Status of Achterbahn and Tweaks

Berndt M. Gammel, Rainer Göttfert and Oliver Kniffler

Infineon Technologies AG
81726 Munich
Germany

berndt.gammel@infineon.com
rainer.goettfert@infineon.com
oliver.kniffler@infineon.com

Abstract

We report on the results of computations concerning the linear complexities of the NLFSRs deployed in Achterbahn's keystream generator. We outline a probabilistic algorithm for estimating the linear complexities of binary sequences of period $2^N - 1$. We define Achterbahn-Version 2 whose keystream generator consists of ten shift registers. We introduce the new combining function. We discuss recent cryptanalysis results against Achterbahn-Version 1. The last part of the paper is concerned with hardware optimization of the feedback functions of the deployed nonlinear primitive shift registers.

Keywords: Stream cipher, NLFSR, linear complexity, probabilistic algorithm, keystream generator.

1 Introduction

Achterbahn is a binary additive stream cipher. The keystream generator (KSG) of Achterbahn-Version 1 consists of eight nonlinear primitive binary feedback shift registers of lengths N between 22 and 31. The KSG of Achterbahn-Version 2 consists of ten primitive shift registers of lengths between 19 and 32. We call an N-stage feedback shift register *primitive* if it produces a sequence of least period $2^N - 1$ for every nonzero initial state $\mathbf{s}_0 \in \mathbb{F}_2^N = \{0,1\}^N$. Both versions of Achterbahn were designed for 80-bit secret key size and support initial values up to 80 bits.

The sequences produced by the eight, respectively ten, nonlinear feedback shift registers (NLFSRs) are combined by a Boolean combining function $R : \mathbb{F}_2^8 \to \mathbb{F}_2$, respectively $S : \mathbb{F}_2^{10} \to \mathbb{F}_2$, to produce the keystream $\zeta = (z_n)_{n=0}^{\infty}$. In reduced Achterbahn the sequences to be combined are the standard output sequences of the NLFSRs (corresponding to given initial states of the shift registers). The standard output sequence of a feedback shift register is obtained by emitting the content of the right-most cell D_0 of the shift register at each clock pulse (assuming that the shifts are performed from left to right).

In full Achterbahn each NLFSR is endowed with a configurable linear feedforward output function controlled by the secret key and the initial value. The produced output sequence $\tau = (t_n)_{n=0}^{\infty}$ is a linear combination of the standard output sequence $\sigma = (s_n)_{n=0}^{\infty}$ and some shifted versions thereof. For instance, let us assume that $t_n = s_n + s_{n+1} + s_{n+4}$ for $n \geq 0$. We then write $\tau = f(T)\sigma$, where $f \in \mathbb{F}_2[x]$ is called the *filter polynomial* and T denotes the shift operator on the $\mathbb{F}_2$-vector space $\mathbb{F}_2^{\infty}$ under termwise operations on sequences. That is, $T\sigma = (s_{n+1})_{n=0}^{\infty}$ for all binary sequences $\sigma = (s_n)_{n=0}^{\infty}$. In the above example, $f(x) = 1 + x + x^4$.

Notice that if all applied filter polynomials are equal to the constant polynomial $f(x) = 1$, the keystream produced by full Achterbahn—under this specific configuration of the output functions—is identical to the keystream produced by reduced Achterbahn. In other words, the KSG of reduced Achterbahn is contained in the KSG of full Achterbahn as a special case. An implementation of full Achterbahn can, therefore, also be operated in the *reduced Achterbahn mode.* A millionaire possessing full Achterbahn can exchange secret information with a pauper who can only afford low cost reduced Achterbahn.

2 Linear complexity of the keystream

Any two nonzero standard output sequences of a primitive feedback shift register have the same minimal polynomial and, therefore, the same linear complexity, which we call *the linear complexity* of the shift register.

Throughout this report, we use the following abbreviations. The lengths of the shift registers are denoted by $N_1, N_2, \ldots$. The linear complexities of the shift registers are designated by $L_1, L_2, \ldots$. The least periods of the nonzero output sequences of the shift registers are denoted by $P_1, P_2, \ldots$. Thus, $P_i = 2^{N_i} - 1$ for all i. A nonzero standard output sequence of the ith shift register is denoted by σ_i. The filter polynomials defining the linear feedforward output functions are denoted by $f_1, f_2, \ldots$. The Boolean combining functions of Achterbahn-Version 1 and Version 2 are designated by $R(x_1, \ldots, x_8)$ and $S(x_1, \ldots, x_{10})$, respectively. The keystream is denoted by $\zeta = (z_n)_{n=0}^{\infty}$. Thus, for instance, in the case of reduced Achterbahn-Version 1, we have $\zeta = R(\sigma_1, \ldots, \sigma_8)$, and in the case of full Achterbahn-Version 2, $\zeta = S(f_1(T)\sigma_1, \ldots, f_{10}(T)\sigma_{10})$.

Suppose we are given $t \geq 1$ primitive binary NLFSRs of lengths $N_1, \ldots, N_t$ and linear complexities $L_1, \ldots, L_t$. Let $\sigma_1, \ldots, \sigma_t$ be standard output sequences of the t shift registers corresponding to any nonzero initial states. Let $F(x_1, \ldots, x_t)$ be an arbitrary Boolean function of t variables. Let $\zeta = R(\sigma_1, \ldots, \sigma_t)$, that is $\zeta = (z_n)_{n=0}^{\infty}$ with $z_n = F(\sigma_1(n), \ldots, \sigma_t(n))$ for $n = 0, 1, \ldots$.

If the lengths $N_1, \ldots, N_t$ of the t shift registers are pairwise relatively prime, then the linear complexity $L(\zeta)$ of ζ can be expressed as

$$L(\zeta) = F(L_1, \ldots, L_t) \tag{1}$$

with the understanding that F is now regarded as a function over the integers. Formula (1) is well known for primitive LFSRs under less restrictive assumptions on the lengths of the shift registers [10]. For primitive NLFSRs of pairwise relatively prime lengths, the formula is implicitly contained in [10, Corollary 6], [9, Theorem 5], and [2, Theorem 3].

If the lengths of the primitive NLFSRs are not pairwise relatively prime, then equation (1) does not hold. In this case, $F(L_1, \ldots, L_t)$ provides only an upper bound for $L(\zeta)$. However, in many cases, it is still possible to derive a reasonable lower bound for the linear complexity of ζ.

Lemma 1. *Let $\sigma_1, \ldots, \sigma_t$ be nonzero output sequences of primitive binary NLFSRs of lengths $N_1, \ldots, N_t$, respectively, and with linear complexities $L_1, \ldots, L_t$, respectively. Let $F(x_1, \ldots, x_t)$ be a Boolean function of algebraic degree $d \geq 1$. A lower bound for the linear complexity of the sequence $\zeta = F(\sigma_1, \ldots, \sigma_t)$ can be given if the following two conditions are fulfilled:*

1. *The algebraic normal form (ANF) of $F(x_1, \ldots, x_t)$ contains a monomial $x_{i_1} x_{i_2} \cdots x_{i_d}$ of degree d for which the corresponding shift register lengths $N_{i_1}, \ldots, N_{i_d}$ are pairwise relatively prime.*
2. *For all other monomials of degree d, which have the form $x_{i_1} \cdots x_{i_{j-1}} x_k x_{i_{j+1}} \cdots x_{i_d}$, we have $\gcd(N_{i_j}, N_k) = 1$.*

If both assumptions are true, then

$$L_{i_1} L_{i_2} \cdots L_{i_d} \leq L(\zeta). \tag{2}$$

Proof. We only give a sketch of the proof. See [2] for more details. We first recall some facts of [11, Chap. 4]. Let $f, g, \ldots, h$ be binary polynomials of positive degree and with nonzero constant terms. Then $f \vee g \vee \cdots \vee h \in \mathbb{F}_2[x]$ is defined to be the polynomial whose roots are the distinct products $\alpha\beta \cdots \gamma$, where α is a root of f, β a root of g, and γ a root of h. The polynomial $f \vee g \vee \cdots \vee h$ is irreducible if and only if the polynomials $f, g, \ldots, h$ are all irreducible and of pairwise relatively prime degrees. In this case, $\deg(f \vee g \vee \cdots \vee h) = \deg(f) \deg(g) \cdots \deg(h)$.

Let the canonical factorization of the minimal polynomial of σ_k over $\mathbb{F}_2$ be given by

$$m_{\sigma_k} = \prod_{j_k=1}^{c_k} h_{j_k} \qquad \text{for } k = 1, \ldots, t.$$

The polynomials h_{j_k} are distinct binary irreducible polynomials with $\deg(h_{j_k}) > 1$ and $\deg(h_{j_k})$ divides N_k.

Consider d sequences of $\{\sigma_1, \ldots, \sigma_t\}$. For simplicity of notation, say, $\sigma_1, \ldots, \sigma_d$. We associate to the sequences $\sigma_1, \ldots, \sigma_d$ the polynomial

$$f_{12\ldots d} = \prod_{j_1=1}^{c_1} \cdots \prod_{j_d=1}^{c_d} (h_{j_1} \vee \cdots \vee h_{j_d}). \tag{3}$$

If $N_1, \ldots, N_d$ are pairwise relatively prime, then $f_{12\ldots d}$ is the minimal polynomial of the product sequence $\sigma_1 \ldots \sigma_d$. In fact, (3) represents the canonical factorization of the minimal polynomial. Using $\deg(h_{j_1} \vee \cdots \vee h_{j_d}) = \deg(h_{j_1}) \cdots \deg(h_{j_d})$, we obtain for the linear complexity of $\sigma_1 \cdots \sigma_d$:

$$\begin{aligned} L(\sigma_1 \cdots \sigma_d) = \deg(f_{12\ldots d}) &= \sum_{j_1=1}^{c_1} \cdots \sum_{j_d=1}^{c_d} \deg(h_{j_1} \vee \cdots \vee h_{j_d}) \\ &= \prod_{k=1}^{d} \left(\sum_{j_k=1}^{c_k} \deg(h_{j_k}) \right) = \prod_{k=1}^{d} L(\sigma_k) = \prod_{k=1}^{d} L_k. \end{aligned}$$

This explains why we need the first requirement in the theorem. The second requirement guarantees that no other products of sequences appearing in $\zeta = F(\sigma_1, \dots, \sigma_t)$ will cancel out some irreducible factors of the the polynomial in (3) $\square$

In order to assign a numerical value to to lower bound for $L(\zeta)$ derived in Lemma 1, we need to know either the exact numerical values or at least lower bounds for the linear complexities $L_1, \dots, L_t$ of the deployed shift registers.

It should be mentioned that a general nontrivial lower bound for the linear complexity L of a nonzero output sequence of a primitive binary N-stage feedback shift register is not known. We have, of course, $N \leq L \leq 2^N - 2$. The trivial lower bound $L = N$ is attained if and only if the primitive shift register is linear. For nonlinear primitive shift registers experimental results show that mostly the upper bound $L = 2^N - 2$ is attained (in over 50% of our observations). We also observed that occasionally the linear complexity L drops below the value 2^{N-1}. This happened in 0.00003% of our observations comprising about 10^8 primitive NLFSRs. The situation is different compared to de Bruijn sequences [8], where the linear complexity of the sequence never drops below the value $2^{N-1} + N$.

Since no nontrivial lower bounds for binary primitive NLFSR-output sequences have been proved in the literature, we have to roll our sleeves up and determine lower bounds for the numbers L_i by way of computation. We did this in two ways, using the Berlekamp-Massey algorithm and using a new probabilistic algorithm.

The KSG of Achterbahn-Version 1 consists of eight NLFSRs of lengths $N = 22$, 23, 25, 26, 27, 28, 29, and 31. For the first three shift registers we found, applying the Berlekamp-Massey algorithm, $L_1 = 2^{22} - 13$, $L_2 = 2^{23} - 2$, and $L_3 = 2^{25} - 2$. For the remaining five shift registers we verified that $L_i \geq 2^{25.8}$ for $i = 5, \dots, 8$, using again the Berlekamp-Massey algorithm. Using the probabilistic algorithm [5], we found that with probability $> 1 - 2^{-100}$ all eight NLFSRs have linear complexities $L \geq 2^{N-1}$, if N denotes the length of the shift register.

The KSG of Achterbahn-Version 2 consists of ten primitive NLFSRs of lengths $N =$ 19, 22, 23, 25, 26, 27, 28, 29, 31, and 32. With the Berlekamp-Massey algorithm we found $L_1 = 2^{19} - 2$, $L_2 = 2^{22} - 2$, $L_3 = 2^{23} - 2$, $L_4 = 2^{25} - 2$, and verified that $L_i \geq 2^{25.2}$ for $i = 5, \dots, 10$. Using the probabilistic algorithm, we verified for all ten shift registers that $L \geq 2^{N-1}$ with probability of error $< 2^{-100}$.

We outline the basic ideas of the used probabilistic algorithm. Let us use a primitive NLFSR of length $N = 31$ as an example. Let $\sigma = (s_n)_{n=0}^{\infty}$ be any standard output sequence of the shift register corresponding to a nonzero initial state. We want to verify that the linear complexity of σ is greater than half the period of σ. The least period of σ is $P = 2^{31} - 1$. The polynomial $x^P - 1$ is a characteristic polynomial of σ. We have

$$x(x^P - 1) = x^{2^{31}} - x = x(x-1) \prod_{\substack{f \text{ irred.} \\ \deg(f)=31}} f(x),$$

where the product is extended over all binary irreducible polynomials of degree 31. It is easily seen that the minimal polynomial m_σ of σ does not contain the polynomials x or $x - 1$ as factors. Since the minimal polynomial of a periodic sequence divides any characteristic polynomial of the sequence, we conclude that m_σ is the product of distinct irreducible binary polynomials of degree 31. If m_σ contains more than one half of all irreducible polynomials of degree 31, then we know that the linear complexity of σ must be greater than half the period of σ.

Given a certain irreducible polynomial f of degree 31, we can check whether or not f is a factor of m_σ in the following way:

1. Compute the polynomial $g_f(x) = (x^P - 1)/f(x)$;
2. Check whether $g_f(T)\sigma \neq \mathbf{0}$.

Again, T denotes the shift operator, and $\mathbf{0}$ represents the zero sequence. The following lemma is crucial.

Lemma 2. *The polynomial f divides m_σ if and only if $g_f(T)\sigma \neq \mathbf{0}$. Furthermore, $g_f(T)\sigma \neq \mathbf{0}$ if and only if the first $N = \deg(f)$ terms of the sequence $\tau = g_f(T)\sigma$ are not all zero.*

Algorithm:

1. Choose at random a binary irreducible polynomial f of degree $N = 31$.
2. Check whether $g_f(T)\sigma \neq \mathbf{0}$.
3. Repeat the first two steps k times.

If in all k experiments $g_f(T)\sigma \neq \mathbf{0}$, then the statement $L(\zeta) \geq 2^{N-1}$ is true with probability $\geq 1 - 2^{-k}$.

The Boolean combining function $S(x_1, \ldots, x_{10})$ for Achterbahn-Version 2, defined in equation (9) below, has algebraic degree $d = 4$. The ANF of S contains the following 22 monomials of degree 4:

$$\begin{gathered} x_1x_3x_6x_8,\ x_1x_3x_6x_9,\ x_1x_4x_6x_8,\ x_1x_4x_6x_9,\ x_1x_5x_6x_8,\ x_1x_5x_6x_9,\ x_2x_3x_6x_8, \\ x_2x_3x_6x_9,\ x_2x_4x_6x_8,\ x_2x_4x_6x_9,\ x_2x_5x_6x_8,\ x_2x_5x_6x_9,\ x_4x_5x_8x_{10},\ x_4x_5x_9x_{10}, \\ x_4x_6x_7x_8,\ x_4x_6x_7x_9,\ x_4x_6x_8x_{10},\ x_4x_6x_9x_{10},\ x_5x_6x_7x_8,\ x_5x_6x_7x_9,\ x_5x_7x_8x_{10}, \\ x_5x_7x_9x_{10}. \end{gathered} \tag{4}$$

We use Lemma 1 to lower bound $L(\zeta)$. The monomial with highest indices satisfying condition 1 of Lemma 1 is

$$x_4x_6x_9x_{10}. \tag{5}$$

The lengths of the corresponding shift registers, $N_4 = 25$, $N_6 = 27$, $N_9 = 31$, $N_{10} = 32$, are pairwise relatively prime. There are exactly two monomials in (4) that overlap with the monomial in (5) in three positions, namely the monomials

$$x_4x_5x_9x_{10} \qquad \text{and} \qquad x_4x_6x_8x_{10}.$$

We have $\gcd(N_5, N_6) = \gcd(26, 27) = 1$ and $\gcd(N_8, N_9) = \gcd(29, 31) = 1$. Thus condition 2 in Lemma 1 is satisfied. Using $L_i \geq 2^{N_i - 1}$ for $i = 1, \ldots, 10$, we conclude that

$$L(\zeta) \geq L_4L_6L_9L_{10} > 2^{24} \cdot 2^{26} \cdot 2^{30} \cdot 2^{31} = 2^{111}.$$

Those of us who only trust results derived by the application of a deterministic algorithm, can use $L_i \geq 2^{25.2}$. It then follows that

$$L(\zeta) > 2^{100}.$$

Otherwise we can use the afore mentioned results derived by the described probabilistic algorithm.

Theorem 1. *The linear complexity of the keystream of Achterbahn-Version 2 satisfies* $L(\zeta) > 2^{100}$ *with certainty and* $L(\zeta) > 2^{111}$ *with probability* $> 1 - 2^{-100}$.

3 Definition of Achterbahn-Version 2

The Boolean combining function in the initial proposal of Achterbahn [3] is given by

$$R(x_1, \ldots, x_8) = x_1 + x_2 + x_3 + x_4 + x_5x_7 + x_6x_7 + x_6x_8 + x_5x_6x_7 + x_6x_7x_8. \qquad (6)$$

Johansson, Meier and Muller [6] described two attacks against Achterbahn exploiting certain weaknesses of R. We responded in posting the following "improved combining functions" at the eSTREAM page [4]

$$R'(x_1, \ldots, x_8) = R(x_1, \ldots, x_8) + x_5x_6 + x_5x_8 + x_7x_8. \qquad (7)$$

and

$$R'' = x_1 + x_2 + x_3 + \sum_{4 \le i < j \le 8} x_ix_j + \sum_{4 \le i < j < k \le 8} x_ix_jx_k + \sum_{4 \le i < j < k < l \le 8} x_ix_jx_kx_l. \qquad (8)$$

Although the functions R' and R'' were meant as examples and never declared to be successor functions for R, in a recent report [7], Johansson, Meier and Muller demonstrated that Achterbahn with its initial combining function replaced by R' or R'' can also be broken.

Before we discuss the attacks found in [7] in detail, we make some general observations regarding desired properties of combining functions to be used in NLFSR-based combining generators, like the KSG of Achterbahn.

3.1 Some general remarks

A joint weakness of the three combining functions R, R' and R'' is that they all contain several variables linearly. This fact was exploited in the first attack in [6] and in the TMO-attack in [7] as well.

The following argument shows why variables should not appear linearly. Consider the function $R(x_1, \ldots, x_8)$ in (6) and the polynomial

$$g(x) = (x^{P_1} - 1)(x^{P_2} - 1)(x^{P_3} - 1)(x^{P_4} - 1),$$

where $P_i = 2^{N_i} - 1$ are the periods of the shift register output sequences $\sigma_1, \ldots, \sigma_4$. The polynomial $g(x)$ is a characteristic polynomial of $\sigma = \sigma_1 + \sigma_2 + \sigma_3 + \sigma_4$, that is $g(T)\sigma = \mathbf{0}$. Therefore, if we apply the linear operator $g(T)$ to the keystream

$$\zeta = \sigma_1 + \sigma_2 + \sigma_3 + \sigma_4 + \sigma_5\sigma_7 + \sigma_6\sigma_7 + \sigma_6\sigma_8 + \sigma_5\sigma_6\sigma_7 + \sigma_6\sigma_7\sigma_8,$$

we obtain

$$g(T)\zeta = g(T)(\sigma_5\sigma_7 + \sigma_6\sigma_7 + \sigma_6\sigma_8 + \sigma_5\sigma_6\sigma_7 + \sigma_6\sigma_7\sigma_8),$$

a sequence depending only on the states of the last four shift registers.

Even in the case when a variable does not appear linearly in the ANF of a Boolean function, but still with low degree, the influence of the corresponding shift register can be undone by applying the linear operator $g(T)$ to the keystream, were g is sparse and has relatively small degree. For instance, if $F(x_1, x_2, x_3, x_4) = x_1x_2 + x_2x_3 + x_1x_3x_4$, the sequence $\tau = g(T)\zeta$ is independent of σ_2 (and thus, independent of the contents of the second shift register) for

$$g(x) = (x^{P_1P_2} - 1)(x^{P_2P_3} - 1).$$

Therefore, another requirement for the Boolean function should be that it contains each variable in a monomial of maximal degree.

Yet another important rule is that for each variable there exists a monomial in the ANF of the function which has maximum degree and has the property that the shift register lengths corresponding to the variables in that monomial are pairwise relatively prime. The last requirement implies that no polynomial of small degree (compared to the linear complexity of the keystream) exists—dense or sparse—that could cancel out the influence of one or several shift registers, when applied to the keystream in the above sense.

3.2 The combining function

While most attacks in [7] could easily be avoided by making sure that the used Boolean function has maximum nonlinearity (for the given order of resiliency) and contains all of its variables in a monomial of maximum degree, there is one attack described in [7] which is quite aggressive. In this attack one guesses the content of one shift register and uses a linear approximation as a mean to confirm or reject the guess. The authors use only linear approximations in [7]. However, if we also take into account quadratic and cubic approximations in combination with the described guessing trick, we see that Achterbahn-Version 1 can always be successfully attacked no matter what Boolean combining function has been chosen. The reason is that the small number of eight variables imposes a severe restriction to the order of correlation immunity and nonlinearity of the function.

In order to avert attacks based on quadratic approximations, we need a combining function of ten variables. As a consequence, the KSG of Achterbahn-Version 2 will consist of ten primitive NLFSRs.

The combining function for Achterbahn-Version 2 is given by

$$\begin{aligned} S(x_1, \ldots, x_{10}) = x_1 + x_2 + x_3 + x_9 + G(x_4, x_5, x_6, x_7, x_{10}) \\ + (x_8 + x_9)(G(x_4, x_5, x_6, x_7, x_{10}) + H(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_{10})) \end{aligned} \tag{9}$$

with

$$\begin{aligned} G(x_4, x_5, x_6, x_7, x_{10}) = x_4(x_5 \vee x_{10}) + x_5(x_6 \vee x_7) + x_6(x_4 \vee x_{10}) \\ + x_7(x_4 \vee x_6) + x_{10}(x_5 \vee x_7) \end{aligned}$$

and

$$\begin{aligned} H(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_{10}) = x_2 + x_5 + x_7 + x_{10} + (x_3 + x_4)\bar{x}_6 \\ + (x_1 + x_2)(x_3\bar{x}_6 + x_6(x_4 + x_5)), \end{aligned}$$

where $a \vee b = a + b + ab$ and $\bar{a} = a + 1$ for $a, b \in \mathbb{F}_2$.

Function S has resiliency 5 and nonlinearity 448. The ANF of S contains 77 monomials, 22 thereof have degree 4. The function can be implemented in hardware with 63 GE. Each of the ten variables of S appears in a monomial of degree 4.

Since S has ten variables, we need another two NLFSRs. We choose shift registers of lengths 19 and 32.

Tweak: The KSG of Achterbahn-Version 2 consists of ten primitive binary NLFSRs of lengths 19, 22, 23, 25, 26, 27, 28, 29, 31, and 32. The maximum degrees of the corresponding filter polynomials describing the linear feedforward output functions of full Achterbahn are 3, 3, 3, 5, 6, 7, 8, 9, 10, 10.

Theorem 2. *The keystream ζ produced by the KSG of reduced Achterbahn-Version 2, as well as all 2^{64} translation distinct keystream sequences produced by full Achterbahn-Version 2, have least period*

$$\mathrm{Per}(\zeta) = \frac{1}{135} \prod_{i=1}^{10} \left(2^{N_i} - 1\right) > 2^{254}.$$

Consider the 22 monomials in (4). Each of the ten variables $x_1, \ldots, x_{10}$ appears in at least one monomial for which the corresponding shift register lengths are pairwise relatively prime. Due to this property and the verified fact that $L_i \geq 2^{N_i - 1}$ for $i = 1, \ldots, 10$, the following theorem can be proved.

Theorem 3. *Let ζ be a keystream produced by reduced or full Achterbahn-Version 2. For each polynomial $g \in \mathbb{F}_2[x]$ with $\deg(g) < 2^{80}$, the sequence $\tau = g(T)\zeta$ depends on all ten NLFSRs.*

3.3 Cryptanalysis of Achterbahn-Version 1

We now compare the complexities of all attacks described in [7] that were successfully applied against Achterbahn-Version 1 with combining functions R, R', or R'' with the complexity of the attack against Achterbahn-Version 2 with combining function S.

The attack described in [7, Sec. 4] makes use of the the fact that the function $R(x_1, \ldots, x_8)$ in (6) becomes linear for $x_5 = x_6 = 0$. The lengths of the corresponding shift registers are 27 and 28, which are the relevant parameters for the complexity of the attack. The complexity is $O(2^{27+28+1}) = O(2^{56})$ for reduced and $O(2^{73})$ for full Achterbahn-Version 1. The function $S(x_1, \ldots, S_{10})$ in (9) becomes only linear if we set at least five of the variables $x_4, x_5, x_6, x_7, x_8, x_9, x_{10}$ to constant values. Thus the length of the shift registers and the maximum degrees of the filter polynomials corresponding to the five variables that cause S to become linear are relevant for the complexity of this attack. We obtain the complexities $O(2^{139})$ and $O(2^{176})$ for reduced and full Achterbahn-Version 2, respectively.

The attack described in [7, Sec. 5] is a distinguishing attack, which exploits the fact that $R(x_1, \ldots, x_8)$ can be approximated by a linear function of eight variables containing five nonzero terms with probability 3/4. The attack requires the examination of 2^{64} keystream bits. The Boolean function $S(x_1, \ldots, x_{10})$ can at best be approximated by a linear function containing six nonzero terms and with probability 9/16. It follows that in order to detect the bias, $O(2^{384})$ keystream bits are necessary. As the keystream ζ of Achterbahn-Version 2 has least period $< 2^{255}$, the attack does not make sense.

The attack described in [7, Sec. 5.3] and [7, Sec. 7] is the most threatening attack in [7]. In Section 5.3, the function $R(x_1, \dots, x_8)$ is attacked. Function R agrees with

$$L(x_1, \dots, x_8) = x_1 + x_2 + x_3 + x_4 + x_6 \tag{10}$$

with probability $p = \frac{3}{4} = \frac{1}{2}(1+\frac{1}{2}) = \frac{1}{2}(1+\epsilon)$. The attacker guesses the first register. This step has complexity $O(2^{22})$. By guessing the first register, the approximation in (10) reduces from five to four nonzero terms. Consider the polynomial

$$g(x) = (x^{P_2} - 1)(x^{P_3} - 1)(x^{P_4} - 1)(x^{P_6} - 1).$$

The sequence $\tau = g(T)\zeta$ is the sum if 16 shifted versions of ζ. The bias for the sequence τ therefore is

$$\epsilon^{16} = \left(\frac{1}{2}\right)^{16} = 2^{-16}.$$

To take advantage of the bias one has to examine 2^{32} keystream bits. Altogether, the complexity of the attack is $2^{22} \cdot 2^{32} = 2^{54}$ for reduced and 2^{60} for full Achterbahn-Version 1.

The same method is used to attack R'' in [7, Sec. 7]. The time complexities of the attack against Achterbahn-Version 1 with R'' are $O(2^{70})$ for the reduced, and $O(2^{76})$ for the full version.

If we apply the attack to Achterbahn-Version 2, we observe that the best linear approximation to S has six nonzero terms and agrees with S with probability 9/16. This yields the complexity $O(2^{211})$, respectively $O(2^{214})$ if the attacker guesses the first register. A better strategy is to guess the contents of the first two registers. This attack has complexity $O(2^{137})$ for reduced and $O(2^{143})$ for full Achterbahn-Version 2. The best strategy consists in guessing the first three registers, which yields complexities $O(2^{112})$ and $O(2^{121})$.

The attack described in [7, Sec. 6.1] against R' takes advantage of the fact that R' contains the first four variables only linearly. The other four variables appear in the nonlinear part of R'. These four variables correspond to the last four shift registers which together can store 115 bits. A TMO-attack is described with time complexity $2^{57.5}$ requiring $2^{57.5}$ keystream bits.

The Boolean combining function S in Achterbahn-Version-2 does not depend linearly of any of its ten variables. Thus the nonlinear part of S coincides with the entire internal state of the KSG which has 262 bits. The complexity of the above attack is comparable with the complexity of a classical TM0-attack which here has time and data complexity 2^{131}.

The attack described in [7, Sec. 6.2] against full Achterbahn-Version 1 makes use of the fact that the function R' reduces to the affine function $L = x_1 + x_2 + x_3 + x_4 + x_7 + 1$ if the variables x_5 and x_6 are both set to 1. The attack requires some more keystream bits (approximately 2^{45}) than the attack described in [7, Sec. 4]. Otherwise the attacks are identical. The time complexity of the attack is $O(2^{73})$, since the lengths of the shift registers corresponding to variables x_5 and x_6 are 27 and 28. The maximum degrees of the corresponding filter polynomials are 8 and 9, respectively. This yields $27 + 28 + 8 + 9 + 1 = 73$, the exponent in the complexity estimation. The same attack applied to Achterbahn-Version 2 has time complexity $O(2^{176})$.

3.4 Quadratic approximations

Quadratic approximation attacks seem to be more threatening to our stream cipher than correlation attacks based on linear approximations. To estimate the threat, we have to consider all quadratic functions of ten variables which have a nonzero correlation coefficient with $S(x_1, \ldots, x_{10})$. The most threatening approximation is given by the quadratic function

$$Q(x_1, \ldots, x_{10}) = x_1 + x_2 + x_3x_4 + x_6x_{10}, \tag{11}$$

which agrees with S with probability

$$\frac{33}{64} = \frac{1}{2}\left(1 + \frac{1}{32}\right) = \frac{1}{2}(1 + \epsilon).$$

If we guess the first two registers of lengths $N_1 = 19$ and $N_2 = 22$, we have only two summands left in (11). The bias of the appropriately filtered keystream sequence is $\epsilon^4 = 2^{-20}$, so that 2^{40} keystream bits must be processed in order to confirm the guess. The overall complexity of the attack is $2^{19} \cdot 2^{22} \cdot 2^{40} = 2^{81}$, still above the complexity of exhaustive key search.

3.5 Cubic approximations

The most threatening cubic approximation is given by

$$C(x_1, \ldots, x_{10}) = x_4 + x_6x_9 + x_1x_2x_3, \tag{12}$$

which agrees with S with probability

$$\frac{63}{128} = \frac{1}{2}\left(1 - \frac{1}{64}\right) = \frac{1}{2}(1 + \epsilon).$$

We guess the the content of the fourth shift register, whose length is $N_4 = 25$. The terms of the sequence $\tau = g(T)\zeta$, where

$$g(x) = (x^{P_6P_9} - 1)(x^{P_1P_2P_3} - 1),$$

are biased with $\epsilon^4 = 2^{-24}$. Thus the time complexity to determine the contents of fourth shift register is $O(2^{73})$ and below the complexity of exhaustive key search. The degree of the polynomial g in (12) is greater than 2^{63}. The attacker needs more than 2^{63} keystream bits in order to run the attack. We counter such an attack by restricting the maximum frame length for our stream cipher to 2^{63} bits.

Tweak: The maximum length of a frame that can be used in the encryption process for Achterbahn-Version 2 is 2^{63} bits.

4 Hardware tweaks

In this section we show how the feedback logics of the driving NLFSRs can be improved with regard to their hardware efficiencies. The goals are:

— to reduce the gate count;

— to increase the frequency at which Achterbahn can be operated.

Both goals can be achieved without sacrificing security.

In the following, the design size is given in gate equivalents. One gate equivalent (GE) is the design size of a 2-input NAND gate. The reported figures have been derived from a synthesis of Achterbahn using high level description language VHDL and mapping the design on 130 nm CMOS standard cell library.

The design size of the KSG can be divided into the following four parts (compare 2):

1. The memory cells including one multiplexor per memory cell for the parallel key-loading.
2. The feedback logics of the ten NLFSRs.
3. The logic that implements the Boolean combining function.
4. The control logic.

How can we save hardware? We cannot shorten the lengths of the shift registers or use a sparser Boolean combining function without lowering the security level, nor can we reduce the control logic. However, there is room for savings in the circuits that implement the feedback functions of the shift registers.

4.1 Reducing the implementation costs of the feedback functions

In this section we describe a way how the implementation costs of the feedback functions can be reduced and at the same time the clock rates for the shift registers increased. The average design size of the feedback functions of the eight driving NLFSRs in the initial proposal of Achterbahn was 42.75 GE. This average value can be reduced to 24.7 GE per shift register in Achterbahn-Version 2.

The objective is to reduce the implementation costs of the feedback functions without thinning out their algebraic normal forms. This is important because a very sparse algebraic normal form would increase the required number of warm-up shifts in the last step of the key-loading algorithm and, thereby, extend resynchronization times. Considering that in many applications the resynchronization intervals are relatively short, this would not be acceptable. Besides, a very sparse feedback function provides less resistance against algebraic attacks [1] than a function of moderate sparsity does.

The objective is achieved by choosing primitive NLFSRs whose feedback functions can be implemented using less expensive gates. Also, 3-input gates are more efficient than 2-input gates. Table 1 lists the hardware costs for the implementation of various logical operations.

HW-Tweak: The initial feedback functions of the NLFSRs are replaced by more efficient feedback functions. The new feedback functions can be implemented at approximately half the hardware costs of the old ones and each function has logical depth three.

For the sake of illustration, let us consider the new NLFSR A. Its feedback function is given by

$$\begin{aligned} A(x_0, x_1, \ldots, x_{18}) = \mathrm{XOR}(&\mathrm{XOR}(x_0, x_3, \mathrm{MUX}(x_5, x_1; x_6)), \mathrm{XOR}(x_8, x_{12}, \mathrm{NAND}(x_4, x_7)), \\ &\mathrm{MUX}(\mathrm{NAND}(x_9, x_{11}), \mathrm{MUX}(x_6, x_{10}; x_4); \mathrm{MUX}(x_2, x_{10}; x_9))). \end{aligned}$$

Logical operation	**Binary function**	**Hardware cost**
NAND(a, b)	$ab + 1$	1.00 GE
NOR(a, b)	$1 + a + b + ab$	1.00 GE
AND(a, b)	ab	1.25 GE
OR(a, b)	$a + b + ab$	1.25 GE
XOR(a, b)	$a + b$	2.25 GE
NAND(a, b, c)	$abc + 1$	1.25 GE
NOR(a, b, c)	$1 + a + b + c + ab + ac + bc + abc$	1.50 GE
AND(a, b, c)	abc	1.50 GE
OR(a, b, c)	$a + b + c + ab + ac + bc + abc$	1.75 GE
XOR(a, b, c)	$a + b + c$	4.00 GE
MAJ(a, b, c)	$ab + ac + bc$	2.25 GE
MUX$(a, b; c)$	$a + ac + bc$	2.50 GE

Table 1: Hardware costs of logical operations

The algebraic normal form of the feedback function is

$$\begin{aligned} A(x_0, x_1, \ldots, x_{18}) = {} & x_0 + x_2 + x_3 + x_5 + x_8 + x_{12} + x_1x_6 + x_2x_6 + x_2x_9 \\ & + x_4x_7 + x_5x_6 + x_9x_{10} + x_9x_{11} + x_2x_4x_6 + x_2x_4x_{10} \\ & + x_2x_6x_9 + x_4x_9x_{10} + x_6x_9x_{10} + x_9x_{10}x_{11} \\ & + x_2x_4x_6x_9 + x_2x_4x_9x_{10} + x_4x_6x_9x_{10}. \end{aligned}$$

The implementation costs for the feedback function $A(x_0, x_1, \ldots, x_{18})$ are 24 GE. A switching circuit for shift register A is shown in Figure 1. Shift register A has linear complexity $2^{19} - 2$.

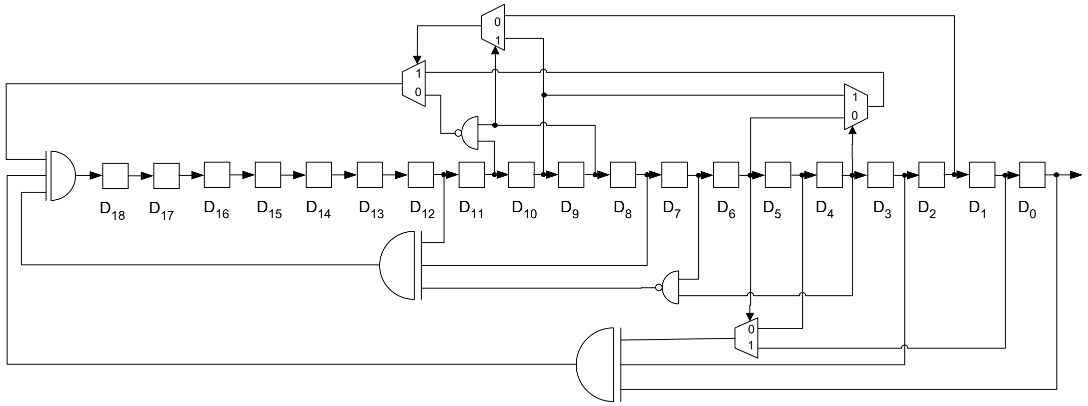

Figure 1: Switching circuit for the new NLFSR A

	Version 1 with DPA protection	**Version 2 with DPA protection**	**Version 2 without DPA protection**
Memory	1002 GE	1245 GE	1245 GE
DPA counter measure	528 GE	655 GE	—
Feedback functions	342 GE	247 GE	247 GE
Combining function	13 GE	63 GE	63 GE
Control logic	288 GE	298 GE	323 GE
Total	**2173 GE**	**2508 GE**	**1878 GE**

Table 2: Design sizes of reduced Achterbahn: Version 1 and Version 2

4.2 Design sizes of parallel implementations of Achterbahn-Version 2

Like the initial NLFSRs of Achterbahn, the new shift registers were chosen in order to facilitate parallel implementations of the KSG. While in a straightforward implementation of the KSG, one bit of keystream is produced per clock cycle, in the parallel implementations two, four, or eight keystream bits are generated per clock cycle. We list the design sizes of the parallel implementations of the KSG for reduced Achterbahn in Table 3. For the sake of comparison, we also list the design sizes of Achterbahn-Version 1. The table contains also the hardware efficiencies of the various implementations. This is the number of keystream bits produced per clock cycle divided by the design size in units of 1000 GE.

Besides the implementations in which countermeasures against the leakage of side channel information are taken (in Table 3 referred to as "Achterbahn with DPA protection"), we also include the design sizes of implementations in which no such counter measures are implemented (in the table referred to as "Achterbahn without DPA protection").

Recall the first part of Achterbahn's key-loading algorithm. In this part all memory cells of the KSG are loaded simultaneously with key bits. The first register, for instance, receives the 19 key bits $k_0, k_1, \ldots, k_{18}$, and the last register, of length 32, the key bits $k_0, k_1, \ldots, k_{31}$. In the next step, the remaining key bits and IV bits are fed serially into the shift registers via an XOR gate in the feedback loop of each shift register. In the third step, the content of one cell of each shift register is overwritten with the bit 1 so that no shift register can be in the all-zero state thereafter. In the last step of the key-loading algorithm, each shift register performs a certain number of warm-up shifts for diffusion purposes.

The intent of the parallel key-loading in step 1 is to avoid the leakage of side channel information in the initialization phase and during resynchronization. Unfortunately, one has to pay a relatively high price in hardware for this feature, to be precise: 655 GE for 262 multiplexors.

In some applications, protection against side channel attacks is not required. For such applications, we can implement the KSG using flip-flops (without reset-capability) which cost 4.75 GE rather than the more expensive scan flip-flops (7.25 GE). The task of the first step of the key-loading algorithm is now accomplished by inserting the key bits serially into each shift register. Contrary to step 2, in this step no feedback values are added to the introduced key bits. The possibility to disable the feedback logic costs one extra multiplexor per shift register resulting in an increase of the control logic by 25 GE. Thus the total saving amounts to 630 GE. See Table 2.

	Achterbahn-Version 1 with DPA protection		**Achterbahn-Version 2 with DPA protection**		**Achterbahn-Version 2 without DPA protection**	
	Design size	Hardware efficiency	Design size	Hardware efficiency	Design size	Hardware efficiency
1-bit impl.	2173 GE	0.46	2508 GE	0.40	1878 GE	0.53
2-bit impl.	2412 GE	0.83	2820 GE	0.71	2188 GE	0.91
4-bit impl.	3113 GE	1.28	3852 GE	1.04	3274 GE	1.22
8-bit impl.	4778 GE	1.67	4888 GE	1.64	4386 GE	1.82

Table 3: Design size and hardware efficiency of parallel implementations of reduced Achterbahn

5 Conclusion

We reported on the results of our computations concerning the linear complexities of the initial and the new NLFSRs constituting the core of Achterbahn's KSG. We outlined a new probabilistic algorithm for estimating the linear complexities of primitive binary NLFSRs. We described tweaks on Achterbahn-Version 1 as specified in [3] that led to Achterbahn-Version 2. The reported cryptanalytic attacks of Johansson, Meier and Muller [7] were discussed and it was shown that the four attacks described in [7] are either not feasible against Achterbahn-Version 2 or have complexities above the complexity of exhaustive key search. We introduced new feedback functions of the shift registers that are more efficient in hardware. All feedback functions now have logical depth three. Properties of the Boolean combining function S for Achterbahn-Version 2 were discussed. The design sizes and hardware efficiencies for the parallel implementations of reduced Achterbahn were updated.

Acknowledgment: We wish to thank Thomas Johansson for sending us a copy of the preprint [7], which drew our attention to the potential threats arising from the combination of divide and conquer attacks with correlation attacks.

References

[1] N. Courtois and W. Meier: Algebraic attacks on stream ciphers with linear feedback, *Advances in Cryptology – EUROCRYPT 2003* (E. Biham, ed.), Lecture Notes in Computer Science, vol. 2656, pp. 345–359, Springer-Verlag, 2003.

[2] B. M. Gammel and R. Göttfert: Linear filtering of nonlinear shift register sequences, *Proc. of The International Workshop on Coding and Cryptography WCC '2005* (Bergen, Norway, 2005), P. Charpin and Ø. Ytrehus, eds., pp. 117-126.

[3] B. M. Gammel, R. Göttfert, and O. Kniffler: The Achterbahn stream cipher, eSTREAM, ECRYPT Stream Cipher Project, Report 2005/002, 29 April 2005. http://www.ecrypt.eu.org/stream/papers.html

[4] B. M. Gammel, R. Göttfert, and O. Kniffler: Improved Boolean combining functions for Achterbahn, eSTREAM, ECRYPT Stream Cipher Project, Report 2005/072, 14 October 2005. http://www.ecrypt.eu.org/stream/papers.html

[5] R. Göttfert: A probabilistic algorithm to determine the linear complexity of a periodic sequence of period $q^n - 1$, manuscript, Oct. 2005.

[6] T. Johansson, W. Meier, and F. Muller: Cryptanalysis of Achterbahn, eSTREAM, ECRYPT Stream Cipher Project, Report 2005/064, 27 September 2005. http://www.ecrypt.eu.org/stream/papers.html

[7] T. Johansson, W. Meier, and F. Muller: Cryptanalysis of Achterbahn, Preprint, Jan. 2006.

[8] A. H. Chan, R. A. Games, and E. L. Key: On the complexities of de Bruijn sequences, *J. Combin. Theory Ser A* **33**, 233–246 (1982).

[9] J. Dj. Golić: On the linear complexity of functions of periodic GF(q) sequences, *IEEE Trans. Inform. Theory* **35**, 69–75 (1989).

[10] R. A. Rueppel and O. J. Staffelbach: Products of linear recurring sequences with maximum complexity, *IEEE Trans. Inform. Theory* **IT-33**, 124–131 (1987).

[11] E. S. Selmer: *Linear Recurrence Relations over Finite Fields*, Univ. of Bergen, 1966.

[12] T. Siegenthaler: Correlation-immunity of nonlinear combining functions for cryptographic applications, *IEEE Trans. Inform. Theory* **IT-30**, 776–780, 1984.