-linear feedback shift registers are widely used in lightweight cryptographic primitives. For such constructions we propose a general analysis technique based on differential cryptanalysis. The essential idea is to identify conditions on the internal state to obtain a deterministic differential characteristic for a large number of rounds. Depending on whether these conditions involve public variables only, or also key variables, we derive distinguishing and partial key recovery attacks. We apply these methods to analyse the security of the eSTREAM finalist Grain v1 as well as the block cipher family KATAN/KTANTAN. This allows us to distinguish Grain v1 reduced to 104 of its 160 rounds and to recover some information on the key. The technique naturally extends to higher order differentials and enables us to distinguish Grain-128 up to 215 of its 256 rounds and to recover parts of the key up to 213 rounds. All results are the best known thus far and are achieved by experiments in practical time.

**Keywords:** differential cryptanalysis, NLFSR, distinguishing attack, key recovery, Grain, KATAN/KTANTAN

## 1 Introduction

For constrained environments like RFID tags or sensor networks a number of cryptographic primitives, such as stream ciphers and lightweight block ciphers have been developed, to provide security and privacy. Well known such cryptographic algorithms are the stream ciphers Trivium [5] and Grain [12, 13] that have been selected in the eSTREAM portfolio of promising stream ciphers for small hardware [9], and the block cipher family KATAN/KTANTAN [6]. All these constructions build essentially on non-linear feedback shift registers (NLFSRs). These facilitate an efficient hardware implementation and at the same time enable to counter algebraic attacks.

Stream ciphers and block ciphers both mix a secret key a and public parameter (the initial value for stream ciphers and the plaintext for block ciphers) in an involved way to produce the keystream or the ciphertext, respectively.

* Supported by the Hasler Foundation www.haslerfoundation.ch under project number 08065.

** Supported by an ERCIM "Alain Bensoussan" Fellowship Programme.

In cryptanalysis, such systems are often analysed in terms of boolean functions that to each key $k$ and public parameter $x$ assign an output bit $f(k,x)$. Several cryptanalytic methods analyse derived functions from $f$. They can be roughly divided into algebraic and statistical methods. The cube attack presented in [8] is an algebraic method. It consists in finding many derivatives of $f$ that are linear in the key bits such that the key can be found by solving a system of linear equations. The $d$-monomial test introduced in [10] provides a statistical framework to analyse the distribution of degree $d$ monomials in the algebraic normal form of $f$. Another statistical approach is presented in [11, 14], where the concept of probabilistc neutral key bits is applied to derivatives of $f$. The notion of cube testers introduced in [2] covers many of these methods. All of them have in common that they interact with $f$ mainly in a black box manner, exploiting the structure of the underlying primitive only indirectly.

In this paper we propose a general analysis principle that we call *conditional differential cryptanalysis*. It consists in analysing the output frequency of derivatives of $f$ on specifically chosen plaintexts (or initial values). Differential cryptanalyis, introduced in [4] for the analysis of block ciphers, studies the propagation of an input difference through an iterated construction and has become a common tool in the analysis of initialization mechanisms of stream ciphers, see [3, 7, 18]. In the case of NLFSR-based constructions, only few state bits are updated at each iteration, and the remaining bits are merely shifted. This results in a relatively slow diffusion. Inspired by message modification techniques introduced in [17] for hash function cryptanalysis, we trace the differences round by round and identify conditions on the internal state bits that control the propagation of the difference through the initial iterations. From these conditions we derive plaintexts (or initial values) that follow the same characteristic at the initial rounds and allow us to detect a bias in the output difference. In some cases the conditions also involve specific key bits which enables us to recover these bits in a key recovery attack.

The general idea of conditional differential cryptanalysis has to be elaborated and adapted with respect to each specific primitive. This is effected for the block cipher family KATAN and its hardware optimized variant KTANTAN as well as for the stream ciphers Grain v1 and Grain-128. The analysis of the block cipher family KATAN/KTANTAN is based on first order derivatives and nicely illustrates our analysis principle. For a variant of KATAN32 reduced to 78 of the 254 rounds we can recover at least two key bits with probability almost one and complexity $2^{22}$. Comparable results are obtained for the other members of the family. We are not aware of previous cryptanalytic results on the KATAN/KTANTAN family. The analysis of Grain v1 is similar to that of KATAN, however the involved conditions are more sophisticated. We obtain a practical distinguisher for up to 104 of the 160 rounds. The same attack can be used to recover one key bit and four linear relations in key bits with high probability. Grain v1 was previously analysed in [7], where a sliding property is used to speed up exhaustive search by a factor two, and in [1], where a non-randomness property for 81 rounds could be detected.

Conditional differential cryptanalysis naturally extends to higher order derivatives. This is demonstrated by our analysis of Grain-128, which, compared to Grain v1, is surprisingly more vulnerable to higher order derivatives. We get a practical distinguisher for up to 215 of the 256 rounds and various partial key recovery attacks for only slightly less rounds. For a 197 round variant we recover eight key bits with probability up to 0.87, for a 213 round variant two key bits with probability up to 0.59. The previously best known cryptanalytic result was a theoretical key recovery attack on 180 rounds, and was able to speed up exhaustive key search by a factor $2^4$, but without the feasibility to predict the value of single key bits, see [11]. Moreover, a result in [7] mentions key recovery for up to 192 rounds and in [1] a non-randomness property was detected in a chosen key scenario.

The paper is organised as follows. Section 2 recalls the definition of higher order derivatives of boolean functions and discusses the application of frequency tests to such derivatives. Section 3 provides the general idea of conditional differential cryptanalysis of NLFSR-based cryptosystems. In the Sections 4, 5 and 6 this idea is refined and adapted to a specific analysis of the KATAN/KTANTAN family, Grain v1 and Grain-128.

## 2 Notation and Preliminaries

In this paper $\mathbb{F}_2$ denotes the binary field and $\mathbb{F}_2^n$ the $n$-dimensional vector space over $\mathbb{F}_2$. Addition in $\mathbb{F}_2$ is denoted by $+$, whereas addition in $\mathbb{F}_2^n$ is denoted by $\oplus$ to avoid ambiguity. For $0 \leq i \leq n-1$ we denote $e_i \in \mathbb{F}_2^n$ the vector with a one at position $i$ and zero otherwise.

We now recall the definition of the $i$-th derivative of a boolean function introduced in [15, 16] and we discuss the application of a frequency test to such derivatives.

### 2.1 Derivatives of Boolean Functions

Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a boolean function. The *derivative of $f$ with respect to $a \in \mathbb{F}_2^n$* is defined as

$$\Delta_a f(x) = f(x \oplus a) + f(x).$$

The derivative of $f$ is itself a boolean function. If $\sigma = \{a_1, \ldots, a_i\}$ is a set of vectors in $\mathbb{F}_2^n$, let $L(\sigma)$ denote the set of all $2^i$ linear combinations of elements in $\sigma$. The *$i$-th derivative of $f$ with respect to $\sigma$* is defined as

$$\Delta_\sigma^{(i)} f(x) = \sum_{c \in L(\sigma)} f(x \oplus c).$$

We note that the $i$-th derivative of $f$ can be evaluated by summing up $2^i$ evaluations of $f$. We always assume that $a_1, \ldots, a_i$ are linearly independent, since otherwise $\Delta_\sigma^{(i)} f(x) = 0$ trivially holds. If we consider a keyed boolean function $f(k, \cdot)$ we always assume that the differences are applied to the second argument and not to the key.

### 2.2 Random Boolean Functions and Frequency Test

Let $D$ be a non-empty subgroup of $\mathbb{F}_2^n$. A *random boolean function on $D$* is a function $D \to \mathbb{F}_2$ whose output is an independent uniformly distributed random variable. If $f$ is a random boolean function on $D$, the law of large numbers says that for sufficiently many inputs $x_1, \dots, x_s \in D$ the value

$$t = \frac{\sum_{k=1}^{s} f(x_k) - s/2}{\sqrt{s/4}}$$

approximately follows a standard normal distribution. Denoting

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}u^2} du$$

the standard normal distribution function, a boolean function is said to *pass the frequency test* on $x_1, \dots, x_s$ at a significance level $\alpha$ if

$$\Phi(t) < 1 - \frac{\alpha}{2}$$

A random boolean function passes the frequency test with probability $1 - \alpha$. If the frequency test is used to distinguish a keyed boolean function $f(k, \cdot)$ from a random boolean function, we denote by $\beta$ the probability that $f(k, \cdot)$ passes the frequency test for a random key $k$. The distinguishing advantage is then given by $1 - \alpha - \beta$.

### 2.3 Frequency Test on Derivatives

If $\sigma = \{a_1, \dots, a_i\}$ is a set of linearly independent differences, the $i$-th derivative of a boolean random function is again a boolean random function. Its output is the sum of $2^i$ independent uniformly distributed random variables. But for any two inputs $x, x'$ with $x \oplus x' \in L(\sigma)$ the output values are computed by the same sum and thus $\Delta_\sigma^{(i)} f(x) = \Delta_\sigma^{(i)} f(x')$. Hence, the $i$-th derivative is not a random function on $D$, but on the quotient group $D/L(\sigma)$. A frequency test of $\Delta_\sigma^{(i)} f$ on $s$ inputs needs $s2^i$ queries to $f$.

## 3 Conditional Differential Cryptanalysis of NLFSR

This section provides the general idea of our analysis. It is inspired by message modification techniques as they were introduced in [17] to speed up the collision search for hash functions. We trace differences through NLFSR-based cryptosystems and exploit the non-linear update to prevent their propagation whenever possible. This is achieved by identifying conditions on the internal state variables of the NLFSR. Depending on whether these conditions involve the public parameter or also the secret key, they have to be treated differently in a chosen plaintext attack scenario. The goal is to obtain many inputs that satisfy the conditions, i.e. that follow the same differential characteristic at the initial rounds. In more abstract terms, we analyse derivatives of keyed boolean functions and exploit that their output values are iteratively computed.

We briefly explain NLFSR-based cryptosystems and why our analysis principle applies to them. Then we define three types of conditions that control the difference propagation in NLFSR-based cryptosystems and we explain how to deal with each of these types in a chosen plaintext (chosen initial value) attack scenario. The basic strategy is refined and adapted in the later sections to derive specific attacks on KATAN/KTANTAN, Grain v1 and Grain-128.

### 3.1 NLFSR-based Cryptosystems

An NLFSR of length $l$ consists of an initial state $s_0, \ldots, s_{l-1} \in \mathbb{F}_2$ and a recursive update formula $s_{l+i} = g(s_i, \ldots, s_{l+i-1})$ for $i \geq 0$, where $g$ is a non-linear boolean function. The bit $s_{l+i}$ is called the *bit generated at round* $i$ and $s_i, \ldots, s_{l+i-1}$ is called the *state of round* $i-1$. Our analysis principle applies to any cryptographic construction that uses an NLFSR as a main building block. These constructions perform a certain number of rounds, generating at each round one or more bits that non-linearly depend on the state of the previous round. It is this non-linear dependency that we exploit in conditional differential cryptanalysis.

Let $f : \mathbb{F}_2^m \times \mathbb{F}_2^n \to \mathbb{F}_2$ denote the keyed boolean function that to every key $k$ and public parameter $x$ assigns one output bit $f(k, x)$ of an NLFSR-based construction. If we consider a first order derivative of the function $f$, we apply a difference $a \in \mathbb{F}_2^n$ to the public parameter. The value $\Delta_a f(k, x)$ then denotes the output difference $f(k, x) + f(k, x \oplus a)$. If $s_i$ is a state bit of our construction, we denote $\Delta_a s_i(k, x)$ the difference in this state bit for the key $k$, the public parameter $x$ and the difference $a$.

### 3.2 Conditions and Classification

We now introduce the concepts of our analysis principle. In general, the difference of a newly generated state bit depends on the differences and the values of previously generated state bits. Each time that $\Delta_a s_i(k, x)$ non-linearly depends on a bit that contains a difference, we can identify conditions on previously generated state bits that control the value of $\Delta_a s_i(k, x)$. In most cases, the conditions are imposed to prevent the propagation of the difference to the newly generated state bits. In particular it is important to prevent the propagation at the initial rounds. Since we want to statistically test the frequency of $\Delta_a f(k, \cdot)$ on inputs that satisfy the conditions, there is an important tradeoff between the number of imposed conditions and the number of inputs that we can derive. The conditions can not only involve bits of $x$, but also bits of $k$. We classify them into three types:

- Type 0 conditions only involve bits of $x$.
- Type 1 conditions involve bits of $x$ and bits of $k$.

– Type 2 conditions only involve bits of $k$.

In a chosen plaintext (chosen initial value) scenario, type 0 conditions can easily be satisfied by the attacker, whereas he cannot control type 2 conditions at all. In most cases, type 2 conditions consist of simple equations and the probability that they are satisfied for a uniformly random key can easily be determined. Since we do not assume that our attacks can be repeated for more than one key, type 2 conditions generally decrease the advantage of distinguishing attacks and define classes of weak keys for this kind of attacks. On the other hand we specifically exploit type 2 conditions to derive key recovery attacks based on hypothesis tests. This is explained in Section 6 where we analyse Grain-128.

In a different way, also type 1 conditions can be used to recover parts of the key. To deal with the type 1 conditions, we introduce the concept of free bits. Suppose that the state bit $s_i$ depends on $x$ as well as on some bits of $k$, and suppose that we want to satisfy the type 1 condition $s_i = 0$. In a chosen plaintext scenario, we cannot control this condition in a simple way. We call those bits of $x$ that do not influence the value of $s_i$ for any key $k$, the *free bits* for the condition. The remaining bits of $x$ are called *non-free*. Together with $k$ the non-free bits determine whether the condition is satisfied or not. We call $x$ a *valid input* if, for a given key $k$, it satisfies the imposed condition. If we define the set $\varphi$ as $\varphi = \{e_i \in \mathbb{F}_2^n | x_i \text{ is a free bit}\}$ then we can generate $2^{|\varphi|}$ valid inputs from a single valid input $x$: these are the elements of the coset $x \oplus L(\varphi)$. In general, more than one type 1 condition are imposed. In that case, the free bits are those that are free for all of these conditions. In some cases it may be possible to give a finite number of configurations for the non-free bits such that at least one configuration determines a valid input. Otherwise, if $t$ type 1 conditions are imposed, we expect that about one of $2^t$ different inputs is valid and we just repeat the attack several times with different random inputs.

In some cases we can not obtain enough inputs only by the method of free bits. We then try to find non-free bits that only must satisfy a given equation but otherwise can be freely chosen. This provides us with more degrees of freedom to generate a sample of valid inputs. We refer to the analysis of KATAN and Grain v1 for concrete examples of this method.

### 3.3 Choosing the Differences

The choice of a suitable difference for conditional differential cryptanalysis is not easy and strongly depends on the specific construction. In particular this holds for higher order derivatives, but also for first order ones. In general, the difference propagation should be controllable for as many rounds as possible with a small number of conditions. In particular, there should not be too many type 1 and type 2 conditions at the initial rounds. Differences which can be controlled by isolated conditions of type 1 or type 2 are favorable for key recovery attacks.

The set of differences for higher order derivatives can be determined by combining first order differences whose characteristics do not influence each other at the initial rounds. In a non-conditional setting, [1] describes a genetic algorithm for finding good sets of differences. This black-box approach did not yield particularly good sets for our conditional analysis.

## 4 Analysis of KATAN/KTANTAN

KATAN/KTANTAN is a family of lightweight block ciphers proposed in [6]. The family consists of six ciphers denoted by KATAN$n$ and KTANTAN$n$ for $n = 32, 48, 64$ indicating the block size of the cipher. All instances accept an 80-bit key and use the same building blocks, namely two NLFSRs and a small LFSR acting as a counter. The only difference between KATAN$n$ and KTANTAN$n$ is the key scheduling.

In the following we describe KATAN32 and provide the details of our analysis for this particular instance of the family. Our analysis of the other instances is very similar. We only sketch the differences and provide the empirical results.

We emphasize that our analysis does not reveal a weakness of any of the original KATAN/KTANTAN ciphers. In contrary, with respect to our method, it seems that the number of rounds is sufficiently large to provide a confident security margin.

### 4.1 Description of KATAN32

The two NLFSRs of KATAN32 have length 13 and 19 and we denote their states by $l_i, \ldots, l_{i+12}$ and $r_i, \ldots, r_{i+18}$, respectively. A 32-bit plaintext block $x$ is loaded to the registers by $l_i = x_{31-i}$ for $0 \leq i \leq 12$ and $r_i = x_{18-i}$ for $0 \leq i \leq 18$. The LFSR has length 8 and we denote its state by $c_i, \ldots, c_{i+7}$. Initialization is done by $c_i = 1$ for $0 \leq i \leq 6$ and $c_7 = 0$. The full encryption process takes 254 rounds defined by

$$\begin{aligned}
c_{i+8} &= c_i + c_{i+1} + c_{i+3} + c_{i+8}, \\
l_{i+13} &= r_i + r_{i+11} + r_{i+6}r_{i+8} + r_{i+10}r_{i+15} + k_{2i+1}, \\
r_{i+19} &= l_i + l_{i+5} + l_{i+4}l_{i+7} + l_{i+9}c_i + k_{2i},
\end{aligned}$$

where $k_0, \ldots, k_{79}$ are the bits of the key and $k_i$ is recursively computed by

$$k_{j+80} = k_j + k_{j+19} + k_{j+30} + k_{j+67}$$

for $i \geq 80$. Finally, the states of the two NLFSRs are output as the ciphertext. If we consider a round-reduced variant of KATAN32 with $r$ rounds, the bits $l_{r+i}$ for $0 \leq i \leq 12$ and $r_{r+i}$ for $0 \leq i \leq 18$ will be the ciphertext.

### 4.2 Key Recovery for KATAN32 Reduced to 78 Rounds

Our analysis is based on a first order derivative and uses the concept of free bits to satisfy type 1 conditions. Here, to obtain enough inputs, we will identify non-free bits that only must satisfy an underdefined system of linear equations, which gives us more freedom degrees generate the samples.

We consider a difference of weight five at the positions 1,7,12,22 and 27 of the plaintext block. Let $a = e_1 \oplus e_7 \oplus e_{12} \oplus e_{22} \oplus e_{27}$ denote the initial difference. At round 0 we have

$$\begin{aligned} \Delta_a l_{13}(k,x) &= 1 + x_{10}, \\ \Delta_a r_{19}(k,x) &= x_{24} + 1 \end{aligned}$$

and impose the conditions $x_{10} = 1$ and $x_{24} = 1$ to prevent the difference propagation. Similarly at the rounds 1, 2, 3 and 5, we impose the bits $x_2, x_6, x_5, x_9, x_{19}, x_{25}$ to be zero. At round 7 we have

$$\Delta_a l_{20}(k,x) = r_{22}$$

and we impose the first type 1 condition

$$r_{22} = x_{28} + x_{23} + x_{21} + k_6 = 0. \tag{1}$$

At round 9 we impose $x_3 = 0$. Then three additional type 1 conditions

$$\begin{aligned} r_{19} &= x_{31} + x_{26} + x_{27} + x_{22} + k_0 = 1, && (2) \\ r_{23} &= x_{27} + x_{22} + x_{23}x_{20} + x_{18} + x_7 + x_{12} + k_1 + k_8 = 0, && (3) \\ r_{26} &= 1 + x_{20}(x_{17} + k_3) + k_{14} = 0 && (4) \end{aligned}$$

are imposed at the rounds 11, 13 and 20.

The free bits for these conditions can be directly read from the equations. They are:

$$x_0, x_4, x_8, x_{11}, x_{13}, x_{14}, x_{15}, x_{16}, x_{29} \text{ and } x_{30}.$$

So far, for any valid plaintext we can derive a sample of $2^{10}$ valid plaintexts. Since, in this case, this is not enough to perform a significant frequency test, we try to obtain larger samples by better analysing the non-free bits. Looking at the equations (1) to (4), we note that the non-free bits $x_7, x_{12}, x_{18}, x_{21}, x_{22}, x_{26}, x_{27}, x_{28}$ and $x_{31}$ only occur linearly. They can be freely chosen as long as they satisfy the system of linear equations

$$\begin{cases} x_{28} + x_{21} = A \\ x_{31} + x_{26} + x_{27} + x_{22} = B \\ x_{27} + x_{22} + x_{18} + x_7 + x_{12} = C \end{cases}$$

for constants $A, B, C$. This system has $2^6$ different solutions that can be added to each valid plaintext. In total this gives a sample of size $2^{16}$ that we can generate from a valid plaintext. Since we imposed 9 type 0 conditions we are left with $2^5$ different samples of plaintexts for a given key. The conditions are satisfied for at least one of these samples. On this sample the difference in bit 18 of the ciphertext after 78 rounds (this is bit $r_{78}$) is strongly biased. We perfom a frequency test of $\Delta_a r_{78}(k,\cdot)$ on each of the $2^5$ generated samples. At significance level $\alpha = 10^{-4}$ the frequency test fails on at least one of them with probability almost one, and if it fails, all four type 1 conditions are satisfied with probability almost one. This allows us to recover $k_0$, $k_6$, the relation $k_1 + k_8$ and either $k_{14}$ (if $x_{20} = 0$) or the relation $k_3 + k_{14}$ with high probability. The complexity of this attack is $2^{22}$.

### 4.3 Analysis of KATAN48 and KATAN64

All the three members of the KATAN family perform 254 rounds, they use the same LFSR and the algebraic structure of the non-linear update functions is the same. The differences between the KATAN$n$ ciphers are the block size $n$, the length of the NLFSRs, the tap positions for the non-linear update and the number of times the NLFSRs are updated per round.

For KATAN48 the NLFSRs have length 19 and 29 and each register is updated twice per round. We obtained our best result with a difference of weight four at the positions 1, 10, 19 and 28 in the plaintext block. Imposing four type 0 conditions and two type 1 conditions we are able to derive a sample of size $2^{31}$ from a valid plaintext. This allows us to recover the key bit $k_{12}$ and the relation $k_1 + k_{14}$ after 70 rounds (this corresponds to 140 updates of the NLFSRs) with a complexity of $2^{34}$.

For KATAN64 the NFLSRs have length 25 and 39 and each register is updated three times per round. We obtained our best result with a difference of weight three at the positions 0, 13 and 26. Imposing six type 0 conditions and two type 1 conditions we are able to derive a sample of size at least $2^{32}$ from a valid plaintext. This allows us to recover $k_2$ and $k_1 + k_6$ after 68 rounds (204 updates of the NLFSRs) with a complexity of $2^{35}$

### 4.4 Analysis of the KTANTAN family

KTANTAN$n$ is very similar to KATAN$n$. They only differ in the key scheduling part. In KATAN the key is loaded into a register and linearly expanded to the round keys after round 40. Until round 40 the original key bits are used as the round keys. In KTANTAN, from the first round, the round keys are a linear combination of key bits (depending on the state of the counter LFSR, which is entirely known). Hence, our analysis of KATAN$n$ directly translates to KTANTAN$n$, but instead of recovering a single key bit, we recover a linear relation of key bits. For instance in KATAN32 we recover the relation $k_7 + k_{71}$ instead of bit $k_0$.

## 5 Analysis of Grain v1

Grain v1 is a stream cipher proposed in [13] and has been selected for the final eSTREAM portfolio [9]. It accepts an 80-bit key $k$ and a 64-bit initial value $x$.

The cipher consists of three building blocks, namely an 80-bit LFSR, an 80-bit NLFSR and a non-linear output function. The state of the LFSR is denoted by $s_i, \dots, s_{i+79}$ and the state of the NLFSR by $b_i, \dots, b_{i+79}$. The registers are initialized by $b_i = k_i$ for $0 \leq i \leq 79$, $s_i = x_i$ for $0 \leq i \leq 63$ and $s_i = 1$ for $64 \leq i \leq 79$ and updated according to

$$\begin{aligned} s_{i+80} &= f(s_i, \dots, s_{i+79}), \\ b_{i+80} &= g(b_i, \dots, b_{i+79}) + s_i, \end{aligned}$$

where $f$ is linear and $g$ has degree 6. The output function is taken as

$$z_i = \sum_{k \in \mathcal{A}} b_{i+k} + h(s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63}),$$

where $\mathcal{A} = \{1, 2, 4, 10, 31, 43, 56\}$ and $h$ is defined as

$$\begin{aligned} h(s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63}) = {}& s_{i+25} + b_{i+63} \\ &+ s_{i+3}s_{i+64} + s_{i+46}s_{i+64} + s_{i+64}b_{i+63} \\ &+ s_{i+3}s_{i+25}s_{i+46} + s_{i+3}s_{i+46}s_{i+64} + s_{i+3}s_{i+46}b_{i+63} \\ &+ s_{i+25}s_{i+46}b_{i+63} + s_{i+46}s_{i+64}b_{i+63} \end{aligned}$$

The cipher is clocked 160 times without producing any keystream. Instead the output function is fed back to the LFSR and to the NLFSR.

If we consider round-reduced variants of Grain v1 with $r$ initialization rounds, the feedback of the output stops after $r$ rounds and the first keystream bit is $z_r$.

Our analysis is similar to the one of KATAN32, but the equations for the conditions are more complex. We first present an attack on 97 rounds and then extend it to 104 rounds.

### 5.1 Distinguishing Attack and Key Recovery for 97 Rounds

Our analysis is based on the first order derivative with respect to a single difference in bit 37 of the initial value. Let $a = e_{37}$ denote the difference. The first conditions are defined at round 12, where the difference in $s_{37}$ eventually propagates to the state bits $s_{92}$ and $b_{92}$ via the feedback of $z_{12}$. We have

$$\Delta_a z_{12}(k, x) = 1 + x_{15}x_{58} + x_{58}k_{75}.$$

We impose the type 0 condition $x_{58} = 1$ and we define the type 1 condition $x_{15} + k_{75} = 0$ to prevent the propagation. The next conditions are determined at round 34, where we have

$$\Delta_a z_{34}(k, x) = s_{98} + x_{59}s_{80} + s_{80}s_{98} + s_{80}b_{97}.$$

We define the conditions $s_{80} = 0$ and $s_{98} = 0$. Similarly we determine $s_{86} = 0$ and $s_{92} = 0$ at the rounds 40 and 46, respectively. So far, we imposed one type 0 condition at round 12 and we have five type 1 conditions at the rounds 12, 34, 40 and 46. The type 1 conditions jointly have 25 free bits:

$$x_7, x_8, x_{10}, x_{11}, x_{14}, x_{16}, x_{17}, x_{20}, x_{22}, x_{24}, x_{28}, x_{30}, x_{32}, x_{33},$$
$$x_{34}, x_{36}, x_{39}, x_{42}, x_{45}, x_{49}, x_{54}, x_{55}, x_{59}, x_{60} \text{ and } x_{61}.$$

In average we expect that one out of $2^5$ randomly chosen initial values satisfies the conditions. We define a distinguisher that chooses $2^5$ random initial values and for each performs a frequency test of $\Delta_a z_{97}(k, \cdot)$ on the sample of $2^{25}$ inputs generated by the free bits. Instead of randomly choosing $2^5$ initial values we can choose $2^4$ and test each of them for $x_{15} = 0$ and $x_{15} = 1$. This guarantees that the condition from round 12 is satisfied for at least one of them. Experiments with $2^{10}$ keys at a significance level $\alpha = 0.005$ show that at least one of the $2^5$ tests fails with probability 0.99. This gives a distinguisher with complexity $2^{31}$ and advantage of about 0.83 for Grain v1 reduced to 97 rounds.

The two conditions $x_{15}+k_{75} = 0$ and $s_{86} = 0$ are crucial to obtain a significant bias after 97 rounds. In a key recovery scenario this reveals information about the key. Experiments show that both conditions hold with probability almost one if the frequency test fails. This recovers the key bit $k_{75}$ and the value of $k_7 + k_8 + k_{10} + k_{37} + k_{49} + k_{62} + k_{69}$ (coming from $s_{86} = 0$).

### 5.2 Extension to 104 Rounds

Using the same conditions as before, we extend the attack to 104 rounds. We use the same idea as for KATAN32 to increase the size of the sample that can be generated from one initial value. We gain four additional degrees of freedom by noting that the non-free bits $x_6$, $x_{19}$, $x_{29}$, $x_{44}$ and $x_{57}$ influence only the condition imposed at round 40 and must only satisfy the linear equation

$$x_6 + x_{19} + x_{29} + x_{44} + x_{57} = A$$

for a constant $A$. In total, we can now derive a sample of size $2^{29}$ from one initial value.

The distinguisher defined above has now a complexity of $2^{35}$ and advantage of about 0.45. When the frequency test fails, the conditions $x_{15} + k_{75} = 0$ and $s_{92} = 0$ are satisfied with a probability almost one, which gives us $k_{75}$ and the value of $k_{13} + k_{14} + k_{16} + k_{22} + k_{43} + k_{55} + k_{68}$ (coming from $s_{92} = 0$). The remaining three conditions are satisfied with a probability about 0.70 and give us similar relations in the key bits.

The sample size can be further increased, because also the non-free bits $x_{13}, x_{23}, x_{38}, x_{51}$ and $x_{62}$ only must satisfy a linear equation. This gives a distinguisher with complexity $2^{39}$ and advantage of about 0.58.

## 6 Analysis of Grain-128

Grain-128 was proposed in [12] as a bigger version of Grain v1. It accepts a 128-bit key $k$ and a 96-bit initial value $x$. The general construction of the cipher is the same as for Grain v1, but the LFSR and the NLFSR both contain 128-bits. The content of the LFSR is denoted by $s_i, \ldots, s_{i+127}$ and the content of the NLFSR is denoted by $b_i, \ldots, b_{i+127}$. The initialization with the key and the initial value is analogous to Grain v1 and the update is performed according to

$$s_{i+128} = f(s_i, \ldots, s_{i+127}),$$
$$b_{i+128} = g(b_i, \ldots, b_{i+127}) + s_i,$$

where $f$ is linear and $g$ has degree 2. The output function is taken as

$$z_i = \sum_{k \in \mathcal{A}} b_{i+k} + h(b_{i+12}, s_{i+8}, s_{i+13}, s_{i+20}, b_{i+95}, s_{i+42}, s_{i+60}, s_{i+79}, s_{i+95}),$$

where $\mathcal{A} = \{2, 15, 36, 45, 64, 73, 89\}$ and $h$ is defined as

$$h(x) = b_{i+12}s_{i+8} + s_{i+13}s_{i+20} + b_{i+95}s_{i+42} + s_{i+60}s_{i+79} + b_{i+12}b_{i+95}s_{i+95}$$

The cipher is clocked 256 times without producing any keystream. Instead the output function is fed back to the LFSR and to the NLFSR.

If we consider round-reduced variants of Grain-128 with $r$ initialization rounds, the feedback of the output stops after $r$ rounds and the first keystream bit is $z_r$.

For the analysis of Grain-128 we use higher order derivatives. The general idea of conditional differential cryptanalysis naturally extends. As in the case of first order derivatives we always assume that the differences are applied to the initial value and not to the key.

### 6.1 Distinguishing Attack up to 215 Rounds

Our attack is based on a derivative of order thirteen with respect to the set of differences

$$\sigma = \{e_0, e_1, e_2, e_{34}, e_{35}, e_{36}, e_{37}, e_{65}, e_{66}, e_{67}, e_{68}, e_{69}, e_{95}\}.$$

These differences are chosen because they do not influence each other in the initial rounds. As a consequence the corresponding differential characteristic (of order thirteen) is zero for as many as 170 rounds. This can be extended to 190 rounds by imposing simple type 0 conditions that control the propagation of each single difference. As an example we derive the conditions for the difference $e_{65}$. The first condition is derived from round 5, where we have

$$\Delta_{e_{65}} z_5(k, x) = x_{84}.$$

We impose $x_{84} = 0$. In the same way the conditions $x_{58} = 0$ and $x_{72} = 0$ prevent difference propagation at rounds 45 and 52. At round 23 we have

$$\Delta_{e_{65}} z_{23}(k, x) = k_{118}.$$

As we will see below, the type 2 condition $k_{118} = 0$ determines a class of weak keys for the distinguishing attack.

Proceeding the same way for the other differences we derive 24 type 0 conditions that consist in setting the following bits to zero: $x_{27}$, $x_{28}$, $x_{29}$, $x_{30}$, $x_{41}$, $x_{42}$, $x_{43}$, $x_{44}$, $x_{58}$, $x_{59}$, $x_{60}$, $x_{61}$, $x_{62}$, $x_{72}$, $x_{73}$, $x_{74}$, $x_{75}$, $x_{76}$, $x_{77}$, $x_{84}$, $x_{85}$, $x_{86}$, $x_{87}$, $x_{88}$. In addition to $k_{118}$ the key bits $k_{39}, k_{119}, k_{120}$ and $k_{122}$ can be identified to define classes of weak keys.

There are $2^{96-13-24} = 2^{59}$ initial values that are different in $\mathbb{F}_2^n/L(\sigma)$ and satisfy all type 0 conditions. We define a distinguisher that performs a frequency test of $\Delta_\sigma^{(13)} z_r(k,\cdot)$ on $2^{12}$ of these inputs. Table 1 summarizes the empirical results obtained for $2^{12}$ different keys tested at a significance level $\alpha = 0.005$. The indicated values denote the probabilty $1-\beta$, where $\beta$ denotes the probability that $\Delta_\sigma^{(13)} z_r(k,\cdot)$ passes the frequency test. Our distinguisher has complexity $2^{25}$ and advantage $1-\alpha-\beta$. The values in the first row are obtained without any condition on the key. They show that we can distinguish Grain-128 reduced to 215 rounds with an advantage of about 0.008. The other rows indicate the probabilities for the classes of weak keys defined by the indicated type 2 conditions.

**Table 1.** Distinguishing attack on Grain-128 reduced to $r$ rounds: Probability $1-\beta$ for $\alpha = 0.005$ and complexity $2^{25}$. Type 2 conditions define classes of weak keys.

| type 2 condition | $r = 203$ | $r = 207$ | $r = 211$ | $r = 213$ | $r = 215$ |
|---|---|---|---|---|---|
| – | 1.000 | 0.587 | 0.117 | 0.173 | 0.013 |
| $k_{39} = 0$ | 1.000 | 0.630 | 0.128 | 0.275 | 0.017 |
| $k_{118} = 0$ | 1.000 | 0.653 | 0.177 | 0.231 | 0.024 |
| $k_{119} = 0$ | 1.000 | 0.732 | 0.151 | 0.267 | 0.025 |
| $k_{120} = 0$ | 1.000 | 0.876 | 0.234 | 0.249 | 0.026 |
| $k_{122} = 0$ | 1.000 | 0.668 | 0.160 | 0.285 | 0.015 |

### 6.2 Key Recovery up to 213 Rounds

In this section we specifically exploit type 2 conditions to recover single key bits with high probability. The attack is explained by a prototypical example that recovers three bits of Grain-128 reduced to 197 rounds with a probability up to 0.87. It is based on a derivative of order five and can easily be extended to recover more bits by using slightly other derivatives. This is demonstrated by an attack that recovers eight bits using two additional derivatives (both of order five). A second attack uses the derivative of order thirteen from the previous section and recovers three bits for Grain-128 reduced to 213 rounds with a probability up to 0.59.

*Prototypical Example.* We use a derivative of order five with respect to the differences $\sigma = \{e_1, e_{36}, e_{66}, e_{67}, e_{68}\}$. In the same way as in the distinguishing attack, we impose conditions on the initial value to control the propagation of each difference. Altogether we impose 12 type 0 conditions and denote by $W$

the set of initial values satisfying all of them. The crucial observation is the following. The key bit $k_{121}$ controls the characteristic of $e_{68}$ in the very early phase of initialization, namely at round 26. If $k_{121} = 1$ the difference propagates, otherwise it does not. This strongly influences the frequency of $\Delta_{\sigma}^{(5)} z_r(k, \cdot)$ after $r = 197$ rounds. Similar strong influences can be found for $k_{40}$ after $r = 199$ rounds and for $k_{119}$ after $r = 200$ rounds. This allows to recover these bits by a binary hypothesis tests.

*Key Recovery by Hypothesis Test.* Let $X$ be a uniformly distributed random variable taking values in $W/L(\sigma)$ and define

$$p_r(k) = \Pr[\Delta_{\sigma}^{(5)} z_r(k, X) = 1].$$

If the key is considered as a uniformly distributed random variable $K$, $p_r(K)$ is a random variable in the interval $[0, 1]$. Our attack is based on the observation that the conditional distributions of $p_r(K)$ conditioned on $K_i = 0$ and $K_i = 1$, for well chosen $i$, strongly differ even for a large number of rounds. This can be exploited to perform a binary hypothesis test on the value of $K_i$. An attacker can estimate a single observation $\hat{p}_r$ of $p_r(K)$ to take her decision. Since in all our attacks the expectation of $p_r(K)$ conditioned on $K_i = 0$ is significantly smaller than the conditional expectation conditioned on $K_i = 1$, we determine a parameter $\pi \in [0, 1]$ and take our decision according to the rule defined as

$$K_i = \begin{cases} 0 & \text{if } \hat{p}_r < \pi \\ 1 & \text{otherwise.} \end{cases}$$

The success probability of the attack essentially depends on the choice of $\pi$. If we denote $\alpha = \Pr[p_r(K) \geq \pi | K_i = 0]$ the probability that we falsely guess $K_i = 1$ and $\beta = \Pr[p_r(K) < \pi | K_i = 1]$ the corresponding probability that we falsely guess $K_i = 0$, then the *probability of a correct decision*, denoted $\mathrm{P}_c$, is given as

$$\mathrm{P}_c = 1 - (\alpha + \beta)/2.$$

An optimal $\pi$ maximizes $\mathrm{P}_c$. Since the conditional distributions of $p_r(K)$ are not known explicitely, we empirically determine $\pi$ in a precomputation phase of the attack.

*Back to the Example.* The first row of Table 2 shows the precomputed parameters $\pi$ and the resulting probability $\mathrm{P}_c$ for our prototypical example. The precomputation of each $\pi$ was done for $2^{14}$ key pairs and $2^{14}$ initial values for each key. This gives an overall precomputation complexity of $6 \cdot 2^{33}$ since we have to compute two histograms for each key bit. The attack itself consists in estimating $\hat{p}_r$ for $r = 197, 199$ and 200. Note that all three estimates can be obtained by the same computation which has complexity $2^{19}$ when estimating over $2^{14}$ initial values. The probabilities $\mathrm{P}_c$ are not completely independent and the probability of correctly guessing all three bits together is about 0.463.

**Table 2.** Key recovery for reduced Grain-128: $\mathrm{P}_c$ is the probability of correctly guessing key bit $k_i$. The attack complexity is $2^{19}$ for $|\sigma| = 5$ and $2^{25}$ for $|\sigma| = 13$.

| Difference set | $k_i$ | $r$ | $\pi$ | $\mathrm{P}_c$ |
|---|---|---|---|---|
| $\sigma = \{e_1, e_{36}, e_{66}, e_{67}, e_{68}\}$ | $k_{40}$ | 199 | 0.494 | 0.801 |
| | $k_{119}$ | 200 | 0.492 | 0.682 |
| | $k_{121}$ | 197 | 0.486 | 0.867 |
| $\sigma = \{e_0, e_1, e_2, e_{34}, e_{35}, e_{36}, e_{37},$<br>$e_{65}, e_{66}, e_{67}, e_{68}, e_{69}, e_{95}\}$ | $k_{39}$ | 213 | 0.490 | 0.591 |
| | $k_{72}$ | 213 | 0.488 | 0.566 |
| | $k_{119}$ | 206 | 0.356 | 0.830 |
| | $k_{120}$ | 207 | 0.486 | 0.807 |
| | $k_{120}$ | 211 | 0.484 | 0.592 |
| | $k_{122}$ | 213 | 0.478 | 0.581 |

*Recovering 8 Bits after 197 Rounds.* The prototypical example can be extended by using two other sets of differences which are obtained by shifting all differences by one position to the left and to the right, respectively. This allows to recover five additional bits of the key, namely $k_{39}, k_{40}, k_{118}, k_{120}$ and $k_{122}$. The complexities of this extended attack are $9 \cdot 2^{34}$ for the precomputation and $3 \cdot 2^{19}$ for the attack itself. We recover all eight bits correctly with a probability of 0.123. This can be improved up to 0.236 by first determining $k_{121}$ and $k_{122}$ and then recovering the remaining bits conditioned on the values of $k_{121}$ and $k_{122}$.

*Recovering Bits up to 213 Rounds.* If we use the derivative of order thirteen that we already used in the distinguishing attack, after 213 rounds we can recover two key bits with probability of almost 0.6. The last row of Table 2 summarizes the results. Here, the precomputation was done for $2^{12}$ key pairs and $2^{12}$ initial values for each key which gives a precomputation complexity of $2^{38}$. The complexity of the attack itself is $2^{25}$.

## 7 Conclusion

We presented a first analysis of the KATAN/KTANTAN family as well as the best known cryptanalytic results on Grain v1 and Grain-128. This was obtained by conditional differential cryptanalysis which also applies to other NLFSR-based contructions and provides further hints for choosing an appropriate number of rounds with regard to the security/efficiency tradeoff in future designs of such constructions.

## Acknowledgements

This work was partially supported by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II.

## References

1. Aumasson, J.P., Dinur, I., Henzen, L., Meier, W., Shamir, A.: Efficient FPGA Implementations of High-Dimensional Cube Testers on the Stream Cipher Grain-128. In: SHARCS (2009)
2. Aumasson, J.P., Dinur, I., Meier, W., Shamir, A.: Cube Testers and Key Recovery Attacks on Reduced-Round MD6 and Trivium. In: Dunkelman, O. (ed.) FSE. LNCS, vol. 5665, pp. 1–22. Springer (2009)
3. Biham, E., Dunkelman, O.: Differential Cryptanalysis in Stream Ciphers. Cryptology ePrint Archive, Report 2007/218 (2007), http://eprint.iacr.org/
4. Biham, E., Shamir, A.: Differential Cryptanalysis of DES-like Cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO. LNCS, vol. 537, pp. 2–21. Springer (1990)
5. Cannière, C.D.: Trivium: A Stream Cipher Construction Inspired by Block Cipher Design Principles. In: Katsikas, S.K., Lopez, J., Backes, M., Gritzalis, S., Preneel, B. (eds.) ISC. LNCS, vol. 4176, pp. 171–186. Springer (2006)
6. Cannière, C.D., Dunkelman, O., Knezevic, M.: KATAN and KTANTAN - A Family of Small and Efficient Hardware-Oriented Block Ciphers. In: Clavier, C., Gaj, K. (eds.) CHES. LNCS, vol. 5747, pp. 272–288. Springer (2009)
7. Cannière, C.D., Küçük, Ö., Preneel, B.: Analysis of Grain's Initialization Algorithm. In: Vaudenay, S. (ed.) AFRICACRYPT. LNCS, vol. 5023, pp. 276–289. Springer (2008)
8. Dinur, I., Shamir, A.: Cube Attacks on Tweakable Black Box Polynomials. In: Joux, A. (ed.) EUROCRYPT. LNCS, vol. 5479, pp. 278–299. Springer (2009)
9. ECRYPT: The eSTREAM project, http://www.ecrypt.eu.org/stream/
10. Englund, H., Johansson, T., Turan, M.S.: A Framework for Chosen IV Statistical Analysis of Stream Ciphers. In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) INDOCRYPT. LNCS, vol. 4859, pp. 268–281. Springer (2007)
11. Fischer, S., Khazaei, S., Meier, W.: Chosen IV Statistical Analysis for Key Recovery Attacks on Stream Ciphers. In: Vaudenay, S. (ed.) AFRICACRYPT. LNCS, vol. 5023, pp. 236–245. Springer (2008)
12. Hell, M., Johansson, T., Maximov, A., Meier, W.: A Stream Cipher Proposal: Grain-128. In: ISIT. pp. 1614–1618 (2006)
13. Hell, M., Johansson, T., Meier, W.: Grain: A Stream Cipher for Constrained Environments. IJWMC 2(1), 86–93 (2007)
14. Khazaei, S., Meier, W.: New Directions in Cryptanalysis of Self-Synchronizing Stream Ciphers. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT. LNCS, vol. 5365, pp. 15–26. Springer (2008)
15. Knudsen, L.R.: Truncated and Higher Order Differentials. In: Preneel, B. (ed.) FSE. LNCS, vol. 1008, pp. 196–211. Springer (1994)
16. Lai, X.: Higher order derivatives and differential cryptanalysis. In: Blahut, R.E., Costello, D.J., Maurer, U., Mittelholzer, T. (eds.) Communicationis and Cryptography: Two Sides of one Tapestry. pp. 227–233. Kluwer Academic Publishers (1994)
17. Wang, X., Yu, H.: How to Break MD5 and Other Hash Functions. In: Cramer, R. (ed.) EUROCRYPT. LNCS, vol. 3494, pp. 19–35. Springer (2005)
18. Wu, H., Preneel, B.: Resynchronization Attacks on WG and LEX. In: Robshaw, M.J.B. (ed.) FSE. LNCS, vol. 4047, pp. 422–432. Springer (2006)